Методичні вказівки
до виконання курсових робіт з дисципліни «Програмування»
зі спеціальності «Кібербезпека та захист інформації»
(освітні програми «Безпека інформаційних і комунікаційних систем» та «Етичний хакінг і кібербезпека»)

Міністерство освіти і науки України
Вінницький національний технічний університет

Методичні вказівки до виконання курсових робіт з дисципліни «Програмування» зі спеціальності «Кібербезпека та захист інформації» (освітні програми «Безпека інформаційних і комунікаційних систем» та «Етичний хакінг і кібербезпека»)

Вінниця
ВНТУ
2026

Рекомендовано до видання Радою з якості освіти Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 11 від 23.04.2026 р.)

Рецензенти:
Д. Х. Штофель, кандидат технічних наук, доцент
Д. І. Катєльніков, кандидат технічних наук, доцент
Л. М. Куперштейн, кандидат технічних наук, доцент

Методичні вказівки до виконання курсових робіт з дисципліни «Програмування» зі спеціальності «Кібербезпека та захист інформації» (освітні програми «Безпека інформаційних і комунікаційних систем» та «Етичний хакінг і кібербезпека») / уклад.: В. А. Каплун, В. Є. Радченко. Електрон. текст. дані. Вінниця : ВНТУ, 2026. 67 с.

Методичні вказівки призначені для надання допомоги при виконанні курсових робіт з дисципліни «Програмування». У роботі зроблено акцент на використанні сучасних технологій програмування при виконанні завдання курсової роботи, на необхідності використання додаткових можливостей мов програмування і засобів вебпрограмування, знання яких має неабияке значення при створенні засобів захисту у галузі інформаційних технологій і при захисті програмного забезпечення. Крім того, у методичних вказівках дано рекомендації щодо правильності подання інформації у пояснювальній записці та грамотного оформлення текстової та ілюстративної частин.

Методичні вказівки призначені для студентів спеціальності «Кібербезпека та захист інформації» освітньо-професійних програм «Безпека інформаційних і комунікаційних систем» та «Етичний хакінг і кібербезпека».

ЗМІСТ

ВСТУП

Курсова робота (КР) з дисципліни «Програмування» – це самостійна робота, яка охоплює весь матеріал, викладений під час вивчення дисципліни «Програмування». Завдання на курсову роботу передбачає використання усього матеріалу, який було опрацьовано під час лекційних, лабораторних, практичних та самостійних занять протягом всього курсу вивчення дисципліни: різноманітні види операторів, роботу з файлами, роботу з масивами, застосування різних технологій програмування, використання різноманітних засобів для побудови графічного інтерфейсу, роботу у візуальних середовищах програмування, елементи вебпрограмування тощо.

В курсовій роботі з дисципліни «Програмування» здобувач повинен показати набуті навички:

- знання мов програмування, розуміння основних концепцій і технологій програмування,
- вміння самостійно розробляти схему роботи програмного застосунку в цілому та алгоритми його складових відповідно до поставленої задачі;
- підібрати засоби для програмної реалізації та подати розробку у вигляді, зручному для його використання сторонніми користувачами;
- правильно оформити текстову і ілюстративну частини курсової роботи згідно ЄСПД;
- грамотно представити і захистити свою роботу.

Курсова робота включає в себе:

1) **програмну частину**, яка може бути виконана з використанням будь-яких програмних засобів, обґрунтування вибору яких здобувач повинен пояснити;
2) **текстову частину**, яка докладно описує процес створення програмного застосунку, включає необхідні приклади, схеми, діаграми тощо;
3) **ілюстративний матеріал**, який допоможе успішно представити курсову роботу під час її захисту.

1 ТЕМАТИКА КУРСОВОЇ РОБОТИ

Тематика курсової роботи пов’язана з майбутньою спеціальністю студентів повинна відповідати завданням дисципліни «Програмування» і тісно пов’язуватися з практичними потребами конкретного фаху. Тематика курсових робіт затверджується на засіданнях кафедр.

Для програмної реалізації даної курсової роботи пропонуються найпростіші традиційні шифри для криптографічного захисту інформації: шифри перестановок, заміни, гамування тощо. Крім того, в якості об’єкта програмування можуть бути розробки ігрових програм, реалізація тестових програм, розробка лабораторних практикумів для інших дисциплін тощо.

Зміст курсової роботи визначається завданням, яке видається викладачем кожному студенту на перших двох тижнях навчального семестру.

Курсова робота включає декілька послідовних етапів, які, в загальному випадку, пов’язані зі змістовною постановкою задачі, вибором форми подання задачі, розробкою математичної моделі, вибором оптимального алгоритму реалізації задачі, проведенням досліджень режимів роботи програми та формулюванням обґрунтованих висновків щодо отриманих в роботі результатів. Кожен етап роботи обов’язково має знайти своє відображення в пояснювальній записці, що містить вхідні, вихідні та пояснювальні матеріали, які пов’язані з виконанням курсової роботи.

Завдання для курсових робіт зазвичай визначаються викладачем із загального списку завдань на курсову роботу (додаток А). Заохочуються пропозиції студентів щодо самостійного, за узгодженням з викладачем, вибору теми КР поза межами запропонованого в методичних вказівках переліку. Самостійний вибір предметної області, в якій доцільно використовувати сучасні методи програмування та оригінальні алгоритми, дозволяє зробити висновок щодо рівня творчої активності здобувача, його вміння самостійно здійснити попередній аналіз предметної області і ставити перед собою конкретні задачі.

Оскільки здобувачі вищої освіти з перших років навчання можуть брати участь в науковій діяльності університету і кафедри, у різноманітних наукових гуртках, заохочується вибір студентами тем, пов’язаних саме з науковою тематикою кафедри.

У випадку повного збігання тем курсових робіт індивідуальне завдання має містити не тільки різні вихідні дані, але й передбачати самостійне викладення здобувачем тексту пояснювальної записки з метою уникнення використання одного і того ж електронного варіанта.

2 ВИМОГИ ДО ПРОГРАМНОЇ ЧАСТИНИ

Програмна частина курсової роботи повинна складатися з двох частин:

- десктопний програмний застосунок;
- вебсторінку, що супроводжує програмний застосунок.

Перша частина – програмний засіб, який реалізує індивідуальне завдання на курсову роботу і є повноцінним застосунком для будь-якої операційної системи – Windows, Unix, IOS, Android. Програма може бути реалізована будь-якою мовою програмування: C/C++/Java/C#/Python тощо, з використанням будь-якого програмного середовища. Не обов'язково використовувати лише ті знання і навички, які набуті під час вивчення дисципліни. Вітається застосування додаткових програмних засобів і мов програмування, вивчених самостійно.

Розроблена програма повинна продемонструвати вміння і знання, отримані під час лекційних, практичних і лабораторних робіт, а також під час самостійної роботи:

- глибоке *знання базових концепцій основних мов програмування* і вміння обґрунтувати вибір тієї чи іншої мови програмування;
- застосування *основних технологій програмування* і принципів об'єктно-орієнтованого та функціонального програмування;
- знання *основ алгоритмізації*: побудова алгоритмів роботи програмних засобів в цілому і розробка алгоритмів роботи певних функцій;
- дотримання *технології структурного і модульного програмування*, що свідчить про вміння грамотно виділяти основні змістовні модулі програмних засобів та організовувати взаємозв'язок між ними;
- *реалізація дружнього інтерфейсу*: використання багаторівневого меню, діалогових вікон, різноманітних елементів керування роботою програми, попередження про можливі помилки при введенні інформації, підказки під час інтерактивного режиму роботи і т. д.;
- *використання файлів* для зберігання та зчитування інформації. Це може бути або введення початкової інформацій з файлу (файлів) і виведення результуючої інформації у файл (файли), або зберігання ключової інформації, або зберігання необхідних таблиць та алфавітів для шифрування;
- *перевірка цілісності даних та обробка виключних ситуацій* на рівні перевірки правильності введеної інформації (числової, символьної, великі літери, малі літери, належність до алфавіту тощо), перевірка існування потрібних файлів і т. д.;

– *подання інформації* (як вхідної, так і результуючої) повинно бути зрозумілим, мати необхідні пояснення. Всі результати вхідних, проміжних, результуючих дій повинні бути виведені на екран у вигляді, зручному для розуміння та аналізу стороннім користувачем (а не лише розробником), з поясненнями, допоміжними вікнами повідомлень тощо.

Друга частина – вебсайт (вебсторінка), яка супроводжує програмний застосунок. На цій сторінці необхідно надати деякі додаткові відомості, пов'язані з розробленим програмним засобом. Наприклад, наповненням сайту може бути така інформація:

– сутність розробленого програмного застосунку (мета, тема);
– історична довідка або короткий аналітичний огляд по темі роботи;
– відомості про автора розробки (представлення, фото, хобі, посилання на особисту сторінку тощо);
– алгоритми, за якими розроблялася програма;
– рекомендації по роботі з програмою, контрольні приклади тощо.

Тобто, здобувач повинен продемонструвати володіння навичками з вебпрограмування, використовуючи при цьому такі засоби, як HTML, CSS, JavaScript і інші.

Контент вебсторінки повинен бути зрозумілим, навігація по складових сайту – зручною.

Заохочується використання при розробці сайту інших засобів вебпрограмування, таких, наприклад, як Python (Django, FastAPI), PHP, Go/Rust, якими студент оволодів самостійно, а не використав можливості штучного інтелекту.

При цьому не можна використовувати готові конструктори та CMS на кшталт WordPress, Webflow, Framer та інші, в яких можна зібрати сайт по готових темах (Elementor, Divi) майже без коду.

Програмна частина курсової роботи обов'язково демонструється під час захисту, а текстова частина роботи повинна бути логічно пов'язана з програмною частиною, і надавати повну інформацію про те, за якими етапами, яким чином, за допомогою яких програмних засобів і технологій виконувалась реалізація програмної частини.

3 ОФОРМЛЕННЯ ТЕКСТОВОЇ ЧАСТИНИ

Текстову частину курсової роботи оформляють згідно з вимогами стандарту ДСТУ 3008:2015 «Звіти у сфері науки і техніки. Структура та правила оформлювання», ДСТУ 1.5:2015 «Правила розроблення, викладення та оформлення національних нормативних документів».

3.1 Обсяг, шрифти, відступи

Текстова частина курсової роботи повинна мати обсяг **20-25 сторінок** основного тексту. При підрахунку цього обсягу не враховуються: індивідуальне завдання, анотації, зміст, список використаних джерел, додатки. Разом з тим, нумерація аркушів додатків продовжує нумерацію сторінок основної частини пояснювальної записки.

Текст курсової роботи набирається у будь-якому текстовому редакторі.

Параметри шрифта при цьому повинні бути такі: гарнітура – Times New Roman, розмір – 14, міжрядковий інтервал – 1,5.

Відступи встановлюють такої ширини: зліва – 2.5 см, справа – 1 см, згори і знизу – 2 см.

Абзацний відступ повинен бути однаковий по всьому тексту КР і складати не більше 1,25 см.

3.2 Нумерація сторінок

Сторінки КР повинні бути пронумеровані **у правому верхньому кутку** сторінки. Для правильного оформлення нумерації сторінок рекомендується скористатися можливостями Microsoft Word:

1) На перших трьох аркушах (титульний аркуш, індивідуальне завдання і анотації) номери сторінок не проставляються. Тому для цих сторінок створити окремий розділ, без нумерації.
2) Аркуші, починаючи зі змісту, повинні бути пронумеровані. Тому для них рекомендується створити новий розділ, проставляючи номери сторінок у верхньому колонтитулі справа.
3) Нумерація сторінок додатків продовжує загальну нумерацію текстової частини КР, у тому числі і перший аркуш ілюстративної частини. Тому для них немає необхідності створювати окремий розділ.
4) Аркуші вмісту ілюстративної частини не нумеруються. Для їх оформлення рекомендується створити окремий розділ, не вказуючи номери сторінок.

3.3 Оформлення розділів і підрозділів

Текст курсової роботи має бути викладений в лаконічному обґрунтувальному стилі і відповідати індивідуальному завданню. Структурними елементами основної частини пояснювальної записки є розділи, підрозділи, пункти, підпункти, переліки. Про структуру і вміст текстової частини курсової роботи йтиметься далі.

Розділ – головна ступінь поділу тексту, позначена номером і має заголовок. *Підрозділ* – частина розділу, позначена номером і має заголовок. *Пункт* – частина розділу чи підрозділу, позначена номером і може мати заголовок. *Підпункт* – частина пункту, позначена номером і може мати або не мати заголовок. Заголовки структурних елементів необхідно нумерувати тільки арабськими числами.

Для коректного оформлення розділів і підрозділів, а також для правильного формування у подальшому аркуша «ЗМІСТ» рекомендується скористатись можливостями оформлення стилів у Microsoft Word. Це набагато полегшить здійснення навігації по документу.

Так, для написання безпосередньо тексту варто надати стилю «Основний» (або «Звичайний») необхідних параметрів і далі саме цей стиль використовувати для викладення матеріалу.

Стиль «Основний» або «Звичайний»	*Шрифт*: Times New Roman, 14 *Інтервал*: 1,5 *Відступ*: ≤ 1,25 *Вирівнювання*: за шириною

Кожен ***розділ*** рекомендується починати з нової сторінки. Заголовок розділу (***заголовок 1-го рівня***) записують посередині великими літерами з більш високою насиченістю – напівжирним шрифтом). Після заголовку розділу пропускають один рядок. Розділи нумерують порядковими номерами в межах всього документа (1, 2, і т. д.). Після цифри крапка не ставиться, пропускається один пробіл.

Стиль «Заголовок 1»	*Шрифт*: Times New Roman, 14 напівжирний, ВЕЛИКІ ЛІТЕРИ *Інтервал*: перед – 0 пт після – 18 пт (пропуск одного рядка) *Відступ*: відсутній *Вирівнювання*: по центру

Заголовки ***підрозділів***, пунктів та підпунктів записують з абзацу малими літерами, починаючи з великої. Перед заголовком і після нього пропускають один рядок. ***Заголовки 2-го рівня*** (2.1, 3.2, …) записують напівжирним шрифтом.

Стиль «Заголовок 2»	*Шрифт*: Times New Roman, 14, напівжирний з великої літери малими, як в реченні *Інтервал*: перед – 18 пт (пропуск одного рядка) після – 18 пт (пропуск одного рядка) *Відступ*: на рівень абзацного відступу *Вирівнювання*: по лівому краю абзацного відступу

Заголовки 3-го рівня, якщо він має назву, записують звичайним шрифтом.

Стиль «Заголовок 3»	*Шрифт*: Times New Roman, 14 з великої літери малими, як в реченні *Інтервал*: перед – 6 пт (маленький пропуск) після – 6 пт (маленький пропуск) *Відступ*: на рівень абзацного відступу *Вирівнювання*: по лівому краю абзацного відступу

Підрозділи нумерують в межах кожного розділу, пункти – в межах підрозділу і т. д. за формою (3.1, 3.2, 3.2.1, 3.2.2 і т. д.). Цифри, які вказують номер, не повинні виступати за абзац. Після номера крапку не ставлять, а пропускають один знак. Наприклад, це може виглядати так:

2 РОЗРОБКА ВЕБЗАСТОСУНКУ

Вебзастосунок розроблений як інформаційно-інтерактивна платформа на базі технологій HTML, CSS та JavaScript. Він слугує для …

2.1 Програмні засоби для реалізації сайту

Для створення вебзастосунку було обрано комплекс сучасних клієнтських вебтехнологій, які забезпечують необхідну структуру, візуальну привабливість, динаміку та адаптивність.

2.1.1 Використання HTML для розмітки вебсторінок

HTML використовується як незмінний каркас та основа для розмітки всього контенту вебсторінок і забезпечує чітку семантичну структуру тексту необхідну для коректного відображення інформації.

…

…

Якщо розділ складається лише з одного підрозділу, або підрозділ містить лише один пункт, або пункт містить лише один підпункт, то дрібніший елемент не нумерують.

Для розділів та підрозділів наявність заголовків обов'язкова. Пункти і підпункти можуть не мати заголовка.

Допускається розміщувати текст між заголовками розділу і підрозділу, між заголовками підрозділу і пункту. Посилання в тексті на розділи і підрозділи виконується за формою: «...наведено в розділі 3» або «… представлено у підрозділі 2.1».

3.4 Оформлення формул

Кожну формулу записують з нового рядка, симетрично до тексту, курсивом. Між формулою і текстом пропускають один рядок. Умовні літерні позначення в формулі наводять в тексті або одразу під формулою. Для цього після формули ставлять кому і записують пояснення до кожного символу з нового рядка в тій послідовності, в якій вони наведені у формулі, розділяючи крапкою з комою. Перший рядок повинен починатися з абзацу зі слова «де» і без будь-якого знаку після нього. Наприклад,

Функцію Ейлера можна подати у вигляді добутку Ейлера:

$$\varphi(n) = n \cdot \prod_{p|n} \left(1 - \frac{1}{p}\right), \qquad (3.1)$$

де p – просте число;
n – натуральне число.

