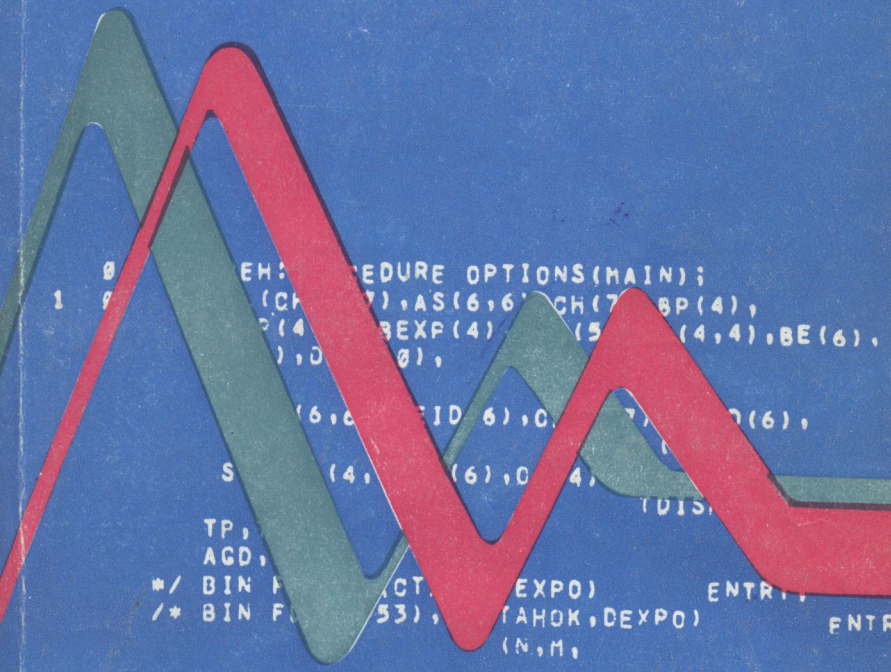
А.С. КУТЬКОВА

ЧЕЛОВЕК – КОМПЬЮТЕР – БУДУЩЕЕ



пособие по английскому языку для вузов

ВШ

А. С. Кутькова

ЧЕЛОВЕК — КОМПЬЮТЕР — БУДУЩЕЕ

Пособие по английскому языку

Допущено Министерством
высшего и среднего
специального образования СССР
в качестве учебного пособия
для студентов высших технических
учебных заведений

МОСКВА «ВЫСШАЯ ШКОЛА» 1987

Рецензенты:

кафедра иностранных языков Обнинского филиала Московского инженерно-физического института (зав. кафедрой И. А. Воробьева);

канд. филол. наук, доц. К. А. Иванова (Ленинградский электротехнический институт им. В. И. Ульянова (Ленина))

Кутькова А. С.

К 95 Человек — компьютер — будущее: Пособие по английскому языку: Учеб. пособие для втузов. — М.: Высш. шк., 1987.—127 с.; ил.— На англ. яз.

Пособие построено на текстах по архитектуре ЭВМ, программированию, видам программ, операционным системам, мини- и микроЭВМ, автоматизированным системам управления и др.

Цель пособия — совершенствование навыков чтения и понимания литературы по специальности для получения информации, ведения беседы в пределах пройденной тематики.

К $\frac{4602010000-106}{001(01)-87}$ 275—87

ББК 81.2 Англ-923

ПРЕДИСЛОВИЕ

В настоящее время в нашей стране огромное внимание уделяется развитию ЭВМ, микропроцессорной техники, а также автоматизированным системам управления, гибким производствам с использованием роботов.

В материалах XXVII съезда КПСС одним из важнейших направлений научно-технического прогресса названа автоматизация производства. Современный этап автоматизации опирается на революцию в электронно-вычислительной технике, электронизацию народного хозяйства. В двенадцатой пятилетке предусмотрено создание новых поколений ЭВМ. Определены конкретные задачи по разработке и массовому освоению современной компьютерной техники.

Понимание роли вычислительной техники в развитии общества, способность применить ее в своем деле, умение решать задачи с помощью ЭВМ становится второй грамотностью.

Настоящее учебное пособие предназначено для студентов технических вузов, обучающихся по специальностям «Электронно-вычислительная техника», «Автоматизированные системы управления», «Прикладная математика», «Робототехника» и продолжающих изучение английского языка в вузе на II и III этапах обучения. Оно составлено в соответствии с требованиями Программы по английскому языку для неязыковых специальностей высших учебных заведений (М.: 1986). Цель пособия — сформировать у студентов навыки и умения в чтении, переводе оригинальной научно-технической литературы по специальности, а также развить навыки устной речи в пределах проработанной тематики. Учебным пособием могут пользоваться также научные работники, аспиранты и переводчики, работающие в области вычислительной техники.

Пособие состоит из трех частей: первая часть содержит учебные тексты и упражнения к ним, вторая — тексты для внеаудиторного чтения и третья — вспомогательный материал. Первая часть, в свою очередь, состоит из двух тематических разделов: аппаратное обеспечение ЭВМ и программное (математическое) обеспечение ЭВМ.

Все тексты пособия взяты из оригинальной литературы. Они сокращены, но не адаптированы. В них описываются последние достижения отечественной и зарубежной науки и техники в области создания и использования ЭВМ. Некоторые из них сопровождаются рисунками. Тексты А предназначены для формирования навыков изучающего чте-

ния. Дополнительные тексты B и B′ способствуют формированию навыков просмотрового и ознакомительного чтения, а также лучшему усвоению лексики по специальности. Форма и место выполнения заданий к ним определяется преподавателем. Каждый из 11 уроков заканчивается диалогом или текстом C, записанными на магнитную ленту. Они предназначены для активной проработки и аудирования в лаборатории устной речи.

В методическом плане пособие построено так, что лексико-грамматический материал прорабатывается комплексно. Подтекстовые упражнения, как правило, лексические; после текста, помимо упражнений, связанных непосредственно с его проработкой, помещены, в основном, грамматические упражнения.

Пособие рассчитано на 120—150 часов аудиторных занятий.

Автор выражает глубокую благодарность профессору Е. М. Синельникову, доцентам В. О. Голубинцеву и М. Е. Синельниковой, старшему преподавателю И. Д. Удовенко за консультации и советы при подготовке рукописи данной книги, а также выражает особую признательность рецензентам: зав. кафедрой иностранных языков ЛЭТИ им. В. И. Ульянова (Ленина) канд. филол. наук, доценту К. А. Ивановой и кафедре иностранных языков Обнинского филиала Московского инженерно-физического института за ценные рекомендации и предложения, которые были учтены при подготовке рукописи к печати.

Автор

CONTENTS

Part I

PART I

SECTION I. HARDWARE

LESSON ONE

Grammar. The Verb (basic forms: Indefinite, Continuous, Perfect Active)
Text A. What is Computing?
Dialogue. Meeting a Friend.

Exercises

I. Read the international words and guess their meaning:

centre; figure; final; to stop; to cross; real; machine; information; ordinary; business; complex; modern; automatic; second; multiplication; division; arithmetical; mathematical; logical; display; computer; operation; physical; number; analog; hybrid; press; to combine

II. Pronounce the following words correctly:

suppose [sə'pouz] *v* предполагать
digit ['dɪdʒɪt] *n* цифра, число; разряд; символ
column ['kɔləm] *n* столбик; столбец
compute [kəm'pju:t] *v* считать, вычислять
take in ['teɪk'ɪn] *v* принимать *(информацию)*
data ['deɪtə] *n pl* данные; информация
perform [pə'fɔ:m] *v* выполнять; производить *(действие)*
reasonable ['rɪ:znəbl] *a* разумный
produce [prə'dju:s] *v* производить; порождать; синтезировать
sequence ['si:kwəns] *n* последовательность; ряд
receive [rɪ'si:v] *v* получать
put out ['put'aut] *v* выдавать *(информацию)*
print [prɪnt] *v* печатать
number ['nʌmbə] *n* число; номер; цифра
keyboard ['ki:bɔ:d] *n* клавишный пульт; клавиатура
press [pres] *v* нажимать
total ['toutl] *n* сумма, итог
key [ki:] *n* кнопка; клавиша; ключ
addition [ə'dɪʃ(ə)n] *n* сложение
subtraction [səb'trækʃ(ə)n] *n* вычитание
multiplication [,mʌltɪplɪ'keɪʃ(ə)n] *n* умножение
division [dɪ'vɪʒ(ə)n] *n* деление
store [stɔ:] *v* хранить; запоминать
hardware ['hɑ:dweə] *n* аппаратное обеспечение ЭВМ
express [ɪks'pres] *v* выражать
relation [rɪ'leɪʃ(ə)n] *n* отношение; соотношение; зависимость; связь
compare [kəm'peə] *v* сравнивать
select [sɪ'lekt] *v* выбирать
display [dɪs'pleɪ] *n* дисплей; *v* показывать
numerical [nju(:)'merɪk(ə)l] *a* числовой, численный
measurement ['meʒəmənt] *n* измерение; вычисление
property ['prɔpətɪ] *n* свойство, особенность

III. Read and memorize the following word combinations:

a set of figures — ряд цифр, набор цифр
the right-hand column — правый столбик
to take in (put in) — принимать *(информацию)*
a sequence of reasonable operations — последовательность разумных операций
the ordinary business adding machine — обычная канцелярская суммирующая машинка

TEXT A. WHAT IS COMPUTING?

1. Suppose you sit down with pencil and paper and centre your attention on adding a set of figures. You add first all the digits in the right-hand column, then all the digits in the next column, and so on—until you finally arrive at the answer.[1] When you do this, you are computing.

2. When you stop at a street corner, looking first to the left for any coming car, then to the right, to cross the street or to wait on the sidewalk [2]—you are computing.

3. When you are walking along a poorly marked path [3] in the woods, thinking if you are really on the path or have lost it [4]—you are computing.

4. When you are taking in information or data, performing reasonable operations (mathematical or logical operations) on the data, and are producing one or more answers—you are computing.

5. A machine can also do this. It can take in information or data, perform a sequence of reasonable operations on the information which it has received, and put out answers. When it does this, it is computing.

6. A very simple example of a computer is the ordinary business adding machine which prints on paper tape the number entered into its keyboard, and also prints a total when you press the total key. A complex example of a computer is a modern automatic digital computer which in each second can perform more than 100,000,000,000 additions, subtractions, multiplications, or divisions.

7. A computing machine can take in and store information because the hardware inside the machine expresses arithmetical and logical relations, such as [5] adding or subtracting, comparing or selecting. A computer can also put out information, display the answers when it receives them. Hardware is useless without software which is computer instructions and programs.

8. The modern computers are of three kinds called analog, digital, and hybrid. An analog computer computes by using physical analogs of numerical measurements. A digital computer computes by using the numbers (digits) and yeses and noes expressed usually in 1's and 0's.[6] A hybrid computer is a machine which combines some of the properties of digital and analog computers.

Notes

[1] **until you finally arrive at the answer** — до тех пор, пока вы наконец не получите ответ
[2] **to cross the street or to wait on the sidewalk** — чтобы перейти улицу или подождать на тротуаре
[3] **a poorly marked path** — едва заметная тропинка
[4] **if you are really on the path or have lost it** — действительно ли вы находитесь на тропинке или заблудились
[5] **such as** — такие как
[6] **yeses and noes expressed usually in 1's and 0's** — да и нет, выраженных обычно единицами и нулями

Exercises

IV. Give the Russian equivalents:

to centre attention on; to the right; to the left; a set of figures; and so on; a sequence of reasonable operations; to put out answers; to take in information; to store information; such as; to express mathematical and logical relations; hardware is useless without software; by using physical analogs; numerical measurements

V. Give the English equivalents:

налево; направо; выдавать информацию или данные; последовательность разумных операций; правый столбик; ряд цифр; и так далее; сосредоточивать внимание

VI. Translate the words of the same root. Define speech parts:

to centre—a centre; to add—addition; to compute—a computer—computation; to mark—a mark; to inform—information; datum—data (mind! a date); to perform—performance; reasonable—a reason; to operate—operation; business—busy; to subtract—subtraction; to multiply—multiplication; to divide—division; logic—logical; arithmetic—arithmetical; measurement—to measure; relation—to relate

VII. Memorize the following definitions:

1. A machine which performs a sequence of reasonable operations on information is a computer. 2. An analog com-

puter is a machine which computes by using physical analogs of numerical measurements. 3. A digital computer is a machine which computes by using digits or numbers. 4. A hybrid computer is a machine which combines some properties of digital and analog computers.

VIII. Answer the following questions:

1. How many operations does a modern computer perform? 2. What kinds of operations does an automatic digital computer perform? 3. What is a computer? 4. What is a very simple example of a computer? 5. Why does a computer take in and store information or data? 6. What kinds of modern computers do you know? 7. How does an analog computer compute? 8. How does a digital computer compute? 9. What is a hybrid computer? 10. What is hardware? 11. What is software?

IX. Choose the 3rd form of the given verbs:

took, takes, take, taken; thought, think, thinks; arrive, arrives, arrived; did, done, do, does; speaks, speak, spoke, spoken; saw, seen, see, sees; was, were, is, am, be, been, are; writes, wrote, written; has, had, have; expresses, expressed, express; receive, received, receives

X. Translate the following sentences paying attention to the use of Indefinite, Continuous, Perfect Tenses in the Active Voice:

1. My friend studies at the department of Electrical Engineering. I study at the same department. We do research. Comrade Ivanov also does his research (научное исследование). We discussed various (различный) problems at the seminar. They did many operations on the computer ES-1045. Yesterday we went to the Institute library. I answered all the questions at the exam in mathematics. We shall solve this algebraic problem. They will do their work in time. I shall compare my results with yours. 2. When you came he was calculating (вычислять) his problem. She was discussing some questions with her instructor. I shall be waiting for you at 6 o'clock. 3. He has given a short answer to my question. I have just (только что) come from Leningrad. They have already gone home. They had written the program for the computer by June.

XI. Fill in the blanks with the following verbs in the Present Indefinite Active:

to express; to compute; to receive; to perform; to store; to calculate; to put out; to take in; to do; to display

1. A computer ... or .... 2. Modern automatic digital computers ... more than 100 mln. additions, subtractions, multiplications, or divisions in each second. 3. A computer ... information or data, ... a sequence of reasonable operations on information and ... answers. 4. The hardware inside the computer ... arithmetical or logical relations such as adding or subtracting, comparing or selecting. 5. When computers ... results, they ... them.

XII. Give the proper tense of the verbs given in brackets:

1. An ordinary business adding machine (to be) a very simple example of a computer. 2. When a computer (to do) reasonable operations, it (to compute) or (to calculate). 3. When I (to come) into the room my friend (to compile — составлять) a program. 4. A computer (to store) information which it (to receive). 5. ... (to come) to the laboratory, I ... (to show) you how to calculate by using a computer. 6. The electronic computer ES-1045 just (to perform) its calculations.

XIII. Listen to the dialogue from the magnetic tape and reproduce it.

Notes

[1] **Haven't seen you for ages!** — Не видел тебя целую вечность!
[2] **to enter the Institute** — поступать в институт
[3] **Good for you!** — Молодец!
[4] **the Electrical Engineering Department** — *эд.* электромеханический факультет
[5] **Automated Management Systems** — автоматизированные системы управления
[6] **Applied Mathematics** — прикладная математика
[7] **They say** — Говорят
[8] **one should know** — нужно знать
[9] **You're (are) right.**— Ты прав.
[10] **With pleasure** — С удовольствием
[11] **to be called** — называться
[12] **too** — также, тоже
[13] **In what capacity will you work?** — Кем ты будешь работать?
[14] **to graduate from the Institute** — окончить институт
[15] **a system-engineer** — инженер-системотехник
[16] **an engineer-mathematician** — инженер-математик
[17] **I am sorry.**— Прости.
[18] **to hurry** — спешить, торопиться
[19] **I'm (am) afraid to be late** — Боюсь опоздать
[20] **So long!** — Пока!
[21] **Wish you luck** — Желаю тебе удачи

MEETING A FRIEND

A: Hallo, B.! Where are you going?
B: Hallo, A.! Haven't seen you for ages! [1] I am going to the Institute.
A: Oh! You are a student now! What Institute have you entered? [2]
B: I've entered the Polytechnic Institute.
A: Good for you! [3] And what department do you study at?
B: I study at the Electrical Engineering Department.[4] My speciality is Electronic Computers (Automated Management Systems,[5] Applied Mathematics [6]).
A: You liked mathematics at school, didn't you? They say [7] one should know [8] that subject well to study your speciality.
B: You're right.[9] Mathematics was my favourite subject at school.
A: And you know already what a computer is, don't you?
B: Yes, I do. A computer is a machine which performs a sequence of reasonable operations on information. I also know what simple and complex computers are.
A: It's very interesting! Tell me, please.
B: With pleasure,[10] listen to: a simple computer is an ordinary business adding machine, and a complex computer is a modern electronic digital computer which performs more than 100 mln. operations a second.
A: Can you say, please, what kinds of operations a modern electronic computer performs?
B: Yes, I can. It performs addition, subtraction, multiplication, and division. These operations are called [11] arithmetical operations. But modern computers perform logical operations too.[12]
A: Well, what are they?
B: Logical operations which I know are comparing and selecting.
A: In what capacity will you work [13] after graduating from the Institute? [14]
B: I shall work as a system-engineer [15] (an engineer-mathematician[16]). But I am sorry.[17] I must hurry [18] to the Institute, I'm afraid to be late [19] for the lectures. So long![20]
A: Wish you luck [21] in your studying. So long!

LESSON TWO

Grammar. Indefinite, Continuous, Perfect Passive. Modal Verbs and Their Equivalents. Gerund. There Be.

Text A. Information, Machine Words, Instructions, Addresses and Reasonable Operations.
Text B. How Much Should an Educated Man Know about Computers?

Exercises

I. Read the international words and guess their meaning:

character; reaction; to react; instruction; register; code; bit; byte; typically; group; address; magnetic; polarize; differentiation; integration; binary; basic; physically; physical; selection; to sort; separate; part; standard; correct

II. Pronounce the following words correctly:

sign [saɪn] *n* знак, обозначение; символ; признак
character ['kærɪktə] *n* знак; символ; цифра; буква; признак
typewriter ['taɪp,raɪtə] *n* пишущая машинка
react [rɪ'ækt] *v* реагировать
unit ['juːnɪt] *n* единица; блок; устройство; элемент
rotation [rou'teɪʃən] *n* вращение; чередование
voltage ['voultɪdʒ] *n* напряжение; разность потенциалов
variable ['vɛəɾɪəbl] *n* переменная (величина)
represent [,reprɪ'zent] *v* представлять
include [ɪn'kluːd] *v* включать
zero ['zɪərou] *n* нуль; нулевая точка
separate ['seprɪt] *a* отдельный
call [kɔːl] *v* называть
consist [kən'sɪst] *v* **(of)** состоять (из)
handle ['hændl] *v* управлять; оперировать
the same [seɪm] *a* тот же самый
arrangement [ə'reɪnd(ʒ)mənt] *n* расположение, размещение
some [sʌm] *a* некоторый
polarize ['poulərаɪz] *v* поляризовать
surface ['sə:fɪs] *n* поверхность; плоскость
state [steɪt] *v* устанавливать, указывать
address [ə'dres] *n* адрес; *v* адресовать
contain [kən'teɪn] *v* содержать
square [skwɛə] *n* квадрат; площадь
power ['pauə] *n* мощность; сила; энергия; степень; способность
sorting ['sɔːtɪŋ] *n* сортировка
matching ['mætʃɪŋ] *n* согласование
determine [dɪ'tə:mɪn] *v* определять

III. Read and memorize the following word combinations:

a number of = a set of — ряд; несколько; некоторое количество
per second — в секунду
a particular memory location or cell — особая ячейка памяти
its own unique address — свой собственный уникальный адрес
instead [ɪn'sted] of saying — вместо того, чтобы сказать
to take a square root — извлекать квадратный корень
to raise to a power — возводить в степень
as well as — так же как

TEXT A. INFORMATION, MACHINE WORDS, INSTRUCTIONS, ADDRESSES AND REASONABLE OPERATIONS

1. Information is a set of marks or signs that have meaning. These consist of letters or numbers, digits or characters, typewriter signs, other kinds of signs, and so on. A computer reacts differently to different digits or characters, and reacts to them as units that have meaning. For example, information for an analog computer has to be in the form of distances, or rotations, or voltages, or other physical variables. And for a digital computer information has to be in the form of digits or numbers.

2. Any information may be represented by the binary system including two digits: one (1) and zero (0). Each 1 and 0 is a separate binary digit called a bit. A bit is the smallest part of information. Bits are typically grouped in units that are called bytes. A byte is the basic unit of information used in modern computers and consists of eight bits.

3. The bytes are handled usually in standard groups called machine words or just words. There are two basic types of information or words that can be put into a memory cell or location: words that are numerical quantities [1] and words that are computer instructions. Regularly, an instruction to the machine is expressed as a word; and so the same set of characters may have meaning sometimes as a number, sometimes as an instruction. A speed of 96,000 characters per second is the same as a speed of 8,000 words per second. Most human beings could not take even 12-digit number per second.[2]

4. Physically the set of bits is a set of arrangements of some physical equipment. One of the ways of storing information in a computer is storing by using a set of small magnetically polarized spots (пятна) on a magnetic surface.

5. The computer is told what operations to perform by means of instructions. An instruction is a command to the computer. It consists of a verb (an operational code) and a noun (an operand). For example, if the computer is instructed "Add 365 the number of times (раз) stated in the register R", and if the register R stores the code for number 3, then the computer will perform that operation three times. An instruction word looks like a number, and there is no way to tell from the word itself whether it is a quantity or an instruction. The computer must be told exactly (точно) which address contains an instruction and which contains a quantity.

6. An address is the name of particular memory location or cell. Each memory location (word or byte) has its own unique address or number just like a post office box.[3] For example, if the computer contains 100 memory cells, their respective addresses might be the numbers[4] from 1 to 100 (or 0 through 99). And instead of saying "A word is in a memory cell", the computer personnel say, "The contents (содержимое) of an address is a word."

7. Reasonable operations are mathematical and logical. Mathematical operations include arithmetic and algebraic operations. Arithmetic operations are addition, subtraction, multiplication, division, taking a square root, etc.; and algebraic operations are called raising to a power as well as differentiating and integrating.

8. Logical operations include comparing, selecting, sorting, matching, etc. These are operations which may be performed either on numbers, or on expressions consisted of letters such as ordinary words. A very important logical operation performed by a computer is determining, i.e., which of two operations is to be performed next.

Notes

[1] **numerical quantities** — численные значения (величины)
[2] **Most human beings could not take even 12-digit number per second** — Большинство людей не смогли бы воспринять даже 12-значное число в секунду
[3] **just like a post office box** — точно так же как почтовый ящик
[4] **their respective addresses might be the numbers** — их соответственные адреса могли бы быть числами

Exercises

IV. Find the equivalents:

1. numerical measurements	1. физическое оборудование
2. by using physical analogs	2. скорость в секунду
3. the binary system	3. численные величины
4. any information	4. численные измерения
5. other physical variables	5. двоичная система
6. numerical quantities	6. любая информация
7. a speed per second	7. один из способов
8. physical equipment	8. особая ячейка памяти
9. one of the ways	9. используя физические аналоги
10. a particular memory cell	10. другие физические переменные

V. Arrange (a) synonyms and (b) antonyms in pairs and translate them:

a) to perform; to compute; to take in; a figure; to arrange; to show; to carry out; a digit; to calculate; little; to display; small; to receive; to position; instruction; data; location; command; information; cell

b) inside; to the right; addition; multiplication; simple; to the left; division; outside; small; complex; big; subtraction; differentiating; at the right; integrating; at the left; the right-hand column; the left-hand column

VI. Form nouns from the verbs by adding suffixes:

-ment [-mənt]

M o d e l: to equip—equipment

to arrange, to require, to measure, to state, to develop

-sion [-ʒn]

M o d e l: to divide—division

to decide, to include, to conclude, to exclude

-ion [-(ʃ)n]

M o d e l: to direct—direction

to subtract, to select, to react, to construct, to act

-ation [-eɪʃn]

M o d e l: to compute — computation

to inform, to combine, to determine, to represent, to differentiate, to integrate, to polarize

VII. Complete the following sentences:

1. Information is a set of marks or signs that .... 2. Reasonable operations are .... 3. Logical operations include .... 4. A very important logical operation is .... 5. Physically the set of bits is a set of arrangements of some .... 6. Determining means which of two operations is to be performed .... 7. The computer must be told exactly which address contains ... and which contains .... 8. The computer is told what operations to perform by means of .... 9. Any information may be represented by the binary system including two digits: ... and .... 10. A bit is the smallest part of ....

VIII. Memorize the following definitions:

1. Information is a set of marks or signs that have meaning. 2. Reasonable operations are mathematical and logical operations. 3. Mathematical operations include arithme-

tic and algebraic operations. 4. The smallest unit of information for a computer is called a byte. A byte consists of eight bits. 5. The word "bit" is formed from the letter "b" in the word "binary" and two letters "it" in the word "digit".

IX. Answer the following questions:

1. What is information? 2. What does a set of marks consist of? 3. What is one of the ways of storing information in a computer? 4. What is a byte? 5. What is a bit? 6. How many bits does a byte usually contain? 7. What is an instruction? 8. By means of what is the computer told what operations to perform? 9. What are machine words? 10. What does a machine word consist of? 11. What are addresses? 12. What are reasonable operations? 13. What do mathematical operations include? 14. What do logical operations include? 15. In what system is information represented for the modern electronic computer? 16. What are the two basic types of words that are put into a memory?

X. Translate the following sentences paying attention to the words in bold type:

a) 1. What **kind** of computer was it?—It was a digital computer. 2. The computer can perform different **kinds** of operations. 3. This **kind** of logical problem is very simple. 4. Will you **kindly** explain to us the operation of this **kind** of machine? 5. Be so **kind**, show us the new equipment.

b) 1. **As** you already know logical relations in a computer are expressed by hardware. 2. **As** the ordinary business adding machine has the special equipment inside it, it can store information. 3. An electronic computer can add, subtract, multiply, and divide **as well**. 4. Natural sciences include mathematics, physics, chemistry, medicine, geology, biology, bionics, **as well as** the engineering sciences and other fields of knowledge. 5. **As** our professor was speaking of the history of computers, he mentioned P. L. Chebyshev and his great inventions. 6. My friend works at a plant **as** an engineer.

c) 1. Progressive people everywhere in the world must fight **for** peace. 2. Yesterday we heard an interesting lecture on modern electronics which was arranged **for** the students. 3. He asked me **for** a book on microcomputers' organization. 4. The first computing machines were not reliable (надежный), **for** there were no good electrical units. 5. The instructions are placed inside the computer, **for** the computer itself can select the numbers of instructions.

XI. Read and translate the following sentences paying attention to the predicates in the Passive Voice:

1. These digits are easily multiplied. 2. I was asked many questions about my work. 3. They were explained how to solve this problem on a computer. 4. The sequence of reasonable operations has been performed by the computer. 5. The new department of mathematics has just been opened. 6. Many books on computers' organization and architecture had been translated from Russian into English by the end of last year. 7. The experiments on the new microcomputer were being carried out during the whole month. 8. All the digits are recorded on the paper tape when addition is performed. 9. The new key adding machine was transferred into the next room yesterday. 10. The sequence of reasonable operations is now being carried out by this microcomputer. 11. The conference was addressed by a well-known scientist. 12. The invention of computers was spoken of at the last lecture. 13. Modern personal computers are always looked at with interest. 14. Many new branches of industry have been developed in our country since World War II.

XII. Fill in the blanks with the verbs given below. Use them in the Passive Voice:

to express; to carry out; to invent; to record; to polarize; to tell; to store; to represent; to require; to construct

1. All the digits inside the hardware ... by the arranging of the special equipment. 2. Complex calculations ... with the help of a computer. 3. A special counter wheel (счетное колесо) for an arithmometer ... by a Russian engineer V.T. Ordner in 1874. 4. The answers of computations ... often in the form of tables. 5. Small spots on a surface inside a computer ... magnetically. 6. By means of instruction any computer ... what operations to perform. 7. All instructions ... in registers, the units of hardware. 8. Any information ... by the binary system. 9. Numbers or instructions ... for solving a problem by a computer. 10. Several computing units ... by M. V. Lomonosov for computational science.

XIII. Read and translate the following sentences paying attention to the modal verbs and the equivalents:

1. Information or data can be stored in the computer's memory or storage. 2. An analog computer is able to calculate by using physical analogs of numerical measurements. 3. The first automatic computers could operate at the low speed. 4. Your paper may be published at our Institute.

5. My friend was happy when at last he might work at the computing centre. 6. Our students are allowed to visit the computing centre to see the operation of the computer ES-1045. 7. Every student must know that a digital computer performs reasonable operations. 8. Some operations for this computer have to be changed and new instructions have to be added. 9. The instructions are recorded in the order in which they are to be carried out. 10. You should know the difference between the digital and analog computers. 11. We ought to help him to solve this problem by a personal computer. 12. According to (согласно) the time-table you are to begin your classes at 8 o'clock. 13. Every student of our speciality has to know what a hybrid computer is. 14. We were permitted to attend the conference on cybernetics [ˌsaɪbə'netɪks].

XIV. Find the sentences in which the verbs 'to have' and 'to be' are translated as «должен»:

1. This ordinary adding machine has ten keys for each column of digits. 2. The main task of this article was to show the results of research work. 3. This personal computer has been constructed at our lab. 4. The lecture was to begin at 9 o'clock. 5. Our aim is to study hard and master our speciality. 6. Our lab assistant has to construct this electronic device (прибор). 7. The general purpose of this unit (block) is to perform different arithmetic operations. 8. The participants of the scientific conference are to arrive tomorrow. 9. You have to remember the names of the scientists who have contributed to the development of your speciality. 10. The results of the experiment have carefully been checked up today.

XV. Translate the following sentences paying attention to 'there + to be':

1. There are many universities and institutes in our country. 2. There is a students' scientific and technical society at our Institute. 3. There are various computers at our computing centre. 4. There were only four departments in our Institute before the World War II. 5. There will be some engineers at the seminar on programming tomorrow. 6. There is a seminar on the History of the CPSU today. 7. There was a lecture on cybernetics yesterday. 8. There were many ways of solving the problem. 9. There are many complex parts and units in every computer. 10. There will be some new laboratories in our Institute next year.

XVI. State the functions of the Gerund. Translate the sentences:

1. Logical operations consist of comparing, selecting, sorting, matching, and determining. 2. The way of solving this problem is very difficult. 3. After performing calculations a computer displays a result. 4. A set of marks or signs can be stored by polarizing little spots on a magnetic surface. 5. Differentiating and integrating are algebraic operations. 6. Registers are used for storing information. 7. Blaze Pascal's merit consists in his constructing the first mechanical computer. 8. By performing the reasonable operations on a computer we solve different kinds of problems for our national economy.

XVII. Translate into English using the Gerund:

1. Одним из способов хранения информации внутри вычислительной машины является хранение с помощью ряда поляризованных точек на магнитной поверхности. 2. Путем размещения специальных устройств внутри вычислительной машины можно хранить информацию. 3. Возведение в степень и извлечение квадратного корня — математические операции. 4. Компьютер используется для решения сложных задач.