Всі формули нумерують в межах розділу арабськими числами. Номер вказують в круглих дужках з правої сторони, в кінці рядка, на рівні закінчення формули. Номер формули складається з номера розділу і порядкового номера формули в розділі, розділених крапкою. Дозволяється виконувати нумерацію в межах всього документа.

Формула є частиною речення, тому до неї застосовують такі ж правила пунктуації, як і до інших членів речення. Якщо формула знаходиться в кінці речення, то після неї ставлять крапку. Формули, які йдуть одна за одною і не розділені текстом, відокремлюють комою.

Посилання на формули в тексті зазначають в круглих дужках за формою: «... в формулі (3.1)» або «... в формулах (3.7, ... , 3.10)».

Рекомендується для оформлення формул використовувати редактор формул, автоматично вбудований у наявний текстовий редактор.

3.5 Оформлення таблиць

Таблицю розміщують симетрично до тексту після першого посилання на неї на даній сторінці або на наступній (якщо на даній вона не вміщується) і таким чином, щоб зручно було її розглядати без повороту або з поворотом на кут 90°. Таблиці у тексті пояснювальної записки набираються основним шрифтом, в деяких випадках розмір шрифту може бути зменшений до 10-12, а міжінтервальний проміжок – до 1 пт.

Заголовок таблиці розташовується над таблицею з вказанням її номера і назви з абзацного відступу від лівого краю таблиці, починаючи зі слова «Таблиця» і її номера (таблиці нумеруються в межах одного розділу). Перед назвою таблиці і після неї роблять невеликі інтервальні проміжки, щоб назва таблиці відділялась від тексту і від самої таблиці. Рекомендується у Microsoft Word для назв таблиць створити окремий стиль.

Стиль «Надтабличний надпис»	*Шрифт*: Times New Roman, 14 з великої літери малими, як в реченні *Інтервал*: перед – 12 пт (пропуск піврядка) після – 6 пт (маленький пропуск) *Відступ*: на рівень абзацного відступу *Вирівнювання*: по лівому краю абзацного відступу

Назви стовпців (графів) таблиці розташовують по центру. Графу «№ п/п» в таблицю не включають. При необхідності такої нумерації номера вказують в боковику таблиці перед найменуванням рядка, наприклад,

Таблиця 1.1 – Основні типи даних

Тип даних	Опис
1 float	Дійсні числа з плаваючою точкою
2 integer	Цілі числа
...	...

Якщо є можливість, краще розміщувати таблицю цілком на одній сторінці. При перенесенні ж частин таблиці на інші сторінки повторюють або продовжують найменування граф. Якщо таблиця велика, допускається виконувати нумерацію граф на початку таблиці і при перенесенні частин таблиці на наступні сторінки повторювати тільки нумерацію граф. У всіх випадках найменування (при його наявності) таблиці розміщують тільки

над першою частиною, а над іншими частинами зліва пишуть «Продовження таблиці 1.1» без крапки в кінці, наприклад,

Продовження таблиці 1.1

1	2
21 string	Рядок символів
22 char	Символьні літерали
...	...

На всі таблиці повинні бути посилання за формою «... в табл. 1» або в дужках по тексту за формою (табл. 1.1). Посилання на таблицю повинно бути наведено до того, як ця таблиця з'явиться у тексті. Посилання на таблицю, яку було наведено раніше, дають зі скороченим словом «дивись» (див. табл. 1.1) за ходом чи в кінці речення.

!

1) На всі таблиці у тестовій частині повинні бути посилання.
2) Кожна таблиця повинна мати назву над таблицею у вигляді: «Таблиця 1.2 – Назва таблиці…»

3.6. Оформлення рисунків

Усі схеми, діаграми, ілюстрації, графіки оформляються у текстовій частині у вигляді рисунків, під кожним з яких повинен бути підпис, який починається зі слова «Рисунок». Дозволяється нумерувати рисунки в межах всього документа. Схеми виконуються у чорно-білому (з відтінками сірого) вигляді. В інших випадках рисунки можна залишати у довільному кольорі.

В тексті рисунок розміщують симетрично до тексту після першого посилання на нього або на наступній сторінці, якщо на даній сторінці він не уміщується без повороту. Нумерують рисунки в межах розділів, вказуючи номер розділу і порядковий номер рисунку в розділі, розділяючи крапкою. Після номера через тире з великої літери наводять назву рисунка. Крапку в кінці не ставлять, знак переносу не використовують.

Між рисунком і текстом пропускають один рядок. Для того, щоб підрисункові підписи по всій роботі були однаковими і відділялися від тексту, рекомендується створити для підписів рисунків окремий стиль.

Стиль «Підрисунковий підпис»	*Шрифт*: Times New Roman, 14 з великої літери малими, як в реченні *Інтервал*: перед – 6 пт (пропуск піврядка) після – 12 пт (маленький пропуск) *Вирівнювання*: по центру

Якщо найменування рисунка довге, то його продовжують у наступному рядку, починаючи від найменування.

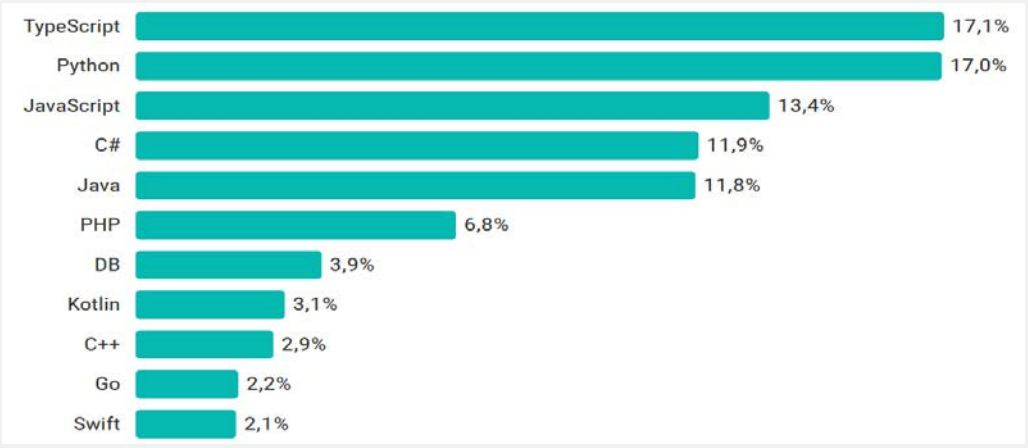
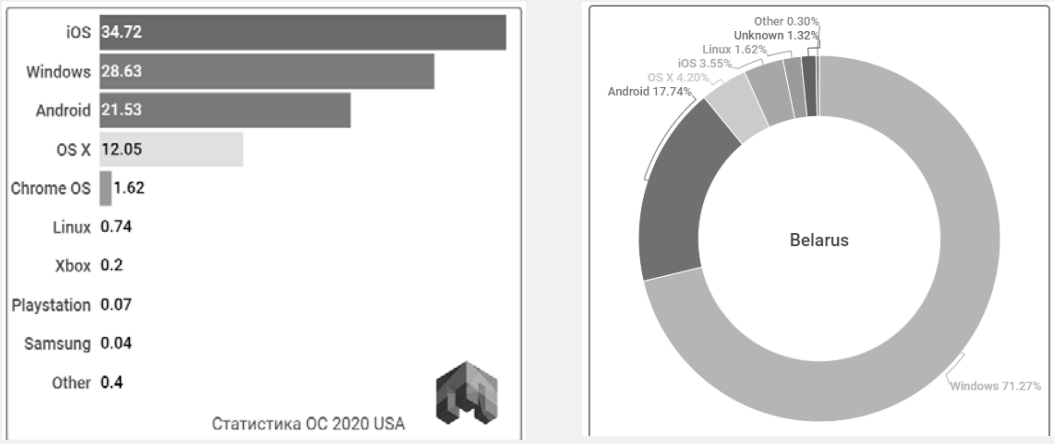


Рисунок 1.2 – Діаграма популярності мов програмування станом на жовтень 2025 року

У випадку, коли рисунок складається з декількох частин, їх позначають малими літерами українського алфавіту з дужкою (а), б), …) під відповідною частиною. В такому випадку після найменування рисунка ставлять двокрапку і дають найменування кожної частини (рис. 1.2).



а) б)

Рисунок 1.2 – Діаграми популярності операційних систем: а) стовпчикова діаграма, б) кругова діаграма

Якщо частина рисунка не вміщається на одній сторінці, то її переносять на наступні сторінки. У цьому випадку під початком рисунка вказують повне його позначення, а під продовженнями позначають «Рисунок 1.2 (продовження)».

Схеми як графічне представлення алгоритмів, структур і т. д. обов'язково повинні бути представлені у курсовій роботі – і в текстовій частині, і в додатках. В індивідуальному завданні повинно бути наведено перелік

обов’язкових схем для даної курсової роботи. Ці схеми можуть бути розміщені як в текстовій частині, так і в додатках. Надалі основні схеми виносяться в ілюстративну частину.

Серед схем можуть бути такі:

– схеми даних;
– схеми роботи програм або конкретних методів;
– схеми взаємодії модулів або методів у програмі;
– схеми ресурсів програмного застосунку.

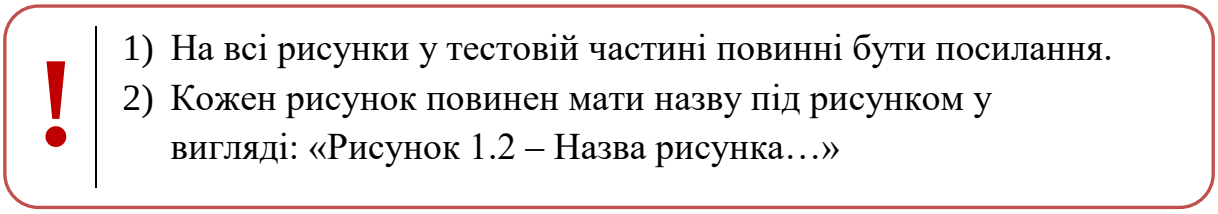
Розробник програми сам повинен вирішувати, які саме схеми доцільно розробляти у своїй роботі.

У випадку використання об’єктно-орієнтованого підходу при розробці програмного застосунку замість схеми взаємодії модулів або методів у програмі рекомендується розробити UML-діаграму класів.

Схеми і діаграми можуть використовуватися на різних рівнях деталізації, причому кількість рівнів залежить від розмірів і складності задачі оброблення даних.

Для побудови схем і діаграм можуть бути використані стандартні програмні засоби, спеціально призначені для цього або вбудовані засоби програмних середовищ, у яких здійснюється реалізація програм.

На всі рисунки повинні бути посилання за формою: «... на рис. 3.3–3.5», або в дужках по тексту за формою (рис. 3.6). Посилання на раніше наведений рисунок дають зі скороченням слова «дивись», наприклад, (див. рис. 3.4), за ходом чи в кінці речення.



3.7 Оформлення переліків

Розрізняють переліки двох рівнів. Перед кожною позицією переліку слід ставити рядкову літеру алфавіту з дужкою, або, не нумеруючи, – дефіс або тире (перший рівень деталізації). Для подальшої деталізації переліку слід використовувати арабські цифри з дужкою (другий рівень деталізації). Переліки першого рівня деталізації друкуються рядковими літерами з абзацного відступу, другого рівня – з відступом щодо місця розташування першого. Наприклад,

Рівні еталонної моделі взаємодії відкритих систем:
- перший – фізичний;
- другий – канальний:
  а) підрівень управління логічним каналом;
  б) підрівень доступу до середовища передачі;
- третій – мережевий.

Для подальшої деталізації переліку слід використовувати арабські цифри з дужкою (другий рівень деталізації). Наприклад,

а) тут пишеться текст 1-го пункту з переліку та його продовження;
б) тут пишеться текст 2-го пункту з переліку і подальша його деталізація:
   1) текст переліку подальшої деталізації вищого рівня та його продовження;
   2) . . . ;
в) останній пункт переліку.

Оформлення текстової частини курсової роботи повинно бути виконано згідно наведених правил. Такі правила оформлення діятимуть і надалі при оформленні курсових робіт з інших дисциплін, а також при виконанні бакалаврських і магістерських кваліфікаційних робіт.

Наповнення текстової частини повинно відповідати розробленому програмному застосунку.

Правильність оформлення, а також структура і вміст розділів текстової частини впливатимуть на кількість балів, отриманих при захисті курсової роботи, а, отже, і на остаточну оцінку за курсову роботу.

4 СТРУКТУРА І ВМІСТ ТЕКСТОВОЇ ЧАСТИНИ

Текст курсової роботи має бути викладений в лаконічному обґрунтувальному стилі і відповідати індивідуальному завданню.

Текстова частина курсової роботи повинна мати обсяг 20-25 сторінок. Текстову частину умовно поділяють на такі складові (табл. 4.1):

- вступну частину,
- основну частину
- додатки.

Таблиця 4.1 – Склад та рекомендований обсяг курсової роботи

Частини	Вміст	Обсяг, стор.
Вступна частина	Титульний аркуш	1
	Індивідуальне завдання	1
	Анотація державною мовою	½
	Анотація іноземною мовою (Abstract)	½
	Зміст	1
Основна частина	Вступ	1
	Розробка програмного застосунку	8-10
	Розробка вебзастосунку	8-10
	Висновки	1
	Список використаних джерел	1
Додатки	Додаток А. Код програмного засобу	Не обмежується
	Додаток Б. Код вебсторінки	
	Інші додатки (при необхідності)	
	Додаток Х. Ілюстративна частина	
Ілюстративна частина	Загальна схема роботи програмного застосунку	1
	Схеми основних алгоритмів …	2-3
	UML-діаграма класів програмного застосунку або структура ПЗ	1
	Схема ресурсів (або схема даних програмного застосунку)	1
	Фрагменти роботи програмного застосунку	2-4
	Структура вебсторінки (або схема взаємозв’язку файлів вебпроєкту)	1
	Фрагменти роботи вебзастосунку	2-4

Даний перелік розділів є лише рекомендованим. Здобувач може самостійно поділяти текстову частину на розділи на власний розсуд. Головне, щоб розділи були логічно пов’язані між собою і представляли весь процес розробки програмного застосунку.

4.1 Вступна частина

4.1.1 Титульний аркуш

Титульний аркуш *(не нумерується, не входить до загального обсягу сторінок)* виконується за встановленим зразком.

На титульному аркуші здобувач зазначає тему курсової роботи, своє прізвище, ім’я та по батькові із зазначенням групи, спеціальності, освітньо-професійної програми, а також прізвище, ініціали, науковий ступінь, учене звання та посаду керівника, а також членів комісії.

Зразок правильного оформлення титульного аркуша курсової роботи наведено у додатку Б.

4.1.2 Індивідуальне завдання

Індивідуальне завдання на курсову роботу *(не нумерується, не входить до загального обсягу сторінок, не вноситься у зміст)* оформляється певним чином і розміщується за титульним аркушем.

Індивідуальне завдання до курсової роботи видається керівником і визначає зміст текстової та ілюстративної частин. Обов’язковим в індивідуальному завданні є зазначення теми, вхідних і вихідних даних, приблизний перелік складових текстової і ілюстративної частин КР. Крім того, індивідуальне завдання до курсової роботи містить термін видачі, підписи керівника курсової роботи та здобувача.

Індивідуальне завдання затверджується на засіданні кафедри, візується завідувачем кафедри і видається здобувачу не пізніше другого тижня від початку навчального семестру.

> **!** Індивідуальне завдання:
> – розміщують на одному аркуші (для цього можна зменшити міжінтервальний проміжок);
> – не нумерується і не входить до загальної кількості сторінок.

Аркуш з індивідуальним завданням роздруковується, збираються необхідні підписи і зберігається у здобувача освіти до моменту захисту курсової роботи. Надалі цей аркуш вставляється в електронну версію курсової роботи у вигляді ксерокопії або якісної фотографії. Зразок оформлення типового індивідуального завдання наведено у додатку В.

4.1.3 Анотація

Анотацію розміщують відразу після індивідуального завдання перед змістом з нової сторінки (*не нумерується, не входить до загального обсягу сторінок, не вноситься у зміст*).

Анотація подається двома мовами (українською та англійською): АНОТАЦІЯ та ABSTRACT. Вона призначена для ознайомлення з текстовим документом курсової роботи. Анотація повинна коротко характеризувати мету роботи, засоби, використані для досягнення поставленої задачі, коротку інформацію про досягнуті результати.

Розмір анотації повинен становити приблизно ⅓ частину сторінки (не перевищувати ½ сторінки).

Анотації розміщують безпосередньо за аркушем з індивідуальним завданням, починаючи з нової сторінки, нумерація якої не зазначається і в зміст не входить. Анотації двома мовами можуть бути розташовані на окремих аркушах або на одному аркуші, якщо вони на ньому поміщаються.

Крім того, в анотації зазначаються автор роботи, її назва, а також перелік ключових слів. Зразок оформлення анотації наведено у додатку Г.

4.1.4 Зміст

Зміст розташовують безпосередньо після анотації, починаючи з нової сторінки, розташувавши по центру слово ЗМІСТ.

До змісту включають: вступ; послідовно перелічені назви всіх розділів, підрозділів, пунктів і підпунктів (якщо вони мають заголовки) суті роботи; висновки; список використаних джерел; назви додатків і номери сторінок, які містять початок матеріалу. До змісту не включають титульний аркуш, індивідуальне завдання на курсову роботу та анотації.

Нумерація у змісті починається зі ВСТУПУ (відповідно до нумерації у пояснювальній записці). Сам зміст за нумерацією є сторінкою з номером 2 або 3 в залежності від того, розміщені анотації на одній або на двох сторінках. Нумерація сторінок повинна бути наскрізною, включаючи додатки (сторінки додатків також входять до змісту).

Для того, щоб контролювати правильність підготовки змісту, рекомендується під час виконання текстової частини встановити опцію «Область навігації» в пункті меню «Подання». У цьому випадку буде зручно слідкувати за формуванням змісту і здійснювати навігацію по тексту документа.

Назви заголовків змісту повинні однозначно відповідати назвам заголовків пояснювальної записки за текстом. Формування змісту у текстовому документі необхідно здійснювати автоматично, використовуючи засоби обраного текстового редактора.

При формуванні змісту в курсовій роботі назви заголовків і підзаголовків розташовувати з різними відступами таким чином, щоб номери розділів і підрозділів нависали над назвою підрозділів меншого рівня.

Приклад оформлення змісту наведено у додатку Д.

4.2 Основна частина

4.2.1 Вступ

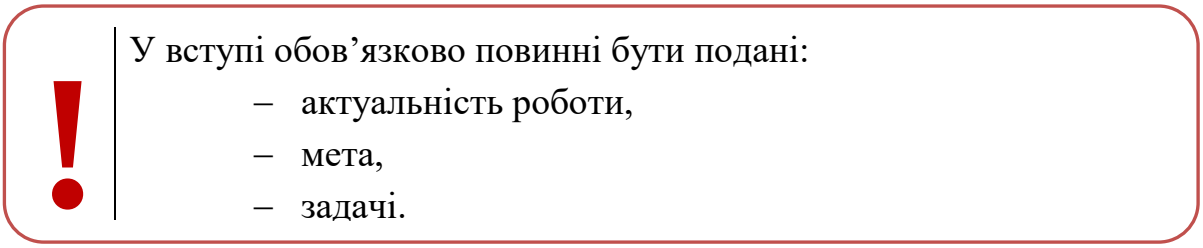
Вступ пишуть з нової пронумерованої сторінки з заголовком ВСТУП по центру великими літерами з більш високою насиченістю *(заголовок 1-го рівня, без номера)*.

Текст вступу повинен бути коротким і висвітлювати питання актуальності, значення, сучасний рівень і призначення курсової роботи. У вступі і далі за текстом не дозволяється використовувати скорочені слова, терміни, крім загальноприйнятих. Якщо у вступі і далі за текстом використовується деяке загальновживане поняття у вигляді **абревіатури**, то при першій появі цього поняття воно наводиться повністю, а поруч у дужках наводиться скорочення. При повторному використанні введеного поняття можна наводити лише скорочення у вигляді абревіатури. Наприклад,

> «… Програмний засіб (ПЗ) може знайти своє застосування при шифрування повідомлень. Для реалізації ПЗ використовувались …»

Вступ зазвичай висвітлює:

- стан розвитку проблеми в даній галузі, до якої має відношення розробка (важливість нових технологій програмування, програмних середовищ, мов програмування тощо);
- ***мету*** курсової роботи. Для даної КР метою не може бути «розробити метод шифрування …», «розробити гру ….» і т. і. Оскільки курсова з дисципліни «Програмування», метою повинно бути щось на кшталт «удосконалення знань ….», «покращення практичних навичок …»;
- ***перелік задач***, які необхідно виконати задля досягнення мети (бажано у вигляді списку);
- ***актуальність***, яку рекомендується подавати в останньому абзаці вступу з метою стислого викладання суті обраної розробки.



Обсяг вступу не повинен перевищувати 1-2 сторінок.

4.2.2 Розділи пояснювальної записки

Таких розділів може бути декілька. Для даної курсової роботи рекомендується два розділи:

– розділ, що стосується розробки програмного застосунку;
– розділ, присвячений створенню вебзастосунку.

Хоча кількість розділів і їх сутність може визначати сам автор. Головне, щоб дані розділи чітко відображали результат виконання КР, були логічно пов’язані і представляли весь процес виконання курсової роботи.

У розділі, що стосується ***розробки програмного застосунку***, повинно бути представлено:

– загальну схему функціонування програми та інших схем і діаграм, які необхідні для досягнення результату, а також математичну модель (якщо це необхідно);
– обґрунтування вибору програмних засобів для реалізації;
– опис розроблених пакетів, класів та методів програми;
– результати тестування розробленого програмного застосунку.

Описуючи основні етапи створення застосунку, необхідно наводити фрагменти коду, що підтверджують певні моменти програмної реалізації, фрагменти роботи застосунку у вигляді зображень, що дають візуальне уявлення про роботу того або іншого фрагмента коду.

У випадку, якщо тема курсової роботи стосується *реалізації певного методу шифрування*, перелік підрозділів може бути, наприклад, таким:

1.1 Формування вимог до програмного застосунку (яким розробник бачить свій застосунок, що програма повинна робити).
1.2 Обґрунтування вибору програмних засобів для реалізації (яка мова програмування і чому, яке середовище і чому і т. д.).
1.3 Розробка і реалізація інтерфейсу (загальна схема роботи застосунку з поясненнями, навігація, які пакети і класи, які вікна, інтерфейси цих вікон, які елементи керування і їх обробка і т. д.).

1.4 Побудова алгоритмів шифрування (сутність методу, математична модель, контрольний приклад, алгоритм і схеми зашифрування і розшифрування).

1.5 Програмна реалізація шифрування (який клас, які методи і т. д.).

1.6 Вхідні і вихідні дані (звідки і як вводити дані, де зберігаються результати, обробка виключних ситуацій тощо, схема даних, робота з файлами, колекціями тощо).

1.7 Структура проєкту (склад проєкту, зв'язки між класами, UML-діаграма класів).

1.8 Тестування програмного застосунку (результати перевірки програми на всіх її режимах роботи).

Якщо тема стосується розробки *ігрового застосунку*, необхідно акцентуватись на правильній побудові її стратегії гри, на рівнях гри, на інтерфейсі, можливості збереження результатів гри та їх статистичній обробці (найкращий гравець, сортування за кількістю бонусів тощо).

У випадку розробки ігрового застосунку перелік підрозділів може бути, наприклад, таким:

1.1 Розробка стратегії гри (сутність гри, правила, очікувані рівні, результати, …).

1.2 Обґрунтування вибору програмних засобів для реалізації (яка мова програмування і чому, яке середовище і чому і т. д.).

1.3 Розробка і реалізація інтерфейсу (загальна схема роботи застосунку з поясненнями, навігація, які пакети і класи, які вікна, інтерфейси цих вікон, які елементи керування і їх обробка і т. д.).

1.4 Побудова алгоритму проходження ігрового процесу (етапи, вибір рівнів складності тощо).

1.5 Програмна реалізація процесу гри (який клас, які методи і т. д.).

1.6 Структура проєкту (склад проєкту, UML-діаграма класів тощо).

1.7 Аналіз роботи програмного засобу (перевірка роботи гри з різними параметрами, обробка виключних ситуацій, збереження результатів тощо).

Розробка алгоритмів повинна супроводжуватись відповідними схемами (схема роботи програми або функції, схема даних тощо), які надалі виносяться в ілюстративну частину для використання їх під час захисту курсової роботи.

У розділі, що стосується ***розробки вебзастосунку***, необхідно відобразити покроково процес створення сайту:

– етап розмітки, семантика сайту (скільки сторінок, які, підготовка контенту, які складові сторінок тощо);

– реалізація навігації (якоря, посилання, переходи …);
– візуальне оформлення інформації на сайті;
– динамічні елементи, елементи керування, анімація тощо.

Перелік підрозділів, наприклад, може бути такий:

2.1 Вимоги до вебзастосунку і обґрунтування вибору засобів для його реалізації (що в ньому повинно бути представлено, як вона повинна виглядати, якими засобами буде реалізовано тощо).

2.2 Розробка структури вебсайту (тут переважно про HTML: якими елементами-тегами реалізується основа сторінки, як і на які частини розбито вебсторінку, навігація, текстове та графічне наповнення, …).

2.3 Стилізація та візуальне оформлення (тут про застосування CSS: які файли стилів створено, як вони підключалися, які стилі використано і для чого і т. д.).

2.4 Реалізація динамічних елементів (тут про використання JavaScript: які js-файли створено, як підключено, які функції і для чого розроблено і т. д.).

2.5 Аналіз роботи вебзастосунку (тут може йтися про структуру проєкту і зв'язків між складовими вебсторінки тощо).

Розділи основної частини КР мають бути змістовними, конкретними, саме вони демонструють знання та навички у галузі програмування.

Наприкінці кожного розділу необхідно навести короткий (1-3 речення) ***висновок по даному розділу***. Наприклад, в кінці першого розділу висновок може бути сформульований таким чином:

…Таким чином, у даному розділі проведено детальне представлення процесу розробки і програмної реалізації застосунку. Проведене тестування показало, що програма працює коректно у всіх запропонованих режимах. Функціональне навантаження виконується належним чином.

4.2.3 Висновки

Висновки оформляють з нового пронумерованого аркуша, вгорі якого по центру великими літерами більш високої насиченості пишуть слово «ВИСНОВКИ» *(заголовок 1-го рівня, без номера)*.

Висновки мають бути конкретними, а не загальними. У висновках наводяться основні результати курсової роботи. Слід зазначити, яким чином і з яким результатом було ***виконано ті задачі, про які йшлося у вступі***. За результатами виконання курсової роботи надаються обґрунтовані висновки щодо переваг та недоліків застосування тієї чи іншої мови програмування, того чи іншого засобу програмування, недоліки та переваги

даного програмного продукту, труднощі при розробленні програми та причини, що їх обумовили і можливі шляхи їх подолання, можливі рекомендації прикладного застосування та шляхи (перспективи) удосконалення розробленого програмного забезпечення.

Рекомендований обсяг висновків для даної КР – 1-1,5 сторінок.

4.2.4 Список використаних джерел

Перелік використаної літератури та інших інформаційних джерел оформляють з нової пронумерованої сторінки, розташовуючи заголовок “СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ” *(заголовок 1-го рівня, без номера)* по центру великими літерами більш високої насиченості.

Джерела подаються у вигляді їх бібліографічних посилань відповідно до ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання» або інших міжнародних стилів бібліографічного опису (IEEE style, MLA style, APA style, Harvard style, Chicago style та ін.).

У перелік використаних джерел можуть входити книжки з програмування відповідною мовою (C/C++, C#, Java, Java FX, …), адреси офіційних сайтів, посилання на матеріали навігатора навчальних ресурсів (HHP) в системі JetIQ, методичні вказівки до виконання лабораторних робіт і курсової роботи тощо.

У переліку наводять найменування використаних інформаційних джерел, патентів, нормативно-технічних документів, інформації з інтернету (назви і адреси сайтів при цьому – обов’язково) і т. д. Перелік містить список джерел, які було використано в процесі виконання роботи, і на які повинні бути обов’язкові посилання в тексті пояснювальної записки.

Усі джерела в загальний список ***записуються в порядку посилання*** на них в тексті. ***Посилання на джерело*** наводять в тексті пояснювальної записки в квадратних дужках **[…]**, вказуючи порядковий номер за списком. У списку кожне джерело записують мовою оригіналу з абзацу, нумерують арабськими цифрами, починаючи з одиниці.

> **!** Посилання на джерела по тексту виставляються у зростаючому порядку, а джерела у списку використаних джерел нумеруються у порядку цих посилань, а не навпаки.

Приклад оформлення переліку використаних джерел різного характеру наведено у додатку Ж.

Рекомендований обсяг списку використаних джерел для курсової роботи становить 12-15 найменувань.

4.3 Додатки

Додатки повинні містити матеріал, який не увійшов в основні розділи пояснювальної записки: коди програм, підпрограм та функцій, результати тестування програми у вигляді образів екранів, таблиць, графіків, схеми роботи програм і функцій, схеми алгоритмів, схеми ресурсів і даних, схеми взаємодії програм – ***те, що не увійшло безпосередньо в основні розділи*** пояснювальної записки.

Кожен додаток необхідно починати з нової сторінки, вказуючи зверху посередині рядка слово «Додаток» (рекомендується використовувати заголовок 2-го рівня, але без нумерації, з вирівнюванням по центру). Кожен додаток повинен мати тематичний (змістовний) заголовок, який записують посередині рядка малими літерами з першої великої (рекомендується назви додатків також оформляти стилем заголовків 2-го рівня, змінивши лише вирівнювання по центру).

Приклад правильного оформлення додатків можна подивитись у даних методичних вказівках.

Додатки позначають послідовно великими українськими літерами, за винятком літер Ґ, Є, З, І, Ї, Й, О, Ч, Ь, наприклад, Додаток А, Додаток Б. Останнім додатком повинен бути додаток з назвою «Ілюстративна частина».

Сторінки додатків нумеруються, продовжуючи загальну нумерацію у пояснювальній записці.

> **!** Всі додатки включають у ЗМІСТ, вказуючи номер додатка, заголовок і номер сторінки, з яких вони починаються.

Якщо кількість додатків три і більше, перед додатками розміщують окремий аркуш, по центру якого стилем заголовку 1-го рівня пишуть слово «ДОДАТКИ».

4.3.1 Лістинги програм у додатках

В даній курсовій роботі в додатках обов’язково повинні бути повні коди програм і вебсторінок. Це можна оформити як два окремих додатки.

Тексти програм, вміст файлів таблиць css-стилів, js-файлів і т. д. слід копіювати безпосередньо з програмних середовищ. Розмір шрифта – не більше 8 пт, гарнітура шрифта – довільна (бажано використовувати шрифт безпосередньо з програмного середовища), міжрядковий інтервал – не більше одинарного.

Якщо програма складається з декількох файлів, перед друком кожного файлу слід написати його назву. Наприклад,

// Клас SoundPlayer.java

```
class SoundPlayer {
     public static final String CORRECT_SOUND_PATH = "\\665b1658a530bef.wav";
    public static final String INCORRECT_SOUND_PATH = "\\online-audio-
converter.com.wav";
    private static Clip correctClip;
    private static Clip incorrectClip;
    public static void loadSounds() {
        new Thread(() -> {
            try {
                correctClip = getClip(CORRECT_SOUND_PATH);
                incorrectClip = getClip(INCORRECT_SOUND_PATH);
                System.out.println("Аудіо ресурси успішно завантажені.");
            } catch (Exception e) {
                System.err.println("Помилка ініціалізації аудіо: " +
e.getMessage());
            }
        }).start();
   ...
```

// Файл myStyles.css

```
body {font-family: Arial, sans-serif;
      font-size: 1.2rem;
      transition: margin-left 0.5s;
      background-color: lightblue;
```

4.3.2 Інструкції по роботі з програмою

Метою даної курсової роботи є розробка програмного застосунку, тому рекомендується передбачити розробку інструктивних матеріалів щодо роботи з програмою, які виносять у додаток, що може мати назву: «Інструкції для роботи з програмою ...», або «Рекомендації для ...» тощо.

Інструкцію слід підготувати у вигляді сукупності певних підпунктів:

1) *Призначення програми*. Тут слід вказати назву програмного засобу, для чого призначено даний програмний засіб, хто може його використовувати.
2) *Системні вимоги*, а саме:
   – вимоги до технічних засобів (вказують мінімальний склад технічних засобів, що забезпечують роботу програми);
   – вимоги до операційного середовища (яка операційна система і, можливо, додаткові програми).
3) *Налаштування програмного засобу*. Тут слід вказати:
   – склад програмного засобу (які файли необхідні для його роботи), тобто, що і куди скопіювати для правильної роботи програми;
   – настроювання програми і перевірка правильності роботи програми (опис дій для налаштування програми на умови конкретного застосування, опис способів перевірки, що дозволяють дати загальний висновок про роботоздатність програми – контрольні приклади, методи прогону, результати. При необхідності наводять пояснювальні приклади);

– повідомлення (тут повинні бути вказані тексти повідомлень в ході виконання налаштування, перевірки програми, а також в ході виконання програми, опис їх змісту і дій, які необхідно виконати після аналізу цих повідомлень).