XVIII. Listen to Text B from the tape recorder. Give its contents in short (in Russian).

Notes

[1] **an educated man** — образованный человек
[2] **basic knowledge** — основные знания
[3] **at least** — по крайней мере
[4] **well enough** — достаточно хорошо
[5] **suitable** — подходящий, соответствующий
[6] **to enjoy** — получать удовольствие
[7] **views and opinions** — точки зрения и мнения
[8] **on evidence** — при наличии доказательств
[9] **to be desirable** — быть желательным
[10] **significant** — существенный
[11] **could be put down on ten sheets of paper** — могли бы поместиться на десяти страницах

TEXT B. HOW MUCH SHOULD AN EDUCATED MAN KNOW ABOUT COMPUTERS?

Some years ago in the United States of America a discussion on the question what an educated man [1] is was organized. At this discussion one of the definitions of an 'educated man' was this:

An educated man:

— is able to read, write, and do arithmetics;

— has a basic knowledge [2] of the history and geography of world and man;

— understands the scientific method, and has an elementary knowledge of at least [3] one science;

— has an elementary knowledge of mathematics and logic, what they are, and how to use them;

— knows at least one other language besides his own well enough [4] to read it and talk a little in it;

— can say what he means in suitable [5] words both speaking and writing;

— is able to listen, knows how to learn, and enjoy [6] learning;

— never forgets that his views and opinions [7] may be wrong, and is always ready to change them on evidence; [8]

— has an elementary knowledge of computers and programming, and some active experience (опыт) with a whole computer is highly desirable.[9]

An educated man of 200 years ago in the United States did not need to know anything about science. The educated man of 25-30 years ago did not need to know anything about computers. But the educated man of today needs to have at least some significant [10] knowledge of science, and at least a little significant knowledge about computers.

The summary knowledge that an educated man should know about computers could be put down on ten sheets of paper,[11] in about 3,000 words.

LESSON THREE

Grammar. The Infinitive and Its Constructions. Emphatic Construction 'It is (was) . . . that (who, which) . . .'
Text A. Input and Output Units.
Text B. Terms.
Text B'. Reliability.
Text C. The ES Electronic Computers.

Exercises

I. Read the following international words and guess their meaning:

active; passive; practical; problem; programming; experiment; to consult; form; regularly; to construct; arithmometer; to energize; to control; peripheral; efficient; buffer; pulse; impulse; communication; interesting; to operate; functional; minute; record; line; factor; process

II. Pronounce the following words correctly:

input ['ınput] *n* ввод; входное устройство; вход
device [dı'vaıs] *n* устройство; прибор, механизм; элемент
provide [prə'vaıd] *v* обеспечивать; снабжать; давать
means [mi:nz] *n pl* средство
communication [kə,mju:nı'keıʃ(ə)n] *n* связь; сообщение
accept [ək'sept] *v* принимать (информацию)
purpose ['pə:pəs] *n* цель; намерение; назначение
north-south ['nɔ:θ'sauθ] север — юг
vice versa ['vaısı'və:sə] и наоборот
output ['autput] *n* вывод; выходное устройство; выход
acceptable [ək'septəbl] *a* приемлемый
message ['mesıdʒ] *n* сообщение; передаваемый блок информации
vary ['vɛərı] *v* менять(ся)
according to [ə'kɔ:dıŋtə] *phr cj* согласно
capacity [kə'pæsıtı] *n* способность; емкость
auxiliary [ɔg'zıljərı] *a* вспомогательный; дополнительный
rate [reıt] *n* скорость; степень; норма; коэффициент; частота
control [kən'troul] *v* управлять; контролировать
speed [spi:d] *n* скорость; быстродействие
consequently ['kɔns(ı)kwəntlı] *adv* следовательно
tremendous [trı'mendəs] *a* огромный
release [rı'li:s] *v* освобождать
ratio ['reıʃıou] *n* отношение; коэффициент; соотношение
advantage [əd'vɑ:ntıdʒ] *n* преимущество

III. Memorize the following word combinations:

the means of communication — средство связи
punched holes and blanks — перфорированные отверстия и пробелы
a punched card — перфокарта
a punched tape — перфолента
a human being — человек
data-handling equipment — оборудование по управлению данными
a high-speed line-printer — высокоскоростное построчно-печатающее устройство
a factor of advantage — коэффициент преимущества
the peripheral or auxiliary devices — вспомогательные устройства
the proper speed — надлежащая скорость

TEXT A. INPUT AND OUTPUT UNITS (I/O UNITS)

1. The part of the computer that takes in information is called the input unit. The input unit or device provides the means of communication between the computer and the people who are interested in its operation.

2. To be accepted by the machine, information for a digital computer has to be in the form of digits 0, 1, 2, 3, 4, ... 9 or characters A, B, C, D, ... These characters are regu-

larly expressed for the computer's purposes as six or seven 1's and 0's. The 1's and 0's may be expressed for the computer: as punched holes (1) and blanks (0) in a card or a paper tape; as presence (1) and absence (0) of electrical pulse; or as polarized spots on a magnetic surface; for example, south-north is 1 and north-south is 0, or vice versa, etc.

3. So, the input unit makes possible communication from the other data-handling equipment and human being [1] to the computer. It is the functional part of the computer that accepts the data to be operated on and programs for operating. It may consist of a keyboard operated tape punch,[2] a paper tape reader, a card reader, and an electric typewriter.

4. The part of a computer that puts out information is called the output unit. The computer can easily put out information in a form acceptable to human beings. For example, the computer may give impulses to an electric typewriter, so that the keys are energized in the proper sequence to type out a message in ordinary typed characters which human beings can read, etc.

5. The output of a computer is known to vary according to the capacity of the auxiliary equipment receiving the information. A computer can record on a magnetic tape at the rate of 1,000,000 characters per second. It can also control: a paper tape punch which will punch a paper tape at the rate of 100 characters per second; or a card punch which will punch per second about 300 standard punch cards of 80 columns; or a high-speed line-printer which will punch 20 lines per second, each of 80 to 120 characters. Input and output devices are usually called peripherals.

6. All this peripheral equipment is slow as compared with the computer. Consequently, for efficient use of the computer's tremendous calculating speed, devices called buffers may be used. A buffer is known to be a storage device which is able to take in information at a very high speed from the computer and release the information at the proper speed for the peripheral equipment.

7. A human being is known to write by hand at the rate of about 30 words per minute, or to type at the rate of about 60 words per minute, or to talk at the rate of 200 or 250 words per minute. The ratio between a computer speed of about 40,000 words per second, and the top output speed of a human being of about 4 words per second, gives a factor of advantage to the computer of about 10,000 to 1 at the beginning of the 60's. Nowadays this ratio is much more.

Notes

[1] **the input unit makes possible communication from the other data-handling equipment and human being** — блок ввода позволяет установить связь от другого оборудования по управлению данными и человеком

[2] **of a keyboard operated tape punch** — ленточного перфоратора, работающего от клавишного пульта

Exercises

IV. Find the equivalents:

1. the input of a computer	1. наличие и отсутствие
2. an auxiliary equipment	2. перфорированные отверстия
3. at the rate of	3. устройство вывода
4. a high-speed line-printer	4. высокоскоростное построчнопечатающее устройство
5. for efficient use	5. коэффициент преимущества
6. a storage device	6. со скоростью
7. a factor of advantage	7. для эффективного использования
8. the output unit	8. вспомогательное оборудование
9. punched holes	9. устройство ввода компьютера
10. presence and absence	10. запоминающее устройство

V. Insert prepositions where necessary:

1. A computer can record ... magnetic tape ... the rate ... 1,000,000 characters ... second. 2. As compared ... the computer the auxiliary or peripheral equipment is rather slow. 3. A human being can write ... hand ... the rate ... 30 words ... minute. 4. ... ... the capacity ... the peripheral equipment receiving information the output ... a computer varies very much. 5. A factor ... advantage ... the computer compared ... a human being is ... 10,000 ... 1.

VI. Complete the following sentences:

1. The peripheral equipment is slow as compared with ... . 2. Devices called buffers may be used for efficient use of the computer's ... . 3. A human being is known to type at the rate ... . 4. The ratio between a computer speed and the output speed of a human being gives a factor of ... to

a computer. 5. The input unit accepts the data ... and programs for ... .

VII. Memorize the following definitions:

1. Input unit is a section of the computer which accepts information from outside the computer. 2. Output is device or devices which put out information of the computer. 3. Buffers are storage devices accepting information at a very high speed from the computer and releasing information at the proper speed for the peripheral equipment.

VIII. Answer the following questions:

1. What is the general purpose of the input unit? 2. How may the 1's and 0's be expressed for the computer? 3. What is the general purpose of the output unit? 4. What does the peripheral equipment consist of? 5. What is the general purpose of a buffer? 6. What is the ratio between a computer's speed and the top output speed of a human being? 7. How are input and output devices usually called?

IX. Form adverbs from adjectives by adding the suffix '-ly':

easy; reasonable; usual; special; physical; functional; real; regular; magnetical; different; logical; mathematical; subsequent; consequent

X. Form nouns from verbs by adding the suffixes '-er', '-or':

to work; to invent; to compose; to calculate; to operate; to act; to react; to receive; to transmit; to select; to use; to combine

XI. Arrange (a) synonyms and (b) antonyms in pairs and translate them:

a) speed; peripheral; to control; to write; auxiliary; to do; to receive; rate; to record; to get; to make; to handle; device; unit; instruction; part; to accept; command; section; information; data; to take in

b) to add; presence; hole; input; north; decimal; to multiply; to divide; binary; south; output; blank; absence; to subtract

XII. Translate the following sentences paying attention to the Infinitive:

1. The assistant came to instruct students how to handle instructions. 2. The assistant came to be instructed by the professor. 3. The main purpose of the computers is to solve complex problems. 4. To perform reasonable operations a computer must have a way of accepting data. 5. To add and

to subtract means to perform mathematical operations. 6. P.L. Chebyshev, a Russian scientist, was the first to construct an arithmometer. 7. The input unit to be described here is a new device. 8. Punched holes in a card or a paper tape are used to represent 1's and 0's. 9. To carry out the instruction, the computer must accept the data in the form of punched holes and blanks. 10. In order to program in a good way, the programmer needs detailed data about the program and the way it is to be done.

XIII. State the functions of the Infinitive and translate the sentences:

a) 1. To do the program the programmer must have a good understanding of the problem for the computer. 2. To do the program for a computer is the main duty of a programmer. 3. The programmer must do a program to give accurate instructions to the computer. 4. Electric typewriters are very slow and are used only by operators to communicate with a computer. 5. To make possible communication from a human being and a computer is the main purpose of the input unit.

b) 1. The experiments to be carried out will be very important. 2. M. V. Lomonosov was the first to receive the higher education among peasants in Russia. 3. Information to be computed is stored usually in registers—units of hardware. 4. The machine to operate with the keys is named an ordinary adding machine. 5. A sequence of reasonable operations to be performed will be done by computer "M-220". 6. The programmer to do the program for a computer must have a good knowledge of mathematics.

XIV. Translate the sentences paying attention to the Subjective Infinitive Construction:

1. The input and output units are known to be the parts of a computer. 2. The human being seems not to be able to add or to multiply without using auxiliary devices such as pencil and paper. 3. Historically, linear programming proved to be especially effective in analyzing industrial processes. 4. This type of the output unit is said to use a punched paper tape. 5. Devices for accepting information are said to have been described in some magazines. 6. Automated Management Systems are known to have appeared quite recently. 7. Our programmers are known to be studying the theory of programming. 8. In ancient ['eɪnʃ(ə)nt] (древний) times the sun was thought to be revolving round the Earth. 9. Y. A. Gagarin is known to be the first cosmonaut who

made an orbital flight around the Earth. 10. The French mathematician Pascal is known to construct the first mechanical computer.

XV. Translate the following sentences with the emphatic construction 'It is (was) . . . that (who, which, when) . . .':

1. It is electrical engineering that deals with (иметь дело с) all kinds of electrical instruments and devices. 2. It was the invention of the computer that revolutionized the thinking process. 3. It was in 1944 when the first automatic computer appeared and began to operate. 4. It is an input unit that accepts information from outside the computer. 5. It was B. Pascal who invented the first mechanical computer. 6. It is the Soviet Union which launched the first man into the outer space. 7. It is in industrial technology and scientific development that electronic devices contributed greatly. 8. It was Norbert Wiener who was considered to be the father of cybernetics.

XVI. Find the equivalents to the following Russian sentence:

Именно П. Л. Чебышев в России в 1882 году изобрел первый арифмометр, выполняющий автоматически умножение и деление.

1. It was in 1882 when P. L. Chebyshev invented the first arithmometer performing automatically multiplication and division. 2. It was in Russia where P. L. Chebyshev invented the first arithmometer performing automatically multiplication and division. 3. It was P. L. Chebyshev who invented in Russia in 1882 the first arithmometer performing automatically multiplication and division. 4. It was the first arithmometer performing automatically multiplication and division that was invented in Russia by P. L. Chebyshev.

XVII. Read Text B and translate it without a dictionary. Express its contents in Russian:

TEXT B. TERMS

The special terms of any subject are the keys to understanding it. The special terms of the field of computers are accordingly the keys to understand this field.

Among (среди) the many special terms in any field of knowledge there are two kinds: those that are essential (существенный), that convey (передавать) the key ideas of the subject to a person interested in understanding it; and those that are helpful but not essential. An example of the

first kind of terms in the computer field is "binary notation" (двоичное исчисление); it would be very hard to understand much [1] in the field of computers without knowing the meaning of "binary notation". An example of the second kind is "minimum latency programming";[2] for many purposes it is not necessary to know exactly (точно) what this term means.

Notes

[1] **it would be very hard to understand much** — было бы очень трудно понять многое
[2] **minimum latency programming** — минимальное скрытое программирование

XVIII. Read Text B′ and translate it without a dictionary. Retell its main contents in Russian:

TEXT B′. RELIABILITY

The first automatic computers of the 1940's were not reliable. The equipment of which they were made had not been accurate and reliable. The programmer for the problem usually had to program the check (проверка) by doing the same operation in another way. For example, in multiplication A times B he used the equipment differently from B times A, and so both (обе) operations might be programmed, and then the computer was given an instruction to compare the results. If a difference in results was more than a tolerance (допуск), the machine was stopped, and the operator in charge of (ответственный за) the computer and the mathematician in charge of the program consulted on how to get rid (избавиться) of the error (ошибка).

Those days have long since gone. Now computers can operate with extraordinary (чрезвычайный) reliability, with as many as a billion or ten billions operations between errors. Automatic checking of different kinds is built into the machine.

XIX. Listen to Text C from the tape recorder. Give its contents in short (in English).

Notes

[1] **The Unified System (ES)** — Единая Система (ЕС)
[2] **variety** — множество, многообразие
[3] **program-compatible models** — программно-совместимые модели
[4] **a wide range** — широкий диапазон (ряд)
[5] **application** — применение
[6] **to design** — конструировать, проектировать
[7] **the Commonwealth** — Содружество

TEXT C. THE ES ELECTRONIC COMPUTERS

The Unified System [1] of electronic computers is represented by a variety [2] of program-compatible models [3] intended for a wide range [4] of scientific, technical and economic problems, for application [5] in computing centres and automated management systems. This system is designed [6] and produced by the Commonwealth [7] of the Socialist countries. Nowadays the models ES 1033, 1040, 1045, 1055, 1065, etc. are widely used all over the world.

LESSON FOUR

Grammar. The Participle. The Absolute Participle Constructions.
Text A. Memory or Storage Unit.
Text B. The Floppy Disks.
Text B'. Bubble Memory.
Text C. Chips.

Exercises

I. Read the international words and guess their meaning:

final; base; container; register; object; subject; transmission; million; billion; disk; serial; series; track; resistor; transistor; millisecond; nanosecond; reaction; concentric; integration; technology; bipolar; diode; diameter; result; correct; to generate; element

II. Pronounce the following words correctly:

storage ['stɔ:rɪdʒ] *n* запоминающее устройство (ЗУ); память
intermediate [ˌɪntə(:)'mɪdjət] *a* промежуточный
final ['faɪnl] *a* окончательный; конечный
generate ['dʒenəreɪt] *v* генерировать; (вос)производить
access ['ækses] *n* обращение *(к памяти)*; доступ; выборка *(из памяти)*
require [rɪ'kwaɪə] *v* требовать
transmit [trænz'mɪt] *v* передавать
use [ju:s] *n* использование; [ju:z] *v* использовать
quantity ['kwɔntɪtɪ] *n* количество; величина
hold [hould] *v* держать; удерживать; проводить
external [eks'tə:nl] *a* внешний, наружный
similar ['sɪmɪlə] *a* подобный, похожий
circle ['sə:kl] *n* круг; окружность
internal [ɪn'tə:nl] *a* внутренний
main [meɪn] *a* главный, основной
core [kɔ:] *n* сердечник; память на магнитных сердечниках
inch [ɪntʃ] *n* дюйм (2,5 см)
slow [slou] *a* медленный
fast [fɑ:st] *a* быстрый, скорый
circuit ['sə:kɪt] *n* схема; цепь; контур
integer ['ɪntɪdʒə] *n* целое число
differ ['dɪfə] *v* отличаться, различаться
development [dɪ'veləpmənt] *n* разработка; развитие
semiconductor ['semɪkən'dʌktə] *n* полупроводник
creation [krɪ(:)'eɪʃ(ə)n] *n* создание
density ['densɪtɪ] *n* плотность; концентрация
allocation [ˌælə'keɪʃ(ə)n] *n* размещение; распределение
chip [tʃɪp] *n* чип; кристалл

achievement [ə'tʃi:vmənt] *n* достижение
layer ['le(ɪ)ə] *n* слой
microcomputer ['maikroukəm'pju:tə] *n* микрокомпьютер
primary ['praiməri] *a* первичный, первоначальный
permanent ['pə:mənənt] *a* постоянный
programmable ['prougræməbl] *a* программируемый
erasable [ɪ'reɪzəbl] *a* стираемый
firmware ['fə:mwɛə] *n* программно-аппаратное обеспечение, встроенное в компьютер фирмой

III. Memorize the following word combinations:

during the course of a computation — во время вычисления
an access time — время обращения (к памяти)
a floppy disk — гибкий диск
a series of concentric circles — ряд концентрических окружностей
a read/write head — считывающая — записывающая головка
internal (main) memory — оперативная память (ОЗУ)
a film memory device — ЗУ на тонких пленках
general-purpose registers — регистры общего назначения
floating-point registers — регистры с плавающей точкой
control registers — регистры управления
either ... or ... — или ..., или; либо ..., либо
both ... and ... — как ..., так и ...; и ..., и ...
in the latter — в последнем (из двух упомянутых)
by spraying layers — путем напыления слоев

TEXT A. MEMORY OR STORAGE UNIT

1. The part of a digital computer which stores information is called storage or memory. The computer's memory stores the numbers to be operated on; it stores intermediate results that are generated during the course of a computation; and it stores the final results. The instructions themselves are also stored in the computer's memory.

2. There are two important factors about the memory unit: an access time and a capacity. The time required to transmit one computer word out of the memory to where it will be used is called the memory access time; it usually amounts to a few millionths of a second or less in modern fast computers. The speed of modern computers is the speed of access to their memories. The capacity of a computer is the quantity of data that its memory unit can hold.

3. There are many ways of memorizing information in memory cells of a digital computer. External memory or storage units may use magnetic tapes, magnetic drums, magnetic disks and floppy disks. The magnetic drum and

magnetic disk are called a Direct Access, or Random Access, Storage Device (DASD).

4. The magnetic disk is very similar to the magnetic drum but is based upon the use of a flat disk with a series of concentric circles of magnetizable material, one read/write head being for each concentric circle, i.e., for each track. Memory units on magnetic disks may store more than 100,000,000 bytes. The magnetic disk is illustrated in Figure 1.

5. Internal or main memory units were constructed of magnetic cores about 8 hundredths of an inch in diameter,

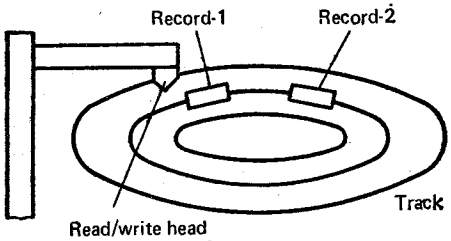


Figure 1. Magnetic Disk

each core storing one 'yes' or 'no', that is, each core representing one bit of information.

6. Information that is stored inside a computer is stored in registers, electronic units of hardware in which the positioning of physical objects stores information. Each register holds one machine word consisting usually of 32 bits or 4 bytes. Registers hold information temporarily during processing. The slower models of registers use magnetic cores; the faster models use special electronic circuits or film memory devices.

7. Usually the registers are of three types:

1. **General-Purpose Registers** are sixteen registers, each being able to contain one word. These registers are used for storing the integer operands taking part in binary arithmetic operations.

2. **Floating-Point Registers** are four registers, each being able to contain a doubleword. These registers hold the operands taking part in arithmetic operations on floating-point numbers.

3. **Control Registers** form a group of registers differing from one model to another.

8. The development of semiconductor integration technology has led to creation of memories on LSI circuits. For

constructing memory units on LSI circuits either bipolar or MOS memory are used. The access time of bipolar memory is about 100 nanoseconds, while the access time of MOS memory is 500 nanoseconds. But on the other hand the density of memorizing elements allocation in the latter is very high and amounts to 4 thousand memorizing elements for one chip. The latest achievements of modern electronics is creation of memories on electronic circuits made by spraying layers of different memorizing materials.

9. Nowadays the main memory RAM which is regularly used in microcomputers can accept new instructions or information from a peripheral device. Terms synonymous with the computer's working memory RAM are: core, core storage, main memory, main storage, primary storage, read/write memory. Other memories, such as ROM or PROM, which are used in microcomputers as well, store instructions or information permanently. ROM, PROM, EPROM, and EEPROM are all together called firmware which is 'hard' software.

Exercises

IV. Find the equivalents:

1. the numbers to be operated on	1. числа с плавающей точкой
2. a magnetic core	2. самые последние достижения
3. an access time	3. напылением слоев
4. a computer's memory	4. во время обработки
5. intermediate results	5. крупномасштабная интеграция
6. Large Scale Integration	6. магнитный сердечник
7. during processing	7. память компьютера
8. by spraying layers	8. промежуточные результаты
9. the latest achievements	9. время обращения
10. floating-point numbers	10. числа, которые будут обрабатываться

V. Memorize the following definitions:

1. Memory unit is a part of a computer which stores information. 2. The memory access time is the time required to transmit one computer word out of the memory to where it will be used. 3. The capacity of a computer memory is the quantity of data that the memory unit can hold.

VI. Answer the following questions:

1. What is the general purpose of the memory or storage unit? 2. What information is stored in the computer memory? 3. Where is information stored inside the computer? 4. What is the memory access time? 5. What ways of memorizing words in a computer do you know? 6. What has led to creation of memories on LSI circuits? 7. What is the latest achievements of modern electronics? 8. What memories are used in microcomputers?

VII. What do you call a unit which:

1) accepts information from outside a computer? 2) memorizes information to be operated on? 3) brings information out of the computer? 4) is able to take in information at the very high speed from a computer and then release it at the proper speed for the peripheral equipment?

VIII. Translate the following sentences paying attention to the words in bold type:

a) 1. Every student must know **that** reasonable operations are logical and mathematical operations. 2. His experiment is simpler than **that** of yours. 3. The capabilities of a digital computer are greater than **those** of an analog computer. 4. It is clear **that** these diagrams are like **those** shown in Fig. 4.

b) 1. **As** science progresses the difference between man-made systems and natural systems may be reduced infinitely (бесконечно). 2. **As** you know information is a set of marks that have meaning. 3. **As** new operations can be composed of sub-operations, no new programming is needed. 4. The development of computers **as** machines for handling information has gone a long way. 5. **As** is known, program is a set of instructions.

c) 1. **One** of these problems has been solved by a computer. 2. By means of a computer **one** can easily solve any problem. 3. There is only **one** solution, the **one** stated above, when n and m are equal. 4. **One** should remember all these rules, while solving a problem.

IX. Translate the following sentences paying attention to the form and function of the Participle:

1. The input unit consists of some devices using different means. 2. Performing addition the computer must have two numbers to be added. 3. When pressing the keys the operator makes the adding machine operate. 4. The operator pressing the keys makes the adding machine operate. 5. A

device invented by the German mathematician Leibnitz could control automatically the amount of adding to be performed by a given digit. 6. Logical operations performed by a computer are comparing, selecting, sorting, and determining. 7. Discussing the advantages of the new memory unit the professor gave the students all the necessary explanations. 8. Having punched holes in a card the operator put it into the computer. 9. When passed through the reading equipment the characters are read in a way similar to a way used for a magnetic tape. 10. The density of memorizing elements in MOS memory is very high.

X. Translate the following sentences paying attention to the Absolute Participle Construction:

1. The first automatic computers of the 1940's not being very reliable, scientists went on improving them. 2. Specialists use computers widely, the latter helping in performing computations at great speeds. 3. Personal computers being used for many purposes, scientists go on improving their characteristics. 4. The computer SM-100 is used in industrial processes and scientific researches, its main function being to carry out reasonable operations with numbers and to calculate complex problems. 5. With the current on, the computer automatically begins operating. 6. A printer's line is usually between 60 and 150 characters long, with 120 characters being a common length.

XI. Find the sentences in which the Absolute Participle Construction is used:

1. Register R storing the code for number 3, the computer will perform the operation 3 times. 2. When storing the code for number 3 register R makes the computer perform the operation 3 times. 3. The computer performs the operation 3 times, the code for number 3 being stored in one of the registers. 4. The register storing the code for number 3 is arranged in the computer's memory. 5. The code for number 3 being stored in register R, the operation will be performed by the computer 3 times.

XII. Read Text B without a dictionary. Express its contents by 3-4 sentences:

TEXT B. THE FLOPPY DISKS

Floppy disks are a magnetic storage medium (среда) which can be recorded, erased and used over and over again. Floppy disks are flexible (гибкий) plastic disks which have

several standard sizes. Full size floppy disks are 8 inches in diameter; minifloppy disks are 5 inches in diameter. Both full size floppy disks and minifloppy disks are housed in a paper-like plastic envelope, usually black, and remain (оставаться) in the envelope at all times.

Microfloppy disks are in sizes from 3 to 4 inches. They are housed in a rigid (жесткий) plastic shell (оболочка) of different design. The number of bytes that can be recorded on a floppy disk are about 80,000 to 1,000,000 and more. Floppy disks are used extensively in personal computers, small business computers, word processing, etc. All kinds of floppy disks are illustrated in Fig. 2.

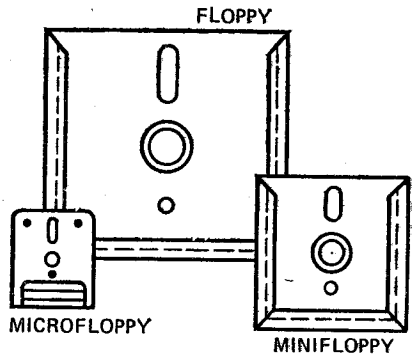


Figure 2. Floppy Disks

XIII. Read Text B′ and translate it without a dictionary. Retell its main contents in Russian and in English:

TEXT B′. BUBBLE MEMORY

Bubble memory [1] is a storage for programs and information. It is a storage technology which combines both semiconductor and magnetic recording techniques [2] to create a solid state [3] storage device. Bubble memory is unique, because it is a disk that doesn't spin.[4] The bits on the surface spin around the disk instead. Bubble memory units are only two square inches in size, and contain a thin film recording layer. The bits, called bubbles of their globular shape,[5] are electromagnetically generated in circular strings [6] inside this layer. In order to read or write the bubbles, the strings of bubbles are made to rotate past the equivalent (the string of bubbles) of a read/write head in a disk.

Bubble memory holds its contents without power, like disk and tape. It is considerably [7] faster than floppy disks and many hard disks. It is often used in portable terminals [8] and computers instead of disks.

Notes

[1] **bubble memory** — память на магнитных доменах (пузырьковая память)
[2] **techniques** [tek'ni:ks] — технические приемы
[3] **solid state** — полупроводниковый
[4] **to spin** — вращаться
[5] **their globular shape** — их округлая форма
[6] **string** — цепочка; последовательность
[7] **considerably** — значительно
[8] **portable terminals** — портативные терминалы

XIV. Listen to Text C from the tape recorder. Give its contents in short (in Russian).

Notes

[1] **electronic components** — электронные элементы
[2] **integrated circuits** — интегральные схемы
[3] **single** — одиночный
[4] **entire** — полный, целый

TEXT C. CHIPS

Chips are miniaturized electronic circuits. A chip contains several hundred thousand electronic components [1] (transistors, resistors, etc.). The terms synonymous to chip are integrated circuits [2] and microelectronics. There are some types of chips. **Logic chip** is a single [3] chip which can perform some or all the functions of a processor. A microprocessor is an entire [4] processor on a single chip. One or more microprocessors are used in a portable or desk computer. Larger computers may use several types of microprocessors. **Memory chips** contain from several hundred to hundreds of thousands of bits (storage cells). **RAM chips** are the computer's primary internal working storage and require constant power to keep their bits. Chips, such as ROMs, PROMs, EPROMs and EEPROMs are permanent memory chips and hold their contents without power.

LESSON FIVE

Grammar. Revision of Non-Finite Forms of the Verb.
Text A. Central Processing Unit.
Text B. Generations of Computer Systems.
Dialogue. Telephone Talk.

Exercises

I. Read the international words and guess their meaning:

nerve; system; to coordinate; to control; activity; central; processor; separate; section; role; to discuss; function; to interpret; actual; interpretation; signal; to decode; generator; automatically; accumulator; argument; decoder

II. Pronounce the following words correctly:

activity [æk'tıvıtı] *n* деятельность
apply [ə'plaı] *v* применять; прилагать, прикладывать
execute ['eksıkju:t] *v* выполнять; осуществлять
carry out ['kærı'aut] *v* выполнять; проводить
process ['prouses] *n* процесс; *v* обрабатывать
load [loud] *n* нагрузка; загрузка; ввод; *v* загружать
convenient [kən'vi:njənt] *a* удобный
consider [kən'sıdə] *v* считать; полагать; рассматривать
previous ['pri:vjəs] *a* предыдущий; предшествующий
obtain [əb'teın] *v* получать; достигать
accomplish [ə'kʌmplıʃ] *v* выполнять; совершать
separate ['sepəreıt] *v* разделять; отделять
sense [sens] *v* считывать; воспринимать
choose [tʃu:z] *v* выбирать
cause [kɔ:z] *v* заставлять; причинять; вызывать
sequentially [sı'kwenʃəlı] *adv* последовательно
design [dı'zaın] *n* проект, конструкция; *v* проектировать; конструировать
happen ['hæp(ə)n] *v* случаться
transfer [træns'fə:] *v* передавать; переносить; выполнять команду перехода
hence [hens] *adv* следовательно
involve [ın'vɔlv] *v* включать в себя; вовлекать
argument ['a:gjumənt] *n* аргумент
accumulator [ə'kju:mjuleıtə] *n* накапливающий сумматор
refer [rı'fə:] *v* (to) относить(ся); ссылаться (на)
facilitate [fə'sılıteıt] *v* облегчать; способствовать
contents ['kɔntənts] *n pl* содержимое; содержание
relate [rı'leıt] *v* относиться; иметь отношение
record ['rekɔ:d] *n* запись; регистрация; [rı'kɔ:d] *v* записывать

III. Memorize the following word combinations:

a central processor — центральный процессор
a control unit — блок управления
a control generator — генератор управления
an instruction decoder — дешифратор команд
an instruction register — регистр команд
a current-address register — регистр текущего адреса
an arithmetic and logic unit — арифметическое/логическое устройство
at the proper time — в надлежащее время
control signals — сигналы управления
in its turn — в свою очередь
on the basis of — на основе
at a time — за одно время (одновременно)

what to do next — что делать дальше
in this way — таким образом
while an instruction is being executed — во время выполнения команды

TEXT A. CENTRAL PROCESSING UNIT

1. The central processing unit (CPU) or central processor is the nerve centre of any digital computer system, since it coordinates and controls the activities of all the other units and performs all the arithmetic and logic processes to be applied to data. All program instructions to be executed must be held within the CPU, and all the data to be processed must be loaded first into this unit. It is convenient to consider the central processor to have three separate hardware sections: an internal or main memory, an arithmetic and logic unit, and a control unit. The role of the internal memory was discussed more detailed in the previous lesson.

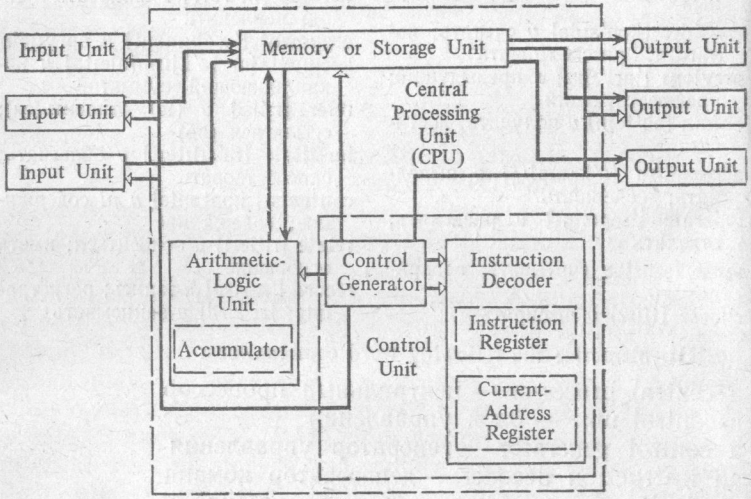


Figure 3. The Simplest Block Diagram of a Computer

Now let us discuss the latter two sections. (All the sections of the CPU are shown in Fig. 3.)

2. The CPU has two functions: it must (1) obtain instructions from the memory and interpret them, as well as (2) perform the actual operations. The first function is executed by the control unit. This unit in its turn must perform two functions: it must (1) interpret the instruction; then, on the

basis of this interpretation (2) tell the arithmetic and logic unit what to do next. The latter function is accomplished through the use of electronic signals. According to these two functions we can separate the part of the control unit that interprets or decodes the instruction called the instruction decoder from the part that generates the control signals called the control generator.

3. An instruction having been transmitted to the instruction decoder, where it is interpreted, the control generator senses this interpretation and then produces signals that tell the arithmetic unit which operation to perform. It also generates signals that choose the proper numbers from the memory and sends them to the arithmetic and logic unit at the proper time; and when operation has been performed, other control signals take the result from the arithmetic and logic unit back to the internal memory. After an instruction has been executed, the control generator produces signals that cause the next instruction to go from the memory to the instruction decoder. In this way the instructions are performed sequentially.

4. The second function of the CPU is performed by the arithmetic and logic unit which does the actual operations. This unit is capable of performing automatically addition, subtraction, multiplication, division, comparing, selecting, and other mathematical and logical operations. Consider now what happens in the arithmetic and logic unit while an instruction is being executed. In most computers only one word at a time can be transferred between the arithmetic/logic unit and the memory. Hence, to perform an operation involving two arguments, the first argument must be transferred from the memory to the arithmetic/logic unit and stored there temporarily while the second argument is being transferred. The special memory cell in the arithmetic/logic unit for this purpose is called the accumulator. The operation being performed, the result is formed in the accumulator before it is transmitted back to memory.

5. Next consider the instruction decoder that interprets instruction. In order that the instruction decoder perform its function, it must constantly refer to the instruction being interpreted during the time control signals are being set up.[1] To facilitate this, while an instruction is being executed it is stored in a special memory cell called the instruction register, located in the instruction decoder.

6. There is another memory cell located in the instruction decoder called the current-address register. The contents

of this register is always the memory address from which the instruction being executed came. The reason for this is related to the fact [2] that the address of the present instruction was given as part of the previous instruction.

For the computer designer to understand the work of the CPU is quite necessary.

Notes

[1] **during the time control signals are being set up** — во время формирования сигналов управления

[2] **The reason for this is related to the fact** — Причина этого заключена в том

Exercises

IV. Find in (b) the Russian equivalents to the English words and word combinations in (a):

a) hence; for example; according to; by means of; i.e.; etc.; always; just; on the other hand; since; any; in its turn; sometimes; the same; while; in this way; instead (of); usually

b) то есть; например; всегда; только что; с другой стороны; так как; в свою очередь; иногда; тот же самый; любой; следовательно; и так далее; в то время как; таким образом; вместо; согласно; обычно; посредством

V. Arrange synonyms in pairs:

semiconductor technology; to execute; to write; to control; memory; to sense; to choose; to form; to feel; storage; to store; to set up; to handle; solid-state technology; to perform; to keep; to select; research; to put in; investigation

VI. Complete the following sentences:

1. The arithmetic/logic unit is capable of ... . 2. The access time is the time required for transmitting one computer ... out of the ... to where it ... . 3. The actual computations are executed in a central ... . 4. The part of the control that interprets the instruction is called ... . 5. The part of the control that generates the control signals is called ... . 6. The control signals choose the proper numbers from ... and send them to ... at the proper time.

VII. Memorize the following definitions:

1. The CPU is the nerve centre of any computer since it coordinates and controls the activities of all the other units. 2. The arithmetic/logic unit is that part of the CPU in which

the actual computations take place. 3. The control unit is that part of the CPU which obtains instructions from the memory, interprets them and generates the control signals. 4. The instruction decoder is the part of the control unit which interprets or decodes the instruction. 5. The control generator is the part of the control unit which generates the control signals. 6. The instruction register is the part of the instruction decoder in which the address of current instruction is stored. 7. The current-address register is the second special memory cell located in the instruction decoder. 8. The accumulator is the special memory cell in the arithmetic/logic unit in which the result is formed before it is transferred back to the memory unit.

VIII. Answer the following questions:

1. What is the general purpose and function of the CPU? 2. How many parts is the CPU composed of? 3. What is the general purpose of the control? 4. What is the arithmetic/logic unit? 5. What is the instruction decoder? 6. What is the general function of the control generator? 7. What happens in the CPU while an instruction is being executed? 8. What is the accumulator? 9. Where is the accumulator located? 10. Where are the instruction register and the current-address register located?

IX. What do you call a unit which:

1) interprets instructions? 2) senses the interpretation of instructions and produces control signals? 3) performs mathematical and logical operations? 4) chooses the proper numbers from the internal memory and sends them to the arithmetic/logic unit at the proper time? 5) obtains instructions from the main memory, interprets them and accomplishes the actual operations?

X. Compare:

a) a memory and a CPU; b) an accumulator and an instruction decoder; c) an arithmetic/logic unit and a control unit

XI. Read and translate the following sentences paying attention to the objective infinitive constructions and for-phrases with the infinitive:

1. Our engineers want the complex problems to be solved by computers. 2. In the laboratory we saw the perforator punch holes in the cards of standard size. 3. It is quite necessary for the programmer to understand the work of all units of a computer. 4. We watched the floppy disk begin

to operate. 5. The speed of the computer may be found by measuring the time which is required for it to transmit one word out of the memory to where it will be used. 6. We asked the lab assistant to show us the computer SM-2 operate. 7. There is a good reason for us to use this kind of the bubble memory in a personal computer. 8. Information has to be in the form of digits or characters for a digital computer to perform reasonable operations.

XII. Define the infinitive constructions and translate the sentences:

1. We know B. Pascal to be the first inventor of the mechanical computer. 2. B. Pascal is known to be the first inventor of the mechanical computer. 3. In the middle of the 17th century it was possible for B. Pascal to invent only the mechanical computer. 4. The possibility for the problem to be solved is illustrated by the given formula. 5. Human beings seem to be able to find facts or even logical consequence of facts in their memory according to association. 6. The magnetic recording is done on a disk which permits an information to be stored or read at one or several points on it. 7. The students were explained the high-speed memory unit to use the LSI circuits. 8. For results to be obtained an instruction has first been put into a computer.

XIII. Translate the following sentences paying attention to the ing-forms:

1. According to the principles of their work computers are subdivided into three parts: analog, digital, and hybrid. 2. When applying mathematical methods to the solving of technical problems engineers are most often interested in obtaining a finite numerical results. 3. Proper relation between theory and practice must be observed in training young specialists. 4. Mathematical tables are necessary aids (необходимые средства) for performing computational work. 5. The students get the practical training when they are working at various plants. 6. When performing the problems in mathematical practice, the students use desk calculators such as arithmometers, key computing machines and slide rules (логарифмическая линейка). 7. In modern computers LSI circuits and RAM/ROM memories are used for executing sophisticated (усложненный) operations. 8. A memory unit is used for storing information. 9. Electronics being used not only in industry but in many other fields of human activity as well, one should have an idea of what it is. 10. The processing of messages can be performed sequentially, i.e., a new task is not given until the current one is completed.

11. Having conducted many experiments scientists proved that electricity had an atomic character. 12. The fast electronic machines, such as microcomputers, are effective for carrying out complicated (сложный) computations.

XIV. Read Text B and translate it with a dictionary (see the List of Acronyms and Abbreviations as well). Write a short summary:

TEXT B. GENERATIONS OF COMPUTER SYSTEMS

The 1st-generation computer systems (beginning approximately in 1950) employed vacuum tubes as the primary switching component [1] in the processor. Memories were constructed of liquid mercury delay lines [2] or magnetic drums (BESM-1, MINSK-1, IBM 650 are examples).

The 2nd-generation systems began around the late 1950's and used transistors in place of vacuum tubes and memories were made of magnetizable cores (BESM-4 and 6, MINSK-32, M-220, IBM 1401 are examples). Size was reduced and reliability was improved significantly in the 2nd-generation systems. The 2nd generation was primarily a batch processing environment [3] with a single program running at one time.

The 3rd-generation computers, beginning in the mid 1960s introduced processors made of integrated circuits (ES-1033, -1045, M-6000, NAIRI-3, SM-3, IBM-360 and 370 are examples). The 3rd generation also introduced system software technologies like Operating Systems [4] and Data Base Management Systems. On-line systems [5] were widely developed throughout the 3rd generation, although most processing was still batch oriented.

The 4th generation is more evolutionary than revolutionary. Starting in the mid 1970s, the 4th-generation computer logic and memory were built almost entirely of chips which contain extremely large numbers of electronic components. The 4th generation embraces extensive integration of small and large computers together in a distributed processing environment.[6] (The examples of the 4G[7] computers in our country are ES-1065, NAIRI-4, ELEKTRONIKA-60, ISKRA-1256, etc.)

The 5th generation ought to become formalized by 1990s. VLSI and SLSI technologies will put mainframes [8] on everyone's desk. Optical fibers,[9] videodisks and technologies now in the research labs, will be used to construct the 5th-generation computer systems. Artificial intelligence [10] techniques will be incorporated into every type of application. By the turn of the century, a computer should be able to converse rather intelligently with a person.

Notes

[1] **as the primary switching component** — в качестве первичного переключающего элемента
[2] **liquid mercury delay lines** — линии задержки на жидкой ртути
[3] **a batch processing environment** — функциональные средства пакетной обработки
[4] **Operating Systems** — операционные системы
[5] **on-line systems** — системы, работающие от ЦПУ
[6] **a distributed processing environment** — функциональные средства распределенной обработки
[7] **the 4G=the 4th generation** — четвертое поколение
[8] **mainframes** — *эд.* большие компьютеры
[9] **optical fibers** — оптические волокна
[10] **artificial intelligence** — искусственный интеллект

XV. Listen to the following dialogue from the magnetic tape. Try to reproduce it.

Notes

[1] **to be anxious** ['æŋkʃəs] — беспокоиться
[2] **'flu = influenza** — грипп
[3] **a headache** — головная боль
[4] **to feel upset** — расстраиваться
[5] **to miss** — пропускать (занятия)
[6] **to catch the idea** — понять
[7] **by the way** — между прочим
[8] **I need it badly** — она мне очень нужна
[9] **to keep (kept)** — держать
[10] **to borrow (from the library)** — брать (из библиотеки)
[11] **a copy** — экземпляр
[12] **Give my best regards** — Передай мой сердечный привет

TELEPHONE TALK

A: Hallo!
B: Is that you, Alec? This is Boris speaking. Why didn't you attend lectures? The whole group's anxious [1] about you, you know.
A: Oh, I'm not very well and I have to stay at home.
B: What's the matter with you?
A: It's the 'flu [2]. My temperature's 38. I've a bad headache.[3] I feel so upset[4] —in some days we'll have the exam in computers.
B: What does the doctor say?
A: He says I must stay in bed for three or four days and take some medicine.
B: Can I do anything for you?
A: Please, bring me your notes. I missed [5] some lectures and now I can't catch the idea [6] of what difference between bipolar and bubble memory is, and what random

access memory and read-only memory are ... and what chips are...

B: All right. By the way [7] I've got a textbook "Microcomputer Design". I'll bring it if you like.

A: Thank you, I need it badly.[8] How long may I keep [9] it?

B: Keep it as long as you like. I can borrow [10] another copy [11] from the library. Good-bye. See you in the evening.

A: Good-bye. Give my best regards [12] to the group, please.

LESSON SIX

Grammar. Revision of the Non-Finite Forms of the Verb.
Text A. Microcomputer and Microprocessor Design.
Text B. Soviet Supercomputer.
Text C. RAM and ROM.

Exercises

I. Read the international words and guess their meaning:

electronics; microcomputer; microprocessor; chip; function; system; application; intelligent; evolution; revolution; combination; personal; maximum; minimum; individual; industrial; product; limited; attraction; type; component; compact; to orient

II. Pronounce the following words correctly:

invent [ɪn'vent] *v* изобретать
single ['sɪŋgl] *a* один; единый
tiny ['taɪnɪ] *a* крошечный
application [ˌæplɪ'keɪʃ(ə)n] *n* применение; использование
sophisticated [sə'fɪstɪkeɪtɪd] *a* усложненный; тонкий *(о приборе и т. п.)*
occur [ə'kə:] *v* случаться, происходить
inexpensive [ˌɪnɪks'pensɪv] *a* недорогой, дешевый
implement ['ɪmplɪmənt] *v* выполнять, осуществлять
significantly [sɪg'nɪfɪkəntlɪ] *adv* значительно; существенно
reduce [rɪ'dju:s] *v* уменьшать; сокращать
cost [kɔst] *n* стоимость, цена
support [sə'pɔ:t] *v* обеспечивать
user ['ju:zə] *n* пользователь; абонент
increase [ɪn'kri:s] *v* увеличивать
supply [sə'plaɪ] *n* источник; питание; подача
complete [kəm'pli:t] *a* полный; *v* заканчивать, завершать
mainframe ['meɪnfreɪm] *n* центральный блок обработки данных
consumer [kən'sju:mə] *n* потребитель
ability [ə'bɪlɪtɪ] *n* способность; умение
fetch [fetʃ] *v* вызывать, выбирать *(данные из памяти)*
counter ['kauntə] *n* счетчик
increment ['ɪnkrɪmənt] *n* приращение, инкремент; *v* увеличивать(ся)
capability [ˌkeɪpə'bɪlɪtɪ] *n* способность; *pl* возможности
reside [rɪ'zaɪd] *v* размещать(ся) в памяти
appear [ə'pɪə] *v* появляться
appliance [ə'plaɪəns] *n* устройство, приспособление; прибор
attraction [ə'trækʃ(ə)n] *n* привлекательность; притяжение

retain [rɪ'teɪn] *v* удерживать; сохранять

allow [ə'laʊ] *v* позволять, разрешать

variety [və'raɪətɪ] *n* разнообразие; **(of)** ряд, множество

junction ['dʒʌŋkʃən] *n* переход; плоскостной транзистор

available [ə'veɪləbl] *a* доступный; имеющийся в наличии; пригодный

conventional [kən'venʃənl] *a* обычный; общепринятый

III. Memorize the following words and word combinations:

since — так как; с тех пор, как; когда; с; со
large (small) scale — крупномасштабный, большой
assembly line — линия сборки
a personal computer — персональный компьютер
a user terminal — терминал пользователя
64 KB of memory — память объемом в 64 килобайта
power supply — электропитание; подача энергии
in addition to — кроме того, дополнительно к
just as so ... — точно так же как ..., так и ...
moreover — более того

TEXT A. MICROCOMPUTER AND MICROPROCESSOR DESIGN

1. Since the transistor was invented, no single electronics innovation [1] has had such an impact [2] as microcomputer. Powered by tiny semiconductor chips [3] containing computing elements with the same power and functions previously found only in large scale computers, these microcomputers systems are now being applied to literally [4] thousands of applications. Microcomputers are automating assembly lines, providing the heart of sophisticated electronic games, making "intelligent" [5] computer peripherals even faster. This revolution is occurring because microcomputers are very inexpensive, easy to implement into a system, and significantly reduce the time and cost of product development.[6]

2. The microcomputers were the first computers to use a single microprocessor chip as the processor. Personal computers and small business computers are microcomputers. 8-bit microcomputers usually support only one user terminal and have a maximum of 64 KB of memory. 16-bit microcomputers may support several user terminals and usually have a maximum of 1 million bytes of memory. As the power of microcomputers increases, they can be used in two ways. They can be used either as a central computer (providing processing for several user terminals), or as a more powerful single computer for an individual user (see Fig. 4).

3. A microprocessor is the tiny processor used, as mentioned above, in microcomputers. The microprocessor requires

a power supply and memory to be a complete computer. Microprocessors are also used in minicomputers, mainframes and peripheral devices, as well as in all industrial and consumer products which use a computer.

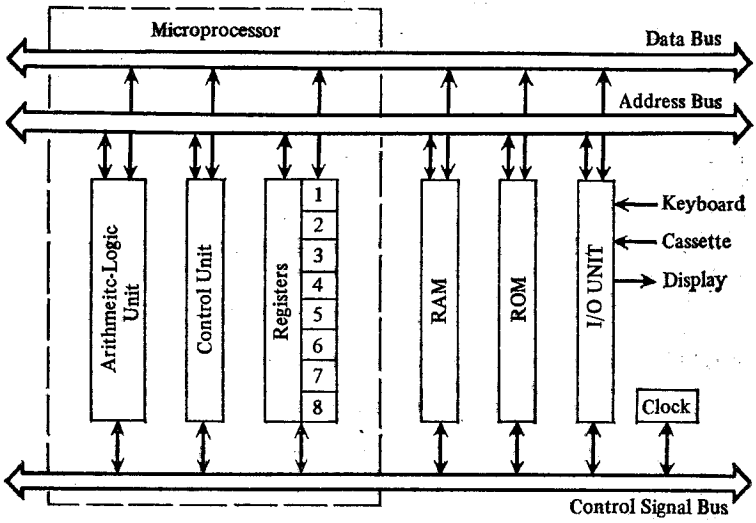


Figure 4. Block Diagram of a Typical Microcomputer with a Microprocessor

4. As usual processors all microprocessors have the ability to fetch (obtain from memory) and execute a limited set of instructions, to perform addition and subtraction on a binary word, and to input (take in) and output (put out) binary data. After fetching an operational code from the program in memory, while this operation is being executed, the program counter is automatically incremented by one unit. At the completion of the instruction, it is then ready with the address for the following program instruction. In many situations the microprocessors have many more capabilities that application requires.

5. In conclusion [7] some words more about single-chip microcomputers. They, where control functions and memory circuits reside on one chip, are beginning to appear in consumer appliances. One of their attractions is the fact that programmable read-only memories (PROMs) that can be included on these chips require no electrical power to retain data. In addition to providing permanent storage, PROMs also allow one type of micro [8] to be stocked [9] as a standard component that can be programmed individually to provide a variety of control functions.

6. Just as computer circuits have grown smaller, so too have memory circuits. Moreover, new technologies such as the bubble memory and Josephson-junction memory [10] show the ability for providing more compact memories that retain data with little or no applied power. These new memories allow micros to be used where power is not always available and conventional read-only memories do not have enough capacity.

Notes

[1] **innovation** — новшество
[2] **an impact** — *зд.* влияние
[3] **Powered by tiny semiconductor chips** — Приводимые в действие крошечными полупроводниковыми кристаллами (чипами)
[4] **literally** — буквально
[5] **"intelligent"** — «умный» (*зд.* применительно к вычислительной машине)
[6] **product development** — *зд.* выпуск продукции
[7] **in conclusion** — в заключение
[8] **micro = microcomputer** — микрокомпьютер
[9] **to stock** — запасать
[10] **Josephson-junction memory** — запоминающее устройство с переходом Джозефсона (прогрессивная технология изготовления полупроводниковых кристаллов (чипов) пересечением слоев сверхпроводящих металлов)

Exercises

IV. For the nouns in column II find suitable attributes in column I and translate:

I	II
1. user	1. computer
2. power	2. lines
3. "intelligent"	3. products
4. large-scale	4. terminal
5. permanent	5. supply
6. assembly	6. peripherals
7. consumer	7. storage
8. tiny	8. counter
9. program	9. semiconductor chips
10. computing	10. microprocessor
11. single-chip	11. element

V. Find the Russian equivalents to the following English word combinations:

1. no single innovation; 2. the ability to fetch; 3. a variety of control functions; 4. more powerful; 5. mainframes

and peripheral devices; 6. a limited set of instructions; 7. 8-bit microcomputers; 8. significantly reduce the time and cost; 9. to retain data; 10. in addition to; 11. an individual user; 12. a personal computer; 13. literally thousands of applications; 14. a standard component

1. персональный компьютер; 2. кроме того; 3. стандартный элемент; 4. удерживать данные; 5. множество функций управления; 6. ограниченный набор команд; 7. большие компьютеры и периферийные устройства; 8. способность выбирать (команду из памяти); 9. разрядные микрокомпьютеры; 10. индивидуальный пользователь; 11. более мощный; 12. значительно уменьшают время и стоимость; 13. буквально тысячи применений; 14. ни одно новшество

VI. Memorize the following definitions:

1. A microcomputer is a small scale computer using a single microprocessor chip in its architecture. 2. A microprocessor is a tiny processor on a single chip. 3. A personal computer is a computer used for home or personal use. Personal computers are microcomputers.

VII. Answer the following questions:

1. What is the reason for advent (приход) of microcomputers? 2. What component do the microcomputers use as the processor? 3. How many user terminals do 8-bit microcomputers support? 4. What is the density of memory of 16-bit microcomputers? 5. What is a microcomputer? 6. What is a microprocessor? 7. Where are microprocessors used? 8. What ability have the microprocessors? 9. What new progressive technologies are used in memory units? 10. What memories are used in microcomputers?

VIII. Translate the words of the same root. Define parts of speech:

to use—user—a use—useful—useless; to process—a process—processed—processor; inexpensive—expensive; type—typical; to invent—inventor—invention; power—powerful—powerless—powered; to conduct—conductor—semiconductor—conducted; to contain—container; to combine—combiner—combination—recombination; electron—electronic—electronics; to apply—application—applied—appliance—applicant; automatic—automatically—automatical—automaton—automated—automation—to automatize; to implement—implementation; operational—to operate—operator—operation; to complete—complete—completion; to attract—attrac-

tive—attraction; capable—capability; convention—conventional—conventionally; capacity—capacitor—capacitance

IX. Read and translate the following sentences paying attention to ing-forms:

1. The operand length, being powers of two, permit maximum efficiency in the use of binary addressing and in matching the physical word size of storage. 2. Floating-point arithmetic is designed to allow easy translation between the two formats. 3. The fraction of a floating-point number is expressed in hexadecimal digits, each consisting of four binary bits and having the values 0—15. 4. Bytes locations in storage are consecutively numbered starting with 0; each number is considered the address of corresponding byte. 5. The floating-point instruction set provides for loading, adding, subtracting, comparing, multiplying, dividing, and storing as well as the sign control of short and long operands. 6. The specific meaning of any setting is significant only to the operation setting in the condition code. 7. The process of discovering, locating and correcting errors proved to be one of main difficulty, often taking considerably longer time than writing the program in the first place and using much machine time. 8. Although programming techniques have improved greatly since the early days, the process of finding and correcting errors in programs known as "debugging" still remains a most difficult and unsatisfactory operation. 9. We are reaching the limit of the human ability to write complicated programs, and the software crisis really is the result of attempting making the human beings to write them. 10. The central processing unit contains the facilities for addressing main storage, for fetching or storing information, for arithmetic and logic processing data, for sequencing instructions in the desired order, and for initiating the communication between storage and external devices. 11. It is the compiling of a program that requires great attention of the programmer. 12. Compiling a program requires great attention of a programmer.

X. Read Text B and translate it without a dictionary. Render it in Russian:

TEXT B. SOVIET SUPERCOMPUTER

The USSR has started production of PS-2000 computers capable of performing 200 million operations per second.

Any computer performs many functions: in addition to

number processing the computer controls the whole computing process, and determines the sequence in which information is to arrive at its processors. In fact, the computer takes only one-tenth of its total operating time to do the computing. The Soviet supercomputer is a computer with computing elements free from all other functions. This becomes possible if supervision (контроль) is assigned (назначать) to special processors acting as managers (руководитель). Operating with computing circuits these processors give instructions and run the queue [1] of information to enter the processor from storage, etc.

The new computer concept attracted interest from the American Control Data Corporation (CDC),[2] and resulted in the production of the Cybers,[3] the fastest computers in the USA. It works like an assembly line with individual operations. Data have to pass through the entire length of the conveyer, no matter how many processors are there to process them. The Soviet approach (подход) differs radically from the above (указанный выше). We suggested (предлагать) a principle whereby (посредством которого) all processors respond (реагировать) to a single control command which leaves them a certain margin [4] of freedom, with the possibility to sort out their data independently. Receiving a command "command" they all start off doing similar operations, later switching over to successive (последующий) operations until the whole problem is solved.[5] The efficiency of this parallel system is obvious (очевидный).

First, similar operations can be handled at any speed as it depends on (зависеть от) the number of processors involved.

Second, a single control system for all processors is simple and, consequently, low-cost.

So far (до сих пор) there are no computers in this class in the West. It is high-speed and low-cost. It can be used in weather forecasting,[6] to predict (предсказывать) patients' post-operational states or the behaviour (поведение) of an airfoil (профиль крыла) which only exists in a blue-print (синька), and also helps to pinpoint (указывать) drilling sites [7] for oil and gas, etc. The American Cyber-73 takes nearly two hours for pinpointing drilling sites. The Soviet computer performs this operation in ten minutes.

Notes

[1] **to run the queue** [kju:] — управлять очередностью
[2] **the CDC** — фирма, выпускающая вычислительное оборудование
[3] **cyber** ['saɪbə] — кибер (от слова «кибернетика»)
[4] **a certain margin** — определенный предел
[5] **until the whole problem is solved** — до тех пор, пока задача не будет решена полностью
[6] **weather forecasting** — прогноз погоды
[7] **drilling sites** — места для бурения скважин

XI. Listen to Text C from the tape recorder. Give the contents in short (in English and in Russian).

Notes

[1] **once** [wʌns] — как только, поскольку
[2] **to fail** — иссякать, кончаться
[3] **to lose** — терять
[4] **manufacture** — изготовление
[5] **to alter** ['ɔ:ltə] — меняться

TEXT C. RAM AND ROM

Once [1] the information is stored in a computer's memory, the computer can calculate, compare and copy this information. The memories are of different kinds. RAM and ROM are examples of the newest ones.

RAM stands for Random Access Memory, because information can be transferred into and out of any single byte of memory. The storage cells, i.e., bits of a RAM chip require power to retain their contents. If the power fails,[2] the contents are lost.[3]

ROM is firmware and stands for Read Only Memory. It is permanent memory chip for program storage. Instructions and information are stored in the ROM at the time of their manufacture [4] and cannot be altered.[5] All microcomputers use RAMs and ROMs.

SECTION II. SOFTWARE

LESSON SEVEN

Text A. The Basic Principles of Programming.
Text B. Programming.

Exercises

I. Read the international words and guess their meaning:

fundamental; principle; tactics; student; format; utilization; term; routine; actual; practice; list; planning; personal; to concentrate; personnel; detective; coding; programming; symbolic; final; procedure

II. Read and translate the following sentences paying attention to the meaning of the words and word combinations given below:

a) **a number** — число, количество; номер
to number — насчитывать
a number of — ряд, множество

1. This professor's lectures on microprocessors and microprogramming are attended by a great number of students. 2. The students' body of our electrical engineering department numbers more than 1,500 students. 3. The first automatic digital computer could add numbers at the speed of 5000 additions per second. 4. The address of our Institute is the Prosveschenya Street, number 132. 5. A number of computing devices were arranged in our laboratory.

b) **mean** — средний; среднее число
means — средство
to mean — значить, означать
meaning — значение
by means of — посредством
by no means — ни в коем случае

1. The year mean temperature in our town is about +8°. 2. Electrical typewriters and keyboard devices are the common means of input into a computer. 3. This means that the ROM is the permanent memory chip for program storage. 4. By means of arranging memory registers inside hardware it is possible to store information and instructions. 5. By no means the computer can substitute (заменить) a human being in all respects (во всех отношениях). 6. The meaning of the word "means" is «средство», «состояние».

c) **term** — термин; срок; семестр
to term — называть
in terms of — с точки зрения; на языке; в терминах

1. The term "programming" means the process by which a set of instructions is produced for a computer to make it performing specified activity. 2. President of the USA is elected for a four-year term. 3. Each academic year at insti-

tutes and universities in our country consists of two terms. 4. A code can be written in terms of automatic language for then it is easy to make changes in it. 5. A code may be termed a program or a routine because they are synonyms. 6. If the language being described is called simply "the language", then the language in terms of which the description is being made is called "metalanguage".

d) **available** — доступный, имеющийся в наличии; пригодный

1. A number of different computing devices available in our Institute's lab is very great. 2. The computer ES-1045 is now available for students' use. 3. Of all the instruments available the control generator is the most suitable for producing electrical impulses.

e) **any** — любой, всякий (в повествовательных предложениях)

1. Any operation performed by a computer must be interpreted into a machine code. 2. Any sort of likeness (подобие) to a human being is simply irrational while constructing robots.

f) **rather** ['rɑːðə] — довольно
rather than — скорее чем; предпочтительнее

1. It would be rather difficult to predict the place where the moon will be at any particular time without computers. 2. The instructions deal directly with addresses rather than with the numbers themselves.

g) **instead** — вместо; взамен
instead of — вместо; взамен

1. Boolean Algebra is an algebra like ordinary one but dealing instead with classes, propositions, on-off circuit elements, etc. 2. In digital computers octal notation (восьмеричное счисление) is sometimes used instead of binary numbers.