4) *Виконання програми*.
   – звернення до програми (опис процедури виклику програми, способи передачі управління, параметрів та ін.);
   – вхідні і вихідні дані (опис організації використовуваної вхідної і вихідної інформації і, при необхідності, її кодування);
   – виконання програми (послідовність дій, що забезпечують завантаження, запуск, виконання і завершення програми, опис функцій, формату і можливих варіантів команд, за допомогою яких оператор здійснює завантаження і управління ходом виконання програми);
   – повідомлення користувачу (тексти повідомлень в ході виконання програми, опис дій, які слід виконати після аналізу цих повідомлень).

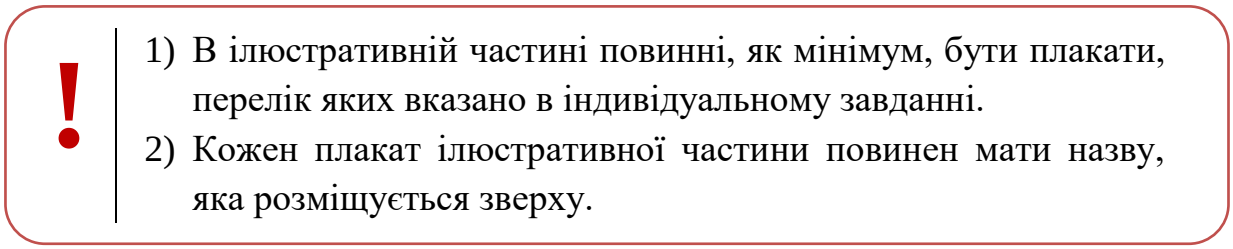
4.4. Ілюстративна частина як додаток

Ілюстративна частина (ІЧ) призначена для того, щоб допомогти студенту вдало захистити свою роботу. В ілюстративну частину слід виносити усі ті матеріали, які допоможуть представити етапи розробки курсової роботи, основні схеми, алгоритми, математичну модель, фрагменти інтерфейсу розробленої програми. Вимоги до переліку плакатів ілюстративної частини зазначаються в індивідуальному завданні.

Ілюстративна частина не потребує обов’язкового дотримання вимог стандартів, оформляється як один додаток з назвою «Ілюстративна частина» і вноситься у зміст пояснювальної записки, тобто, ілюстративна частина є одним (як правило, останнім) з додатків КР.

Ілюстративна частина починається з нового аркуша, що продовжує нумерацію, на якому по центру написано «Ілюстративна частина» і тема КР. Зразок оформлення першого аркуша ілюстративної частини наведено у додатку К.

Наступні аркуші ІЧ не нумеруються і оформляються довільним чином. Кожен плакат повинен мати назву, зазначену в індивідуальному завданні.

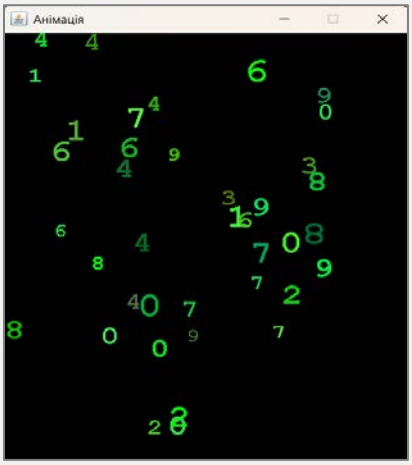


В ілюстративну частину може бути винесено більше плакатів – це на розсуд автора КР. Це можуть бути фрагменти інтерфейсу, проміжні результати, додаткові таблиці, використані в програмі, наприклад, для шифрування.

Якщо деякі плакати можуть містити декілька рисунків, ілюстрацій, зображень і т. д. У цьому випадку плакат повинен мати загальну назву зверху, а далі, під кожним рисунком, окрема назва саме цього рисунку. Наприклад, у випадку, коли плакат представляє фрагменти роботи програмного застосунку (або вебсайту), він може виглядати таким чином.

Фрагменти інтерфейсу програмного застосунку

Вигляд головного вікна програми



Вигляд вікна з анімацією

Вигляд ігрового вікна програми

В ілюстративну частину обов’язково виносяться схеми, що представляють основні алгоритми, реалізовані у програмному застосунку. Це можуть бути:

– *схеми даних* (відображають шлях даних при розв’язанні задач і визначають етапи обробки, а також різні використовувані носії даних);
– *схеми роботи* програм або конкретних методів (відображають послідовність операцій в програмі або у методі);

– *схеми взаємодії* модулів або методів у програмі (відображає шлях активацій певних модулів і методів і взаємодій з відповідними даними, що передаються і повертаються);
– *схеми ресурсів* програмного застосунку або вебзастосунку (відображає конфігурацію блоків даних і оброблювальних блоків, яка потрібна для розв’язання задачі).

Розробник програми сам повинен вирішувати, які саме схеми доцільно розробляти у своїй роботі.

Перелік символів у схемах до програм наведено у додатку Л.

У випадку використання об’єктно-орієнтованого підходу при розробці програмного застосунку замість схеми взаємодії модулів або методів у програмі рекомендується розробити UML-діаграму класів. Приклад UML-діаграми класів наведено у додатку М.

Схеми і діаграми можуть використовуватися на різних рівнях деталізації, причому кількість рівнів залежить від розмірів і складності задачі оброблення даних.

Для побудови схем і діаграм можуть бути використані стандартні програмні засоби, спеціально призначені для цього або вбудовані засоби програмних середовищ, у яких здійснюється реалізація програм.

5 ОРГАНІЗАЦІЯ ВИКОНАННЯ І ЗАХИСТУ КУРСОВОЇ РОБОТИ

5.1 Рекомендований графік виконання курсової роботи

Рекомендується такий графік виконання курсової роботи, який враховує самостійну роботу студентів під час 3-го семестру (18 тижнів) (табл. 5.1).

Таблиця 5.1 – Рекомендований графік виконання курсової роботи

Зміст розділу	Термін виконання (тиждень)
Отримання завдання на курсову роботу, розробка і оформлення індивідуального завдання	1-2
Розробка структури програмного застосунку: дослідження алгоритму задачі, виконання вручну контрольних прикладів, розробка інтерфейсу, обґрунтування необхідності додаткових засобів, розробка структури вхідних і вихідних даних, підбір необхідних елементів керування і т. д.	3-4
Програмна реалізація застосунку і налагоджування його: програмування та тестування основних процедур та функцій, програмна реалізація інтерфейсу, програмна реалізація роботи з файлами, з елементами керування, реалізація захисту та перевірки цілісності даних і т. д.	5-11
Тестування розробленого програмного застосунку та виправлення виявлених недоліків. Підготовка контрольних прикладів.	11-12
Розробка вебсторінки для супроводження програми, реалізованої в курсовій роботі	13-14
Оформлення текстової частини курсової роботи, підготовка ілюстративного матеріалу	14-15
Представлення курсової роботи для попередньої перевірки: демонстрація роботи програми та чернетки текстової частини (електронний варіант)	15-16
Коригування і доповнення (при необхідності) програми згідно із зауваженнями керівника курсової роботи, врахування і виправлення текстової частини	15-16
Перевірка курсової роботи на наявність текстових запозичень	17
Захист курсової роботи	17-18

Готовність до захисту курсової роботи визначає керівник за результатами попередньої перевірки якості текстової та ілюстративної частин і роботоздатності програмного застосунку.

Обмін файлами КР між здобувачами вищої освіти та керівником здійснюється виключно через інструмент «Файл-Експрес» системи JetIQ.

Готові курсові роботи подаються здобувачами вищої освіти через інструмент «Файл-Експрес» системи JetIQ у вигляді одного файлу у форматі Portable Document Format (*.pdf).

У процесі формування електронної відомості успішності в системі JetIQ файли захищених КР вивантажуються в електронний архів ВНТУ, де зберігаються в електронному архіві ВНТУ протягом одного року.

Доступ до електронного архіву ВНТУ, де зберігаються файли захищених КР, має відповідальний за моніторинг якості та удосконалення курсового проєктування Ради з якості освіти ВНТУ.

5.2 Академічна доброчесність при виконанні курсової роботи

Курсові роботи не повинні містити списування, академічного плагіату, фальсифікації, фабрикації, елементів, створених за допомогою систем генеративного штучного інтелекту, або інших ознак академічної недоброчесності відповідно до «Положення про академічну доброчесність у ВНТУ».

Контроль за дотриманням вимог академічної доброчесності при виконанні курсової роботи здійснює її керівник. Підтвердженням того, що керівник не виявив ознак порушення академічної доброчесності здобувачем вищої освіти, вважається факт прийняття роботи до захисту.

З метою забезпечення якості вищої освіти та відповідно до «Положення про запобігання академічному плагіату та порядок його виявлення у наукових, кваліфікаційних, навчальних та науково-методичних роботах у ВНТУ» ***електронні версії курсових робіт підлягають перевірці на наявність текстових запозичень*** за допомогою програмно-технічних засобів, які дозволяють згенерувати звіт за результатами перевірки зі встановленням факту наявності чи відсутності текстових та інших запозичень. Перевірці підлягає текстова частина КР, яка передбачає написання унікального тексту.

Відповідальність щодо дотримання академічної доброчесності у прийнятих курсових робіт несе керівник. Необхідність перевірки на наявність текстових запозичень встановлюється на засіданні кафедри разом з обсягом перевірки. Наявність протоколів не є обов’язковою. Прийняття викладачем КР до захисту автоматично означає відсутність плагіату в роботі. Тому наявність або відсутність протоколу перевірки визначає кафедра.

У разі виявлення ознак академічного плагіату, інших ознак академічної недоброчесності або якщо рівень унікальності тексту складає менше 60 %, така КР не допускається до захисту.

Якщо файл недопущеної до захисту КР було надіслано здобувачем вищої освіти через інструмент «Файл-Експрес» системи JetIQ не пізніше останнього дня теоретичного семестру (міжсесійного періоду для здобувачів вищої освіти заочної форми навчання), то така КР повертається керівником здобувачу вищої освіти на доопрацювання з відповідними

зауваженнями. При цьому здобувачу вищої освіти надається можливість повторного надсилання КР на перевірку керівнику.

Консультування здобувачів вищої освіти з питань академічної доброчесності при виконанні КР здійснює керівник, а в разі його тимчасової відсутності – інший науково-педагогічний працівник кафедри, визначений розпорядженням завідувача кафедри.

5.3 Критерії оцінювання виконання курсових робіт

При оцінюванні курсової роботи враховуються (табл. 5.2):

- ступінь і якість виконання індивідуального завдання;
- ступінь і якість виконання програмної частини курсової роботи;
- відповідність текстової частини встановленим вимогам до змісту і оформлення КР ступінь і якість виконання ілюстративної частини;
- рівень і якість представлення результатів курсової роботи здобувачем вищої освіти під час захисту;
- відповіді на запитання в процесі захисту КР.

Таблиця 5.2 – Розподіл бальної оцінки за виконання курсової роботи

Складові КР	Вимоги	Бали
Текстова частина	– дотримання вимог щодо оформлення згідно ДСТУ 3008:2015; – відповідність текстової частини і програми індивідуальному завданню; – відповідність текстової частини програмному забезпеченню; – логічна пов’язаність розділів текстової частини КР.	30
Програмна частина	– виконання основної задачі КР (функціональність); – зрозумілість і зручність інтерфейсу програми; – якість написання коду програми (читабельність, структура програми, використання функцій, класів, бібліотек тощо); – передбачення системи допомоги, обробки виключних ситуацій; – якість виконання вебсторінки до курсової роботи; – наявність у додатках інструкцій по роботі з програмою і їх відповідність програмному забезпеченню.	40
Ілюстра-тивна частина	– дотримання вимог щодо оформлення схем; – правильність оформлення плакатів; – повнота підготовки матеріалу для захисту.	10
Захист роботи	– демонстрація працездатного програмного застосунку або інших результатів розробки; – вміння пояснити основні моменти розробки; – обґрунтованість відповідей на запитання членів комісії; – представлення розробки (презентація, висновки, …).	20
РАЗОМ:		100

5.4 Порядок захисту

Захист курсових робіт відбувається під час заліково-екзаменаційної сесії згідно з розкладом контрольних заходів, визначених деканатом. До захисту допускаються курсові роботи, які оформлені відповідно до вимог, викладених в даних методичних вказівках.

Якщо здобувач вищої освіти не з'явився на захист КР у зв'язку з поважною причиною (перешкода стихійного характеру, хвороба, воєнні дії на території перебування здобувача або інші обставини, які позбавили його можливості особисто і своєчасно прибути на захист), такому здобувачу вищої освіти надається можливість захистити КР в інший день за узгодженням з керівником, але не пізніше останнього дня заліково-екзаменаційної сесії. Про поважну причину неявки на захист здобувач вищої освіти повинен повідомити керівника і деканат відповідного факультету протягом доби від призначеного часу захисту.

Якщо здобувач вищої освіти не з'явився на захист КР без поважної причини, такий здобувач вищої освіти отримує незадовільну оцінку (від 0 до 59 балів за стобальною шкалою) з виставленням її у відомість успішності.

Якщо оцінка за стобальною шкалою склала від 35 до 59 балів включно, то здобувач вищої освіти має право на доопрацювання і захист КР з виставленням оцінки у другу відомість.

Якщо оцінка за стобальною шкалою склала від 0 до 34 балів включно, то здобувач вищої освіти вважається таким, що має академічну заборгованість. Для її ліквідації здобувач повинен виконати курсову роботу за новою темою (або зміненим індивідуальним завданням) у відповідності до «Положення про порядок ліквідації академічної заборгованості, академічної різниці та надання платної послуги з проведення занять з вивчення окремої навчальної дисципліни понад обсяги, встановлені навчальним планом».

Захист курсових робіт повинен мати публічний характер і прийматися комісією, до складу якої входять не менше 2-х осіб з-поміж викладачів кафедри, один з яких є керівником КР.

Процедура захисту включає коротку доповідь здобувача вищої освіти щодо основних результатів виконання КР, демонстрування роботи програмного застосунку та вебзастосунку з необхідними поясненнями, а також відповіді на запитання членів комісії або інших присутніх на захисті осіб.

За результатами захисту КР комісія визначає оцінку, яку керівник оголошує здобувачам вищої освіти в день захисту та виставляє у відомість успішності.

В результаті курсова робота оцінюються за 100-бальною шкалою:

Сума балів за за КР	Оцінка ECTS
90 - 100	A
82 – 89	B
75 – 81	C
64 – 74	D
60 – 63	E
35 – 59	FX
0 – 34	F

У разі незгоди з отриманою оцінкою здобувачі вищої освіти мають право оскаржити її у відповідності до «Порядку організації та проведення заліків, диференційованих заліків, екзаменів у ВНТУ».

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Положення про курсове проектування у Вінницькому національному технічному університеті / уклад. Д. Штофель. Вінниця : ВНТУ, 2024. 51 с.
2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. [Чинний від 2015-06-22]. Вид. офіц. Київ : УкрНДНЦ, 2016. 26 с.
3. ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання». [Чинний від 2016-07-01]. Вид. офіц. Київ : УкрНДНЦ, 2016. 16 с.
4. Положення про запобігання академічного плагіату та порядок його виявлення у наукових, кваліфікаційних, навчальних та науково-методичних роботах у Вінницькому національному технічному університеті : наказ ВНТУ від 03.04.2020 р. № 95. URL: https://vntu.edu.ua/uploads/2020/plag.pdf (дата звернення: 12.02.2026).
5. ДСТУ 3974–2000 Правила виконання дослідно-конструкторських робіт. Загальні положення. [Чинний від 2000-11-27]. Вид. офіц. Київ : Держстандарт України, 2001. 34 с.