III. Pronounce the following words correctly:

introduce [ˌɪntrə'djuːs] *v* вводить; знакомить; представлять
subroutine ['sʌbruː'tiːn] *n* подпрограмма; часть программы
entire [ɪn'taɪə] *a* весь, целый; полный
routine [ruː'tiːn] *n* (стандартная) программа
define [dɪ'faɪn] *v* определять; формулировать *(задачу)*
diagram ['daɪəgræm] *n* диаграмма; схема; *v* строить диаграмму (схему)
flow-chart ['flou'tʃɑːt] *n* блок-схема; схема потока информации

change [tʃeɪndʒ] *n* перемена; изменение; *v* изменять
assign [ə'saɪn] *v* назначать; присваивать
procedure [prə'sɨdʒə] *n* процедура; методика проведения *(опыта)*
debugging [dɪ'bʌgɪŋ] *n* наладка; отладка программы
error ['erə] *n* ошибка, погрешность
invalidate [ɪn'vælɪdeɪt] *v* выводить из строя
technique [tek'nɨk] *n* метод; методика; технический прием
shorthand ['ʃɔ:thænd] *n* стенография; сокращенная запись
evidently ['evɪd(ə)ntlɪ] *adv* очевидно
precisely [prɪ'saɪslɪ] *adv* точно
observation [ˌɔbzə(:)'veɪʃ(ə)n] *n* наблюдение; соблюдение
reemphasize ['rɪ'emfəsaɪz] *v* вновь подчеркнуть
erase [ɪ'reɪz] *v* стирать *(запись)*; разрушать *(информацию)*
remain [rɪ'meɪn] *v* оставаться
replace [rɪ'pleɪs] *v* заменять; подставлять; перемещать

IV. Memorize the following word combinations:

a computer manual — руководство к компьютеру
it should be pointed out — следует указать
commercially available computers — серийно выпускаемые компьютеры
the complete list of — полный список (перечень)
the over-all planning — общее планирование
so-called — так называемый
the actual coding — действительное кодирование
debugging the code — откладка кода
running the code on the computer — прогон кода на компьютере
a single error — единственная ошибка
the rest of the bits — оставшиеся биты
octal numbers — восьмеричные числа
the previous contents — предыдущее содержание

TEXT A. THE BASIC PRINCIPLES OF PROGRAMMING

1. **Introduction.** The purpose of this chapter is to introduce the student to the fundamental principles of coding and programming. These principles are connected with the stages of programming, the flow-charting, using the subroutines and the computer manual, etc. In order to leave students free to concentrate on these principles,[1] the four-address format, with a minimum of instruction types, is utilized. However, it should be pointed out that the four-address format is used in this chapter for pedagogical reasons only. In practice commercially available computers use only three-, two-, or one-address formats, the latter perhaps being the most common.

2. **The Terms 'Coding' and 'Programming'** are often used as synonyms. However, 'a code' is more specifically a short

list of instructions that direct the computer to perform only a part of the entire calculations, whereas the term 'program' refers to the complete list of instructions used for the problem. Hence the term 'programming' usually includes the over-all planning of the use of the computer for a particular problem as well as the writing of the instruction lists, or codes, whereas 'coding' is usually limited in meaning to the writing of the instruction lists. Sometimes a code is called a routine.

3. **Stages in Programming.** There are five stages in programming. First, the computations to be performed must be clearly and precisely defined. The over-all plan of the computations is diagramed by means of a so-called flow chart. The second stage is the actual coding. It is often best to write a code in terms of a symbolic language first, for then changes are easily made. Numbers are assigned to the symbols, and the final code is prepared. In the third stage some procedure is used to get the code into the memory of the computer. The fourth stage consists of debugging the code, i.e., detecting and correcting any errors. The fifth and final stage involves running the code on the computer and tabulating the results. In fact, it is well known that a single error in one instruction invalidates the entire code. Hence, programming is a technique requiring attention to details without loosing sight of [2] the over-all plan.

4. **Instruction Format.** Some bits of the instruction are set aside for the operation code designation [3]—i.e., they tell the instruction is 'add', 'multiply', 'divide', etc. The rest of the bits usually defines the four addresses. For the more usual operations that involve two operands, such as addition, multiplication, etc., two of the addresses are the addresses of operands. The third address tells where the result is to be put; the fourth address tells where to obtain the next instruction. So, the instruction format is the way in which the different digits are allocated to represent specific functions.

5. **Octal Shorthand.** The first important detail of coding is the fact that the actual bits in an instruction are not written out in the binary code; rather, some shorthand is written instead, i.e., the octal equivalent would be written out. In other words, two octal numbers represent the instruction, and each address would be represented by three octal numbers. Thus, if 101 011 is the binary code for the command 'add' then the instruction that says, "Add the contents of address 011 010 110 to the contents of address 011 100 101, put the

result into address 011 110 100, and take the next instruction from address 100 000 001," is written in octal notation as: operation—53, the first operand address—326, the second operand address—345, the third address—364, and the fifth address—401. In such cases it evidently facilitates matters [4] to call addresses in the memory by their octal numbers. Also, numerical quantities will be written on the code sheet in octal (i.e., they will have to be converted from decimal to octal before being written on the code sheet).

6. **The Computer Manual.** For the computer we must have a computer manual that gives the operation codes of different instructions and also defines precisely the meaning of the addresses for each instruction type. The coding manual must always be at the coder's side.[5] Two further observations must be reemphasized: first, when a word is called into the arithmetic/logic unit from the memory, it is not erased from its memory address, but remains there also; second, when a word is put into a memory address, it replaces the previous contents of this address, i.e., it erases what had been there.

(To be continued)

Notes

[1] **In order to leave students free to concentrate on these principles** — Чтобы дать возможность студентам сосредоточиться на этих принципах
[2] **without loosing sight of** — не теряя из виду
[3] **are set aside for the operation code designation** — откладываются для обозначения кода операции
[4] **it evidently facilitates matters** — это, очевидно, облегчает дело (ситуацию)
[5] **must always be at the coder's side** — должен всегда быть у программиста под рукой

Exercises

V. Find the Russian equivalents to the following English words and word combinations:

1. then; 2. thus; 3. hence; 4. however; 5. whereas; 6. perhaps; 7. also; 8. sometimes; 9. there; 10. in other words; 11. moreover; 12. instead of; 13. the current chapter; 14. in fact; 15. specifically; 16. often; 17. in addition to; 18. easily; 19. clearly; 20. precisely

1. точно; 2. легко; 3. ясно; 4. вместо чего-либо; 5. кроме того; 6. вероятно; 7. также; 8. следовательно; 9. однако; 10. затем; 11. таким образом; 12. иногда; 13. там; 14. более того; 15. другими словами; 16. в действительности; 17. часто; 18. данная (текущая) глава; 19. тогда как; 20. специфично, специально

VI. Memorize the following definitions:

1. A program is a set of instructions composed for solving a given problems by a computer. 2. A code is the representation of data or instruction in the symbolic form. 3. A procedure is the sequence of steps required to solve a problem. 4. Programming is the process by which a set of instructions is produced for a computer to make it perform specified activity.

VII. Answer the following questions:

1. What is a code? 2. What is a program? 3. What does the term 'programming' include? 4. How many stages are there in programming? 5. Which are the stages in programming? 6. What does programming require? 7. What will happen if there is a single error in one instruction? 8. What does the computer manual give and define? 9. When does a word erase the previous contents of the memory address? 10. When is a word not erased from its memory address? 11. What are some bits of the instruction set aside for? 12. What is an instruction format? 13. Why are the octal code often used instead of the binary code?

VIII. Read Text B and write a brief summary of it:

TEXT B. PROGRAMMING

The word 'program' has come into use to refer to the sequence of instructions which a computer carries out. A program for a computer is an exact (точный) sequence of instructions that it uses to solve a problem. It usually consists of subroutines or subprograms which are portions of it.

Programming for automatic computer requires a good deal of [1] knowledge, common sense,[2] and training. Specially, programming requires: (1) understanding the operations of a business or the steps of a scientific calculation; (2) understanding the best way for having a computer carry out these operations and steps; (3) arriving at a good sequence of commands for the computer to solve the problem; and (4) adequately translating these commands into the computer language.

Programming for the computers has several forms. One form is the construction of compiling programs or compilers (компилятор) — which use the computer to take subprograms out of a library and link (соединять) them together appropriately (соответственно) so as to solve a new problem. A second form is the construction of programs called inter-

preters (интерпретатор) which accept instructions in certain (определенный) standard words and translate these words into a machine language, so that the machine "knows" what the words "mean". A third form is the development of common (обычный) languages for automatic programming for problems, so that any problem when expressed in such (такой) a language can be given to any automatic computer, and the computer will translate the common language into its own (собственный) instruction code, and then solve the problem.

Notes

[1] **a good deal of** — много
[2] **common sense** — здравый смысл

IX. Translate the following dialogue from Russian into English:

НА ВЫЧИСЛИТЕЛЬНОМ ЦЕНТРЕ НАШЕГО ИНСТИТУТА

Студент: Здравствуйте. Я хотел бы видеть оператора.
Оператор: Я оператор. Чем могу быть полезен вам?
Ст.: Видите ли, я впервые на вычислительном центре. Мне хотелось бы посмотреть на вычислительную машину и задать вам несколько вопросов относительно ее устройства и работы.
Оп.: С большим удовольствием отвечу вам. Пойдемте. Вот электронный компьютер ЕС-1045. Он является представителем семейства компьютеров, входящих в Единую Систему (the United System of Computers), которая проектируется и выпускается содружеством социалистических стран (the Commonwealth of Socialist Countries). Его быстродействие — 880000 операций в секунду, а объем ОЗУ — оперативного запоминающего устройства (the main memory capacity) — равен четырем мегабайтам (4 МВ).
Ст.: А что значит Единая Система ЭВМ?
Оп.: Единая Система ЭВМ — это такая система, которая использует программно-совместимые модели (program-compatible models), предназначенные для решения широкого круга научно-технических и экономических задач. В ЭВМ ЕС применяются большие интегральные схемы (LSI circuits).
Ст.: А что собой представляли компьютеры второго поколения? Каково было их быстродействие, вместимость памяти?

Оп.: Компьютеры второго поколения строились на полупроводниках (solid-state computers). Их ОЗУ строились на магнитных ферритовых сердечниках (the magnetic ferrit cores), а ПЗУ — постоянное запоминающее устройство (the read-only memory) — на магнитных лентах. Объем ОЗУ был около 70000—100000 байт. Их быстродействие было от 30000 до 50000 операций в секунду. Обращение к памяти (the access time) составляло 5—7 микросекунд. Машины второго поколения уже устарели (has become obsolete). Компьютеры 3-го и 4-го поколений считают в сотни и тысячи раз быстрее.

Ст.: А это какой компьютер?

Оп.: Это микрокомпьютер «Искра-1256». Объем его ОЗУ 64 КВ. Он построен на БИСах (LSI circuits). А это алфавитно-цифровое печатающее устройство (the alphanumerical printer).

Ст.: Большое спасибо. Разрешите мне прийти еще раз.

Оп.: Пожалуйста. Я расскажу вам об устройстве машин 4-го поколения, например, о компьютере «Наири-4», быстродействие которого увеличилось по сравнению с компьютерами 3-го поколения в 5 раз, благодаря применению многослойных печатных схем (the multilayer printed circuits).

Ст.: С удовольствием приду. До свидания!

X. Listen to the dialogue "At Our Institute Computing Centre" from the tape and compare your version. Reproduce the dialogue in English.

LESSON EIGHT

Text A. The Basic Principles of Programming *(continued)*.
Text B. Kinds of Programs.

Exercises

I. Read the international words and guess their meaning:

to conserve; permanent, constant; location; diagram; code; symbol; function; alternative; line; term; symbolism; specific; actual; specially; subroutine; initial; characteristic; coder; result; to illustrate; programmer; catalogue; to consult

II. Read and translate the following sentences paying attention to the meaning of the words and word combinations given below:

a) **to be late** — опаздывать
later than — позже чем

in the late 60s — в конце 60-х годов
the latest — самый последний
the former . . . the latter . . . — первый (из упомянутых) . . ., последний (из упомянутых)

1. Our professor is never late for his lectures. 2. This program was made later than it was planned. 3. In the late 60s the second generation of computers appeared and began to operate. 4. The latest achievements of up-to-date (современный) computing technique are applied in our computing centre. 5. The CPU consists of two parts: the control unit and the arithmetic/logic unit; the former receives and interprets instructions and generates control signals, the latter performs real computations.

b) **to set** — ставить; помещать; устанавливать
to set aside — откладывать
to set up — основывать, учреждать
a set — набор; комплект; множество; прибор
a set of — ряд
to preset — заранее помещать

1. The conditional code is set as a result of all logical comparing, connecting, testing, and editing operations. 2. The Institute of Automatics and Engineering Cybernetics of the USSR Academy of Sciences was set up in the late 1930s. 3. All logical operations other than editing are the part of the standard instruction set. 4. The set of logical operations include moving, comparing, bit connecting, bit testing, translating, editing. 5. A set of instructions is provided for the logical operation of processing data. 6. Several bits of an instruction are set aside to designate (определять) the operation code. 7. The address of a jump instruction must be present before making a block diagram.

c) **case** — ящик; коробка; корпус; случай
in any case — во всяком случае
this is the case — дело обстоит так

1. These are cases for packing all the units of the computer ES-1045. 2. In this case the bits would be divided into 14 groups of 3 bits each. 3. This is some kind of synchronization of the pulses in a computer, and in most computers this is the case: all pulses are synchronized with respect to each other. 4. In each case, the operation is suppressed; therefore the condition code and data in storage and registers remain unchanged. 5. In any case registers store information in

the memory. 6. The word which comes from the arithmetic unit back to the memory is not erased, but this is not the case with a new word when the previous one in the memory is always erased.

d) **yet** [jet] — (пока) еще; однако, но; хотя

1. In cases where we must write instructions involving addresses of constants that have not yet been specially assigned, we usually use the symbolic coding. 2. Yet the data for processing information appeared to be more important than the data for storing information in this case. 3. The program must be debugged if the errors have not yet been corrected. 4. Yet in practice, programmers desire the computer to take alternative ways of acting.

e) **once** — однажды, (один) раз; как только

1. A subroutine may be used many times during the computation of a program but is written only once in the whole program. 2. Once the execution of a command has been initiated, the indication of the neon bulb can be seen on the control panel. 3. Once set, the condition code remains unchanged until modified by an instruction that reflects a different condition code.

III. Pronounce the following words correctly:

temporaries ['temp(ə)rərɪz] *n pl* рабочие ячейки памяти
conserve [kən'sə:v] *v* сохранять; консервировать
point [pɔɪnt] *n* точка
share [ʃɛə] *v* распределять; делить(ся)
visualize ['vɪzjuəlaɪz] *v* мысленно представлять себе
interrelationship ['ɪntə(:)rɪ'leɪʃ(ə)nʃɪp] *n* взаимосвязь; взаимозависимость
various ['vɛərɪəs] *a* различный, разный; разнообразный
essentially [ɪ'senʃ(ə)lɪ] *adv* по существу
decision [dɪ'sɪʒ(ə)n] *n* решение
associate [ə'souʃɪeɪt] *v* соединять; связывать; объединять
alternative [ɔ:l'tə:nətɪv] *n* альтернатива; вариант; выбор варианта
eliminate [ɪ'lɪmɪneɪt] *v* устранять; исключать, заменять
indicate ['ɪndɪkeɪt] *v* указывать; показывать
statement ['steɪtmənt] *n* утверждение; оператор (в алгоритмических языках)
description [dɪs'krɪpʃ(ə)n] *n* описание; характеристика
particularly [pə'tɪkjuləlɪ] *adv* особенно
proceed [prə'si:d] *v* продолжать (делать что-л.)
occasionally [ə'keɪʒənlɪ] *adv* случайно; нерегулярно; вдруг
jump [dʒʌmp] *n* переход; *v* переходить
locate [lo(u)'keɪt] *v* размещать; помещать; располагать
detour [dɪ'tuə] *n* уход; обход; удаление
entrance ['entr(ə)ns] *n* вход *(в подпрограмму)*
exit ['eksɪt] *n* выход; выходной канал
condition [kən'dɪʃ(ə)n] *n* условие; состояние; режим; ситуация
value ['vælju:] *n* величина; значение; оценка

consider [kən'sɪdə] *v* считать; полагать; рассматривать
depend [dɪ'pend] *v* (**on, upon**) зависеть от
find [faɪnd] *v* находить
quite [kwaɪt] *adv* совсем; полностью, очень
profitable ['prɔfɪtəbl] *a* выгодный

IV. Memorize the following word combinations:

memory space — объем памяти, пространство памяти
point of view — точка зрения
permanent numbers — постоянные числа
temporary numbers — временные числа
intermediate stages — промежуточные этапы
specific instructions — специальные (конкретные) команды
various alternatives — различные альтернативы
crossing lines — линии пересечения
at a later time — в более позднее время, позже
jump (branch) instructions — команды перехода
a function calculation — вычислительные функции
for instance = for example — например

TEXT A. THE BASIC PRINCIPLES OF PROGRAMMING

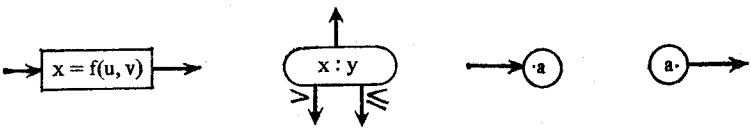
(continued)

1. **The Use of Temporaries.** Memory space in a computer should be conserved. From this point of view there are two kinds of numbers contained in addresses: permanent, or constant numbers, such as 2 and *g*, and temporary numbers, i.e., numbers that appears only in intermediate stages of the computation. These temporary numbers can share their address with other temporary numbers that appear at some other time. Hence we call such memory locations temporaries.

2. **Flow Chart.** A flow chart is a diagram, or picture, of a code that is often helpful for visualizing interrelationship between various parts of a code. Such diagram is almost always made before the specific instructions are written. There are essentially three kinds of symbols used in a flow chart (see Fig. 5). The first represents function calculations, the second represents decisions and the various associated alternatives, the third, called a variable connector, is simply a way to eliminate too many crossing lines in the picture or to indicate which lines to follow when one has to continue the diagram on another page.

3. **Symbolic Coding Aids.**[1] It is another intermediate aid between the statement of the problem and the final code. Symbolic coding consists in writing a code not in terms of

specific numerical addresses, but rather in terms of some name description or other symbolism to represent the addresses. Then, at a later time, specific addresses can be assigned for these symbols, or names, to produce the actual code. The intermediate code in terms of symbols is called the sym-



Function calculation Decision calculation Connector

Figure 5. Flow Chart Symbols

bolic code. This technique is extremely useful particularly in those cases where one must write instructions involving addresses of constants or of other instructions that have not yet been specially assigned.

4. **The Use of a Subroutine.** A subroutine is a subcode that may be used many times during the computation of a program but written only once in the whole code. As the computer proceeds down the main program, the control will occasionally jump to this subroutine and then, after doing the subroutine, will jump back to the main program where it left off. The term "jump" or "branch" is commonly used in the sense of "take the next instruction from". Thus the expression "jump to a subroutine" means "take the next instruction from the address where the first instruction of the subroutine is located".

5. The detour from the main program through the subroutine may occur several times during the computation of the program. Hence a subroutine must have an entrance, a way of getting into it, and an exit, a way of getting out of it. Each time when an entrance is made to a subroutine, some initial conditions [2] must be set up that are characteristic of the place in the main program from which the entrance was made. For instance, if the subroutine calculates some function, the initial values of the independent variables [3] at that point in the main program must be given to the subroutine. In addition, as an entrance to a subroutine is made, the exit must be set up; i.e., the subroutine must be told where to transfer control back to the main program.

6. Hence, in order to use a subroutine, the coder must know (1) the entrance, i.e., the address of the first instruction; (2) the addresses of the temporaries in which the initial

conditions are to be set up; (3) the addresses of the temporaries whose contents will be the results of the subroutine computation; (4) the exit, i.e., the address of some jump (branch) instruction that is to be present (when entering the subroutine) so that when the computation of the subroutine has been completed, the computer will transfer control back to the proper address [4] of the main program.

7. Consider, for instance, the flow chart in Fig. 6, which illustrates the setting up of the initial conditions and differ-

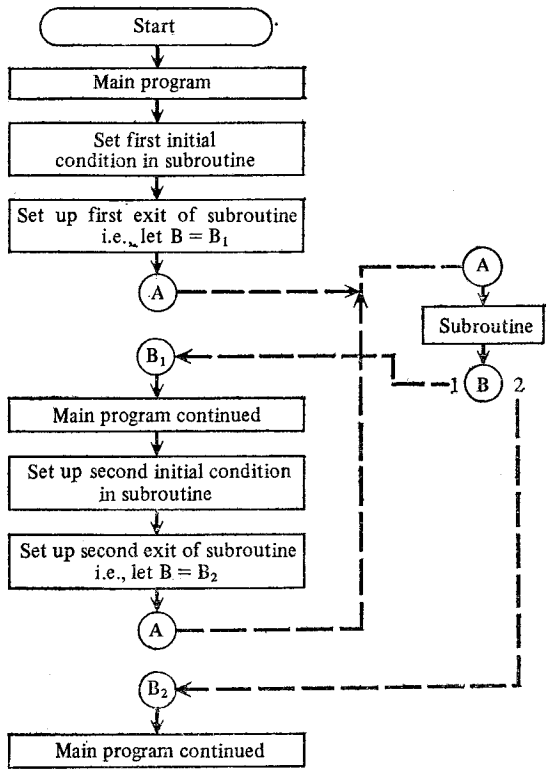


Figure 6. Flow Chart for Setting up Initial Conditions and Different Exits of a Subroutine

ent exits of the subroutine. The A in the circle is a connector and of course means that the program jumps to the subroutine at that point. The B connector is called a variable connector, i.e., it indicates that the program jumps either to B_1 or to B_2, depending on how the "variable" connector is set. When we say that $B=B_1$, we mean that the B connector

is set so that it will go from B to B_1; and when we say that $B=B_{\bar{2}}$, then the B connector is set to go from B to $B_{\bar{2}}$. The rest of the flow diagram is self-explanatory [5] but should be studied carefully by the student.

8. **Library of Subroutines.** Several subroutines may be used in one program. In fact it is found that many common subroutines are used quite often, such as sin x, cos x, e^x, $\sqrt{x}$, etc. Hence it often becomes profitable to have a library of subroutines available to the programmer, stored at all times in some part of the computer memory. There would also be some catalogue kept outside the computer that the programmer can consult when he wants to use a subroutine. This catalogue would tell where each subroutine is located and all data about how to use it, such as where to put the initial values of independent variables, where the computed values of the dependent variables [6] are found, etc.

Notes

[1] **symbolic coding aids** — средства символического кодирования
[2] **initial** [ɪ'nɪʃəl] **conditions** — начальные (исходные) условия
[3] **independent variables** — независимые переменные
[4] **the proper address** — правильный (соответствующий) адрес
[5] **self-explanatory** — ясный, не требующий разъяснения
[6] **dependent variables** — зависимые переменные

Exercises

V. Find the Russian equivalents of the following English word combinations:

1. crossing lines; 2. symbolic coding aids; 3. the statement of the problem; 4. the final code; 5. independent variables; 6. dependent variables; 7. initial conditions; 8. the computed values; 9. the initial values; 10. for visualizing interrelationship; 11. the intermediate code; 12. extremely useful; 13. addresses of constants; 14. memory location temporaries; 15. to transfer control back

1. конечный код; 2. передать управление обратно; 3. вычисленные значения; 4. линии пересечения; 5. средства символического кодирования; 6. адреса констант; 7. промежуточный код; 8. чрезвычайно полезный; 9. для отчетливого представления взаимосвязи; 10. рабочие ячейки памяти; 11. независимые переменные; 12. исходные значения; 13. формулировка задачи; 14. зависимые переменные; 15. исходные (начальные) условия

VI. Arrange the synonyms in pairs and translate them:

permanent numbers; block diagram; to call; memory; for example; a step; a routine; calculation; location; to share; storage; a stage; constant numbers; computation; subroutine; to term; for instance; subcode; to divide; a program; cell; flow chart

VII. Form two sentences of your own with each word combination:

from this point of view; to set aside; at the coder's side; in other words; the rest of

VIII. Memorize the following definitions:

1. A subroutine is a part of a program which performs a logical section of the over-all function of the program. 2. A jump or a branch is a departure [dɪ'pɑ:tʃə] (уход) from the normal sequence of program steps. 3. A library of subroutines is any collection of subroutines which have been written for general application and can be incorporated in different programs when required. 4. A flow chart or a block diagram is a diagrammatic representation of a sequence of events (событие), usually drawn with conventional symbols representing different types of events and their interconnection.

IX. Answer the following questions:

1. How many kinds of numbers does the address contain? 2. What are permanent numbers? 3. What are temporary numbers? 4. What memory locations do we call temporaries? 5. What is a flow chart? 6. When is a flow chart made? 7. What symbols are used in a flow chart? 8. What is symbolic coding? 9. What is a subroutine? 10. What must the coder know in order to use a subroutine? 11. What is a jump or a branch made for? 12. What is a subroutine library? 13. Where is the subroutine library stored?

X. Speak on:

the temporaries; the symbolic coding aids; the library of subroutine; the flow chart

XI. Read Text B and translate it with a dictionary. Write a brief summary of it:

TEXT B. KINDS OF PROGRAMS

There are two main kinds of programs which are subdivided as well. They are control programs and system service programs.

Control Programs.[1] There are some control programs to handle interruptions (прерывание), I/O operations, transition between different jobs (задание) and different phases of the same job, initial program loading (IPL), and symbolic assigment of I/O devices.

Actually, the control program consists of three components:

1. **IPL Loader.** When system operation is initiated, the hardware IPL reads in this program which then clears unused core storage [2] to zeros, generally performs some housekeeping operations,[3] and then reads in the Supervisor.

2. **The Supervisor.** It is resident in the low order part core storage whenever any job is being run. It can handle execution of any I/O operations and can provide standard processing for all interrupts. It may also queue I/O operations so that the operation will start as soon as the required channel and device are free.

3. **Job Control.**[4] When the end of a job is sensed, the user normally turns control over to the Supervisor which then reads in the Job Control program. This program senses and processes all cards; after the new job is loaded, it turns control over to the new job. This system of operation eliminates the necessity of operation intervention between jobs, something which is of particular importance on large computers where the average job time [5] may be less than one minute.

The control program is a large program occupying thousands of bytes. It is usually stored on a magnetic tape or a magnetic disk. The particular unit on which the program resides is called the System Resident unit.[6]

System Service Programs.[7] They are used to maintain the library: to place new programs into the library, to delete, replace, or change existing programs, to read programs from the library into memory, to link segments of programs written at different times into one program, etc. They are called Librarian [8] and Linkage Editor.[9]

There are three libraries maintained by the Librarian program: 1) Core Image library,[10] 2) Relocatable library,[11] 3) Source Statement library.[12]

The core image programs includes the processors [13] and control programs and as many of the application programs [14] as the user desires. The relocatable library contains modules which are stored in such a way that each module can be arbitrarily relocated and still be executed correctly. A module may be a complete program or it may be a relatively independent part of a larger program. The source statement

library contains macro definitions,[15] a sort of collection of standard type statements which can be incorporated into a program to provide some special functions.

Notes

[1] **control programs** — управляющие программы
[2] **unused core storage** — неиспользованная часть ОЗУ
[3] **housekeeping operations** — вспомогательные операции
[4] **Job Control** — управление потоком заданий
[5] **the average job time** — среднее время счета задания
[6] **System Resident unit** — системное резидентное устройство
[7] **System Service program** — системная сервисная программа
[8] **Librarian** — программа «Библиотекарь»
[9] **Linkage Editor** — программа «Редактор связей»
[10] **Core Image library** — библиотека абсолютных модулей
[11] **Relocatable library** — перемещаемая библиотека
[12] **Source Statement library** — библиотека исходных предложений
[13] **processor** — программа «Транслятор»
[14] **application program** — прикладная программа
[15] **macro definitions** — макроопределения

XII. Translate the following dialogue into English:

НА ЭКЗАМЕНЕ ПО ЭВМ

Студент: Здравствуйте, профессор.
Профессор: Здравствуйте. Как вас зовут, из какой вы группы?
Ст.: Иванов, я студент 3-го курса 5-й группы.
Проф.: Берите, пожалуйста, билет и готовьтесь отвечать.

После подготовки к ответу по билету студент Иванов отвечает на все 3 вопроса хорошо.

Проф.: Позвольте задать вам несколько вопросов дополнительно. Во-первых, с какого времени стали выпускать компьютеры на электронных лампах, и какие отечественные машины подобного типа вы знаете?
Ст.: Электронные компьютеры стали выпускать с 1944 года. Представителями этого первого поколения электронных компьютеров в СССР были БЭСМ и «Минск-1».
Проф.: Какова скорость микрокомпьютера?
Ст.: Современный микрокомпьютер может выполнять более 10 биллионов операций в секунду, и это не предел.
Проф.: Кто был изобретателем первого в мире арифмометра?
Ст.: Русский математик П. Л. Чебышев, в 1882 г.
Проф.: Где нашли применение компьютеры?
Ст.: Во всех областях жизни. Особенно в астрономии,

электротехнике, физике, химии, ядерной технике, лингвистике, статистике, банковских операциях, медицине. Компьютеры сочиняют музыку, играют в шахматы, рисуют и чертят.

Проф.: Ваш ответ заслуживает высокой оценки.

Ст.: Спасибо, профессор. Этот предмет мой любимый. До свидания.

XIII. Listen to the dialogue "At the Exam" from the tape, compare your version. Reproduce the dialogue in English.

LESSON NINE

Text A. Low Level Programming Languages: Machine and Assembly Languages.
Text B. Programming Languages.
Text C. The Interface.

Exercises

I. Read the international words and guess their meaning in Russian:

order; sort; to communicate; spectrum; assembler; mnemonic; autocode; designer; repertoire; absolute; pseudo; figure; increment; decrement; container; competition; code; translation; ALGOL; FORTRAN; COBOL

II. Translate the following sentences paying attention to the meaning of the words given below:

a) **to provide** — обеспечивать; снабжать; предусматривать
providing — обеспечение; при условии, если только; в том случае, если
provided — обеспеченный; предусмотренный; снабженный; при условии, если только; в том случае, если

1. The system control section provides the normal CPU operation. 2. A set of instructions is provided for the logical processing of data. 3. All branching operations are provided in the standard instruction set. 4. After providing this command the channel can communicate with other devices on the interface. 5. The process could be repeated, providing we wanted to receive the final results. 6. In this case the channel refers to a location not provided in the system. 7. Any main storage location provided in the system can be used to transfer data to or from an I/O device, provided that during an input operation the location is not protected.

b) **both of** — оба
both . . . and . . . — как ..., так и ...; и ... и ...