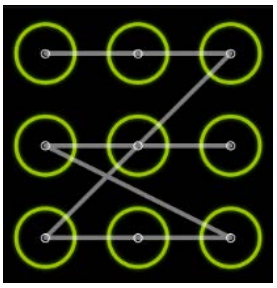
ДОДАТКИ

Додаток А

Варіанти індивідуальних завдань

Теми, пов’язані з управлінням доступом та даними і візуалізацією криптографічних алгоритмів

1.	**Графічна програма-менеджер паролів.** Застосунок повинен дозволяти користувачу створювати, переглядати і видаляти записи з логінами та паролями. Усі дані мають зберігатися в зашифрованому вигляді в локальному файлі (можна використовувати будь-який метод шифрування). Також необхідно використовувати ключовий пароль для доступу до цього файлу.
2.	**Створення візуального симулятора для розмежування прав доступу.** Це може бути програма, яка імітує файлову систему і показує, як різні користувачі мають або не мають доступу до певних файлів і папок, відповідно до заданих прав.
3.	**Симулятор атаки методом перебору (Brute-Force).** Створіть GUI-застосунок, що імітує вхід до системи. Користувач може задати пароль, і програма буде візуально показувати, як іде перебір комбінацій для його зламу, та рахувати час, що на це потрібен.
4.	**Створити застосунок для безпечного знищення інформації.** Звичайне видалення файлу не стирає дані повністю. Розробіть застосунок, що не просто видаляє файли, а спочатку перезаписує їхній вміст випадковими даними, а потім стирає. Це покаже, як насправді відбувається безпечне видалення даних.
5.	**Візуалізація роботи алгоритму зашифрування та розшифрування** (наприклад, алгоритм шифрування Цезаря). Створити застосунок з графічним інтерфейсом, де можна вводити текст і ключ, а програма покроково показуватиме, як кожна літера зашифровується або розшифровується.
6.	**Візуалізація алгоритму RSA.** Створитиграфічний застосунок, який покроково показує, як відбувається обмін відкритим ключем та як повідомлення зашифровується і розшифровується. Можна використовувати дуже малі числа, щоб спростити візуалізацію.
7.	**Програма для демонстрації гешування.** Розробити застосунок, який приймає будь-який рядок, обчислює його геш (наприклад, за алгоритмами MD5, SHA-1, SHA-256) і показує результат. Інтерфейс може візуалізувати, як навіть невелика зміна вхідного тексту повністю змінює хеш.
8.	**Цифровий пароль.** Реалізувати процес автентифікації, де логін – будь-який рядок, а пароль – цифровий. При цьому цифри для введення паролю – кнопки з цифрами у випадковому порядку, який змінюється при кожному вході. Передбачити два режими: режим реєстрації і режим входу.
9.	**Графічний пароль**. Реалізувати процес автентифікації, де логін – будь-який рядок, а пароль – послідовність координат точок, натиснутих мишею на картинці, кількість вибраних точок і назва файлу с картинками. Кількість картинок змінна, кількість точок паролю – не більше 5. Передбачити два режими: режим реєстрації і режим входу. Вся інформація про користувача (логін, параметри графічного паролю, назва файла с картинкою) зберігається у файлі. 

10.	**Графічний ключ**. Реалізувати автентифікацію за графічним ключем (як в пристроях з ОС Android). На екрані – поле для введення логіну та послідовність кружків (прямокутників, трикутників), розташованих на у вигляді матриці. Причому кількість рядків і стовпців змінюється користувачем, а кольори – змінюватимуться у часі. Ключовою інформацією є розмір матриці, індекси вибраних фігур і їх послідовність. Інформація про користувача (логін і параметри паролю) описується за допомогою класу і зберігається у файлі. 
11.	**Капча**. Реалізувати буквенно-цифрову капчу: випадкові літери і цифри, випадковий розмів символів, випадкові кольори, випадкові лінії. Передбачити перевірку правильності введення капчі, підраховуючи при цьому кількість помилкових спрацювань. Інформація про згенеровані капчі, кількість символів у капчі, кількість помилкових спрацювань зберігається у файлі. 
12.	**Симулятор двофакторної автентифікації (2FA).** GUI-застосунок, що демонструє вхід у систему з логіном, паролем та одноразовим кодом (TOTP / SMS-імітація). Показати, як змінюється код у часі.
13.	**Перевірка складності паролів.** GUI-застосунок, який аналізує пароль і візуально показує його стійкість (довжина, ентропія, словникові атаки).
14.	**Система блокування акаунта.** Реалізувати захист від brute-force: обмеження кількості спроб входу, затримка або тимчасове блокування користувача.
15.	**Пароль на основі рухів мишею.** Пароль – це траєкторія руху миші; програма порівнює форму та напрям.
16.	**Аутентифікація за часовими інтервалами.** Пароль вводиться з певними паузами між символами.
17.	**Комбінований пароль (текст + графіка).** Частина пароля вводиться з клавіатури, частина – кліками по зображенню.
18.	**Генератор надійних паролів**. Суть: візуальний конструктор паролів. Функціонал: чек-бокси «Використовувати цифри», «Спецсимволи», «Великі літери», повзунок вибору довжини пароля.
19.	Програма для приховування тексту (Base64 візуалізатор). Це не зовсім шифрування, а кодування, яке часто використовується в IT. Суть: перетворення звичайного тексту в формат Base64 і навпаки. Чому це просто: у більшості мов програмування (Python, C#, Java) є вбудована функція для Base64. Візуалізація: можна зробити два вікна: в одному пишемо «Привіт», в іншому миттєво з'являється `UHJpdml0`. Це дозволяє зрозуміти, як дані готуються до передачі мережею.
20.	Гра «Зламай код» (Mastermind / Бики та Корови). Це логічна гра, яка імітує процес підбору пароля. Суть: комп'ютер загадує 4 цифри, а користувач намагається їх вгадати. Програма підказує, скільки цифр на своїх місцях. Це ідеальна демонстрація логіки автентифікації та спроб доступу, але у формі гри.
21.	Детектор «слабких» паролів. Програма, яка просто перевіряє введені дані за списком найгірших паролів світу. Суть: користувач вводить пароль, а програма миттєво каже, чи є він у базі (наприклад, "123456", "password", "qwerty"). Функціонал: робота зі списком (масивом) слів. Якщо збіг знайдено – виводиться попередження «Ваш пароль дуже популярний, його зламають за 1 секунду».
22.	Візуальний «Детектор підглядання» (Masking Tool) – програма, яка демонструє, як працюють поля введення паролів. Суть: користувач вводить текст, а програма відображає його різними способами (на вибір): як крапки (••••), як зірочки (****), або у бінарному коді (нулі та одиниці). Додати кнопку «Око» (як у браузерах), щоб показати/приховати введений текст. Додатково можна додати функцію «Шредер», яка візуально анімує очищення поля (наприклад, символи зникають один за одним), показуючи, що дані видалено з пам'яті.

23.	Калькулятор часу брутфорсу (Simple Brute-Force Calc). Це полегшена версія симулятора атаки, але без самого процесу перебору. Суть: користувач вводить пароль, а програма просто рахує за математичною формулою, скільки часу знадобиться звичайному комп'ютеру, щоб його вгадати. Наприклад, якщо пароль лише з цифр – це 10^n комбінацій. Якщо додаються літери – 62^n (знайти подібні залежності в Інтернеті). Програма ділить кількість комбінацій на умовну швидкість комп'ютера (наприклад, 1 млн спроб/сек). Як результат, виведення результату: «Цей пароль буде зламано за 2 години» або «за 500 років».

Теми, пов'язані з простими методами шифрування інформації

24.	Реалізувати **шифрування Даніеля Дефо**. Сутність його у тому, що у зашифрованому тексті значення мали лише літери, що стоять на парних (або непарних) місцях. Наприклад, фраза "КУРСОВА РОБОТА" після зашифрування може виглядати так: "КЙУУРЕСИОЛВІАПРЮОЛБФОКТРАС".
25.	Реалізувати **шифрування "слободське письмо"**. Ідея використання літер "чужого алфавіту" для засекречування повідомлень отримала розвиток у XVII-XVIII ст. Вона полягає в написанні українських (російських) слів латинськими літерами.
26.	Реалізувати шифрування **методом прямої перестановки**, для якої ключем слугує розмір таблиці. Наприклад, повідомлення "ЦЕ МОЯ ПЕРША КУРСОВА РОБОТА" записується у таблицю по стовпцях. Результатом заповнення таблиці розміром 5×5 буде: Ц П К В О Е Е У А Т М Р Р Р А О Ш С О Ц Я А О Б Е Після заповнення таблиці текстом повідомлення по стовпцях для формування шифротексту зчитують вміст таблиці по рядках. Якщо шифротекст записувати групами по 5 літер (ключ – розмір таблиці), виходить таку шифроване повідомлення: ЦПКВО ЕЕУАТ МРРРА ОШСОЦ ЯАОБЕ. Передбачити також дане шифрування для випадку, коли запис повідомлення у таблицю відбуватиметься по рядках.
27.	Реалізувати шифрування **методом поодинокої перестановки** за ключем (система шифрування Фальконета). Даний метод базується на методі, наведеному у варіанті 3, але відрізняється тим, що стовпці таблиці переставляються за ключовим словом, фразою або набором чисел довжиною в рядок таблиці. Наприклад, візьмемо як ключ слово "ДИСКЕТА", а текст повідомлення візьмемо з попереднього варіанта. Наведемо дві таблиці: одна відповідає заповненню до перестановки, друга – після перестановки. *До перестановки:* Д И С К Е 1 3 5 4 2 Ц П К В О Е Е У А Т М Р Р Р А О Ш С О Ц Я А О Б Е *Після перестановки:* Д Е И К С 1 2 3 4 5 Ц О П В К Е Т Е А У М А Р Р Р О Ц Ш О С Я Е А Б О У верхньому рядку лівої таблиці записаний ключ, а номери під літерами ключа визначені відповідно до природного порядку літер ключа в алфавіті. У разі, якщо в ключі зустрічаються однакові букви, вони були б пронумеровані зліва направо. У правій таблиці стовпці переставлені за порядком номерів літер ключа. При зчитуванні вмісту правої таблиці по рядках отримуємо повідомлення: ЦОПВКЕТЕАУМАРРРОЦШОСЯЕАБО.

	Ключем для даного методу шифрування служать розміри таблиці та ключова фраза. Для розшифрування дії виконуються у зворотному порядку.
28.	Реалізувати шифр **маршрутної перестановки** (шифр "Сцитала"). Відкритий текст записується у прямокутну таблицю з n рядків і m стовпців. Вважається, що довжина тексту t дорівнює n $\cdot m$ (в іншому випадку залишкова частина тексту шифрується окремо з тим самим шифром). Якщо t<n·m, то решта пустих клітинок заповнюється довільним чином літерами з алфавіту. Шифротекст записується по цій таблиці за заздалегідь обумовленим "маршрутом" – шляхом, що проходить через усі клітинки таблиці. Ключем шифру є числа n, m та вказаний маршрут.
29.	Реалізувати шифрування **методом подвійної перестановки**. Даний метод схожий на метод з варіанта 6, але тут перестановки визначаються окремо для рядків і для стовпців. Спочатку у таблицю записується текст повідомлення, а потім по черзі переставляються стовпці, а потім рядки. При розшифруванні порядок перестановки повинен бути зворотним. Приклад виконання шифрування повідомлення “ЦЕ МОЯ КУРСОВА РОБОТА” методом подвійної перестановки наведений у таблицях: Початкова таблиця: | | 3 | 1 | 2 | |---|---|---|---| | 3 | ц | е | м | | 1 | о | я | к | | 6 | у | р | с | | 4 | о | в | а | | 2 | р | о | б | | 5 | о | т | а | Таблиця після перестановки стовпців: | | 1 | 2 | 3 | |---|---|---|---| | 3 | е | м | ц | | 1 | я | к | о | | 6 | р | с | у | | 4 | в | а | о | | 2 | о | б | р | | 5 | т | а | о | Після перестановки рядків: | | 1 | 2 | 3 | |---|---|---|---| | 1 | я | к | о | | 2 | о | б | р | | 3 | е | м | ц | | 4 | в | а | о | | 5 | т | а | о | | 6 | р | с | у | Зчитуючи шифротекст з правої таблиці по рядках, маємо: ЯКООБРЕМЦВАОТАОРСУ. Ключем до шифру подвійної перестановки служить послідовність номерів стовпців і номерів рядків початкової таблиці (у нашому прикладі 312 та 316425, відповідно).
30.	Реалізувати **шифрування за допомогою магічних квадратів**. Магічними квадратами називають квадратні таблиці з вписаними в їх клітинки послідовними натуральними числами, починаючи з 1, які дають в сумі по кожному стовпцю, кожному рядку і кожній діагоналі одне й те саме число. Повідомлення для шифрування вписується у магічні квадрати відповідно до нумерації їх клітинок. Якщо потім виписати вміст такої таблиці по рядках, то отримаємо шифротекст, сформований завдяки перестановці літер початкового повідомлення. В давні часи вважалось. що за допомогою магічних квадратів шифротекст береже не тільки ключ, але й магічна сила. Приклад магічного квадрата і його заповнення повідомлення “МОЯ КУРСОВА РОБОТА” наведений нижче: *Магічний квадрат:* | 16 | 3 | 2 | 13 | |---|---|---|---| | 5 | 10 | 11 | 8 | | 9 | 6 | 7 | 12 | | 4 | 15 | 14 | 1 | *Заповнення магічного квадрату:* | А | Я | О | Б | |---|---|---|---| | У | А | Р | О | | В | Р | С | О | | К | Т | О | М | Шифротекст, що отримуємо при зчитуванні вмісту правої таблиці по рядках, має цілком загадковий вигляд: АЯОБ УАРО ВРСО КТОМ.
31.	Реалізувати **систему шифрування Цезаря.** Шифр Цезаря є окремим випадком шифру простої заміни (одноалфавітна підстановка). Свою назву цей шифр отримав за іменем римського імператора Гая Юлія Цезаря, який використовував його при переписці з Цицероном (приблизно за 50 р. до н.е.). При шифруванні початкового тексту кожна літера замінювалась на іншу літеру цього ж самого алфавіту за таким правилом. Замінювальна літера визначалась шляхом зміщення за алфавітои початкової літери на К літер. При досягненні кінця алфавіту виконувався циклічний перехід на його початок. Цезар використовував шифр заміни при зміщенні на К=3. | А - Ю | Б - Я | В - А | Г - Б | Д - В | Є - Г | Ж - Д | З - Е | І - Ж | К - З | Л - И | М - К | Н - Л | О - М | П - Н | Р - О | С - П | Т - Р | У - С | Ф - Т | Х - У | Ц - Ф | Ч - Х | Ш - Ц | Щ - Ч | Ї - Ш | И - Щ | Ь - Ъ | Э - Ы | Ю - Ь | Я - Є |

Наприклад, якщо використовувати дану одноалфавітну підстановку повідомлення МОЯ ПЕРША КУРСОВА РОБОТА перетвориться у КМЭ НГОЦЮ ЗСОПМАЮ ОМЯМРЮ

32. Реалізувати **шифрування за допомогою блочних замін**, в якій шифрування відкритого тексту здійснюється блоками. Наприклад, блоку літер «АБА» може відповідати блок «РТК», а блоку літер «ВАЯ» – блок «АСС» і т. д.

33. Реалізувати **афінну систему підстановок Цезаря.** Сутність цієї системи полягає в тому, що літера, яка відповідає числу t, замінюється на літеру, що відповідає числовому значенню (at+b) за модулем m, де m – кількість літер алфавіту; a, b – цілі числа, причому a>=0, b<m, НОД(a, m) = 1. Нехай m=31, a=3, b=5. Тоді очевидно, що НОД(3, 31) = 1, і ми отримуємо таку відповідність між числовими кодами букв:

Початковий алфавіт	А	Б	В	Г	Д	Е	Ж	З	І	К	Л	М	Н	О	П	Р	С
t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3t+5	5	8	11	14	17	20	23	26	29	1	4	7	10	13	16	19	22
Алфавіт підстановки	Е	І	М	П	Т	Х	Ш	И	Ю	Б	Д	З	Л	О	С	Ф	Ч

Початковий алфавіт	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ї	И	Ь	Є	Ю	Я			
t	17	18	19	20	21	22	23	24	25	26	27	28	29	30			
3t+5	25	28	0	3	6	9	12	15	18	21	24	27	30	2			
Алфавіт підстановки	Ї	Є	А	Г	Ж	К	Н	Р	У	Ц	Щ	Ь	Я	В			

Якщо візьмемо для шифрування слово “ПРОГРАМУВАННЯ”, за допомогою даної системи підстановки отримаємо зашифроване повідомлення: “СФОПФЕЗЄЬЕЛЛВ”. Позитивною рисою афінної системи є зручне управління ключами – ключі шифрування і розшифрування подаються у компактній формі у вигляді пари (a, b).

34. Реалізувати **багатоалфавітне шифрування.** Багатоалфавітний шифр відноситься до шифрів складної заміни. Для шифрування кожного символу початкового повідомлення застосовують свій шифр простої заміни. Багатоалфавітна заміна послідовно і циклічно змінює використовувані алфавіти. При r-алфавітній підстановці символ x_0 початкового повідомлення заміняють символом y_0 з алфавіту B_0, символ x_1 – символом y_1 з алфавіту B_1, і т.д., символ x_{r-1} заміняють символом y_{r-1} з алфавіту B_{r-1}, символ x_r – символом y_r знову з алфавіту B_0. Наведемо загальну схему багатоалфавітної підстановки для випадку r=4.

Символи початкового повідомлення	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Алфавіт підстановки	B_0	B_1	B_2	B_3	B_0	B_1	B_2	B_3	B_0

Ефект використання багатоалфавітної підстановки полягає в тому, що забезпечується маскування природної статистики початкової мови, оскільки конкретний символ може бути перетворений в декілька різних символів шифрувальних алфавітів.