1. Both diagrams shown in Fig. 10 are easy to understand. 2. Both of devices have been designed by our postgraduate students. 3. If decimal arithmetic is provided, both operands and results are located in storage. 4. Both punched cards and punched paper tapes are used for accepting information.

c) **but** — но; лишь; только; кроме; однако

1. Individual computers differ, but enough similarity exists to make a general discussion of the more important points helpful. 2. Chess for mathematicians is but a means to show the ability to compile a program. 3. Because of the millions of characters of information that can be stored, magnetic tapes are common with all but the smallest computers. 4. But in future the machines will be able to solve many problems which today are in the competence of man.

d) **since** — так как; с тех пор как; с тех пор; с

1. Ch. Babbage's machine could not operate since there were no reliable and accurate electrical equipment at the beginning of the 18th century. 2. We have been ready to begin our experiments since yesterday. 3. Since the beginning of the 40s the computing technique has started to develop successfully. 4. Since electronics became known, it began to be used in nearly all branches of industry. 5. Our lecturer left for Leningrad and we have never seen him since.

e) **a result** — результат; следствие
as a result of — в результате
to result in — давать в результате; приводить к
to result from — являться результатом; вытекать из

1. After performing computations a computer displays the results. 2. The detection of mistakes (errors) in a program results in a program interruption. 3. The protection of a computer is recorded in bits 0—3 as a result of the channel operation. 4. These data result from the comparison operation. 5. The importance of microprocessors results from their ability to process information with unimaginable (невообразимый) speeds . 6. The achievements of the Soviet science have resulted in a triumph for our country in many fields of science.

III. Pronounce the following words correctly:

wish [wɪʃ] *v* хотеть, желать
spectrum ['spektrəm] *n* спектр
mnemonic [ni(:)'mɔnɪk] *a* мнемонический
bottom ['bɔtəm] *n* низ; дно
autocode ['ɔ:to(u)koud] *n* автокод
immediately [ɪ'mi:djətlɪ] *adv* немедленно
obey [ə'beɪ] *v* повиноваться; подчиняться
adopt [ə'dɔpt] *v* принимать; перенимать
repertoire ['repətwɑ:] *n* набор; состав; система команд; репертуар
alphanumeric [,ælfənju:'merɪk] *a* буквенно-цифровой
assembly language [ə'semblɪ'læŋgwɪdʒ] язык ассемблера
disadvantage [,dɪsəd'vɑ:ntɪdʒ] *n* недостаток
assembler [ə'semblə] *n* программа «Ассемблер»
source [sɔ:s] *n* источник; исток; *a* исходный
specify ['spesɪfaɪ] *v* точно определять, устанавливать
distinguish [dɪs'tɪŋgwɪʃ] *v* различать
relative ['relətɪv] *a* относительный
offset ['ɔ(:)fsət] *n* смещение; сдвиг
distinction [dɪs'tɪŋ(k)ʃ(ə)n] *n* различие, распознавание
therefore ['ðeəfɔ:] *adv* поэтому
existence [ɪg'zɪst(ə)ns] *n* существование
note [nout] *v* замечать; отмечать; упоминать
shift [ʃɪft] *n* сдвиг; смещение; *v* сдвигать; смещать

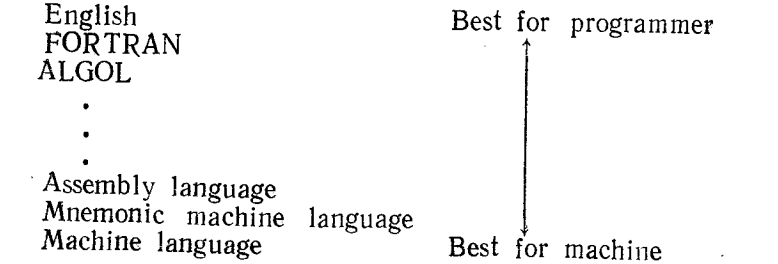
IV. Memorize the following word combinations:

machine-dependent — машинно-зависимый
low level languages — языки низкого уровня
an absolute address — абсолютный адрес
a source program — исходная программа
an object program — конечная программа
pseudo-op (operation) — псевдокоманда
machine-op (operation) — машинная команда
relative addresses — относительные адреса

TEXT A. LOW LEVEL PROGRAMMING LANGUAGES: MACHINE AND ASSEMBLY LANGUAGES

1. In order to communicate with each other, men use languages. In the same way, 'languages' of one sort or another are used in order to communicate instructions or commands to a computer.

2. When the user wishes to communicate with the computer, he uses a spectrum of languages:



Let us discuss the three lowest members of this spectrum, beginning with the first three from the bottom: machine language, mnemonic machine language, and assembly language.

3. A machine language which is sometimes called as a basic programming language or autocode refers to instructions written in a machine code. This machine code can be immediately obeyed by a computer without translation. The machine code is the coding system adopted in the design of a computer to represent the instruction repertoire of the computer. The actual machine language is generated by software, not a programmer. The programmer writes in a programming language which is translated into the machine language.

4. A mnemonic machine language uses symbolic names for each part of instruction that is easier for the programmer to remember than the numeric code for the machine. A mnemonic is an alphanumeric name, usually beginning with a letter rather than a number to refer to fields, files, and subroutines in a program. For example, the operation 'multiplication' might be represented as MULT, the 'load' instruction as L, or DISP NAME ADDR in the mnemonic form means 'display name and address', etc.

5. An assembly language is the most machine-dependent language used by programmers today. There are four advantages to using an assembly language rather than machine language. They are the following: 1) it is mnemonic, e.g.,[1] we write ST instead of the bit configuration 0101 0000 for the STORE instruction; 2) addresses are symbolic, not absolute as in a machine language; 3) reading is easier; 4) introduction of data to a program is easier.

6. A disadvantage of assembly language is that it requires the use of an assembler to translate a source program into object code (program) in order to be directly understood by the computer. The program written in assembly language is called an assembler. The assembler usually uses such instructions as A (ADD), L (LOAD), ST (STORE), START, TEST, BEGIN, USING, BALR (Branch And Link Register), DC (Define Constant), DS (Define Storage), END, etc. Let us consider some of them.

7. The USING instruction is a pseudo-op. A pseudo-op is an assembly language instruction that specifies an operation of the assembler; it is distinguished from a machine-op which represents to the assembler a machine instruction. So, USING indicates to the assembler which general register

to use as a base and what its contents will be. This is necessary because no special registers are set aside for addressing, thus the programmer must inform the assembler which register(s), to use and how to use them. Since addresses are relative, he can indicate to the assembler the address contained to the base register. The assembler is thus able to produce the machine code with the correct base register and offset.

8. BALR is an instruction to the computer to load a register with the next address and branch to the address in the second field. It is important to see the distinction between the BALR which loads the base register, and the USING which informs the assembler what is in the base register. Hence, USING only provides information to the assembler but does not load the register. Therefore, if the register does not contain the address that the USING says it should contain, a program error may result.

9. START is a pseudo-op that tells the assembler where the beginning of the program is and allows the user to give a name to the program, e.g., it may be the name TEST. END is a pseudo-op that tells the assembler that the last card of the program has been reached.

10. Note that in the assembler instead of addresses in the operand fields of the instruction there are symbolic names. The main reason for assemblers coming into existence was to shift the burden [2] of calculating specific addresses from the programmer to the computer.

Notes

[1] **e.g.** = **for example** (for instance) — например
[2] **burden** — нагрузка; тяжесть

Exercises

V. Memorize the following definitions:

1. The machine language is such a language which can immediately be obeyed by a computer without translation. 2. The assembly language is a symbolic programming language which allows the programmers to write their programs at the machine language level. 3. The assembler is a program that translates assembly language into machine language. 4. A source program is a program written in a source language (ALGOL, COBOL, FORTRAN, PL/1, etc.) which cannot be directly processed by a computer but requires

translation into the object program. 5. An object program is a program which is written in the machine code capable of being directly understood by the computer in the form required for running.

VI. Answer the following questions:

1. What spectrum of languages does the user have at his disposal? 2. Which languages do you call "low level"? 3. Which languages are the best for machine? 4. Which languages are the best for programmer? 5. What language do you call a machine language? 6. How is an instruction usually written in a machine language? 7. When is the mnemonic form of the machine language used and why? 8. Does the assembly language use symbolic addresses? 9. What advantages to using the assembly language do you know? 10. What is a disadvantage of the assembly language? 11. Which program can the computer directly understand? 12. What is an assembler? 13. What does a USING instruction indicate? 14. What does the BALR instruction mean? 15. What is the distinction between BALR and USING? 16. What is a source program? 17. What is an object program?

VII. Read Text B without a dictionary. Try to get the main idea of each paragraph. Render the text in Russian:

TEXT B. PROGRAMMING LANGUAGES

How are directions (указания) to be expressed to the computer? The computer is not another human being with whom one can speak easily and clearly in common English. For expressing directions to a computer the programmer in practice uses special programming languages.

Because computers can accept letters and numbers, nearly (почти) all the programming languages express the directions in some combination of letters and numbers.

The programming languages in use [1] fall into three general categories in terms of their similarity (подобие) to ordinary English: machine languages, symbolic languages, and automatic coding languages. In terms of their importance for computer utilization, the machine languages are the most basic, for the computers can use them directly. But symbolic and automatic coding languages are more convenient for the programmer use because they are more similar to English.

Some programming languages are used only with a particular model [2] of computer; some are used with more than one model of computer. For the convenience of the program-

mer, a language that can be used with several different models of computers is the more useful.

Instructions in a machine language are almost always represented by particular combinations [3] of letters and numbers acceptable to a given computer. Programs written in an appropriate (соответствующий) machine language can be directly accepted and used by a computer.

Symbolic languages use symbolic addresses in the operands and usually also as the addresses for the instructions. This is in contrast to machine languages, which use absolute addresses. An absolute address is one expressed in machine language. It identifies (определять) a specific and physical location of data in storage. An indirect (косвенный) address is an absolute or symbolic address which has as its contents the absolute address (usually) of the operand needed by the instruction. Indirect addresses allow greater flexibility (гибкость) in programming because the programmer by changing the contents of indirect addresses can, in effect (в действительности), modify a program.

For data description in the symbolic languages, the programmer uses special commands. Being able to use these commands simplifies (упрощать) the process of data description, because often these commands can be used with symbolic addresses as their operands. Thus in many programming languages the programmer can assign addresses in symbolic, relative, or absolute form, depending upon the character of the language and what is most convenient for the programmer at the time.

Notes

[1] **in use** — используемые (в настоящее время)
[2] **a particular model** — определенная модель
[3] **a particular combination** — определенная комбинация

VIII. Listen to Text C from the tape recorder. Give the contents in short (in Russian).

Notes

[1] **interconnection** — взаимосвязанность
[2] **wire** — провод
[3] **to exchange** — обмениваться
[4] **specific message** — конкретное сообщение
[5] **to establish** — устанавливать
[6] **application program** — прикладная программа
[7] **Data Base Management System** — система управления базами данных
[8] **screen** — экран

TEXT C. THE INTERFACE

The interface is interconnection [1] between hardware, software, and people. Hardware interfaces are physical channels, cables, or wires [2] that must connect and exchange [3] electronic signals between a CPU and peripherals, and between any two units. Software interfaces are specific messages [4] established [5] between programs. Examples of the software interfaces are application programs,[6] the operating system, Data Base Management Systems [7] and communication programs. Interfaces between people and computers are terminal screen [8] and keyboards.

LESSON TEN

Text A. High Level Programming Languages: FORTRAN and PL/1.
Text B. ALGOL.
Text B'. COBOL.
Text C. Data Base Management System (DBMS).

Exercises

I. Read the international words and guess their meaning:

oriented; to orient; combination; algebraic; formulae; standard; segment; constant; master; position; individual; option; representative; range; to combine; compiler; to construct; commercial; application; modular; structure; interpretation; effect; document; notation; accurate

II. Translate the following sentences paying attention to the meaning of the words given below:

a) **structure** — структура; конструкция; строение
to structure — строить; конструировать; создавать

1. The structure of this device is very complex. 2. The block structure of ALGOL is of great value because it permits the parts of a program to be written by different programmers. 3. The words of a language are structured directly from the characters which are usually the smallest units that have meaning by themselves. 4. The program of FORTRAN is structured in segments.

b) **master** — главный; основной
to master — управлять, руководить; совершенствовать

1. A master card is a punched card holding any fixed information about a group of cards. 2. Master data are data elements of a record which seldom change. 3. We have to

master our knowledge in mathematics and English. 4. The CPU masters the actual calculations inside the computer.

c) **to state** — устанавливать
state — государство; штат; состояние

1. Programming languages offer various ways for a programmer to state precisely the order of instructions which is often known as the "sequence of control". 2. Scientists of the Soviet State take an important part in international conferences and symposiums. 3. Lights and signals on the control panel show the electrical state of the computer.

d) **aim** — цель; намерение
to aim — иметь целью

1. The aim of this paper is to show the organization of a memory made on thin films. 2. PL/1 aims at mathematical and economical use. 3. This equipment aims at printing results on cards and displaying them on screens.

f) **only** — только
the only — единственный

1. In computers only two electrical states are used, 1 for one state, and 0 for the other. 2. The idea of an automatic computer that would not only add, multiply, subtract, and divide but perform a sequence of reasonable operations automatically was given by the English scientist Charles Babbage. 3. It was the only way to solve this problem. 4. Petrov is the only student in our group who had worked at the computing centre before entering the Institute. 5. The access time is only a few millionth of a second for magnetic cores. 6. The printers are used only for output unit.

III. Pronounce the following words correctly:

correspond [ˌkɔrɪs'pɔnd] *v* соответствовать
several ['sevr(ə)l] *a* несколько
notation [no(u)'teɪʃ(ə)n] *n* исчисление; запись
familiar [fə'mɪljə] *a* знакомый; известный
orient ['ɔ:rɪənt] *v* ориентировать(ся)
item ['aɪtəm] *n* элемент; единица информации
executable ['eksɪkju:təbl] *a* исполнимый
optional ['ɔpʃənl] *a* произвольный; необязательный
array [ə'reɪ] *n* массив
reference ['refr(ə)ns] *n* ссылка; сноска; эталон
whole [houl] *a* целый; весь
range [reɪn(d)ʒ] *n* область; диапазон; интервал; ряд
imaginary [ɪ'mædʒɪn(ə)rɪ] *a* воображаемый; нереальный; мнимый
true [tru:] *a* истинный
false [fɔ:ls] *a* ложный
appropriate [ə'prouprɪɪt] *a* подходящий; соответствующий
precede [pri(:)'si:d] *v* предшествовать

label ['leɪbl] *n* метка; идентификатор; знак
enable [ɪ'neɪbl] *v* давать возможность
general-purpose ['dʒen(ə)r(ə)l 'pə:pəs] *a* универсальный, общего назначения
feature ['fi:tʃə] *n* черта; особенность; признак; свойство
emphasis ['emfəsɪs] *n* ударение; выразительность
volume ['vɔljum] *n* объем; большое количество; том
facility [fə'sɪlɪtɪ] *n* доступность; *pl* средства; оборудование
neglect [nɪ'glekt] *v* пренебречь; упустить из виду
modular ['mɔdju:lə] *a* модульный
extensive [ɪks'tensɪv] *a* расширенный; протяженный
subset [,sʌb'set] *n* подмножество
default [dɪ'fɔ:lt] *n* упущение; недосмотр
unspecified [(')ʌn'spesɪfaɪd] *a* не установленный; точно не определенный
valid ['vælɪd] *a* действительный; обоснованный
framework ['freɪmwə:k] *n* строение; структура; конструкция; рамки
sensitive ['sensɪtɪv] *a* чувствительный
peculiarity [pɪ,kju:lɪ'ærɪtɪ] *n* специфичность, особенность

IV. Memorize the following word combinations:

master segment — главный сегмент
alphanumeric names — буквенно-цифровые имена
data items — элементы данных
floating-point representation — представление с плавающей точкой (запятой)
imaginary parts — мнимая часть (в комплексном числе)
truth values — истинные значения
executable statements — исполнительные операторы
assign(ment) statements — операторы присвоения
dummy variables — фиктивные переменные
to meet the needs (requirements) — отвечать требованиям
default feature — признак «по умолчанию»
valid interpretation — обоснованная интерпретация

TEXT A. HIGH LEVEL PROGRAMMING LANGUAGES: FORTRAN & PL/1

1. A high level language is a language in which each instruction or statement correspond to several machine code instructions. It is contrasted with a low level language in which each instruction has a single corresponding machine code equivalent. High level languages allow users to write in a notation with which they are familiar, e.g., FORTRAN in mathematical notation, COBOL in English. So, high level languages are oriented to the problem, while low level languages are oriented to the machine code of a computer.

2. **FORTRAN.** FORTRAN is an acronym for FORmula TRANslation. It is a problem oriented high level programming language for scientific and mathematical use, in which the source program is written using a combination of alge-

braic formulae and English statements of a standard but readable form. FORTRAN was the first high level programming language. It was developed in 1954, and was designed to easily express mathematical formulas for computer processing. It is still the most widely used programming language. There were several versions of FORTRAN. Nowadays the most popular and used is FORTRAN-4.

3. A FORTRAN program consists of data items, executable statements and non-executable statements. The program is structured in segments which consist of a master segment and optional function segments and subroutines.

4. Data items in FORTRAN are either variables or constants, and are assigned alphanumeric names by the programmer. Groups of similar items of data can be processed as arrays, or tables of data, in which case the individual items are defined by their position or reference within the array by naming the array followed by one or more subscripts.

5. Data items in FORTRAN may take the following forms: **Integer** is a whole number value falling within a range determined by the capacity of the computer being used. **Real** is a number expressed in floating-point representation accurate to a number of significant digits, the range again depends on the capabilities of the particular machine being used. **Complex** is a number in which two real numbers are used to express the real and imaginary parts. **Logical** is a quantity which can only take two values, true or false. **Text** is character information, which is not used for mathematical operations.

6. The actual operations of the program are expressed by means of 'executable statements'. These can take two forms: 'assignment statement' and 'control statement'. An assign statement takes the form Variable = Expression. The expression may be either arithmetic or logical. An arithmetic expression can include variables, elements, form arrays, constants and a variety of standard functions which are combined by arithmetic operations, e.g., +, —, *(multiplication),/(division), **(exponentiation). A logical expression is similar but include the operations AND, NOT, OR, etc., and the logical operators.

7. An example of an arithmetic assignment statement would be:

ROOT=(—B+SQRT (B**2—4*A*C))/(2*A),

where the word ROOT and the letters A, B, C represent variables and SQRT—the function provided for calculating

square roots. The compiler recognizes these symbols and translates them into appropriate machine code. An example of a logical assignment statement would be: BOOL-A.OR.B. In this expression the variable BOOL would be given the value true or false according to the truth values of variables A and B, and the truth table defined by the Boolean operator OR.

8. Each statement can be preceded by a numerical label, permitting reference to the statement by means of control statements. Control statements enable the program to branch to other statements. Branches themselves may also be constructed which are conditional on results of arithmetic or logical operations.

9. As was said above, a FORTRAN program consists of one or more segments, of which there is one and only one master segment, and optionally, function and subroutine segments. A function segment is used where the same form of function is required several times in a program. The statements describing the operation required to calculate the result of using the function are named and written once, and whenever the function is required in the program [1] it is only necessary to give the function name and a list of parameters to replace the 'dummy' variables used in the function segment.

10. **PL/1.** PL/1 was introduced in 1964. It was developed as a general-purpose programming language, incorporating features from both COBOL and FORTRAN. PL/1 is used primarily on large mainframes. PL/1 stands for Programming Language 1. Commercial applications (COBOL) with their emphasis on efficient handling of large volumes of data have led to the development of languages with sophisticated I/O facilities; scientific problems (FORTRAN) with their emphasis on rapid definitions and descriptions of complex problems have led to the development of highly sophisticated algorithmic languages while neglecting the data handling aspects.[2]

11. PL/1 aims at combining the problem-solving facility of scientific languages with the data-handling capabilities of commercial languages, in order to meet the needs of increasingly mathematical commercial analysis and increasingly large volumes of data being processed by scientific routines.

12. Among the more important features of PL/1 are the following: (1) The language is modular in structure. This means that the user needs only master the set of facilities necessary for his programming needs. More complex problems can use more extensive subsets of the language. (2)

The language has a 'default' feature by which every error or unspecified option is given a valid interpretation, thus minimizing the effects of programming errors. (3) The language structure is 'free form'. No special documents are needed for coding,[3] since the significance of each statement depends on its own format and not on its position within a fixed framework.

13. PL/1 is much less sensitive to the peculiarities of the hardware than the machine language. This makes it possible to use the same program on different types of computers.

14. PL/1 uses 60 symbols: 29 letters from A to Z (capital Roman letters [4]), 10 digits from 0 to 9, and 21 special characters. The PL/1 statements are subdivided into the following logical groups: input/output statements, declare statements,[5] debugging statements,[6] assignment statements, memory allocation statements,[7] program structure statements, and control statements.

(To be continued)

Notes

[1] **whenever the function is required in the program** — всякий раз когда в программе требуется функция
[2] **while neglecting the data handling aspects** — пренебрегая аспектами управления информацией
[3] **No special documents are needed for coding** — никакие специальные документы для кодирования не требуются
[4] **capital Roman letters** — заглавные латинские буквы
[5] **declare statements** — операторы описания
[6] **debugging statements** — операторы отладки
[7] **memory allocation statements** — операторы распределения памяти

Exercises

V. Memorize the following definitions:

1. FORTRAN is a problem oriented high level programming language for scientific and mathematical use. It is a compiler language. 2. PL/1 is a general-purpose high level programming language for scientific and commercial applications.

VI. Answer the following questions:

1. What is FORTRAN? 2. What are low level languages oriented to? 3. What are high level languages oriented to? 4. When was FORTRAN developed? 5. What does a FORTRAN program consist of? 6. Who assigns alphanumeric

names to data items? 7. Which forms may data items in FORTRAN take? 8. By means of what are the actual operations expressed in FORTRAN? 9. How is a program in FORTRAN structured? 10. What do segments in a FORTRAN program consist of? 11. When was PL/1 introduced? 12. What is PL/1? 13. What does PL/1 aim at? 14. What are the more important features of PL/1? 15. How many symbols does PL/1 use? 16. What logical statements does PL/1 use?

VII. Compare FORTRAN and PL/1.

VIII. Speak on:

a) arithmetic and logical assignment statements in FORTRAN;

b) the features of the general-purpose PL/1.

IX. Read Text B without a dictionary. Render the main points of the text in Russian:

TEXT B. ALGOL

ALGOL was developed as an international language for the expression of the algorithms between individuals, as well as a programming language. It was introduced in the early 1960s and gained popularity in Europe more than in the United States.

ALGOL is an acronym for ALGOrithmic Language. It is a problem oriented high level programming language for mathematical and scientific use, in which the source program provides a means of defining algorithms as a series of statements and declarations[1] having a general resemblance (сходство) to algebraic formulae and English sentences.

An ALGOL program consists of data items, statements and declarations, organized in a program structure in which statements are combined to form compound (составной) statements and blocks. Ingredients (составная часть) of the ALGOL, namely (а именно), characters, words, expressions (data items), statements, and declarations are really the hierarchical (иерархический) ingredients, because words are made from combination of characters, expressions are composed of groups of words, and statements consist of combinations of expressions.

Declarations used in ALGOL provide the compiler with information about quantities appearing in the program. The 'type declaration' is used to specify whether a variable is an integer,[2] a real number or a Boolean variable.

ALGOL was originally known as IAL or International Algebraic Language. Improvements (усовершенствования) and modifications are still being offered. There were some versions of ALGOL, the most known of them being ALGOL 60 and ALGOL 68.

Notes

[1] **declaration** — описание (в алгоритмическом языке АЛГОЛ); объявление (в алгоритмическом языке ПЛ/1)
[2] **whether a variable is an integer** — является ли переменная целым числом

X. Read Text B′ and translate it without a dictionary. Write a short summary:

TEXT B′. COBOL

COBOL is an acronym for COmmon Business Oriented Language. It is internationally accepted programming language developed for general commerial use. COBOL is a problem oriented high-level language in which the source program is written using statements in English.

A COBOL program is written in four divisions: Identification Division,[1] Environment Division,[2] Data Division, Procedure Division. The Identification Division contains descriptive information that identifies the program being compiled. The Environment Division deals with [3] the specification of the computer to be used for operating the object program, including such information as the size of memory, the number of tape decks,[4] printers and other peripheral devices that will be used; a description of the computer to be used for compiling the source program is also given here. The Data Division is used to allocate labels to all units of data on which operations are to be performed. All input and output files are defined and associated with the peripheral units to be used for input and output. The Procedure Division gives the step-by-step instructions [5] necessary to solve the problem. These steps are specified by means of instructions expressed in English statements which can be recognized (распознавать; узнавать) by the compiler and translated into a sequence of machine code instructions capable of being used by the computer to solve the problem.

The advantages of using COBOL are that it is relatively simple to learn, and programs can be quickly written and tested; programmers can easily understand programs not written by themselves, and thus associated documentation

can be simplified; and programs can be used on other machines, within the limitations noted above.

The disadvantages are: (1) the relative inefficiency of the resulting object program as compared with a program written in machine code or a machine oriented language and (2) the lack of flexibility (гибкость) imposed by the restrictions (ограничение) on the type of instructions and methods for performing operations in a highly standardized language.

Notes

[1] **Identification Division** — раздел идентификации
[2] **Environment Division** — раздел оборудования
[3] **to deal with** — иметь дело с
[4] **tape decks** — комплекты (наборы) перфолент
[5] **the step-by-step instructions** — поэтапные команды

XI. Listen to Text C from the tape recorder. Give its contents in short (in Russian).

Notes

[1] **a software package** — комплект программного обеспечения
[2] **to request** — запрашивать
[3] **a view** — вид, изображение
[4] **to reside** — размещать(ся) (в памяти)
[5] **a non-procedural language** — непроцедурный язык

TEXT C. DATA BASE MANAGEMENT SYSTEM (DBMS)

The Data Base Management System is a software package [1] which acts as an interface between the user's program and the physical data base. The DBMS makes it easier to access all varieties of data or information stored in a computer. It allows users to request [2] data from the computer, and keeps track of all the data. It also allows each user to have an individual view [3] of the data.

If a DBMS is not used, it requires more detailed programming to access data. The user's application program asks the DBMS to select that user's view and deliver it to the program or user. Only the DBMS knows where and how to get it.

The DBMS acts as a buffer between the programs and the physical structure of the data base. A portion of the DMBS resides [4] in the memory and is called by the application program each time when data must be transferred to or from the data base.

The main DBMS features are: data independence, security, application of high-level non-procedural languages.[5]

LESSON ELEVEN

Text A. High Level Programming Languages *(continued)*: BASIC & PASCAL.
Text B. Ada & C.
Text C. Procedural and Non-Procedural Languages.

Exercises

I. Read the international words and guess their meaning:

BASIC; PASCAL; original; popularity; popular; version; compiler; interpreter; dialect; authority; documentation; to ignore; block; parameters; syntactic; novelty; file; section; line; basic; to limit; interactive; calculator; principal; principle; structure; action; declaration

II. Translate the following sentences paying attention to the meaning of the words and word combinations given below:

a) **a feature** — черта; особенность; признак
to feature — отличать(ся); быть характерной чертой

1. In this chapter we shall discuss the possible features that control panel might have. 2. The main feature this robot has is that it is given "hands" adjusted delicately enough to thread a needle under water. 3. ALGOL 60 has different declarations, such as: type declarations, array declarations, switch declarations, and procedure declarations, whereas ALGOL 68 features the 'identity declarations', whose expressive power includes all of these. 4. The low level languages are oriented to a machine, while the high level languages feature the fact of being problem oriented languages.

b) **a record** ['rekɔ:d] — запись
to record [rɪ'kɔ:d] — записывать

1. A record of bytes on the floppy disks is read by the read/write head. 2. The problem of records of numbers has pressed upon human beings for more than five thousand years. 3. The reading equipment records digits and letters as they appear on a printed page. 4. In the world of business there are large quantities of records which are necessary to be handled by a computer. 5. To record each bit of information a computer must have the special equipment.

c) **whether** ['weðə] — ли
whether . . . or not — ли ... или нет; в любом случае
whether . . . or — или ..., или; то ли ..., то ли; либо ..., либо

1. Whether the control function is executed as an immediate operation depends on the operation and the type of device. 2. Science fiction stories sometimes give rise to questions as to whether a conflict is possible between men and machines. 3. The condition code indicates whether or not the channel has performed the function specified by the instruction. 4. Whether or not the transfer of control operation actually occurs depends upon the results of the previous logical operation. 5. This program is used when it is necessary to define whether the device is out of order, or there are some errors in the previous program. 6. The type declaration is used to specify whether a variable is an integer, or a Boolean variable.

III. Translate the following sentences paying attention to the comparison degrees:

1. The commonest problem is there where a word could mean one of two different things. 2. A television set, a computer, a telephone system are simpler than a human being. 3. Another advantage in this case is that less power is required to run a computer. 4. A computer does arithmetic problems million times faster than any person. 5. The smaller the computer, the faster it can work. 6. The better we know the Universe, the better we know our Earth. 7. The symbols representing intermediate results appear later in the flow chart of the right-hand side of the equation. 8. In this case the instruction puts the number U into the largest number cell. 9. The left-hand function box in the flow chart represents 005 on the next lower level, and the lowest box represents 007.

IV. Pronounce the following words correctly:

solve [sɔlv] *v* решать; разрешать; растворять
interactive [ˌɪntər'æktɪv] *a* взаимодействующий; интерактивный
time-sharing ['taɪm'ʃeərɪŋ] с разделением времени
widely ['waɪdlɪ] *adv* широко
extremely [ɪks'tri:mlɪ] *adv* чрезвычайно
version ['və:ʃ(ə)n] *n* версия; вариант
hand-held ['hændheld] *a* карманный; портативный
easy ['i:zɪ] *a* легкий; нетрудный
conversational [ˌkɔnvə'seɪʃənl] *a* разговорный
however [hau'evə] *adv* однако
inherent [ɪn'hɪər(ə)nt] *a* присущий; свойственный
decipher [dɪ'saɪfə] *v* расшифровывать
coherently [ko(u)'hɪər(ə)ntlɪ] *adv* логически связно
concern [kən'sə:n] *v* касаться, иметь отношение
remove [rɪ'mu:v] *v* удалять; перемещать
cassette [kə'set] *n* кассета
ignore [ɪg'nɔ:] *v* игнорировать
valuable ['væljuəbl] *a* ценный; дорогостоящий
propose [prə'pouz] *v* предлагать
revise [rɪ'vaɪz] *v* пересматривать; перерабатывать

implementation [ˌɪmplɪmen'teɪʃn] *n* реализация *(языков)*; внедрение

successor [s(ə)k'sesə] *n* преемник

novelty ['nɔv(ə)ltɪ] *n* новизна; новинка; новшество

afford [ə'fɔ:d] *v* предоставлять, давать

suitable ['sju:təbl] *a* подходящий; соответствующий

heading ['hedɪŋ] *n* заглавие

body ['bɔdɪ] *n* основная часть, тело программы

compulsory [kəm'pʌls(ə)rɪ] *a* обязательный

identifier [aɪ'dentɪfaɪə] *n* идентификатор

V. Memorize the following word combinations:

to become extremely popular — стать чрезвычайно популярным

the first-time programmer — программист-новичок

structured programming — структурное программирование

a general-purpose language — универсальный язык

TEXT A. HIGH LEVEL PROGRAMMING LANGUAGES: BASIC & PASCAL

1. **BASIC** was developed in 1965 and stands for Beginners All-purpose Symbolic Instruction Code. It is a programming language designed for solving mathematical and business problems. BASIC was originally developed as an interactive programming language for time-sharing on large mainframes. It is widely used on all sizes of computers and has become extremely popular on microcomputers.

2. There are many different versions of BASIC available with limited versions running on small hand-held computers. BASIC is available in both compiler and interpreter form, the latter form being more popular and easier to use, especially for the first-time programmer. In interpreter form the language is conversational and can be used as a desk calculator. In addition, it is easy to debug a program, since each line of code can be tested one at a time.