35. Реалізувати **одноразову систему шифрування.** Одноразова система шифрування винайдена у 1917 р. американцями Дж. Моборном та Г. Вернамом. Дана система шифрує початковий відкритий текст: $X = (X_0, X_1, \ldots, X_{n-1})$ у шифротекст: $Y = (Y_0, Y_1, \ldots, Y_{n-1})$ за допомогою підстановки Цезаря: $Y_i = (X_i + K_i) \bmod m, (0 \le i < n)$, де K_i – i-й елемент ключової послідовності, m – кількість літер алфавіту. Процедура розшифрування описується співвідношенням:

$$Y_i = (X_i - K_i) \bmod m, (0 \le i < n).$$

Для реалізації цієї системи підстановки іноді використовувався одноразовий блокнот. Цей блокнот складений з відривних сторінок, на кожній з яких надрукована таблиця з випад-

ковими ключами. Блокнот виконується у двох екземплярах: один використовується відправником, а другий – одержувачем. Для кожного символу повідомлення використовується свій ключ лише один раз. Після того, як таблиця використана, вона повинна бути видалена з блокнота і знищена. Шифрування нового повідомлення починається з нової сторінки.

36. Реалізувати шифрування за допомогою **шифруючої таблиці Трисемуса.** У 1508 р. абат з Германії Иоганн Трисемус написав друковану роботу з криптології, в якій вперше систематично описав застосування шифруючих таблиць, заповнених алфавітом у випадковому порядку. Для отримання такого шифру заміни зазвичай використовувалась таблиця для запису літер алфавіту і ключове слово (або фраза). У таблицю спочатку вписувалось по рядках ключове слово, причому літери, що повторювались, відкидалися. Потім ця таблиця заповнювалась літерами з алфавіту, які не увійшли у ключову фразу. Оскільки ключову фразу досить легко зберігати у пам'яті, то такий підхід спрощував процес шифрування. Наведемо приклад. Для російського алфавіту шифруюча таблиця може мати розмір 4×8. Оберемо ключем слово "БАНДЕРОЛЬ". Шифруюча таблиця з цим ключем буде такою:

Б	А	Н	Д	Е	Р	О	Л
Ь	В	Г	Ж	З	І	Й	К
М	П	С	Т	У	Ф	Х	Ц
Ч	Ш	Щ	И	Ї	Є	Ю	Я

При шифруванні знаходять в цій таблиці чергову літеру відкритого тексту і записують у шифротекст літеру, що розташована на один рядок нижче (або першу літеру цього стовпця, якщо літера розташована у нижньому рядку). Наприклад, при шифруванні повідомлення КУРСОВА РОБОТА отримуємо шифротекст ЦЇІЩЙПВ ІЙЬЙИВ. Такі табличні шифри є монограмними, оскільки шифрування виконується по одній літері.

37. Реалізувати **систему шифрування Цезаря з ключовим словом.** Ця система шифрування є одноалфавітною системою підстановки. Особливістю її є використання ключового слова для зміщення та зміни порядку символів в алфавіті підстановки. Виберемо деяке число k ($0 \le k \le 25$) і слово або коротку фразу як ключове слово. Бажано, щоб всі букви ключового слова були різними. Нехай вибрано слово "КУРСОВА" як ключове слово і число k=5. Ключове слово записується під буквами алфавіту, починаючи з букви, числовий код якої збігається з вибраним числом k, а решта літер алфавіту підстановки записуються після ключового слова в алфавітному порядку:

0	1	2	3	4	**5**					10					15					20
А	Б	В	Г	Д	Е	Ж	З	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Ы	Ь	Э	Ю	Я	**К**	**У**	**Р**	**С**	**О**	**В**	**А**	Б	Г	Д	Е	Ж	З	И	Л	М

21				25					30
Ц	Ч	Ш	Щ	Ї	И	Ь	Є	Ю	Я
Н	П	Т	Ф	Х	Ц	Ч	Ш	Щ	Ъ

Тепер ми маємо підстановку для кожної літери довільного повідомлення. Так, повідомлення Я ЗАХИСТИВ КУРСОВУ шифрується як ЇРИМСЖЗСЄОІЕЖГЄІ.

Слід зауважити, що вимоги щодо різних літер ключового слова не є обов'язковими. Можна записати ключове слово або фразу без повторення однакових літер.

38. Реалізувати застосування **біграмного шифру Плейфейра.** Основою шифру є шифротаблиця з випадково розташованими літерами алфавіту початкового тексту. Для зручності запам'ятовування шифротаблиці відправник і одержувач повідомлення можуть використовувати ключове слово або фразу при заповненні початкових рядків таблиці. В цілому структура шифруючої таблиці системи Плейфейра повністю аналогічна структурі шифруючої таблиці Трисемуса (див. 2.23). Для пояснення використаємо саме її. Процедура зашифрування включає в себе такі кроки:

	1. Відкритий текст повідомлення розбивається на пари літер (біграми). Текст повинен мати парну кількість літер і в ньому не повинно бути біграм, що містять однакові літери. Якщо ці вимоги не витримані, то текст модифікується, навіть припускаються незначні помилки. 2. Послідовність біграм відкритого тексту перетворюється за допомогою шифруючої таблиці в послідовність біграм шифротексту за такими правилами: - якщо обидві літери біграми відкритого тексту не попадають на один рядок або стовпець, тоді знаходять літери в кутках прямокутника, що визначається даною парою літер(наприклад, пара літер АЙ відображається в пару ОВ); - якщо обидві літери біграми відкритого тексту належать одному стовпці, то літерами шифротексту вважаються літери, що лежать під ними (наприклад, біграма НС відображається в ГЩ). Якщо при цьому літера відкритого тексту знаходиться в нижньому рядку, то для шифротексту береться відповідна буква з верхнього рядка одного й того ж самого стовпця (наприклад, ВШ відображається в ПА); - якщо обидві літери біграми належать одному рядку, то літерами шифротексту вважаються літерами, що лежать справа від них (наприклад, НО відображається ДЛ). Якщо при цьому літера знаходиться у крайньому правому стовпці, то для шифру беруть відповідну літеру з лівого стовпця того ж рядка (наприклад, ФЦ відображається в ХМ). Зашифруємо, наприклад, текст ВСЕ ТАЄМНЕ СТАНЕ ЯВНИМ. Розбиваємо його на біграми: ВС ЕТ АЄ МН ЕС ТА НЕ ТЯ ВН ИМ. Дана послідовність біграм відкритого тексту перетворюється у таку послідовність: ГП ДУ ОВ ДЛ НУ ПД ДР ЦИ ГА ЧТ. При розшифруванні застосовується зворотний порядок дій.
39.	Реалізувати **шифр “Атбаш”**. В Біблії є натяки на шифрування і дешифрування текстів. Сутність їх полягає у тому, що древні євреї використовували декілька систем шифрування за принципом простої заміни. Шифр «Атбаш» задавався таким чином: <table><tr><td>А</td><td>Б</td><td>В</td><td>… … …</td><td>Ь</td><td>Ю</td><td>Я</td></tr><tr><td>Я</td><td>Ю</td><td>Ь</td><td>… … …</td><td>В</td><td>Б</td><td>А</td></tr></table> Зауважимо, що в цьому шифрі заміна має симетричний вигляд: (А-Я, Я-А), (Б-Ю, Ю-Б) і т. д. Тому як і при шифруванні, так і при розшифруванні літери відкритого і шифрованого текстів беруться з одного й того самого верхнього рядка.
40.	Реалізувати **шифр “Альбам”**. Шифр “Альбам” полягає в розбитті алфавіту на дві частини і підписуванні однієї частини під другою: А Б В Г ... Н О П Р С Т У ... Ь Ю Я Тут заміна має, як і в шифрі “Атбаш”, симетричний характер: (А-Р, Р-А), (Б-С, С-Б), ..., (П-Я, Я-П) Як і при шифруванні, так і при розшифруванні літери відкритого і шифрованого текстів беруться з одного й того самого верхнього рядка.
41.	Реалізувати **систему шифрування Віжинера.** Система Віжинера (за іменем французького дипломата XVI ст. Блеза Віжинера) вперше була опублікована в 1586 році і є однією з найстаріших та найбільш відомих багатоалфавітних систем. Даний шифр багатоалфавітної заміни можна описати таблицею шифрування, яка носить назву таблиці (квадрата) Віжинера (Додаток А.1). Таблиця Віжинера використовується для шифрування і розшифрування. Таблиця має два входи: – верхній рядок підкреслених символів, що застосовується для зчитування чергової літери початкового відкритого тексту; – крайній лівий стовпець ключа. Послідовність ключів зазвичай отримують з числових значень літер ключового слова. При шифруванні початкового повідомлення його вписують в рядок, а під ним записують ключове слово або фразу. Якщо ключ виявився коротшим, ніж повідомлення, то його циклічно повторюють. В процесі шифрування знаходять у верхньому рядку таблиці чергову

букву початкового тексту і в лівому стовпці чергове значення ключа. Чергова літера шифротексту знаходиться на перетині стовпця, що визначається літерою, яка шифрується, і рядка, що відповідає значенню ключа.

Розглянемо приклад отримання шифротексту за допомогою таблиці Віжинера. Нехай ми вибрали ключове слово ОЦІНКА. Зашифруємо повідомлення МОЯ КУРСОВА РОБОТА. Випишемо наше повідомлення в рядок і запишемо під ним ключове слово з повторенням. В третій рядок будемо вписувати літери шифротексту за таблицею Віжинера.

Повідомлення	М	О	Я	К	У	Р	С	О	В	А	Р	О	Б	О	Т	А	
Ключ	О	Ц	І	Н	К	А	О	Ц	І	Н	К	А	О	Ц	І	Н	
Шифротекст	И	Є	Д	Ч	Ю	Р	А	Е	З	Н	І	Р	О	Ч	У	А	

42. Реалізувати шифр **“подвійний квадрат” Уітстона**. Шифр “подвійний квадрат” використовує відразу дві таблиці, розміщені по одній горизонталі, а шифрування здійснюється біграмами, як у шифрі Плейфейра (див. варіант 2.25). Нехай є дві таблиці з випадково розташованими в них алфавітами.

Ж	Щ	Н	Ю	Р
И	Т	Ь	Ц	Б
Я	М	Е	.	С
В	Ї	П	Ч	
:	Д	У	О	К
З	Є	Ф	Г	Ш
Х	А	,	Л	Ї

И	Ч	Г	Я	Т
,	Ж	Ь	М	О
З	Ю	Р	В	Щ
Ц	:	П	Е	Л
Ї	А	Н	.	Х
Є	К	С	Ш	Д
Б	Ф	У	Ї	

Перед шифруванням початкове повідомлення розбивають на біграми. Кожна біграма шифрується окремо. Першу літеру знаходять у лівій таблиці, а другу – у правій таблиці. Потім подумки будують прямокутник таким чином, щоб літери біграм належали його протилежним вершинам. Інші дві вершини цього прямокутника дають букви біграми шифротексту. Припустимо, що шифрується біграма ИЛ. Літера И знаходиться у стовпці 1 і у рядку 2 лівої таблиці. Літера Л знаходиться у стовпці 5 і у рядку 4 правої таблиці. Це означає, що прямокутник утворений рядками 2 і 4 та стовпцями 1 лівої таблиці і 5 правої таблиці. Отже, в біграму шифротексту входять літери О (стовпець 5 і рядок 2 правої таблиці) та літера В (стовпець 1 і рядок 4 лівої таблиці), і ми отримуємо біграму ОВ. Якщо обидві літери біграми повідомлення лежать в одному рядку, то і букви шифротексту беруть з цього ж самого рядка. Першу літеру біграми шифротексту беруть з лівої таблиці у стовпці, що відповідає другій літері біграми повідомлення. Друга літера біграми шифротексту береться з правої таблиці у стовпці, що відповідає першій літері біграми повідомлення. Тому біграма повідомлення ТО перетворюється у біграму шифротексту ЖБ.

Наприклад, зашифруємо повідомлення КУРСОВА РОБОТА. Розбиваємо повідомлення на біграми: КУ РС ОВ АР ОБ ОТ А. Отримуємо шифротекст: НІ ГШ .. УМ ЄЛ ХЮ ИД.

43. Реалізувати **шифр “квадрат Полібія”**. Квадрат Полібія – це винахід древніх греків (Полібій – грецький державний діяч, полководець, історик, III ст. до н.е.). Сутність цього шифрування відносно латинського алфавіту з 26 літер (вважалось, що I=J) полягала у тому, що у квадрат розміром 5×5 клітинок вписувались літери алфавіту:

	A	**D**	**C**	**D**	**E**
A	A	B	C	D	E
D	F	G	H	I	K
C	L	M	N	O	P
D	Q	R	S	T	U
E	Y	W	X	Y	Z

Літера, що шифрується, замінювалась на координати квадрату, в якому вона записана. Наприклад, B замінялась на AB, F на BA, R на DB і т.д. При розшифруванні кожна така пара визначала відповідну букву повідомлення. У даному випадку ключ відсутній, оскільки вико-

ристовується фіксований порядок літер. Все виходить занадто просто. Ускладнений варіант шифру Полібія полягає у запису літер алфавіту у довільному порядку. Цей довільний порядок і є ключем шифру. Але довільний порядок складно запам'ятати, користувачам шифру постійно приходиться тримати при собі ключ-квадрат. Як компроміс було запропоновано ключ–пароль. Пароль виписувався без повторів у квадрат, в решту клітинок вписувались в алфавітному порядку літери алфавіту, що не увійшли у парольну фразу. Такий квадрат вже немає необхідності тримати при собі. Досить лише запам'ятати пароль.

44. Реалізувати **шифр Чейза.** В середині XIX століття американець П. Е. Чейз запропонував таку модифікацію шифру Полібія. У прямокутник 3×10 вписуються літери алфавіту. Ключем шифру є порядок розташування літер у таблиці. Як і у шифрі Полібія з ключовим словом (див. варіант 2.20) порядок літер у таблиці можна зробити не зовсім випадковим (щоб не тримати при собі ключ-квадрат), а задати ключову фразу. Нехай. наприклад, ключем буде слово "ПРОГРАМА".

	1	2	3	4	5	6	7	8	9	0
1	П	Р	О	Г	А	М	Б	В	Д	Е
2	Ж	З	И	К	Л	Н	С	Т	У	Ф
3	Х	Ц	Ч	Ш	Щ	Ь	И,Й	Є	Ю	Я

Ключем (вже другим) шифру є порядок розташування літер у таблиці. При шифруванні координати літер виписуються вертикально. Наприклад, слово КУРСОВА прийме вигляд:

2 2 1 2 1 1 1
4 9 2 7 3 8 5

Чейз запропонував ввести третій ключ: заздалегідь обговорене правило перетворення нижнього (верхнього) ряду цифр. Наприклад, число, утворене цим рядом, множимо на 9:

4927385 × 9 = 44346465.

Отримуємо новий дворядковий запис:

2 2 1 2 1 1 1
4 4 3 4 6 4 6 5

Тепер цей дворядковий запис знову переводиться у літери згідно з таблицею; при цьому перше число (4) нижнього рядка визначає літеру першого рядка. Шифротекст набуває вигляду:

Г К И Г Н Г М А

Можуть бути використані і інші перетворення координат. Цей шифр сильніший за шифр Полібія, він вже не є шифром простої заміни. При розшифруванні отримана послідовність переводиться у дворядковий запис:

(1) 2 2 1 2 1 1 1
4 4 3 4 6 4 6 5

Нижній ряд ділиться на 9 (44346465 : 9 = 4927385), утворюється дворядковий запис і за ним згідно з таблицею читається відкритий текст.

45. Реалізувати **шифрування за допомогою таблиць Тритемія.** Реалізація таблиці Тритемія не потребувала використання якихось механічних застосувань. Таблиця складалася з рядків, кожний з яких являв собою літери алфавіту, зсунуті з кожним рядком на одиницю вліво (Додаток А.2). При шифруванні перша літера відкритого повідомлення шифрується по першому рядку (перший рядок є одночасно і рядком літер відкритого тексту), друга літера – по другому рядку і т.д. Після використання останнього рядка знову повертаються до першого. Так, наприклад, слово ПРОГРАМА відкритого повідомлення перетвориться у послідовність ПСРЖФЕТЗ шифротексту.