3. BASIC is considered to be one of the easiest programming languages to learn. For simple problems BASIC programs can be written 'on the fly',[1] at the terminal. However, complex problems require programming technique, as in any conventional programming language. Since BASIC does not require a structured programming approach, like PASCAL, and since there is no inherent documentation in the language, as in COBOL, BASIC programs can be difficult to decipher later if the program was not coherently designed.

4. BASIC is now used almost universally. There is no one BASIC language, but something like 90 different versions

or dialects; however, all have certain common features that make it easy to use any version once the fundamentals are mastered. Since BASIC is job and human oriented, it cannot be understood by the computer as written, but must go through the intermediate step of a compiler or interpreter, as was said above. As far as[2] the programmer is concerned, it makes very little difference whether a compiler or interpreter is used.

5. A compiler, generally used in a large computer, converts the source program written in BASIC to an object program or file in machine language which is then stored in memory. In the compiler each BASIC phrase or statement is converted to one or more machine instructions. An interpreter is similar in result, but conversion is usually done while the program is running, one statement at a time. The difference between the two, which is important to the microprocessor user, is that the interpreter must be preset in memory while the program is being run, while the compiler can be removed once it has done its job.

6. One would think then that a microcomputer would most often use a compiler, but this is not the case.[3] Since interpreter programs can be run line-by-line,[4] they can be debugged simply rather that being recompiled for each correction, and they are more popular in small computers. In certain versions the interpreter is stored in ROM, which is less expensive than RAM, and does not need to be loaded from some external source such as a cassette.

7. The typical example of the algorithm in BASIC can be written as follows:

```
1. REM THIS PROGRAM SEARCHES A LIST AND
PRINTS THE ADDRESS
10 DATA 74, 83, 66, 67, 87, 65, 84, 80, 76, 70
20 LET N=10
30 LET X=65
40 LET J=N
50 IF J=0 GOTO 100
60 READ K
70 IF K=GOTO 100
80 LET J=J—1
90 GOTO 50
100 PRINT J
110 END
```

BASIC features the fact that every line is a statement and every statement must be preceded by a line number followed by space. Any statement on a line beginning with

REM is ignored by the interpreter or compiler. However, these REMarks [5] may be extremely valuable in explaining the purpose and method of the program. Some BASIC variations use the apostrophe (') as an abbreviation for REM.

8. BASIC has various expressions (constants and variables combined by arithmetic and algebraic operators), line numbers, spaces, remarks, data, and statements. BASIC statements may be: LET statement which is the simplest kind of an arithmetic assignment statement, READ statement, GOTO statement, IF statement, IF THEN statement, etc. In the case of the IF statement we are interested in whether the relation between two expressions following the IF is TRUE or FALSE. In other words we are interested in the Boolean value of the expression following IF.

9. **PASCAL.** PASCAL is a general-purpose high level programming language. It is named after the famous French mathematician, Blaise Pascal, who in 1642 designed and built the first mechanical calculator, the "Pascaline". PASCAL is noted for its simplicity and structured programming design. It is available as both a compiler and an interpreter.

10. PASCAL was proposed and defined in 1971, and gained popularity in universities and colleges in Europe and the United States. It was later revised and appeared as standard PASCAL in 1975. Its principal features are on teaching programming and on the efficient implementation of the language.

11. PASCAL may be considered a successor to ALGOL-60, from which it inherits [6] syntactic appearances.[7] The novelties of PASCAL lie mainly in extensive data structuring facilities such as record, set and file structures. It also affords more sophisticated control structures suitable to structured programming.

12. An algorithm of a computer program consists of two essential parts: a description of actions which are to be performed, and a description of the data, which are manipulated by these actions. Actions are described by statements, and data are described by declarations and definitions.

13. The program is divided into a heading and a body, called a block. The heading gives the program a name and lists its parameters. These are file variables and represent the arguments and results of the computation. The file output is a compulsory parameter. The block consists of six sections. They are: label declaration part, constant definition part, type definition part, variable declaration part,

procedure and function declaration part, and statement part.

14. The first section lists all labels defined in this block. The second section introduces identifiers for constants. The third section contains type declarations, and the fourth—variable definitions. The fifth section defines procedures and functions. And the last, the sixth, gives the statements which specifies the actions to be taken.

15. The statements used in PASCAL may be: EMPTY[8] statement, GOTO statement, structured statement, compound statement,[9] conditional statement,[10] repetitive statement,[11] WITH statement, etc. For example, IF statement ::=IF expression THEN statement IF expression THEN statement ELSE statement.[12]

Notes

[1] **on the fly** — с ходу; прямо
[2] **as far as** — поскольку
[3] **this is not the case** — это не так
[4] **line-by-line** — построчно
[5] **remark** — замечание; примечание; пометка
[6] **to inherit** — перенимать
[7] **syntactic appearances** — синтаксические признаки
[8] **empty** — пустой
[9] **compound statement** — составной оператор
[10] **conditional statement** — условный оператор
[11] **repetitive statement** — оператор цикла (повторений)
[12] **ELSE statement** — оператор ИНАЧЕ

Exercises

VI. Read and translate the verbs meaning repetition:

retype; recompile; recycle; reuse; re-emphasize; relocate; reread; rewrite; reoccur; rearrange; reappear; replace; restart; rewind; review; return

VII. Read and translate the words meaning negation:

unusual; unused; unspecified; unlimited; unsatisfactory, unfrequently; unseparated; independent; indirect; indistinguishable; impossible; disadvantage; disjunction; decode; regardless; useless

VIII. Find in (b) the Russian equivalents to the following word combinations in (a):

a) 1. the Boolean value; 2. repetitive statement; 3. identifiers for constants; 4. type declaration; 5. step-by-step; 6. line-by-line; 7. hand-held computers; 8. to debug a prog-

ram; 9. basic features; 10. this is not the case; 11. conditional statement; 12. general-purpose languages

b) 1. основные особенности; 2. идентификаторы постоянных величин; 3. построчный; 4. отладить программу; 5. описание типа; 6. портативные компьютеры; 7. Булево значение; 8. оператор повторений; 9. поэтапный; 10. условный оператор; 11. универсальные языки; 12. это не так

IX. Memorize the following definitions:

1. BASIC is a programming high level language designed for solving mathematical and business problems. It is a problem oriented language extensively used with microcomputers and microprocessors. 2. PASCAL is a general-purpose high level programming language named after the French mathematician Blaise Pascal.

X. Answer the following questions:

1. What is BASIC? 2. What kinds of problems is BASIC designed for? 3. Why is the BASIC language popular on microcomputers? 4. Is it easy to debug a program written in BASIC? 5. What BASIC statements do you know? 6. What is PASCAL? 7. Who is PASCAL named after and why? 8. When did PASCAL appear as standard language? 9. How is a program in PASCAL divided? 10. How many sections does a block consist of? 11. Does PASCAL have the block structure? 12. What statements in PASCAL can you name?

XI. a) Compare BASIC and PASCAL; b) Speak on the difference in BASIC and PASCAL structures.

XII. Read and translate Text B without a dictionary. Render it in Russian:

TEXT B. ADA & C

ADA is a high level programming language. It is a PASCAL-based language, but much more comprehensive (обширный, всесторонний) than PASCAL, being designed for both commercial and scientific problems. ADA is a compiler language which can be compiled in separate segments and is noted for its multitasking capabilites.[1]

ADA was named after Augusta Ada Byron (1815-1852), daughter of the English poet, Lord Byron.

C is a high-level structured programming language. It is a compiler language too which is noted for its ability to handle conditions that normally would have to be written in an Assembly Language. Some operating systems are written in C.

Note

[1] **multitasking capability** — возможность обработки многих задач

XIII. Listen to Text C from the tape recorder and reproduce it in English.

Notes

[1] **looping** — организация циклов
[2] **query language** — язык запросов
[3] **report writer** — язык по написанию отчетов

TEXT C. PROCEDURAL AND NON-PROCEDURAL LANGUAGES

Procedural language is a language requiring the use of programming discipline. Programmers, writing in procedural languages must develop a proper order of actions in order to solve the problem, based on a knowledge of data/information processing operations and programming techniques, such as looping.[1] All conventional programming languages are procedural languages.

Non-procedural language is a language which does not require programming techniques. Non-procedural languages allow a user or a programmer to express a request to the computer in English-like statements, which specify what is to be done rather than how it is to be done. Query languages,[2] report writers,[3] and financial planning languages are examples of non-procedural languages.

Non-procedural languages generate the necessary program logic for the computer directly from a user's description of the problem.

PART II

TEXTS FOR SUPPLEMENTARY READING

FROM THE HISTORY OF THE COMPUTER

The introduction of agriculture revolutionized ancient man's social, economic, and cultural potential. This was the first great step in the evolution of civilization. The more recent Industrial Revolution, vastly increasing man's productive capabilities, was the next great step and brought forth our present highly mechanized economic and interdependent social civilization.

Nowadays we have another new kind of revolution, based on machines that greatly increase man's thinking capabilities of planning, analyzing, computing, and controlling. Hundreds of millions of computers are already in daily use penetrating almost all spheres of our modern society, from nuclear energy production and missile design to the processing of bank checks and medical diagnoses.

The development of mechanical calculating machines made the digital computers necessary. An ordinary arithmometer and a desk key calculator have given rise to electronic digital computers. Digital computers came into being in the first half of the 17th century. Many outstanding Russian and foreign mathematicians of that time created mechanical calculating devices.

The famous Russian scientist M. V. Lomonosov compiled a lot of calculating tables and several computing devices concerning the different fields of science and engineering.

In 1874 the Russian engineer V.T. Odner invented a special counter wheel [1] named after him the Odner's wheel which is used in modern arithmometers and calculators.

P. L. Chebyshev, academician, made a valuable contribution to the field of computing machine. He is known to have many good ideas in mathematics, some of which have been named after him. For example, the Chebyshev's polynomials play a unique role in the field of orthogonal func-

tions. In 1878 he constructed the original computing machine which was exhibited in Paris. In 1882 P. L. Chebyshev invented an arithmometer performing automatically multiplication and division. The principle of automatization put into this computing machine is still widely used all over the world when developing the most modern computers.

In 1884 Russia began to manufacture computing machines. In the period of World War I the output of computing machines ceased and was resumed only in the years of the Soviet Power.

At the end of the 1930s computing engineering began the new era. Electronic computers operating at high speed appeared, with electronic devices and units being applied.

The rapid advance of computers followed the success achieved by electronics. There appeared a possibility to solve complex mathematical problems within an unusually short time. The modern computing engineering enables to do the amount of calculations and researches within a week's time which would have required years of laborious work of large groups of people before.[2]

In the Soviet Union the first electronic digital computer was developed in the Ukrainian Academy of Sciences under the guidance of S. A. Lebedev in 1950. Then, in 1953 the BESM (the large-size electronic computing machine) was designed by the USSR Academy of Sciences. By means of the BESM, the system of 800 unknown values was solved in 20 hours. During the course of computation about 250 million operations were performed. A human being could solve such a problem in 300 years of continuous work.[3]

The BESM was followed by a number of types of large-, medium-, and small-size general and special purpose computers [4] such as Arrow, Ural-1, Ural-2, Minsk-1, etc. It was the first generation computers constructed on electronic tubes.

The second generation computers were solid-state large-powered machines. They were BESM-4, BESM-6, Minsk-22M, Minsk-32, Ural-14, Ural-16, Razdan-3, M-220 and others.

Nowadays more contemporary computers made on the integrated circuits are commercially available. They are the computers of the third and fourth generations. Among them there are such machines as Nairi-3 made on integrated hybrid microcircuits, the Unified System (ES) of electronic computers 1022, 1033, 1045, 1055, 1066, etc., made on integrated circuits with application of advanced designing-technological achievements, which are developed by the

socialist countries—members of the CMEA (the Council for Mutual Economic Assistance).

Nairi-4 is the representative of the fourth generation the speed of which has been increased by 5 times as compared with the speed of the third-generation computers thanks to using the multilayer printed circuits. The computers of the fourth generation are based on LSI circuits containing tens and hundreds of thousands of active electronic devices in tiny elements. Researches are also being done on computers based on superconducting devices.

A major advance in the development of computer technology was the creation of microprocessors and microcomputers. The tiny computing devices are able to control complex operations. Soviet industry has already started mass production of microprocessors and microcomputers which will be soon of vital help to man everywhere.

The fifth-generation computers are expected to appear by the turn of the century. They will be based on VLSI and SLSI technologies, optical fibers, videodisks, and artificial intelligence [5] techniques will be incorporated into them.

Invention of electronic computers is one of the greatest achievements of mankind. The significance of it can be compared with the invention of the steam-engine at the end of the 18th century and the utilization of atomic energy.

Notes

[1] **a special counter wheel** — специальное счетное колесико
[2] **which would have required years of laborious work of large groups of people before** — которые потребовали бы раньше годы трудоемкой работы большой группы людей
[3] **A human being could solve such a problem in 300 years of continuous work.**— Человек мог бы решить такую задачу за 300 лет непрерывного труда.
[4] **general and special purpose computers** — универсальные и специализированные компьютеры
[5] **artificial intelligence** — искусственный интеллект

THE USSR ACADEMY OF SCIENCES

Soviet scientists are making a great contribution to the development of world science and technology in all the main fields. Today there is probably no area of human knowledge [1] in which works of Soviet scientists do not play an important part.

Soviet scientists have reached great successes in mathematics, physics, chemistry, medicine, geology, the engi-

neering sciences and other fields of knowledge. They have constructed new modern automatic production lines, initiated laser technology, and produced new types of computers, microprocessors and microcomputers.

The USSR Academy of Sciences is responsible for [2] the general guidance (руководство) of research in the main areas of the natural and social sciences. Founded in St. Petersburg in 1724, it was moved to Moscow in 1934. The USSR Academy of Sciences has nearly 700 full members (Academicians) and Corresponding Members, and more than 70 honorary members from other countries. The Academy's main tasks are to promote basic research directly connected with production development, to determine potential for the technical progress and contribute to the fullest use of scientific innovations in the USSR.

Notes

[1] **there is probably no area of human knowledge** — вероятно, нет такой области человеческих знаний
[2] **to be responsible for** — быть ответственным за

AT THE COMPUTING CENTRE OF THE USSR ACADEMY OF SCIENCES

The Soviet State gave great support to the development of computing engineering as soon as this branch of science appeared. Several computing centres are known to have been established (основывать) all over the country. It is from these centres that computational mathematics started to penetrate into all other branches of science, economic planning and industrial and agricultural management.

The Academy of Sciences Computing Centre was set up on the initiative of Academician M. Lavrentyev. The centre developed new numerical methods, worked out (разрабатывать) the ways and means of automating programming, solved practical problems for various institutes and perfected (совершенствовать) new computers.

As the centre developed, its work became more and more complex. There are many complicated problems to be solved in science and industry with the help of new methods. Mathematical methods of planning and industrial management involving electronic computers are being used on the increasing scale. Soviet scientists have developed electronic computers capable of performing up to hundred million operations per second and much more.

The Computing Centre of the Academy of Sciences of the USSR pays much attention to training specialists. It holds (проводить) scientific engineering seminars and gives consultations to the students. The Computing Centre maintains (поддерживать) extensive (обширный) scientific contacts with institutes and organizations of other countries of the world. Its leading scientists often make reports at international conferences and symposiums.

NEWS IN THE COMPUTER FIELD

1. **Holography and Computer Memory.** A new polymer to coat holographic plates—reoxane [1]— has been developed in Leningrad. Reoxane will bring about many changes both in holography and in other areas—computers, for instance. These machines required devices with a huge memory capacity. It is now believed that holographic memory is more suitable for these purposes, as any data from it can be retrieved (*зд.* находить информацию) in a microsecond and the cost of this device will be comparatively moderate. The limit of reoxane memory has not been measured, but it is known that it is hundreds and possibly thousands of times larger than electronic memory.

2. **Computing at the Speed of Light.** A group of scientists of the Edinburgh University has been investigating the remarkable properties of an indium-antimony compound with a refractive index [2] that varies with the intensity of light falling on it. They have found that a tiny increase in the light incidence [3] can produce a large change in the light emerging.[4] This brighter emerging beam [5] persists even when the triggering light beam [6] is reduced again in brightness. In this way the crystal acts as a switch and "remembers" the triggering light impulses. The group has incorporated these properties into new computer components, which operate on low-powered laser beams and can respond in picoseconds. So the modern electronic computer may be replaced by much faster equipment that uses optical devices. In the future such computers which include these optical devices will be probably called the optical computers.

3. **Current Injection Logic Circuits.**[7] Scientists from the IBM Company have invented a new family of experimental computer circuits. Called "current injection logic" circuits, they operate in 13 trillionths of a second taking 7 picoseconds to perform their switching function and another 6 picoseconds to send an electric signal from one circuit to another.

In fact, the speed of an electrical signal moving between these new circuits is only limited by the speed of light.

These circuits function at temperatures close to absolute zero (—273° Celsius), where many metals lose all resistance to electrical current flow. But these "superconducting" circuits operate three times faster than any of circuits previously developed. Because of their speed and low heat output, the new circuits promise a future generation of more powerful computers.

Notes

[1] **reoxane** [rɪə'kseɪn] — реоксан (полимер)
[2] **an indium-antimony compound with a refractive index** — соединение индия с сурьмой с показателем преломления
[3] **the light incidence** — угол падения света
[4] **the light emerging** — появление света
[5] **This brighter emerging beam** — Этот более яркий появляющийся луч
[6] **the triggering light beam** — запускающий световой луч
[7] **current injection logic circuits** — схемы с токовой инжекторной логикой

SUPERCOMPUTERS

By 1989, in Japan, the $ 200-million supercomputer project plans to produce a computer that is 1,000 times faster than any currently available.[1] Such a computer would be capable of executing 10 billion floating-point operations per second. These research points in this project are divided into two major areas. One is concerned with very high speed devices, and the project staff [2] is studying three candidate technologies—galium arsenide,[3] high-electron-mobility transistor, and Josephson junction. The other area concerns ultra-parallel processing. Here, they are directing major research efforts not only toward computer architectures but also toward algorithms for parallel processing, operating systems, and programming languages.

This project has stimulated activity in several branches of the United States. Considering Japan's rapid advances in hardware technology and the relatively small technological edge [4] that the United States maintains in supercomputer software, such reactions seem natural.

Notes

[1] **currently available** — *зд.* выпускаемый серийно
[2] **the project staff** — сотрудники, работающие над проектом
[3] **galium arsenide** — арсенид галия
[4] **the technological edge** — технический перевес

THE FIFTH-GENERATION PROJECT

Until May 1982, when the Institute for New-Generation Computer Technology was established, the Fifth-Generation Computer System Project was guided by the Japan Information Processing Development Centre. The project will last for 10 years. It became clear that Japanese computer scientists participate in this project with great pride, viewing it as a way of bringing Japan to the forefront in all aspects of computer technology. But the main thrust [1] will be toward finding solutions to current difficulties in software. Some of the researchers emphasized that Japan had not invented a single programming language and that the time had come for creativity in computer science research.

The project staff has declared that 5G (the fifth-generation) will be a knowledge information processing system [2] based on innovative theories and technologies offering the advanced functions that will be required in the 1990s. As they envision it, 5G will have the following characteristics: high-level user interfaces based on artificial intelligence approaches for interaction via natural languages, including speech input; knowledge bases, containing both general knowledge to support man—machine interactions and specific knowledge of the problem area; flexibility and high reliability and performance.

Clearly, a great deal of attention will be given to artificial intelligence techniques applicable to problem solving and inference [3] and to the man—machine interface. The other key technologies to be dealt with are software engineering, VLSI, and non-von Neumann architecture.[4]

Notes

[1] **the main thrust** — главный стимул (толчок)
[2] **a knowledge information processing system** — система обработки информации на основе знаний (сведений)
[3] **inference** — вывод, заключение
[4] **non-von Neumann architecture** — не-Неймановская архитектура (структура), нетрадиционная архитектура

COMPUTERS COMPETE

The first international chess match was played over 100 years ago. Naturally, the contestants were fellow creatures.[1] But an automatic "chess-playing device" had appeared as back as 1769. Kempelen, its inventor, had toured many European countries demonstrating its power. In 1809 the

machine played in Vienna against Napoleon. The record of the moves [2] has been preserved. Napoleon lost the game.

Kempelen's "computer" didn't operate on radio tubes or transistors. Its secret lay in a crack player [3] being secretly hidden inside.

But the time has come for real computers to have a go.[4] The electronic computer of Stanford University in the USA pitted its "wits" against [5] its counterpart of the Institute of Theoretical Experimental Physics (USSR, Moscow). Four games were played simultaneously. Men acted as "coaches", and they had provided the programs for the machines. Strictly speaking, it is a match between programs. Which is more perfect? That's the answer the match has to provide. Hence the computers are doing things calmly [6] while the people, above all [7] mathematicians, are all worried.[8]

In this case, chess for the mathematicians is but one of means to an end.[9] Some of the most respected publications declared that a machine could never do anything that required thought, that it would never learn to play chess. The mathematicians retorted with: "It all depends on how you teach the machine." Now there can be no talk about whether a computer can or cannot play chess. Because it can.

The Soviet program "Caissa", developed by Moscow scientists headed by the former world chess champion Mikhail Botvinnik, has won the world chess championship for computer programs several times.

Notes

[1] **fellow creatures** — люди
[2] **move** — *зд.* ход (в шахматной игре)
[3] **a crack player** — первоклассный игрок
[4] **to have a go** — сделать шаг
[5] **to pit "wits" against** — сразиться в «остроумии» с
[6] **calmly** — спокойно
[7] **above all** — больше всего, главным образом
[8] **to be worried** — волноваться, беспокоиться
[9] **but one of means to an end** — лишь одно из средств для достижения цели

HAND-HELD COMPUTER

The electronic translator, a calculator-like device, offered travellers a computerized foreign-language dictionary. Two US companies have recently put such devices on the market. One of Japanese companies said it would begin manufacturing a translator device later. However, it said

its machine would be more than just a translator. In fact, the company said it would be making the world's first all-purpose, hand-held computer.[1]

This machine will be the same translator device but it will make use of several attachments: those now of the drawing board [2] include a mini-printer that effectively turns the translator into a typewriter, a video screen [3] that would make the machine function as a miniature computer-display terminal and a voice synthesizer [4] so that users can hear as well as see the information stored in the machine. Through the use of programmed information capsules [5] the machine will also be able to function as an electronic encyclopedia, putting sports, arts, history and other facts at the user's fingertips.[6] With capsules that users can program themselves, the computer applications become even broader, extending to home accounting,[7] telephone and address filling and even recipe (рецепт) indexing.

The heart of this as-yet-unnamed device [8] is a new memory bank. Despite its tiny size it can hold about twice the data of memories used in pocket-size electronic machines. This machine will be available for export sometime in future years.

Notes

[1] **the world's first all-purpose hand-held computer** — первый в мире универсальный портативный компьютер
[2] **now of the drawing board** — сейчас на чертежной доске
[3] **a video screen** — видеоэкран
[4] **a voice synthesizer** — речевой (голосовой) синтезатор
[5] **capsule** — капсюль
[6] **to put facts at smb's fingertips** — *эд.* вводить факты в компьютер кончиками пальцев
[7] **home accounting** — ведение домашних расчетов
[8] **this as-yet-unnamed device** — это еще не названное устройство

VOLUMETRIC ELECTRONICS [1]

"Polytron" is the name of a new vacuum electronic device developed by Soviet scientists. This device possesses outstanding abilities.

Radio valves or semiconductor transistors in previous-generation electronic devices had an impressive number of capacitors, resistors, ferrits, etc. "Polytron" hardly needs any of them. The most complicated processes occur in its "maw" [2] only by means of several auxiliary units. Moreover, it consumes little power.

Two remarkable features of electrons—radiation flows [3] and undulation processes [4]—are known to have been previously used separately in two independent classes of electronic equipment. The first class comprises radio receivers, TV sets and computers. The second one includes microwave devices such as radars, for example.

For a long time scientists, egineers and inventors have been engaged in solving the problem of how to combine the two processes, radiation flows and electrons' undulation properties, in one device. The problem has been successfully solved by the authors of "polytron". When the device was constructed, experiments were set up to study its "abilities". Scientists have discovered valuable physical effects. It turned out to be that electrons obtained and controlled in the new device have demonstrated a volumetric character, i.e., they moved in three directions. Therefore, a new trend in science and engineering was coined (создавать): "stereoelectronics".

Up till now [5] in order to discern a definite sound or a voice, it was necessary to generate first a programme for them, and then insert the voice and sound data into a computer storage. A device developed on the "polytron" basis needs no programme. A sound uttered (произносить) or a voice is instantaneously (мгновенно) memorized and the apparatus itself discerns these signals among many others.

The discovered volumetric effect of the electrons' behaviour opens broad realms (область) of application of new devices in various fields of science and computer engineering.

At present in many laboratories of the world scientists are studying possibilities of creating "artificial intelligence". In their opinion, separate elements of man-made intelligence may turn out to be polytronic systems.

Notes

[1] **volumetric electronics** — объемная электроника
[2] **in its "maw"** — в его «утробе»; внутри
[3] **radiation flow** — поток излучения
[4] **undulation process** — процесс волнообразного движения
[5] **up till now** — до сих пор

ARTIFICIAL INTELLIGENCE

"Artificial intelligence" is a metaphoric figure of speech [1] designating an entire scientific trend which includes mathematicians, linguists, psychologists, engineers, and many

other specialists. The essence (суть) of this trend is the intensification of man's creative activities with the aid of computers.

Artificial intelligence (AI) is machine intelligence. It refers to applications of the computer which, in operation, resemble human intelligence. There are different categories of uses which all fall into the AI area. For example, robots or machines with sensory capabilities [2] which detect and recognize sounds, pictures, etc., are one category. Another category is knowledge based systems,[3] which contain a base of knowledge about a subject and can assist us in solving problems. Knowledge based systems being developed from the experience of human experts are called expert systems and can perform such tasks as medical diagnoses. AI will encompass (охватывать) many areas that have not been easily solved using traditional hardware and software.

AI will be incorporated into the 5th-generation computer systems. Then the average computer system should not require users to remember a lot of complex codes or commands. Rather, the user should ask: "Can you help me with this type of problem?" The master control program or operating system will be able to direct the user to the appropriate expert system through questions and answers.

AI programming is not magical; it does however imply a change in rules and methods for the traditional application programmer. Normal application programs follow a fixed algorithm: if this—do that. Given a set of input conditions, the output can be precisely determined. AI requires program design with more imagination. New methods of program organization and construction must be developed. AI programs may require the use of heuristic techniques,[4] which are exploratory in nature and use trial and error methods.[5] AI programs are often programmed in the LISP programming language, which allows the program designer to concentrate on the problem-solving logic more effectively than common languages like BASIC and COBOL.

It is impossible in principle to develop an artificial intelligence as it is sometimes understood literally (буквально), because the human brain is a very sophisticated system composed of tens of billions of interconnected cells. Each cell is extremely complex in itself. A rather probable hypothesis says that an individual cell processes the signals penetrating it like a computer. Therefore, even the most sophisticated machine we may imagine cannot even be compared to the brain. Man created the machine to fulfil his own re-

quirements. A machine cannot have either human feelings, desires or emotions. Can anyone imagine a machine in love with someone? And what about the experience a person receives during his life in society through dealing all the time with purely human problems? A machine, in general, cannot think, either logically or figuratively.

But nevertheless (тем не менее), the research trend of artificial intelligence will acquire ever greater importance as time goes by, because the programming and technical means of artificial intelligence will ensure us of the opportunity to associate directly with the machine without the aid of a huge crowd of engineers, economists, biologists, chemists, and many other specialists. The question of AI acquires special importance for economic planning and management. In conditions when production is becoming automated, management must become automated as well.

One of the trends in AI now being intensively developed is to design so-called thinking robots, capable of a certain amount of independent activities.

The model of creative processes in computers gave birth to the term "artificial intelligence". But that doesn't mean that the computers possess it. The "intelligence" has been packaged in it by an expert who developed the programme for solving some practical creative problem. Man differs from the machine in that he does not simply fulfil the programs stored in his memory, but also develops them himself, depending on the goals facing him.

Notes

[1] **a metaphoric figure of speech** — метафорический образ речи
[2] **sensory capability** — сенсорная способность (способность воспринимать)
[3] **knowledge based systems** — экспертные системы
[4] **heuristic** [hjuə'rıstık] **techniques** — эвристические приемы (методы)
[5] **trial and error method** — метод проб и ошибок

SOME FACTS ABOUT ROBOTS

It is estimated that by the end of the century the overwhelming majority [1] of people will be occupied in science, education, administration and the service industry. Machines and robots will be the primary producers of material wealth.

1. **A Robot Laboratory Assistant.** When a research worker studies biological objects, the brain, for example, he has to operate with tremendous figures: 15 thousand million neurons,

150 thousand million interneuron elements. One cubic millimetre of blood has 5 million red blood cells.[2]

Our scientists have developed a machine to count and determine these biological microobjects. This robot laboratory assistant does in one to three minutes the work that usually takes a month using conventional technique. This robot, besides counting the objects, also determines their size.

2. **Robot Assists Surgeon.**[3] A cybernetic device which collects information on the main physiological processes taking place in the organism of a patient undergoing an operation has been made at one of Leningrad's clinics. A special screen shows the patient's pulse rate, blood pressure and body temperature. It has been decided to improve the device to give recommendations to the operator of artificial heart. A computer being assembled at the clinic, after collecting and processing all the necessary information about the patient, will be able to control "heart-lung" apparatus [4] automatically.

3. **An Underwater Robot.** An underwater robot with the TV camera, the first of its kind, has been successfully tested in the Soviet Union. Intended for detailed studies of the ocean bed at depths as low as [5] 4,000 metres, it was remote-controlled [6] from the shore and could perform previously programmed operations. It could be given "hands" adjusted delicately enough to thread a needle under water.[7]

4. **Space Robots.** Space research in our country opens up vast perspectives for robots. Artificial satellites are already circulating the globe. The orbital stations are added to them nowadays. This equipment involves maintenance, repair, part replacement and other jobs which cannot be planned beforehand. Spacecrafts fitted with robots are already in use.

The great advantage of the space robots are that they need no life support.[8] They could be left in orbit for ever (навсегда), or sent off in a moon vehicle (летательный аппарат) to explore some remote area without anyone worrying about [9] whether it would come back or not. They are also very useful for repairing the outside of a spacecraft or for assembling a space station in orbit.

5. **Artificial Man for Medical Students.** Science fiction has become reality at a research laboratory in California with the creation of a 6 ft. 2 in. artificial man which breathes and blinks.

"Sim-1"—the first of a series of simulators—was created by several scientists from California universities. He is to serve as a "patient" for medical students. His inventors believe that increasingly complex Sims will be built, shortening from

eight to six the number of years of study required for a medical degree.

"Sim-1" human characteristics include ability to breathe, blink his eyes, open his jaws—which contain a full set of teeth—move his tongue and vocal cords,[10] and be affected by drugs (лекарство) in much the same way as a human being. Speech and mobility have not yet been built in, although the inventors say this would be no problem.