46. Реалізувати **шифр Белазо**. Даний спосіб шифрування винайдений італійцем Жованом Белазо. У 1553 р. виходить у світ його книжка "Шифр синьйора Белазо". В цьому шифрі ключем є так званий пароль – фраза або слово, що легко запам'ятовуються. Пароль записується

періодично над літерами відкритого тексту. Літера паролю, що стоїть над відповідною літерою відкритого тексту, вказує номер рядка у таблиці Тритемія, за якою слід проводити заміну (шифрування) цієї літери (літера відкритого тексту знаходиться у першому рядку таблиці). Наприклад, якщо взяти як пароль слово “ШИФР”, то при шифруванні слова “ПРОГРАМА” отримуємо:

Ключове слово	Ш	И	Ф	Р	Ш	И	Ф	Р
Текст повідомлення	П	Р	О	Г	Р	А	М	А
Шифротекст	И	Ш	Г	У	Й	И	Б	Р

47. Реалізувати **книжковий шрифт Енея**. Існує немало можливостей використовувати книжки для таємного обміну повідомленнями. Наприклад, якщо адресати заздалегідь домовились про використання дублікатів однієї і тієї ж книжки як ключа шифру, то їх таємні послання могли б складатися з таких елементарних одиниць: n/m/t, n – номер сторінки книги, m – номер рядка, t – номер літери в рядку. Так само і читається таємне послання. Ключем такого шифру є книга і використовувана в ній сторінка. Замість книг можуть бути використані окремі файли.

48. Реалізувати **шифр Порта.** Цей шифр являє собою прямокутну таблицю з літер алфавіту у такому порядку:

1	А	а	б	в	г	д	е	Ж	з	и	й	к	л	м	н	о	п
	Б	р	с	т	у	ф	х	Ц	ч	ш	щ	ї	і	ь	є	ю	я
2	В	а	б	в	г	д	е	Ж	з	и	й	к	л	м	н	о	п
	Г	с	т	у	ф	х	ц	Ч	ш	щ	ї	і	ь	є	ю	я	р
3	Д	а	б	в	г	д	е	Ж	з	и	й	к	л	м	н	о	п
	Е	т	у	ф	х	ц	ч	Ш	щ	ї	і	ь	є	ю	я	р	с
4	Ж	а	б	в	г	д	е	Ж	з	и	й	к	л	м	н	о	п
	З	у	ф	х	ц	ч	ш	Щ	ї	і	ь	є	ю	я	р	с	т
5	И	а	б	в	г	д	е	Ж	з	и	й	к	л	м	н	о	п
	Й	ф	х	ц	ч	ш	щ	Ї	і	ь	є	ю	я	р	с	т	у
6	К	а	б	в	г	д	е	Ж	з	и	й	к	л	м	н	о	п
	Л	х	ц	ч	ш	щ	ї	І	ь	є	ю	я	р	с	т	у	ф
7	М	а	б	в	г	д	е	Ж	з	и	й	к	л	м	н	о	п
	Н	ц	ч	ш	щ	ї	і	ь	є	ю	я	р	с	т	у	ф	х
8	О	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	П	ч	ш	щ	ї	і	ь	є	ю	я	р	с	т	у	ф	х	ц
9	Р	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	С	ш	щ	ї	і	ь	є	ю	я	р	с	т	у	ф	х	ц	ч
10	Т	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	У	щ	ї	і	ь	є	ю	я	р	с	т	у	ф	х	ц	ч	ш
11	Ф	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	Х	ї	і	ь	є	ю	я	р	с	т	у	ф	х	ц	ч	ш	щ
12	Ц	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	Ч	і	ь	є	ю	я	р	с	т	у	ф	х	ц	ч	ш	щ	ї
13	Ш	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	Щ	ь	є	ю	я	р	с	т	у	ф	х	ц	ч	ш	щ	ї	і
14	Ї	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	І	є	ю	я	р	с	т	у	ф	х	ц	ч	ш	щ	ї	і	ь
15	Ь	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	Є	ю	я	р	с	т	у	ф	х	ц	ч	ш	щ	ї	і	ь	є
16	Ю	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	Я	я	р	с	т	у	ф	х	ц	ч	ш	щ	ї	і	ь	є	ю

Шифрування здійснюється за допомогою секретного ключа, так званого лозунгу. Лозунг періодично виписується над відкритим текстом. За першою літерою лозунгу розшукується алфавіт (великі літери на початку рядка), у верхньому або нижньому півалфавіті відшукується перша літера відкритого тексту і заміняється відповідною літерою з верхнього або нижнього рядка. Аналогічно шифруються й інші літери (інтерваліи між словами не враховуються). Нариклад, зашифруємо за допомогою ключового слова «ШИФР» послідовність «КУРСОВА РОБОТА».

Ключова фраза	Ш	И	Ф	Р	Ш	И	Ф	Р	Ш	И	Ф	Р	Ш
Відкритий текст	К	У	Р	С	О	В	А	Р	О	Б	О	Т	А
Шифротекст	Ц	П	Ж	Й	І	Ц	Є	З	Е	Ф	И	Ц	Ж

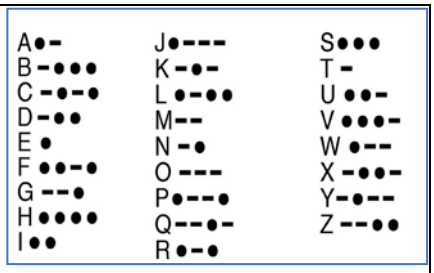
49. **Решето Ератосфена.** Розробити програму, яка дозволяє скласти “решето Ератосфена” на будь-якому інтервалі чисел.

50. Реалізувати **омофонну заміну.** Омофонна заміна аналогічна простій заміні, але має єдину відмінність: кожній букві відкритого тексту ставиться у відповідність декілька символів шифротексту. Наприклад, літера «А» заміняється на цифру 5, 13, 25 або 57; літера «Б» – на 7, 19, 12, 41 і т. д.

51. **Шифр Гронсфельда.** Це одна з модифікацій шифру Цезаря. Алгоритм шифру Гронсфельда (створений в 1734 році бельгійцем Хосе де Бронкхором, графом де Гронсфельда, військовим і дипломатом) полягає в тому, що величина зсуву не є постійною, а задається ключем (гамою). В якості гами можна взяти парольне слово.

52. **Шифрування Транспозицією (спосіб 2).** Реалізувати шифрування, виконавши таку транспозицію. У вхідному повідомленні береться кожна друга літера (символ) і її ASCII-код збільшується на *k*.

53. Реалізувати **шифр Енігми**. Енігма – це шифрувальна машина, яка використовувалася нацистами в часи ІІ Світової. Принцип її роботи такий: є декілька коліс і клавіатура. На екрані оператору показувалася буква, якою шифрувалася відповідна літера на клавіатурі. Те, якою буде зашифрована літера, залежало від початкової конфігурації коліс. Суть в тому, що існувало понад сто трильйонів можливих комбінацій коліс, і при наборі тексту колеса зсувалися самі, так що шифр змінювався протягом усього повідомлення. Всі Енігми були ідентичними, так що при однаковому початковому положенні коліс на двох різних машинах і текст виходив однаковий. У німецького командування були Енігми і список положень коліс на кожен день, тому вони могли з легкістю розшифровувати повідомлення один одного, але вороги, не повідомляючи положень, послання прочитати не могли. Коли Енігма потрапила в руки до союзників, вони все одно спершу не могли нічого з нею зробити, тому що не знали положень-ключів. Справу по злому шифру Енігми було розпочато в польській розвідці і доведено до кінця в британській за допомогою вчених і спеціальних машин (наприклад, Turing Bombe, чия робота полягала в тому, щоб моделювати одночасно роботу відразу декількох десятків Енігм). Відстеження комунікацій нацистів дало армії союзників важливу перевагу у війні, а машини, що використовувалися для злому, стали прообразом сучасних комп’ютерів.

54.	Реалізувати шифрування за допомогою **азбуки Морзе**. В азбуці Морзе кожна буква алфавіту, всі цифри і найбільш важливі знаки пунктуації мають свій код, що складається з низки коротких і довгих сигналів, що їх називають «точками і тире». Так, А – це «• -», В - «- •••», і т.д. На відміну від більшості шифрів, азбука Морзе використовується не для утруднення читання повідомлень, а навпаки, для полегшення їх передачі (за допомогою телеграфу). Довгі й короткі сигнали посилаються за допомогою включення і виключення електричного струму. Телеграф і азбука Морзе змінили світ, зробивши можливою блискавичну передачу інформації між різними країнами, а також сильно вплинули на стратегію ведення війни, адже тепер можна було здійснювати майже миттєву комунікацію між військами. 
55.	**Шифрування Транспозицією (спосіб 1).** У шифрах транспозиції букви переставляються за заздалегідь визначеним правилом. Наприклад, якщо кожне слово пишеться задом наперед, то з «*all the better to see you with*» виходить «lla eht retteb ot ees joy htiw». Інший приклад – міняти місцями кожні дві букви. Таким чином, попереднє повідомлення стане «*la tl eh eb tt re ot es ye uo iw ht*». Подібні шифри використовувалися в Першу Світову і Американську Громадянську Війну, щоб посилати важливі повідомлення. Реалізувати шифрування, виконавши таку транспозицію. У вхідному повідомленні кожне слово модифікувати, замінюючи у ньому порядок літер: перша міняється на останню, друга – на передостанню і т. д. При цьому пробіли і знаки пунктуації зміщуються на *k* позицій вперед. Наприклад, повідомлення «МОЯ КУРСОВА РОБОТА» після шифрування буде мати вигляд (при *k=2*): «ЯОМАВ ОСРУКАТ ОБОР».
56.	**Шифрування гамуванням.** У цьому способі шифрування виконується шляхом складання символів вихідного тексту і ключа за модулем, рівним числу букв в алфавіті. Якщо у вихідному алфавіті, наприклад, 33 символи, то складання проводиться за модулем 33. Такий процес складання початкового тексту і ключа називається в криптографії накладенням гами. Нехай символам вихідного алфавіту відповідають числа від 0 (А) до 32 (Я). Якщо позначити число, відповідне вихідному символу, *x*, а символу ключа - *k*, то можна записати правило гамування наступним чином: $z = x + k \ (mod\ N),$ де *z* – закодований символ, *N* – кількість символів в алфавіті, а складання за модулем N – операція, аналогічна звичайному додаванню, з тією відмінністю, що якщо звичайне підсумовування дає результат, більший або рівний N, то значенням суми вважається залишок від ділення його на N. Наприклад, нехай складемо за модулем 33 символи Г (3) і Ю (31): $3 + 31 \ (mod\ 33) = 1,$ тобто, в результаті отримуємо символ Б, що відповідає числу 1.
57.	**Реалізувати решітку Кардано** представляє собою лист з твердого матеріалу, в якому через неправильні інтервали зроблені прямокутні вирізи висотою одного рядка і різної довжини. Накладаючи цю решітку на лист паперу, можна було записувати в вирізи секретне повідомлення. Решту місць заповнювалися довільним текстом, що маскує секретне повідомлення. Цим методом маскування користувалися багато відомих історичних осіб, наприклад, кардинал Рішельє у Франції. На основі такої решітки Кардано побудував шифр перестановки.
58.	**Шифр перестановки.** При використанні шифрів перестановки вхідний потік вихідного тексту ділиться на блоки, в кожному з яких виконується перестановка символів. Перестановки в класичній "докомп'ютерної" криптографії виходили в результаті запису вихідного тексту і читання шифрованого тексту за різними шляхами геометричної фігури. Найпростішим прикладом перестановки є перестановка з фіксованим періодом d.

У цьому методі повідомлення ділиться на блоки по *d* символів, і в кожному блоці проводиться одна і та ж перестановка. Правило, за яким здійснюється перестановка, є ключем і може бути задано деякою перестановкою перших *d* натуральних чисел.

В результаті самі літери повідомлення не змінюються, але передаються в іншому порядку. Наприклад, для d = 6 в якості ключа перестановки можна взяти 436215. Це означає, що в кожному блоці з 6 символів четвертий символ стає на перше місце, третій – на друге, шостий – на третє і т.д.

59. **Шифрування перемішуванням.** У цьому методі шифрування *k-й* біт кожного байта вхідного повідомлення змінюється на 0 – якщо число, представлене даним байтом, парне, на 1 – якщо це число непарне, або цей біт вилучається з потоку і дописується в кінець зашифрованого повідомлення (або зберігається на окремому носії).

60. **Шифр пропорційної заміни.** Відомо, що при використанні шифру пропорційної заміни кожній літері поставлено у відповідність одне або декілька тризначних чисел по таблиці замін:

А	760	128	350	201		**С**	800	767	105	
Б	101					**Т**	759	135	214	
В	210	106				**У**	544			
Г	351					**Ф**	560			
Д	129					**Х**	768			
Е	761	130	802	352		**Ц**	545			
Ж	102					**Ч**	215			
З	753					**Ш**	103			
І	762	211	131			**Щ**	752			
К	754	764				**Ї**	561			
Л	132	354				**И**	136			
М	755	742				**Ь**	562			
Н	763	756	212			**Є**	750			
О	757	213	765	133	353	**Ю**	570			
П	743	766				**Я**	216	104		
Р	134	532				**Пробіл**	751	769	758	801	849

61. **Шифрування порозрядними операціями.** В такому шифруванні кожний байт вхідного повідомлення замінюється на байт, отриманий шляхом виконання порозрядної операції (>>> – циклічний зсув вправо, <<< – циклічний зсув вліво, ~ – операція інверсії). Потрібна операція обирається, виходячи з пароля.

Наприклад, можна взяти двійкове представлення пароля: 100010100110, а кодування операцій буде таке: 01_2 (1_{10}) – операція >>>, 10_2 (2_{10}) – операція <<<, 11_2 (3_{10}) – операція ~, 00_2 (0_{10}) – ніяка операція не виконується. Тоді порядок виконання операцій над символами у вхідному повідомленні буде такий: <<<, ,<<<, <<<, >>>, <<<. А далі обирається наступний блок байтів, і порозрядні операції виконуються далі.

62. **Шифрування перемішуванням символів.** Сутність даного методу така. Вхідне повідомлення ділиться на 2 (або 4, 6, 8…) частин. Повідомлення зчитується по байтах. Між першим і другим байтами першої частини повідомлення вставляється перший байт з другої частини, між другим і третім – другий байт з другої частини повідомлення і т.д. Наприклад, фраза «КУРСОВА РОБОТА» перетвориться на фразу «К УРРОСБООВТАА».

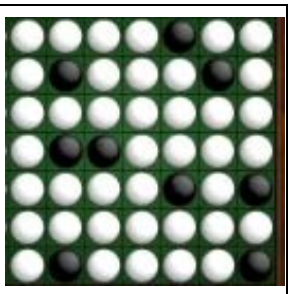
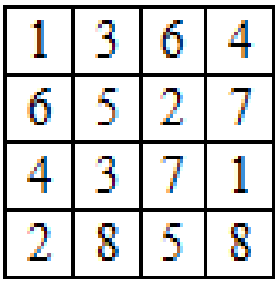
63. **Тарабарська грамота**, проста літорея – шифр, що широко використовувався в давньоруських рукописах. Являє собою найпростіший шифр заміни без ключа. Приголосні в алфавіті ділять на дві рівні частини, і першу пишуть рядком в алфавітному порядку, а другу – під буквами першої в зворотному порядку. Таким чином отримують таблицю:

б	**в**	**г**	**д**	**ж**	**з**	**к**	**л**	**м**	**н**
щ	**ш**	**ч**	**ц**	**х**	**ф**	**т**	**с**	**р**	**п**

Вживають в листі верхні літери замість нижніх і навпаки, а голосні залишаються без зміни. Так, наприклад, *словник* на пташиний грамоті буде *лсошамь*, *великий господар – шесітій чолноцам* і т. п. Для розшифрування використовують той самий спосіб, що і для шифрування (шифр симетричний).

64.	
65.	
66.	

Варіанти індивідуальних завдань на розробку програм ігрового характеру

67.	**Гра «Реверс».** Ігрове поле NxN складається з клітинок, кожна з яких забарвлена в один із двох кольорів – жовтий або синій. На початку гри клітинки фарбуються у випадковий спосіб. У процесі гри гравець може змінювати колір клітинок, натискаючи на них. При натисканні на клітинку одночасно змінюється колір усіх клітинок у її рядку та в її стовпці. Мета гри – пофарбувати всі клітинки в один колір.  Вікно повинно мати текстовий напис, що повідомляє, скільки клітинок кожного кольору наразі є на полі. Після кожного ходу програма повинна автоматично визначати, чи завершена гра. У файлі зберігається поточний стан гри. Для клітинок поля використайте кнопки з картинками. Реалізуйте можливість вибору розміру ігрового поля та режим ручного завдання кольорів клітинок на початку гри.
68.	**Гра на уважність.** На екрані – кнопки, кількість яких кратна двом (трьом, чотирьом, п'яти, …). На кнопках випадкові числа, серед яких обов'язково є пари (трійки, четвірки, …) однакових чисел. Сутність гри у тому, щоб якомога швидше відшукувати однакові числа і цим набирати бонуси. Рівень гри залежить від розміру матриці. Інформація про параметри гри (ім'я гравця, рівень, кількість бонусів і час гри) зберігається у файлі. У меню передбачити можливість виведення рейтингу гравців. Використати кнопки з цифрами; текстові поля – для введення імені ігрока; список – для вибору рівня гри.  1 3 6 4 6 5 2 7 4 3 7 1 2 8 5 8
69.	**Швидка арифметика.** Гравцеві пропонується два випадкових числа і пусте місце (або знак «?») для третього числа. Арифметична операція генерується випадковим чином. Завдання  3+2=5 полягає в тому, щоб ввести третє число, яке доповнить наведену рівність. Чим більше правильних відповідей, тим вище складність проходження тренувань (наприклад, інтервал значень операндів) . Інформація про параметри гри (ім'я гравця, рівень, кількість бонусів і час гри) зберігається у файлі. У меню передбачити можливість виведення рейтингу гравців.
70.	**Цифровий годинник.** На екрані розташовано цифровий годинник, час на якому повинен співпадати з реальним. В окремому вікні – панель для налаштування параметрів годинника (кольори, шрифт, розмір тощо). Доступ до панелі налаштування – лише після  16:53:38 авторизації. Передбачити можливість зупинки і запуску годинника, а також можливість встановлення будильника. Інформація про параметри налаштування годинника зберігається у файлі і застосовується при кожному новому сеансі роботи годинника.