6. **Robots on a Stroll.** In 1910, news was received in Japan that a magic box (a robot using springs) was displayed at the World Exposition in London. This news captured the interest of a boy in the fifth form of an elementary school who started to construct his own robot of tin and cardboard.[11] Ten years passed and the boy, now a young man of 21, had already received a patent for a robot which could speak. This was the young Aizawa who, since then, has spent all his life in making toys.

Aizawa has created 700 robots, ranging from 2 meters tall to those of only 30 centimeters. But he is particularly proud of eight of his creations, which are life-size robot brothers. These eight robots not only walk, but also can shake hands, wink, bow, nod and talk. Numerous requests are received by Aizawa for the services of his eight robots, which are dispatched to children's halls and children's science museums.

Aizawa brought out two of eight brother robots for a stroll outside his home. One is Goro. He is 165 centimeters tall and weighs 123 kilograms. The other is Hachiro, weighing 60 kilograms and 115 centimeters in height.

Notes

1 **the overwhelming majority** — подавляющее большинство
2 **red blood cells** — красные кровяные шарики
3 **a surgeon** — хирург
4 **a "heart—lung" apparatus** — аппарат "сердце—легкие"
5 **at the depths as low as** — на глубинах до
6 **remote-controlled** — дистанционно управляемый
7 **adjusted delicately enough to thread a needle under water** — достаточно точно приспособленные, чтобы вдевать нитку в иголку под водой
8 **life support** — жизненные средства к существованию
9 **without anyone worrying about** — и никто не будет беспокоиться о том
10 **vocal cords** — голосовые связки
11 **tin and cardboard** — жесть и картон

'HERE COMES THE ROBOT ...'

The first robots appeared on Olympus: gold mechanical maids served Hephaestus, the god of fire, divine smith and

patron of craftsmen. Attempts to create a mechanical being looking like a man and capable of carrying out his work have penetrated with the history of civilization. The robots of today, industrial automatons and mechanisms, have no similarity with man but, being faster and more accurate than him when working, replace him in carrying out monotonous, and dangerous work ...

The word "robot", which has become an international technological term, was coined (создавать) in 1920 by the Czech writer Karel Chapek. He used it to designate a mechanical worker, looking like a man, possessing unusual physical strength, infinite capacity for work, and totally devoid of such human drawbacks as fatigue, distraction and emotions. Literature had known about such creatures long before Chapek. It is curious to note that all kinds of mechanical assistants described in literature suffered from anthropomorphism, i.e., they looked like a man. In practice there is absolutely no need for this.

The first robots, perhaps, did not yet have the right to be called such because they could not work by themselves, but worked only on operator's instructions, being in a safe place. The operators controlled a whole set of "multiarticulate" hands,[1] carrying out necessary actions for maintaining, for example, nuclear reactors, manipulated in sterile conditions, etc. The first manipulators were very heavy and clumsy. But the more engineering thought developed, the fewer anthropomorphous elements appeared. There were more flexible and mobile devices which could squeeze into narrow openings, move on flat vertical walls, and even on the ceiling.

One of the outstanding achievements of robot technology was the Soviet Lunokhod (Moon rover). The first space robot, an eight-wheeled apparatus, travelled more than 10 km on the Moon's surface in 1970-1971, and the second in 1973 investigated a 37-km-long route. Since then, space robots have done a lot of work on Mars and Venus, reporting valuable information to the Earth about our neighbours in the solar system.

By freeing man from routine physical work, robots greatly change the content of labour, resolve the manpower shortage [2] problem, and eliminate the need of man's presence in harmful and dangerous conditions. This is where the social significance of robot technology lies.

Most people have come across the term "industrial robots". These are programmable controlled automatic devices which can replace man in performing functions that were once

thought to be purely human operations. Nowadays several thousand industrial robots are already operating at Soviet enterprises across the country. But this is only a first step.

Here are some examples of industrial robots. Robots service a mechanics section at the Elektrosila Association in Leningrad, feeding billets into the clamps of programmed machine tools and then taking out the round shiny cylinders. Robots made at the Kommunar Automobile Factory in Zaporozhye work on assembly lines. There are several robots working at the forging and mechanical assembly shops, on milling machines and aggregate machine tools. The automatic apparatus Sever-1 has proved its worth [3] in laying main pipelines. It was created by scientists and designers of the Ye. Paton Institute of Electrical Welding of the Ukrainian SSR Academy of Sciences. Kirghiz machine builders have developed and successfully use robots by remote control of drilling work in the mountains. The robot working faultlessly at the Dynamo Electrical Engineering Works in Moscow lifts parts weighing up to 160 kg and moves them with a speed of 2.5 m/sec. The robot can easily manipulate heavy shafts, carrying them from lathe to lathe,[4] measure the length and diameter of parts, and accurately stack them.

Thanks to scientists' and designers' efforts, the family of industrial robots, man's friendly mechanical assistants, has been growing rapidly both qualitatively and quantitatively.

Notes

[1] **"multiarticulate" hands** — «многосуставные» руки
[2] **manpower shortage** — нехватка рабочей силы
[3] **to prove one's worth** — доказать свою пригодность, свое достоинство
[4] **from lathe to lathe** — от станка к станку

ROBOTICS AND ROBOT GENERATIONS

Robotics is the art and science of the creation and use of robots, i.e., in other words, robot technology. Today practically all sectors of the economy and industry are looking forward [1] to introducing industrial robots. But robot building is not simple and certainly not cheap. If every sector begins to build its own robots, it will be impossible to avoid unnecessary duplication of research and development, and large sums will be wasted. Therefore the need is to concentrate all efforts in robot technology in one pair of hands, in a powerful intersectoral scientific and technical organization. Only in this way it is possible to ensure the maximum standardization of

production of industrial robots and multipurpose automatic manipulators. There are two rational ways in the field of robotics. The first one is to build standardized modules—unified elements on the basis of which it will be easy to assemble, in different combinations, robots for the most varied of purposes. The second way is to create an inter-sectoral exchange fund [2] of robots so that research and development of designers in different technical fields should be within the reach of all interested organizations and enterprises.

The robots are divided into three generations: programmed, adaptive and intellectual. Characteristic of the first generation—the programmed robots—is that their control system acts according to a rigid oft-repeated programme [3] all the time. But the programmed robots are easily returned to various action programmes.

The adaptive robots, robots of the second generation, have been already worked out and will be widely applied in production at the close of this and the beginning of the next decade. Their fundamental difference from the first robot generation is the appearance of artificial sensors, which give the adaptive robots the ability to see, to hear and feel. The possibilities of them are immeasurably greater than the robots of the first generation.

The third generation—intellectual robots—will be able to perform intricate selective operations, and carry out practically autonomous work, not depending on the operator. Robots with artificial intelligence will be able to identify objects in a pile, select the objects in the appropriate sequence and assemble them into a unit. And then we shall be able to speak about a robot revolution in the economy, about a many-fold increase [4] of labour productivity, and the advent of a new age of industrial production—the age of fully automated enterprises and, maybe, whole branches of industry.

Notes

[1] **to look forward** — ожидать (с удовольствием)
[2] **intersectional exchange fund** — промежуточный обменный фонд
[3] **a rigid oft-repeated programme** — жесткая неоднократно повторяющаяся программа
[4] **a many-fold increase** — многократное увеличение

A

accept [ək'sept] *v* принимать (информацию)
access ['ækses] *n* выборка *(из памяти)*; обращение *(к памяти)*; доступ; **direct** ~ прямая выборка; непосредственный доступ; **memory** ~ выборка из памяти; обращение к памяти; **random** ~ произвольная выборка; произвольное обращение
accumulator [ə'kjuːmjuleɪtə] *n* накапливающий сумматор
accuracy ['ækjurəsɪ] *n* точность, правильность; четкость *(изображения)*; **adequate** ~ требуемая (достаточная) точность; **given** ~ заданная точность
acronym ['ækrənɪm] *n* акроним
add [æd] *v* складывать; суммировать; прибавлять; увеличивать
adder ['ædə] *n* сумматор, суммирующее устройство
address [ə'dres] *n* адрес; *v* адресовать; **call** ~ адрес вызова; адрес запроса; **current** ~ текущий адрес; **dummy** ~ фиктивный адрес; псевдоадрес; **jump** ~ адрес перехода; **load** ~ адрес загрузки
addressable [ə'dresəbl] *a* адресуемый, имеющий адрес
aid [eɪd] *n* помощь; метод; средство; пособие (учебное); **debugging** ~s средства отладки; **symbolic-coding** ~s средства символического кодирования
allocation [ˌælə'keɪʃ(ə)n] *n* размещение; **memory** ~ распределение памяти
alphanumeric ['ælfənjuː'merɪk] *a* буквенно-цифровой
amount [ə'maunt] *n* количество; сумма
amplifier ['æmplɪfaɪə] *n* усилитель
analysis [ə'næləsɪs] *n* анализ; теория; теоретические исследования; **cost-effectiveness** ~ инженерно-экономический анализ
analyst ['ænəlɪst] *n* аналитик; **system** ~ системный аналитик
AND [ænd] И (логическая функция или операция)
application [ˌæplɪ'keɪʃən] *n* применение; использование; прикладная задача; работа; **real-time** ~ работа в истинном масштабе времени
approach [ə'prouʧ] *n* приближение; подход; метод
arrange [ə'reɪndʒ] *v* размещать, располагать; монтировать
array [ə'reɪ] *n* матрица; решетка; сетка; массив; поле; таблица
assembler [ə'semblə] *n* программа «ассемблер»
assembling [ə'semblɪŋ] *n* сборка; монтаж; компоновка
assign [ə'saɪn] *v* назначать; присваивать
assignment [ə'saɪnmənt] *n* назначение; присвоение; распределение; **address** ~ присвоение адреса; **unit** ~ распределение устройств; **value** ~ присвоение значения
attribute ['ætrɪbjuːt] *n* признак; описатель (в ПЛ/1)
automaton [ɔː'tɔmət(ə)n] *n* автомат
auxiliary [ɔːg'zɪljərɪ] *a* вспомогательный; дополнительный
availability [əˌveɪlə'bɪlɪtɪ] *n* пригодность; доступность; наличие
available [ə'veɪləbl] *a* доступный; пригодный; имеющийся в распоряжении; **commercially** ~ реально существующий; серийно выпускаемый

B

batch [bætʃ] *n* группа; партия; пакет

behaviour [bɪ'heɪvjə] *n* режим работы *(машины)*; поведение

binary ['baɪnərɪ] *a* двоичный; бинарный; двучленный

bit [bɪt] *n* бит; (двоичный) разряд

blank [blæŋk] *n* пробел, пропуск; пустое место

Boolean ['bu:lɪən] *n* булево выражение; *a* булев; логический

bracket ['brækɪt] *n* скобка; **round** ~ круглая скобка; **square** ~ квадратная скобка; **statement** ~ операторная скобка

branch [bra:n(t)ʃ] *n* ответвление; (условный) переход

built-in ['bɪlt'ɪn] *a* встроенный, вмонтированный; внутренний

bus [bʌs] *n эл.* шина; канал *(информации)*

button ['bʌtn] *n* кнопка

byte [baɪt] *n* байт; слог

C

calculation [ˌkælkju'leɪʃ(ə)n] *n* вычисление, расчет, подсчет; счет; **decision** ~ вычисление выбора направлений; **function** ~ вычисление функции

call-in ['kɔ:l(')ɪn] *n* вызов

capacitor [kə'pæsɪtə] *n* конденсатор

capacity [kə'pæsɪtɪ] *n* способность; емкость; емкостное сопротивление; объем

card [kɑ:d] *n* карта; перфокарта; плата

carry ['kærɪ] *n* перенос; *v* переносить; вводить *(данные в машину)*; ~ **on** проводить *(исследования)*; ~ **out** выполнять

cell [sel] *n* ячейка; элемент; клетка

change [tʃeɪndʒ] *n* изменение, перемена; *v* изменять; заменять

character ['kærɪktə] *n* знак; символ; цифра; буква; признак

charge ['tʃɑ:dʒ] *n* заряд; *v* заряжать; **to be in** ~ быть ответственным за что-л.

chart [tʃɑ:t] *n* диаграмма; схема; график; карта; чертеж; **flow-**~ блок-схема, графическое представление

check [tʃek] *n* проверка, контроль; *v* проверять, контролировать

chip [tʃɪp] *n* чип, кристалл

circuit ['sə:kɪt] *n* схема; цепь, контур; простой цикл; **digital-switching** ~ цифровая переключающая схема; **large-scale integration** ~ большая интегральная схема; **printed** ~ печатная схема; **solid-state** ~ полупроводниковая схема

circuitry ['sə:kɪtrɪ] *n* схема; компоновка схем; схематика

clock [klɔk] *n* генератор синхронизированных импульсов; таймер

commercial [kə'mə:ʃ(ə)l] *a* серийный *(о машине)*; предназначенный для коммерческих задач

compatible [kəm'pætəbl] *a* совместимый; сочетаемый

compiler [kəm'paɪlə] *n* компилирующая программа; компилятор

computer [kəm'pju:tə] *n* вычислительная машина, компьютер; **general-purpose** ~ универсальный компьютер; **special-purpose** ~ специализированный компьютер

condition [kən'dɪʃ(ə)n] *n* условие; состояние; режим; ситуация

conductor [kən'dʌktə] *n* проводник; провод

connector [kə'nektə] *n* соединитель; (штепсельный) разъем; **variable** ~ переменный (логический) соединитель

contents ['kɔntənts] *n pl* содержимое; содержание

control [kən'troul] *n* управление; контроль; *v* управлять; **job** ~ управление потоком заданий; **off-line** ~ автономное управление; **on-line** ~ управление от центрального процессора

controller [kən'troulə] *n* регулировщик; контро́ллер

conventional [kən'venʃənl] *a* общепринятый; традиционный

conversion [kən'və:ʃ(ə)n] *n* преобразование; превращение; переход

copy ['kɔpɪ] *n* экземпляр; копия; отпечаток; *v* копировать; печатать
core [kɔ:] *n* сердечник; память на магнитных сердечниках
correspondence [,kɔrɪs'pɔndəns] *n* соответствие
counter ['kauntə] *n* счетчик; пересчетное устройство; *a* противоположный
cruncher ['krʌntʃə] *n* большая ЭВМ
current ['kʌr(ə)nt] *n* электрический ток; *a* текущий; **alternating** ~ переменный ток; **direct** ~ постоянный ток
curve [kə:v] *n* кривая; характеристика
cybernetics [,saɪbə(:)'netɪks] *n* кибернетика
cycle ['saɪkl] *n* цикл; период; *v* работать циклами

D

data ['deɪtə] *n pl* данные; информация; **alphanumeric** ~ буквенно-цифровые данные; **built-in** ~ встроенные (в программу) данные
debugging [dɪ'bʌgɪŋ] *n* наладка; отладка *(программы)*
decimal ['desɪm(ə)l] *n* десятичное число; *a* десятичный; **binary coded** ~ двоично-десятичное число
decision [dɪ'sɪʒ(ə)n] *n* решение
deck [dek] *n* пачка; колода перфокарт
declaration [,deklə'reɪʃ(ə)n] *n* описание; объявление (в ПЛ/1)
decoder ['di:'koudə] *n* дешифратор
define [dɪ'faɪn] *v* определять; формулировать *(задачу)*
degree [dɪ'gri:] *n* степень; порядок; градус
delay [dɪ'leɪ] *n* задержка; запаздывание; *v* задерживать; ~**line** линия задержки
density ['densɪtɪ] *n* плотность; концентрация
departure [dɪ'pɑ:tʃə] *n* уход; отклонение *(от заданной величины)*
description [dɪs'krɪpʃ(ə)n] *n* описание; характеристика
design [dɪ'zaɪn] *n* проект; конструкция; *v* проектировать; конструировать
designation [,dezɪg'neɪʃ(ə)n] *n* обозначение; наименование
designer [dɪ'zaɪnə] *n* проектировщик; конструктор; разработчик
desk [desk] *n* пульт; стенд; стол
destination [,destɪ'neɪʃ(ə)n] *n* место назначения (записи) информации
detection [dɪ'tekʃ(ə)n] *n* обнаружение; детектирование
determine [dɪ'tə:mɪn] *v* определять; вычислять; детерминировать
detour [dɪ'tuə] *n* уход; удаление; обход; объезд
develop [dɪ'veləp] *v* разрабатывать; развивать
development [dɪ'veləpmənt] *n* разработка; развитие; усовершенствование; развертывание; разложение *(формулы)*
device [dɪ'vaɪs] *n* устройство; прибор, механизм; элемент
different ['dɪfr(ə)nt] *a* различный, разный
digit ['dɪdʒɪt] *n* цифра, число; разряд; символ; знак
digital ['dɪdʒɪt(ə)l] *a* цифровой
dimension [dɪ'menʃ(ə)n] *n* размер; величина; объем; размерность
direct [dɪ'rekt] *a* прямой, непосредственный; постоянный
direction [dɪ'rekʃ(ə)n] *n* направление; управление; *pl* инструкция
discrimination [dɪs,krɪmɪ'neɪʃ(ə)n] *n* дискриминация; различение
disk [dɪsk] *n* диск; круг; **floppy** ~ гибкий диск
displacement [dɪs'pleɪsmənt] *n* смещение; сдвиг; перемещение
display [dɪs'pleɪ] *n* дисплей; *v* показывать
distinguishable [dɪs'tɪŋgwɪʃəbl] *a* различимый
distributer [dɪs'trɪbju(:)tə] *n* распределитель
divide [dɪ'vaɪd] *v* делить(ся)
division [dɪ'vɪʒ(ə)n] *n* деление; раздел; отделение; **data** ~ раздел данных (КОБОЛ); **environment** ~ раздел оборудования (КОБОЛ); **identification** ~ раздел идентификации (КОБОЛ);

procedure ~ раздел процедур (КОБОЛ)
drum [drʌm] *n* барабан; цилиндр
dummy ['dʌmɪ] *n* макет; замена; *a* фиктивный
duration [dju(ə)'reɪʃ(ə)n] *n* длительность; продолжительность

E

edge [edʒ] *n* фронт *(импульса)*; край *(карты, ленты)*; грань *(кристалла)*
edit ['edɪt] *v* редактировать
editor ['edɪtə] *n* программа «редактор»
effect [ɪ'fekt] *n* влияние; действие; эффект; результат
efficiency [ɪ'fɪʃ(ə)nsɪ] *n* эффективность; коэффициент полезного действия
effort ['efət] *n* усилие; напряжение; попытка; что-то достигнутое
elaboration [ɪ,læbə'reɪʃ(ə)n] *n* реализация; выполнение *(процедур)*
eliminate [ɪ'lɪmɪneɪt] *v* устранять, исключать; заменять
emergency [ɪ'mə:dʒ(ə)nsɪ] *n* непредвиденный случай; авария
emit [ɪ'mɪt] *v* испускать, излучать
emitter [ɪ'mɪtə] *n* эмиттер; источник; генератор
empty ['em(p)tɪ] *a* пустой, незаполненный; *v* освобождать; очищать
end [end] *n* конец, окончание
energize ['enədʒaɪz] *v* возбуждать
engineering [,en(d)ʒɪ'nɪəɪŋ] *n* техника; *a* технический
enter ['entə] *v* вводить; записывать; подавать
enterprise ['entəpraɪz] *n* предприятие; *v* предпринимать; пытаться
entire [ɪn'taɪə] *a* целый, весь
entity ['entɪtɪ] *n* объект; сущность; категория
entrance ['entr(ə)ns] *n* вход *(в подпрограмму)*
entry ['entrɪ] *n* вход; ввод; запись; подача; введенные данные; содержимое; компонент; элемент; **argument** ~ содержимое поля (адреса) аргумента (операнда); **command** ~ содержимое поля кода операции
environment [ɪn'vaɪər(ə)nmənt] *n* функциональные средства; оборудование; окружающая среда
equal ['ɨkw(ə)l] *a* равный; *v* равняться; уравнивать
equation [ɪ'kweɪʃ(ə)n] *n* уравнение; равенство
equipment [ɪ'kwɪpmənt] *n* оборудование; аппаратура; приборы; **computer** ~ аппаратура компьютера; **control** ~ аппаратура управления; **peripheral** ~ внешнее (периферийное) оборудование
erase [ɪ'reɪz] *v* стирать *(запись)*; разрушать *(информацию)*
error ['erə] *n* ошибка, погрешность; ~**-free** свободный от ошибок
establish [ɪs'tæblɪʃ] *v* основывать; устанавливать; учреждать; ~ **a file** выделять файл (массив)
establishment [ɪs'tæblɪʃmənt] *n* установление, основание; создание
estimate ['estɪmɪt] *n* оценка; ['estɪmeɪt] *v* оценивать
evaluation [ɪ,væljʊ'eɪʃən] *n* оценка; вычисление; оценочная функция
even ['ɨv(ə)n] *a* четный; ровный; равномерный; *adv* даже
event [ɪ'vent] *n* событие; исход; результат
exact [ɪg'zækt] *a* верный; безошибочный; точный
exceed [ɪk'si:d] *v* превышать; превосходить
excess [ɪk'ses] *n* избыток, излишек; остаток
exchange [ɪks'ʧeɪn(d)ʒ] *n* обмен; замена; *v* обменивать; заменять
exclusive [ɪks'klu:sɪv] *a* исключающий; исключительный
execute ['eksɪkju:t] *v* исполнять, выполнять; осуществлять
existence [ɪg'zɪst(ə)ns] *n* существование
exit ['eksɪt] *n* выход; выходной канал
expanded [ɪks'pændɪd] *a* расширенный *(о языке)*
experience [ɪks'pɪərɪəns] *n* опыт; опытность; *v* узнать по опыту
exponentiation [,ekspo(u)nenʃɪ'eɪʃ(ə)n] *n* возведение в степень

expose [ɪks'pouz] *v* экспонировать; выставлять
expression [ɪks'preʃ(ə)n] *n* выражение
extensible [ɪks'tensəbl] *a* расширенный; протяженный; экстенсивный
extension [ɪks'tenʃ(ə)n] *n* расширение; продолжение; протяженность
external [eks'tə:nl] *a* внешний, наружный

F

facility [fə'sɪlɪtɪ] *n* устройство; *pl* средства; оборудование
factor ['fæktə] *n* коэффициент; множитель; фактор
failure ['feɪljə] *n* повреждение; сбой; неудача; проигрыш
false [fɔ:ls] *a* ложный
fault [fɔ:lt] *n* повреждение; ошибка; недостаток; дефект
feature ['fi:tʃə] *n* черта, особенность, признак, свойство
feed [fi:d] *n* подача; питание; *v* (**fed**) подавать; питать
feedback ['fi:dbæk] *n* обратная связь
fetching ['fetʃɪŋ] *n* вызов; выборка *(данных из памяти)*
field [fi:ld] *n* поле; пространство; зона; область знаний
figure ['fɪgə] *n* цифра; число; рисунок, чертеж
file [faɪl] *n* файл; массив; картотека; *v* составлять (хранить) файл
film [fɪlm] *n* пленка; тонкий слой *(чего-л.)*; фотопленка
fitting ['fɪtɪŋ] *n* сборка; монтаж; подгонка; сглаживание
fixed [fɪkst] *a* фиксированный; неподвижный; постоянный
flexibility [ˌfleksə'bɪlɪtɪ] *n* гибкость; **software** ~ гибкость программного обеспечения
flip-flop ['flɪpflɔp] *n* триггер; триггерная схема
floating ['floutɪŋ] *a* плавающий; отключенный **~-point** с плавающей точкой (запятой)
floppy disk ['flɔpɪ'dɪsk] гибкий диск
flow [flou] *n* поток; течение, истечение; струя; **~-chart** блок-схема, схема потока информации
form [fɔ:m] *n* форма; бланк; *v* формировать; образовывать
fraction ['frækʃ(ə)n] *n* дробь; дробная часть; доля
framework ['freɪmwə:k] *n* основа; структура; строение
frequency ['fri:kwənsɪ] *n* частота
function ['fʌŋkʃən] *n* функция; зависимость

G

gain [geɪn] *n* усиление; коэффициент усиления; прирост
gate [geɪt] *n* вентиль; логический элемент; *v* пропускать; **AND** ~ схема И, схема совпадения
general-purpose ['dʒen(ə)r(ə)l'pə:pəs] *a* общего назначения, универсальный
generate ['dʒenəreɪt] *v* производить; создавать; генерировать
generation [ˌdʒenə'reɪʃ(ə)n] *n* производство; создание; поколение *(ЭВМ)*
generator ['dʒenəreɪtə] *n* генератор; преобразователь
go [gou] *n* шаг *(в шахматной игре)*; ~ **to** перейти на (оператор в языках программирования)

H

hand [hænd] *n* ручка; рукоятка; стрелка *(прибора)*
handle ['hændl] *n* ручка, рукоятка; *v* управлять; оперировать
hardware ['hɑ:dweə] *n* аппаратура; аппаратное обеспечение ЭВМ
head [hed] *n* головка; заголовок, рубрика; *pl* лицевая сторона *(монеты)*; **read/write** ~ универсальная головка (для считывания записи)
heading ['hedɪŋ] *n* заголовок; рубрика
hexadecimal [ˌheksə'desɪm(ə)l] *a* шестнадцатиричный
high-speed ['haɪ'spi:d] *a* быстродействующий; скоростной
holding ['houldɪŋ] *n* хранение *(информации)*; блокировка
hole [houl] *n* отверстие; пробивка; дыра; *pl* перфорация

I

identification [aɪ,dentɪfɪ'keɪʃən] *n* идентификация; отождествление
image ['ɪmɪdʒ] *n* образ; изображение; отображение; *v* изображать
implement ['ɪmplɪmənt] *n* орудие, инструмент; ['ɪmplɪment] *v* выполнять, осуществлять
implementation [,ɪmplɪmən'teɪʃən] *n* внедрение; ввод в работу; реализация
imply [ɪm'plaɪ] *v* означать; подразумевать, предполагать
impression [ɪm'preʃ(ə)n] *n* отпечаток; оттиск; (стереотипное) издание; впечатление
incompatibility ['ɪnkəm,pætə'bɪlɪtɪ] *n* несовместимость
increment ['ɪnkrɪmənt] *n* приращение, прирост; инкремент
independent [,ɪndɪ'pendənt] *a* независимый
indicator ['ɪndɪkeɪtə] *n* индикатор, указатель
indirect [,ɪndɪ'rekt] *a* косвенный, непрямой
inference ['ɪnf(ə)r(ə)ns] *n* вывод, заключение; следствие
infinity [ɪn'fɪnɪtɪ] *n* бесконечность
initiate [ɪ'nɪʃɪeɪt] *v* начинать; записать; включать; инициировать (АЛГОЛ)
input ['ɪnput] *n* ввод; вход; входное устройство; информация на входе
inquiry [ɪn'kwaɪərɪ] *n* запрос; опрос; исследование
instrument ['ɪnstrumənt] *n* прибор; орудие; инструмент
integer ['ɪntɪdʒə] *n* целое число
intelligence [ɪn'telɪdʒ(ə)ns] *n* сведения; сообщение; логические функции; **artificial** ~ искусственный интеллект
interaction [,ɪntər'ækʃ(ə)n] *n* взаимодействие; взаимосвязь
interconnection [,ɪntə(:)kə'nekʃ(ə)n] *n* внутреннее соединение; межсоединение; разводка; взаимосвязь; взаимозависимость
interface ['ɪntə(:)feɪs] *n* интерфейс; устройство сопряжения; *v* сопрягать
intermediate [,ɪntə(:)'mi:djət] *a* промежуточный
internal [ɪn'tə:nl] *a* внутренний
interrelation [,ɪntə(:)rɪ'leɪʃ(ə)n] *n* взаимозависимость; взаимосвязь
interrupt [,ɪntə'rʌpt] *n* прерывание; *v* прерывать
invalidate [ɪn'vælɪdeɪt] *v* выводить из строя
item ['aɪtem] *n* элемент; единица информации; статья; пункт; позиция
iteration [,ɪtə'reɪʃ(ə)n] *n* итерация; повторение; цикл (ФОРТРАН)

J

job [dʒɔb] *n* задание; задача; работа
jump [dʒʌmp] *n* переход; команда перехода; *v* переходить; **conditional** ~ условный переход; **unconditional** ~ безусловный переход
junction ['dʒʌŋ(k)ʃ(ə)n] *n* соединение; переход; плоскостной (транзистор)

K

key [ki:] *n* ключ; кнопка; клавиша; шифр; код; дескриптор
keyboard ['ki:bɔ:d] *n* клавиатура; коммутационная панель
keypunching ['ki:,pʌntʃɪŋ] *n* перфорирование (вручную)
kilobaud ['kɪlo(u),bɔ:d] *n* килобод (единица скорости передачи информации)
kilobit ['kɪlo(u),bɪt] *n* килобит (единица емкости памяти)
kilobyte ['kɪlo(u),baɪt] *n* килобайт (несколько килобит)
know-how ['nouhau] *n* умение; знание дела; секрет производства

L

label ['leɪbl] *n* метка; идентификатор; знак; *v* маркировать
language ['læŋgwɪdʒ] *n* язык; **object** ~ выходной язык; **problem-oriented** ~ проблемно-ориентированный язык; **reference** ~ эталонный язык; **source** ~ исходный язык; входной язык

large-scale ['lɑ:dʒ'skeɪl] *a* крупномасштабный; большой; крупный
letter ['letə] *n* буква; символ; знак; *v* помечать буквами
level ['levəl] *n* уровень
life-time ['laɪftaɪm] *n* время жизни; срок службы, работы
light [laɪt] *n* свет; освещение; лампа; *pl* световая сигнализация
line [laɪn] *n* линия; провод; шина; строка; **delay** ~ линия задержки; **off-**~ вне ЭВМ, автономно, отдельно от ЭВМ; **on-**~ непосредственно, неавтономно (с управлением от ЭВМ)
link [lɪŋk] *n* звено; связь; команда возврата; *v* связывать, соединять
linkage ['lɪŋkɪdʒ] *n* связь; возврат; переход с возвратом
list [lɪst] *n* список; перечень; *v* составлять список
listing ['lɪstɪŋ] *n* распечатка; листинг
literal ['lɪt(ə)r(ə)l] *a* буквенный; буквальный; текст (АЛГОЛ)
load [loud] *n* нагрузка; загрузка; ввод; *v* загружать, нагружать
loader ['loudə] *n* программа загрузки
location [lo(u)'keɪʃ(ə)n] *n* ячейка; адрес ячейки; размещение; расположение
loop [lu:p] *n* петля; контур; цикл; цепь
loss [lɔs] *n* потеря; проигрыш
low-speed ['lou'spi:d] *a* медленный; малого быстродействия

M

machinery [mə'ʃi:nərɪ] *n* машины; механизмы; оборудование
main [meɪn] *a* главный, основной; *pl* сеть; магистраль
mainframe ['meɪnfreɪm] *n* центральный блок обработки данных; большая ЭВМ
maintain [men'teɪn] *v* поддерживать; обслуживать; содержать в исправности
maintenance ['meɪnt(ɪ)nəns] *n* эксплуатация; уход; **operating** ~ текущее обслуживание и ремонт; **preventing** ~ профилактическое обслуживание; **program** ~ обслуживание программного обеспечения
management ['mænɪdʒmənt] *n* управление; руководство; **industrial process** ~ управление производственными процессами; **job** ~ управление заданиями
man-made ['mæn'meɪd] *a* искусственный
manner ['mænə] *n* метод; способ; образ действия
manual ['mænjuəl] *n* руководство; справочник; *a* ручной; **computer** ~ инструкция (руководство) к ЭВМ
manufacturer [ˌmænju'fæktʃ(ə)rə] *n* производитель; изготовитель
master ['mɑ:stə] *a* главный; ведущий; основной
matching ['mætʃɪŋ] *n* согласование; подгонка; выравнивание
matrix ['meɪtrɪks] *n* матрица; сетка из сопротивлений
mean [mi:n] *n* средняя величина; *pl* средство; *a* средний; *v* значить, означать; иметь значение
meaning ['mi:nɪŋ] *n* значение; символ; содержание
measure ['meʒə] *n* мера; масштаб; *v* измерять; вычислять
measurement ['meʒəmənt] *n* измерение; вычисление; *pl* размеры
medium ['mi:djəm] *n* (*pl* **-dia**) среда; носитель; среднее число
member ['membə] *n* член (*уравнения*); элемент
memorize ['meməraɪz] *v* запоминать; хранить
memory ['memərɪ] *n* память, запоминающее устройство (ЗУ); **addressable** ~ оперативная память (с присвоенными адресами); **external** ~ внешняя память; **internal** ~ внутренняя (оперативная) память; **magnetic bubble** ~ магнитная пузырьковая память
message ['mesɪdʒ] *n* сообщение; передаваемый блок информации
microcircuit ['maɪkro(u)'sə:kɪt] *n* микросхема
minus ['maɪnəs] *n* минус

mnemonics [ni(:)'mɔnɪks] *n* мнемоника; мнемоническая схема
mode [moud] *n* режим *(работы)*; способ; метод; принцип *(работы)*
module ['mɔdju:l] *n* модуль; модульный отсек; блок
monitor ['mɔnɪtə] *n* монитор; диспетчер; *v* управлять; контролировать
movement ['mu:vmənt] *n* движение, перемещение
multiaddress [,mʌltɪə'dres] *a* многоадресный
multiple ['mʌltɪpl] *n* кратное число; *a* многократный; **common** ~ общее кратное; **least common** ~ общее наименьшее кратное
multiprocessing ['mʌltɪ'prousesɪŋ] *n* многопроцессорная обработка

N

negative ['negətɪv] *n* знак минуса; *a* отрицательный
net [net] *n* сеть; сетка; схема
network ['netwə:k] *n* сеть, сетка; схема; контур; сетевой график
no-operation ['nou,ɔpə'reɪʃ(ə)n] *n* холостая команда
NOR [nɔ:] НЕ (логическая функция или операция)
NOT AND ['nɔt'ænd] НЕ И (логическая функция или операция)
notation [no(u)'teɪʃ(ə)n] *n* система счисления; запись; представление; **binary** ~ двоичное счисление; **coded-decimal** ~ кодированное десятичное счисление; **decimal** ~ десятичное счисление; **hexadecimal** ~ шестнадцатиричное счисление; **octal** ~ восьмеричное счисление
number ['nʌmbə] *n* число; номер; цифра; *v* считать; нумеровать; **permanent** ~ постоянное число; **temporary** ~ временное число
numeral ['nju:m(ə)rəl] *n* цифра; числительное; *a* числовой; численный; **Arabic** ~ арабская цифра; **Roman** ~ римская цифра

O

octal ['ɔkt(ə)l] *a* восьмеричный
off-line ['ɔ(:)f'laɪn] *a* автономный; независимый
offset ['ɔ(:)fset] *n* смещение; сдвиг
on-line ['ɔnlaɪn] *a* неавтономный; зависимый
operation [,ɔpə'reɪʃ(ə)n] *n* действие; операция; режим; срабатывание
operator ['ɔpəreɪtə] *n* оператор; операция
option ['ɔpʃ(ə)n] *n* выбор; вариант; версия; опция
order ['ɔ:də] *n* команда; порядок; степень; разряд числа
original [ə'rɪdʒənl] *a* первоначальный; исходный; оригинальный
output ['autput] *n* выход; вывод; результат; мощность; выходной сигнал
overflow ['ouvəflou] *n* переполнение; избыток
own [oun] *a* собственный

P

pack [pæk] *n* узел; блок; пакет; колода перфокарт; *v* уплотнять
package ['pækɪdʒ] *n* модуль; корпус; блок; упаковка; пакет; **application** ~ прикладные пакеты; **application program** ~ пакет прикладных программ
parameter [pə'ræmɪtə] *n* параметр; характеристика
part [pɑ:t] *n* часть; доля; деталь; совокупность (АЛГОЛ)
pass [pɑ:s] *n* проход; прогон; просмотр; *v* пропускать
path [pɑ:θ] *n* траектория; путь; дорожка; цепь; длина пробега
pattern ['pæt(ə)n] *n* образец; модель; шаблон; схема; набор
perform [pə'fɔ:m] *v* выполнять; производить *(действие)*
performance [pə'fɔ:məns] *n* характеристика; к. п. д.; производительность
peripheral [pə'rɪfər(ə)l] *a* периферийный, внешний
picture ['pɪktʃə] *n* изображение; шаблон

pipeline ['paɪplaɪn] *n* труба; шина; трубопровод; *v* укладывать шину
place [pleɪs] *n* разряд; место
plate [pleɪt] *n* пластина; плата; анод; диск *(в памяти ЭВМ)*
playback ['pleɪbæk] *n* воспроизведение; считывание
plot [plɔt] *n* график; кривая; диаграмма
plug-in ['plʌg'ɪn] *a* съемный; сменный
plus [plʌs] *n* плюс; положительная величина
point [pɔɪnt] *n* точка; пункт; место; точка, отделяющая десятичную дробь от целого числа; **binary** ~ двоичная точка (запятая); **decimal** ~ десятичная точка (запятая); **fixed** ~ фиксированная точка (запятая); **floating** ~ плавающая точка (запятая)
pointer ['pɔɪntə] *n* стрелка; указатель *(прибора)*
positive ['pɔzətɪv] *a* положительный
possess [pə'zes] *v* обладать, владеть; ~ **the value** принимать значение
power ['paʊə] *n* мощность; энергия; сила; способность; степень
precede [prɪ(:)'si:d] *v* предшествовать; превосходить
preceding [prɪ(:)'si:dɪŋ] *a* предшествующий
precision [prɪ'sɪʒ(ə)n] *n* точность; ~ **instrument** точный прибор
predetermine ['pri:dɪ'tə:mɪn] *v* предопределять, предрешать
prediction [prɪ'dɪkʃ(ə)n] *n* предсказание; расчет; результат
presentation [ˌprezen'teɪʃ(ə)n] *n* представление; воспроизведение
pre-set [pri:'set] *v* предварительно устанавливать
pressure ['preʃə] *n* давление; **blood** ~ кровяное давление
printer ['prɪntə] *n* печатающее устройство
priority [praɪ'ɔrɪtɪ] *n* приоритет
probability [ˌprɔbə'bɪlɪtɪ] *n* вероятность
procedure [prə'si:dʒə] *n* процедура; методика проведения *(опыта)*; ~ **body** тело процедуры
process ['prouses] *n* процесс; прием; режим; ход; *v* обрабатывать
processing ['prousesɪŋ] *n* обработка
processor ['prousesə] *n* процессор; устройство для обработки данных; программа «транслятор»; **bit-slice-**~ микропроцессор с разрядно-модульной организацией
produce [prə'dju:s] *v* производить; порождать; синтезировать
product ['prɔdəkt] *n* продукт; произведение
program(me) ['prougræm] *n* программа; *v* программировать; ~ **body** тело программы; **application** ~ прикладная программа; **debugging** ~ программа отладки; **general-purpose** ~ универсальная программа; **loading** ~ программа-загрузчик; **object (language)** ~ конечная программа; **supervisor** ~ программа-супервизор
programmer ['prougræmə] *n* программист; ~ **analyst** программист-аналитик
programming ['prougræmɪŋ] *n* программирование; планирование
property ['prɔpətɪ] *n* свойство; особенность
provide [prə'vaɪd] *v* снабжать; обеспечивать; давать
pulse [pʌls] *n* импульс; вибрация; *v* посылать импульсы; **clock** ~**s** синхронизирующие импульсы
punch [pʌntʃ] *n* пробивка; перфорация; *v* перфорировать
purpose ['pə:pəs] *n* назначение; цель; намерение

Q

quality ['kwɔlɪtɪ] *n* качество; достоинство
quantity ['kwɔntɪtɪ] *n* количество; *мат.* величина
query ['kwɪərɪ] *n* запрос на очередь; вопрос
queue [kju:] *n* очередь; очередность *(запросов)*
quatient ['kwouʃ(ə)nt] *n* частное; коэффициент

radix ['reɪdɪks] (*pl* **radices**) *n* основание системы счисления
random ['rændəm] *a* случайный; произвольный
range [reɪndʒ] *n* область; диапазон; интервал; блок (АЛГОЛ)
rapid ['ræpɪd] *a* быстрый, скорый; быстродействующий
rate [reɪt] *n* коэффициент; степень; скорость; частота; норма
rated ['reɪtɪd] *a* расчетный; номинальный
ratio ['reɪʃɪou] *n* отношение; соотношение; коэффициент
reader ['ri:də] *n* считывающее (читающее) устройство
read-in ['ri:d'ɪn] *n* ввод (запись) программы в память ЭВМ
reading ['ri:dɪŋ] *n* отсчет *(по шкале)*; показание *(измерительного прибора)*
read-out ['ri:daut] *n* считывание данных; показание; вывод
real [rɪəl] *a* вещественный; действительный, реальный; истинный; **~-time** в истинном масштабе времени
recognition [ˌrekəg'nɪʃən] *n* опознавание, распознавание; различение
record ['rekɔ:d] *n* запись; регистрация; [rɪ'kɔ:d] *v* записывать
recursive [rɪ'kə:sɪv] *a* рекурсивный; повторяющийся
reduction [rɪ'dʌkʃ(ə)n] *n* сокращение; редукция; превращение *(в другие, более мелкие меры, единицы и т. п.)*; упрощение
reference ['refr(ə)ns] *n* ссылка *(на кого-л., что-л.)*; сноска; сообщение; обращение; эталон
regeneration [rɪˌdʒenə'reɪʃ(ə)n] *n* восстановление; перезапись
region ['ri:dʒ(ə)n] *n* область, зона; диапазон
register ['redʒɪstə] *n* регистр; *v* регистрировать
relation [rɪ'leɪʃ(ə)n] *n* отношение, соотношение, зависимость, связь
relative ['relətɪv] *a* относительный; связанный
relay ['ri:leɪ] *n* реле; ['ri:'leɪ] *v* передавать
release [rɪ'li:s] *n* освобождение; разъединение; *v* освобождать
reliability [rɪˌlaɪə'bɪlɪtɪ] *n* надежность
relocation ['ri:lo(u)'keɪʃ(ə)n] *n* перераспределение *(памяти)*; перемещение
remainder [rɪ'meɪndə] *n* остаток; разность
remote [rɪ'mout] *a* дистанционный; **~ control** дистанционное управление
repertoir ['repətwɑ:] *n* набор; состав; система команд; репертуар
replace [rɪ'pleɪs] *v* заменять; подставлять; перемещать
representation [ˌreprɪzən'teɪʃ(ə)n] *n* представление *(чисел)*
reproduction [ˌri:prə'dʌkʃ(ə)n] *n* воспроизведение; репродукция
request [rɪ'kwest] *n* запрос; требование
requirement [rɪ'kwaɪəmənt] *n* требование, необходимое условие
rerun ['ri:'rʌn] *n* повторный проход или прогон программы
research [rɪ'sə:tʃ] *n* научно-исследовательская работа
resident ['rezɪd(ə)nt] *n* резидент; резидентная часть программы
response [rɪs'pɔns] *n* ответ; реакция; *v* реагировать
retrieval [rɪ'tri:v(ə)l] *n* поиск *(информации)*
return [rɪ'tə:n] *n* возврат, отдача; *v* возвращать, отдавать
root [ru:t] *n* корень
root-mean-square ['ru:t'mi:n'skwɛə] среднеквадратичное значение
rotation [ro(u)'teɪʃ(ə)n] *n* вращение; чередование
routine [ru(:)'ti:n] *n* (стандартная) программа
rule [ru:l] *n* правило; масштаб; масштабная линейка; **slide ~** логарифмическая линейка
run [rʌn] *n* проход; прогон; работа; *v* прогонять *(программу)*; **~ning the code on the computer** прогон программы на вычислительной машине

safety ['seɪftɪ] *n* безопасность; надежность
saturation [ˌsætʃə'reɪʃ(ə)n] *n* насыщение
scale [skeɪl] *n* шкала; масштаб; масштабная линейка
scan [skæn] *n* сканирование; просмотр; поиск; развертка
scheduler ['ʃedjuːlə] *n* планировщик *(программ)*
scope [skoup] *n* индикатор; область действия; сфера рассмотрения
screen [skriːn] *n* экран; экранная сетка; *v* экранировать
searching ['səːtʃɪŋ] *n* поиск
selection [sɪ'lekʃ(ə)n] *n* выбор; выборка; выборочная совокупность
self-running ['self'rʌnɪŋ] *a* синхронизированный; свободноидущий
semiconductor ['semɪkən'dʌktə] *n* полупроводник
sense [sens] *v* считывать; опознавать; воспринимать; определять
sensing ['sensɪŋ] *n* считывание; опознавание; восприятие
sensitivity [ˌsensɪ'tɪvɪtɪ] *n* чувствительность
sensor ['sensə] *n* датчик, чувствительный элемент
sequence ['siːkwəns] *n* последовательность; порядок следования; ряд
service ['səːvɪs] *n* служба; работа; обслуживание; *v* обслуживать
set [set] *n* установка; набор; множество; *v* устанавливать; **~up** образовывать; формулировать *(задачу)*
setting-up ['setɪŋ'ʌp] *n* сборка; монтаж; наладка; настройка
share [ʃɛə] *v* делить(ся), распределять
sharing ['ʃɛərɪŋ] *n* деление, разделение; совместное пользование; **time ~** разделение времени; работа с разделением времени
sheet [ʃiːt] *n* лист; бланк; схема; диаграмма; плата
shift [ʃɪft] *n* сдвиг; смещение; *v* сдвигать; смещать
shorthand ['ʃɔːthænd] *n* стенография; сокращенная запись
sign [saɪn] *n* знак, обозначение; символ; признак
significant [sɪg'nɪfɪkənt] *a* значительный; значащий; значимый
similarity [ˌsɪmɪ'lærɪtɪ] *n* подобие
simulation [ˌsɪmjuˈleɪʃ(ə)n] *n* моделирование; имитация
simultaneous [ˌsɪm(ə)l'teɪnjəs] *a* одновременный; совместный
size [saɪz] *n* размер, величина, объем; емкость *(памяти)*
slicing ['slaɪsɪŋ] *n* разделение; разрезание на части; **time ~** разделение времени
software ['sɔftweə] *n* программное (математическое) обеспечение; **support ~** вспомогательное программное обеспечение
solid-state ['sɔlɪd'steɪt] *a* полупроводниковый
solution [sə'l(j)uːʃ(ə)n] *n* решение; разрешение *(проблемы и т. п.)*; раствор
solve [sɔlv] *v* решать, разрешать; растворять
sophisticated [sə'fɪstɪkeɪtɪd] *a* усложненный; тонкий *(о приборе и т. п.)*
source [sɔːs] *n* источник; исток
space [speɪs] *n* пространство; область; космос; *v* располагать с интервалами
spare [spɛə] *n* запас; резерв; *a* запасной; резервный
special-purpose ['speʃ(ə)l'pəːpəs] *a* специализированный
specific [spɪ'sɪfɪk] *a* особый; характерный; конкретный
speed [spiːd] *n* скорость; быстродействие
spot [spɔt] *n* пятно; место; ячейка *(в памяти)*; **polarized ~** поляризованный участок
square [skweə] *n* квадрат; прямоугольник
stack [stæk] *n* стек; стековая (магазинная) память; набор
stage [steɪdʒ] *n* этап; ступень; разряд
stand-by ['stæn(d)baɪ] *n* резервное (запасное) оборудование
start [stɑːt] *n* пуск; начало программы; *v* запускать; начинать
state [steɪt] *n* состояние, положение; структура; *v* формулировать

statement ['steɪtmənt] *n* оператор; утверждение; высказывание; ~ **body** тело оператора; **assignment** ~ оператор присвоения; **conditional** ~ условный оператор; **declaration** ~ оператор описания; **executable** ~ исполнительный оператор (ФОРТРАН)

status ['steɪtəs] *n* состояние; **quality** ~ знак качества

step [step] *n* шаг; ступень; стадия; этап *(вычислений)*

storage ['stɔ:rɪdʒ] *n* запоминающее устройство (ЗУ); память

store [stɔ:] *v* запоминать; хранить;

stream ['stri:m] *n* поток; ~ **of data** поток данных

string [strɪŋ] *n* строка; цепочка; последовательность

subprogram(me) ['sʌb'prougræm] *n* подпрограмма, часть программы

subroutine ['sʌbru:'ti:n] *n* подпрограмма, часть программы; **library of ~s** библиотека подпрограмм

subscriber [səb'skraɪbə] *n* пользователь; абонент

subsript ['sʌbskrɪpt] *n* нижний индекс; подстрочный символ

subsequence ['sʌbsɪkwəns] *n* последовательность

subset ['sʌb,set] *n* подмножество

substitute ['sʌbstɪtju:t] *n* замена; *v* заменять; замещать

subtraction [səb'trækʃən] *n* вычитание

success [s(ə)k'ses] *n* успех, удача

succession [s(ə)k'seʃ(ə)n] *n* последовательность

sum [sʌm] *n* сумма, количество, итог

summary ['sʌmərɪ] *n* краткое содержание; аннотация

supermarket [,sju:pə'mɑ:kɪt] *n* универсам; большой магазин самообслуживания

supervisor ['sju:pəvaɪzə] *n* супервизор; программа-распределитель

supplementary [,sʌplɪ'ment(ə)rɪ] *a* дополнительный

supply [sə'plaɪ] *n* подача, источник питания; *v* подавать; питать

support [sə'pɔ:t] *n* обеспечение; поддержка; *v* обеспечивать

surface ['sə:fɪs] *n* поверхность; плоскость

switch [swɪtʃ] *n* тумблер; переключатель; коммутатор; *v* переключать

switching ['swɪtʃɪŋ] *n* переключение; коммутация

synchronize ['sɪŋkrənaɪz] *v* синхронизировать; согласовывать

synthesis ['sɪnθɪsɪs] *n* синтез

system ['sɪstɪm] *n* система; совокупность; установка; устройство; **binary-coded decimal** ~ двоично-кодированная десятичная система счисления

T

table ['teɪbl] *n* таблица; стол; доска; плоская поверхность

tabulate ['tæbjuleɪt] *v* сводить в таблицы, табулировать

tape [teɪp] *n* лента; *v* заклеивать (липкой) лентой

task [tɑ:sk] *n* задача; проблема; ветвь *(программы)*

technique [tek'ni:k] *n* метод; методика; технический прием

temporary ['temp(ə)rərɪ] *a* временный

term [tə:m] *n* термин; терм; член; семестр; **in ~s of** на языке; с точки зрения; в пределах; в переводе на язык

terminal ['tə:mɪnl] *n* терминал; оконечное устройство; клемма

test [test] *n* испытание; проверка; *v* проверять; контролировать

theory ['θɪərɪ] *n* теория; метод

three-dimensional ['θri:d(a)ɪ'menʃənl] *a* трехмерный

throughput ['θru:put] *n* производительность; пропускная способность

time [taɪm] *n* время; момент; *v* согласовывать во времени; **access** ~ время выборки; время обращения к памяти

time-table ['taɪm,teɪbl] *n* временная диаграмма; расписание

timing ['taɪmɪŋ] *n* согласование во времени; тактировка

tolerance ['tɔlər(ə)ns] *n* **допуск,** допустимое **отклонение**
total ['toutl] *n* **сумма; итог;** *a* суммарный
track [træk] *n* **дорожка; канал;** тракт; перфорация
traffic ['træfɪk] *n* поток **информации** (сообщений); уличное **движение**
train [treɪn] *n* ряд; серия; **цепочка;** последовательность
transfer ['trænsfə(:)] *n* **передача;** пересылка; перенос; переход; [træns'fə:] *v* переносить; выполнять команду перехода
transform [træns'fɔ:m] *v* преобразовывать; превращать
transient ['trænzɪənt] *a* переходный; переменный
transistor [træn'sɪstə] *n* транзистор; *n-p-n* **junction** ~ плоскостной транзистор *n-p-n* типа
transmit [trænz'mɪt] *v* передавать
transport ['trænspɔ:t] *n* перенос; перемещение; протяжка; лентопротяжный механизм; [træn'spɔ:t] *v* переносить; перемещать
treatment ['tri:tmənt] *n* обработка; исследование; анализ; решение
trigger ['trɪgə] *n* триггер; *v* запускать; отпирать
trouble ['trʌbl] *n* неисправность; повреждение
true [tru:] *a* истинный; *v* настраивать; регулировать
truth [tru:θ] *n* правда; истина
tube [tju:b] *n* электронная лампа; электронно-лучевая трубка
tune [tju:n] *n* тон; *v* настраивать
turn [tə:n] *n* оборот; виток; *v* вращать(ся); превращать(ся); ~ **off** выключать; ~ **on** включать
type [taɪp] *n* тип; род, класс; вид; *v* писать на машинке
typeprinter ['taɪp,prɪntə] *n* буквопечатающий аппарат
typewriter ['taɪp,raɪtə] *n* пишущая машинка

U

understandability [,ʌndə'stændə,bɪlɪtɪ] *n* понятливость
unidirectional ['ju:nɪd(a)ɪ'rekʃənl] *a* однонаправленный
unilateral ['ju:nɪ'læt(ə)r(ə)l] *a* односторонний
unit ['ju:nɪt] *n* единица; устройство; блок; элемент
universe ['ju:nɪvə:s] *n* область; совокупность; вселенная; мир
unknown ['ʌn'noun] *n* неизвестное, неизвестная величина
updating [ʌp'deɪtɪŋ] *n* корректировка; модернизация; обновление
uptime [ʌp'taɪm] *n* рабочее (машинное) время
up-to-date ['ʌptə'deɪt] *a* современный, новейший
usage ['ju:zɪdʒ] *n* использование, применение; эксплуатация
use [ju:s] *n* использование, применение; [ju:z] *v* использовать
useful ['ju:sf(u)l] *a* полезный; эффективный
user ['ju:zə] *n* пользователь; абонент; **high-priority** ~ пользователь с высоким приоритетом; **remote** ~ пользователь, работающий с дистанционного пульта

V

valid ['vælɪd] *a* действительный; обоснованный, достоверный
value ['vælju:] *n* значение; величина; оценка; *v* оценивать
variable ['vɛərɪəbl] *n* переменная (величина); **dependent** ~ зависимая переменная; **dummy** ~ фиктивная переменная; **independent** ~ независимая переменная; *a* переменный; ~**-length** переменной длины
variety [və'raɪətɪ] *n* разнообразие; **(of)** ряд, множество
various ['vɛərɪəs] *a* различный, разный; разнообразный
vary ['vɛərɪ] *v* менять(ся), изменять(ся); разнообразить
vehicle ['vi:ɪkl] *n* летательный аппарат; средство передвижения
velocity [vɪ'lɔs(ɪ)tɪ] *n* скорость; вектор скорости; **variable** ~ переменная скорость
version ['və:ʃ(ə)n] *n* вариант; версия
view [vju:] *n* вид; изображение; проекция

violation [ˌvaɪə'leɪʃ(ə)n] *n* нарушение; *мат.* противоречие
visual ['vɪzjuəl] *a* визуальный; наглядный
vocabulary [və'kæbjuləɪ] *n* лексика; словарь; список команд
voice [vɔɪs] *n* голос
voltage ['voultɪdʒ] *n* напряжение, разность потенциалов

W

wave [weɪv] *n* волна; колебание; сигнал
way [weɪ] *n* способ; путь
wear [wɛə] *n* износ; срабатывание; *v* изнашиваться; срабатывать
wheel [wi:l] *n* колесо; колесико; ролик; диск; **counter** ~ колесо счетчика, цифровое колесо
win [wɪn] *n* выигрыш, победа; *v* выиграть, победить
wind [waɪnd] *n* намотка; *v* наматывать; перематывать
winding ['waɪndɪŋ] *n* обмотка; виток; намотка
wipe-out ['waɪp'aut] *n* стирание *(с магнитной ленты)*
wire [waɪə] *n* проволока; провод; проводник; шина
wiring ['waɪəɪŋ] *n* монтаж; проводка; межсоединения (в интегральной схеме)
word [wə:d] *n* слово; число; код; группа символов; **call** ~ вызывающее слово; **interrupt status** ~ слово о состоянии прерывания; **key** ~ ключевое слово, основное слово; **machine (computer)** ~ машинное слово; **variable (-length)** ~ слово переменной длины
work [wə:k] *n* работа; *v* работать; ~ **out** вырабатывать, разрабатывать; вычислять
working ['wə:kɪŋ] *n* работа, действие; операция
workshop ['wə:kʃɔp] *n* секция; семинар; симпозиум
write-in ['raɪt(')ɪn] *n* запись
writing ['raɪtɪŋ] *n* запись; документ
written ['rɪtən] *a* записанный; письменный

Z

zero ['zɪərou] *n* нуль; нулевая точка
zero-address ['zɪərouə'dres] *a* безадресный
zone [zoun] *n* зона; область; зона перфокарты; **storage** ~ зона запоминающего устройства

LIST OF ACRONYMS AND ABBREVIATIONS

AC, ac, a-c 1. automatic computer автоматический компьютер; 2. alternating current переменный ток

AL Assembly Language язык ассемблера

ALGOL ALGOrithmic Language алгоритмический язык, АЛГОЛ

ALU Arithmetic/Logic Unit арифметическое/логическое устройство (АЛУ)

AMS Automated Management Systems автоматизированные системы управления, АСУ

ASCII American Standard Code for Information Interchange Американский стандартный код для обмена информацией

BASIC Beginners All-Purpose Symbolic Instruction Code универсальный символический язык для начинающих программистов, БЭЙСИК

BESM High-Speed Electronic Computer быстродействующая электронно-вычислительная машина (СССР), БЭСМ

bit binary digit двоичная цифра, двоичный разряд, бит

bpi bit per inch бит на дюйм, разрядов на дюйм

bps bit per second бит в секунду, разрядов в секунду

CAD Computer-Aided Design проектирование с помощью ЭВМ

CAM Computer-Aided Manufacturing производство с помощью ЭВМ

CMEA the Council for Mutual Economic Assistance Совет Экономической Взаимопомощи, СЭВ

COBOL Common Business Oriented Language алгоритмический язык для экономических и коммерческих задач, КОБОЛ

cps cycle per second герц, циклов в секунду

CPU Central Processing Unit центральный блок управления, ЦПУ

DASD Direct Access Storage Device запоминающее устройство с прямым доступом

DBMS Data Base Management System система управления базами данных СУБД

DC, dc, d-c digital computer 1, цифровой компьютер; 2. direct current постоянный ток

EBCDIC Expanded Binary Coded Decimal Interchange Code расширенный двоично-кодированный код для обмена информацией (США)

e.g. for example например

ES Unified System единая система ЭВМ (СССР)

etc. et cetera и так далее

FMS Flexible Manufacturing Systems гибкие системы производства

FORTRAN FORmula TRANslation алгоритмический язык для научных задач, ФОРТРАН

fps feet per second футов в секунду

HYDAC HYbrid Digital-Analog Computer гибридная цифро-аналоговая вычислительная машина

IBM/650 International Business Machine/650) семейство ЭВМ 1-го поколения, выпускаемого фирмой ИБМ (США)

IC integrated circuit интегральная схема, ИС

i.e. that is то есть

in. inch дюйм

I/O Input/Output ввод/вывод, вход/выход, ВВ

IPL initial program loading загрузка начальной программы

KB kilobit, kilobyte килобит, килобайт

L label метка, идентификатор, отметка

LASER Light Amplification by Stimulated Emission of Radiation лазер

LISP LIST Processing, LIST Programming язык программирования для обработки списков и списковых структур, ЛИСП

loc location ячейка, местоположение, размещение

LSI large-scale integration интеграция высокого уровня

MB megabit, megabyte мегабит, мегабайт

MOS metal-oxide-semiconductor метал-окисел-полупроводник, МОП

NC numerical control цифровое управление

n-tuple Н-кратный

op operation операция

OS Operating System операционная система, ОС

PASCAL язык программирования, ПАСКАЛЬ

PL/1 язык программирования, ПЛ/1

PROM Programmable ROM программируемое ПЗУ

pseudo-op pseudo-operation псевдооперация (команда)

R register регистр

RAM Random Access Memory память с произвольной выборкой

ROM Read-Only Memory постоянная память, ПЗУ

Sim 1 Simulator 1 Модель 1 (название робота)

SISI Super Large-Scale Integration интеграция сверхвысокого уровня

SQRT square root квадратный корень

VISI Very Large-Scale Integration интеграция очень высокого уровня

TERMS USED IN COMPUTING FIELD

Space, Bytes (or Bits)	Time Fraction of 1 second
Kilobyte (KB) 1,000	Millisecond (Ms) 1/1,000th
Megabyte (MB) 1,000,000	Microsecond (μs) 1/1,000,000th
Gigabyte (GB) 1,000,000,000	Nanosecond (Ns) 1/1,000,000,000th
Terabyte (TB) 1,000,000,000,000	Picosecond (Ps) 1/1,000,000,000,000th

that is why вот почему
the . . . the чем . . . тем
till then до того времени
to some extent до некоторой степени
twice as much в 2 раза больше
under consideration представленный на рассмотрение
under operating conditions в рабочих условиях
unlike that в отличие от этого; в противоположность этому
up till now до сих пор
up to до; вплоть до
up-to-date новейший; современный
used to (+ Infinitive) бывало
with reference to относительно; не напрасно
with regard to по отношению к; относительно
with relation to относительно; что касается
with respect to по отношению к; относительно
with the result в результате чего
within certain limits в определенных пределах
within the range в пределах
without fail наверняка; непременно; обязательно
without question несомненно
word for word буквальный; дословный

Учебное издание

Алиса Семеновна Кутькова

ЧЕЛОВЕК — КОМПЬЮТЕР — БУДУЩЕЕ

Пособие по английскому языку

Заведующая редакцией И. Э. Волкова. Редактор Л. Н. Белая. Младший редактор Т. В. Пигалева. Художник О. В. Орлова. Художественный редактор В. И. Пономаренко. Технический редактор А. К. Нестерова. Старший корректор Н. А. Каджардузова

ИБ № 6186

Изд. № А-17. Сдано в набор 08.08.86. Подп. в печать 30.12.86. Формат 84×108$^{1}/_{32}$. Бум. кн.-журн. Гарнитура литературная. Печать высокая. Объем 6,72 усл. печ. л. 7,03 усл. кр.-отт. 7,78 уч.-изд. л. Тираж 50 000 экз. Заказ № 839. Цена 25 коп.

Издательство «Высшая школа». 101430, Москва, ГСП-4, Неглинная ул., д. 29/14

Отпечатано с матриц ордена Октябрьской Революции и ордена Трудового Красного Знамени МПО «Первая Образцовая типография» имени А. А. Жданова Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли 113054, Москва, Валовая, 28 во Владимирской типографии Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли. 600000, г. Владимир, Октябрьский проспект, д. 7.