71. **Механічний годинник**. Представлення механічного годинника зі стрілками (годинниковою, хвилинною, секундною), який покузує реальний час.

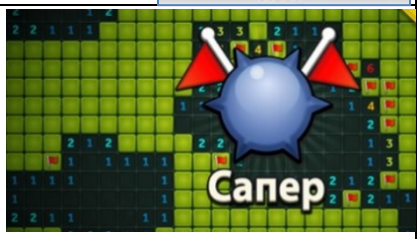
Параметри зображення: форма годинника (квадрат, коло, еліпс, …), розмір і кольори. Передбачити можливість зупинки і запуску годинника, а також можливість встановлення будильника. Вибрані параметри зберігати у файлі і застосовується при кожному новому сеансі роботи годинника.

72. **Годинник з маятником (або з зозулею)**. Представлення механічного годинника з маятником замість секундної стрілки, який показує реальний час, при цьому рух маятника кожну секунду супроводжується «тіканням».

Передбачити можливість зупинки і запуску годинника, а також можливість зміни розмірів та кольорів.

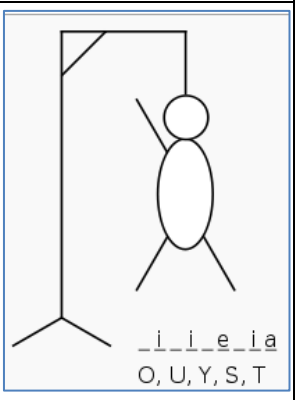
Інформація про параметри налаштування годинника зберігається у файлі і застосовується при кожному новому сеансі роботи годинника.

73. **Гра «Сапер».** Правила гри аналогічні до стандартної гри Windows. Використовуйте ігрове поле NxN. Вікно повинно мати текстовий напис, що повідомляє, скільки «мін» залишилося не знайденими. Після кожного ходу програма повинна автоматично визначати, чи завершена гра і чи виграв гравець. У файлі зберігати поточний стан гри і результати гравців.



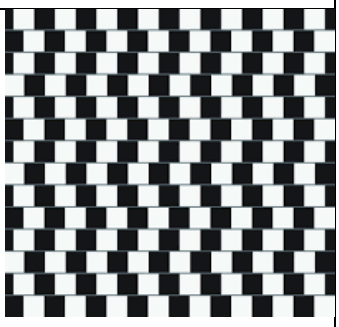
Для комірок використовувати кнопки з картинками. Реалізувати можливість вибору розміру ігрового поля і кількості «мін».

74. **Гра «Шибениця».** Програма загадує слово (підготувати заздалегідь базу слів у файлі з можливістю її поповнення) і виводить на екран першу та останню літери слова й позначає місця для інших букв, наприклад, рисками. Також на екран виводиться шибениця з петлею. Гравець пропонує літеру, яка може входити в це слово. Якщо така літера є в слові, то вона виводиться на відповідному місці – стільки разів, скільки вона зустрічається в слові. Якщо такої літери немає, то до шибениці домальовують коло в петлі, що зображає голову і т. д. Якщо тулуб у шибениці намальований повністю, то гравець вважається повішеним. Якщо гравцеві вдається вгадати слово, він виграє.



Інформація про параметри гри (ім’я гравця, загадане слово, кількість вгаданих і невгаданих літер) описується об’єктом типу клас і зберігається у файлі. У меню передбачити можливість виведення цієї інформації на екран.

75. **Ілюзія обману.** Відтворити картинку. За замовчуванням колір клітинок чорно-білий, а кількість клітинок (полосок) по горизонталі і вертикалі – 12. У подальшому кількість клітинок (полосок) і кольори можуть обиратися користувачем довільно.



Інформація про параметри фігури в останньому сеансі роботи програми (кількість полосок, кольори, величина зсуву) описується об’єктом типу клас і зберігається у файлі. При завантаженні програми ці налаштування зчитуються і фігура будується саме з цими параметрами.

76. **Гра «Знайди пару».** Матриця з картинок (серед яких всі картинки мають пару) з'являється на декілька секунд і зникає (перевертається). Далі гравець повинен згадати і вказати пари картинок. При цьому гравцю нараховують бонуси.

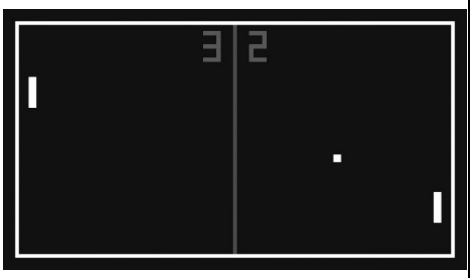
Інформація про параметри гри (ім'я гравця, рівень, кількість бонусів і час гри) зберігається у файлі. У меню передбачити можливість виведення рейтингу гравців.

77. **Імітація дзвінка по мобільному телефону**. Імітація дзвінків на мобільному телефоні: контакти, журнал здійснених дзвінків (кому, коли, тривалість розмови). При цьому відтворити звучання дзвінка.

Передбачити роботу з журналом контактів (додавання, видалення).

Журнал контактів і журнал дзвінків зберігаються в окремих файлах.

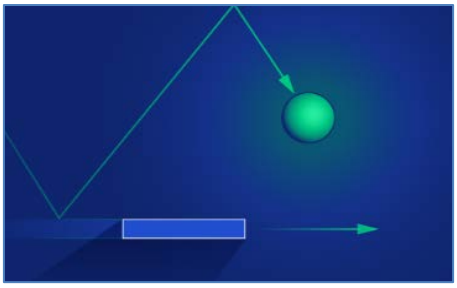
78. **Гра «Пінг-понг» (варіант 1)**. Ping-Pong є найпростішим спортивним симулятором настільного тенісу. Невеликий квадратик (або кружечок), що заміняє пінг-понговий м'ячик, рухається по екрану лінійною траєкторією. Якщо він наштовхується на периметр ігрового поля, його траєкторія змінюється залежно від кута зіткнення. Якщо кулька відбивається ракеткою гравця (за бажанням замість одного з гравців може бути комп'ютер), його рух додатково залежить від швидкості і напряму руху ракетки. Управління ракетками здійснюється за допомогою клавіш.



Ігровий процес полягає в тому, що гравці пересувають свої ракетки вертикально для захисту своїх воріт. На початку кожного раунду м'яч подається одному з гравців, і раунд продовжується доти, доки один із гравців не заробить очко. Це відбувається тоді, коли його супротивник не може відбити м'ячик. З часом гри швидкість руху м'ячика поступово збільшується, і так гра ускладнюється.

Інформація про параметри гри (ім'я гравця, рівень, кількість бонусів) зберігається у файлі. У меню передбачити можливість виведення рейтингу гравців.

79. **Гра «Пінг-понг» (варіант 2)**. Реалізувати спрощений варіант гри «Пінг-понг» - для одного гравця. Внизу (або вгорі) екрана рухається платформа, від якої відбивається м'ячик. Від платформи і від країв екрана м'ячик відбивається під кутом 45^0. Платформою керує гравець за допомогою клавіш або миші. Його задача – не дати м'ячику торкнутись нижнього (верхнього) краю екрана.



Інформація про параметри гри (ім'я гравця, рівень, кількість бонусів) зберігається у файлі. У меню передбачити можливість виведення рейтингу гравців.

80. **Плеєр**. Відтворити спрощений плеєр (вибір жанру, музичного файлу, програвання та інші елементи інтерфейсу, графічні ефекти – за бажанням).

У файлі запам'ятовувати перелік прослуханих файлів, які формуватимуть список відтворення.

81. **Гра на швидкість і концентрацію уваги**. На екрані кнопки, розташовані у комірках матриці (NxM), на яких нанесені цифри (1, 2, …, NxM). Кнопки розташовані у випадковому порядку. На початку гри гравцю надається певна кількість життів. Треба якомога швидше натискати кнопки у порядку зростання цифр. При помилковому натисканні кількість життів зменшується.

3	17	21	8	4
10	6	15	25	13
24	20	1	9	22
19	12	7	14	16
2	18	23	11	5

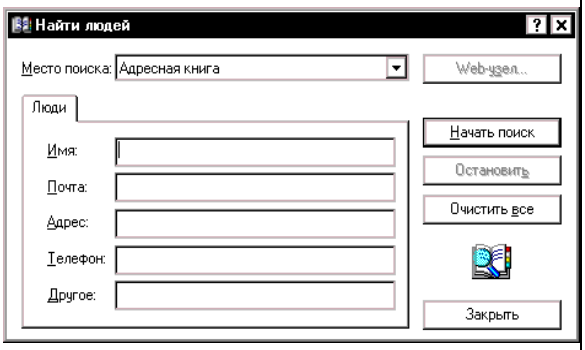
Інформація про параметри гри (ім'я, рівень, кількість життів і час гри) зберігається у файлі. У меню передбачити можливість виведення рейтингу гравців.

82. **Цифропад**. Відтворити «цифропад»: цифри «падають» з різною швидкістю (щось на кшталт «матриці»). При цьому змінювати колір цифр (кнопки селектора), запуск і зупинка руху (кнопки), зміна кольору фону (кнопка для виклику стандартної панелі кольорів). У файл записувати останні значення вибраних параметрів.

83. **Коливання тексту**. Відтворити текст, що коливається на хвилях (по синусоїді або косинусоїді). Текст – змінний, амплітуда – змінна, вид функції та швидкість коливання – змінні, колір – випадковий, що змінюється у часі.
Інформація про параметри зображення зберігається у файлі.

84. **Програма «Адресна книга».** У кожен момент часу вікно додатка має відображати один запис адресної книги (описаний окремим класом), який включає:

- ім'я людини (текстове поле);
- її телефон та e-mail (текстові поля);
- групу (список, що випадає), наприклад, «Друзі», «Знайомі»;
- прапорець «доданий до обраних контактів»;
- день народження (текстове поле).



У вікні мають бути кнопки «Наступний», «Попередній» для переходу між записами.

Має бути можливість редагування всіх полів даних. За кнопкою «Застосувати» усі зміни мають зберігатися у файлі. Реалізуйте можливість сортування записів за різними атрибутами.

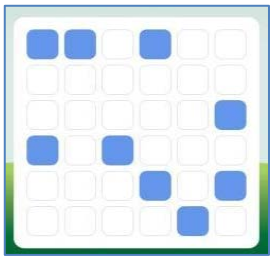
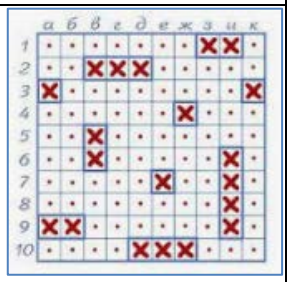
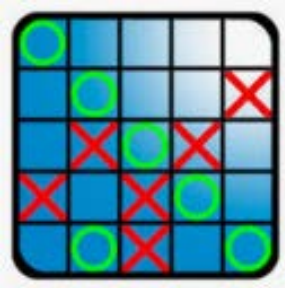
85. **Програма «Щоденник».** У кожен момент часу вікно додатку має відображати один запис щоденника (описаний окремим класом), який включає:

- назву події (текстове поле);
- докладний опис (багаторядкове текстове поле);
- дату, час (текстові поля);
- прапорець «важлива подія».

У вікні мають бути кнопки «Наступна», «Попередня» для переходу між подіями.

У файлі зберігати всі записи щоденника. Передбачити можливість редагування атрибути події, а зміни повинні зберігатися у файлі.

Реалізувати фільтр записів за датою (для пошуку задається діапазон дат), а також за довільною фразою в тексті опису події.

86.	**Матриці пам’яті.** На екрані – матриця, розбита на комірки, частина з яких зафарбовані деяким кольором. Через декілька секунд усі комірки зафарбовуються основним кольором. Гравцю потрібно запам’ятати розміщення зафарбованих клітин, а потім відтворити їх по пам’яті. Інформація про параметри гри (ім’я гравця, рівень, кількість бонусів і час гри) зберігається у файлі. У меню передбачити можливість виведення рейтингу гравців. 
87.	**Гра «П’ятнашки».** Головоломка є набором з 15 однакових квадратних кістяшок з нанесеними на них числами, що лежать у квадратній коробці. Довжина сторони коробки в чотири рази більша за довжину сторони кісточки, тому в коробці залишається незаповненим одне квадратне поле. Мета гри – упорядкувати кісточки за зростанням номерів, переміщуючи їх усередині коробки, бажано зробивши якнайменше переміщень.  У файлі зберігати ім’я гравця і час, витрачений на проходження гри, і кількість переміщень. Додатково можна передбачити можливість зміни розмірів ігрового поля і кольорів кістяшок.
88.	**Гра «Морський бій».** Користувач грає проти комп’ютера. Комп’ютер розміщує «кораблі» на полі у випадковий спосіб, а гравець повинен їх знайти.  Використовуйте ігрове поле NxN. Гравець виграє, якщо він знаходить усі «кораблі» противника, і програє, якщо не знаходить їх за 25 ходів. У вікні мають бути текстові написи, скільки «кораблів» наразі «вбито», «поранено» і скільки ще не знайдено. Після кожного ходу програма має автоматично визначати, чи завершена гра. Для комірок можна використовувати кнопки з картинками. У файл зберігати поточний стан гри та усі параметри налаштування. Додатково (не обов’язково) можна реалізувати режим початкової розстановки «кораблів» гравцями вручну.
89.	**Гра «Хрестики-нулики».** Ігрове поле NxN (N>4) складається з клітинок, у кожній із яких може стояти хрестик, нулик або порожньо. Гравці по черзі ставлять хрестики і нулики в порожні клітинки поля. Гра завершується, коли один з гравців має три хрестики або нулики в ряд. Вікно повинне мати текстовий напис, що повідомляє, який гравець повинен ходити. Після кожного ходу програма повинна автоматично визначати, чи завершена гра, і хто виграв.  Для клітинок поля використовуйте кнопки з картинками. У файл зберігати поточний стан гри. Додатково реалізуйте можливість вибору розміру ігрового поля, кольори хрестиків і нуліків і, за бажанням, можливість гри з комп’ютером.
90.	
91.	
92.	
93.	
94.	

Додаток А.1 – Шифрувальна таблиця Віжинера

Ключ	**А**	**Б**	**В**	**Г**	**Д**	**Е**	**Ж**	**З**	**И**	**І**	**К**	**Л**	**М**	**Н**	**О**	**П**	**Р**	**С**	**Т**	**У**	**Ф**	**Х**	**Ц**	**Ч**	**Ш**	**Щ**	**Ь**	**Ї**	**Э**	**Ю**	**Я**
0	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Э	Ю	Я
1	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Э	Ю	Я	А
2	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Э	Ю	Я	А	Б
3	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Э	Ю	Я	А	Б	В
4	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Э	Ю	Я	А	Б	В	Г
5	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д
6	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е
7	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж
8	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З
9	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И
10	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І
11	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К
12	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л
13	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М
14	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н
15	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О
16	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П
17	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р
18	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С
19	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т
20	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У
21	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф
22	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
23	Ч	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц
24	Ш	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
25	Щ	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
26	Ь	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
27	Ї	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь
28	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї
29	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є
30	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	**Т**	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ї	Є	Ю

Додаток А.2 – Шифрувальна таблиця Тритемія

А	**Б**	**В**	**Г**	**Д**	**Е**	**Ж**	**З**	**И**	**І**	**К**	**Л**	**М**	**Н**	**О**	**П**	**Р**	**С**	**Т**	**У**	**Ф**	**Х**	**Ц**	**Ч**	**Ш**	**Щ**	**Ь**	**Й**	**Є**	**Ю**	**Я**
Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А
В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б
Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В
Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г
Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д
Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е
З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж
И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З
І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И
К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І
Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К
М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л
Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М
О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н
П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П
С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р
Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С
У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т
Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У
Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф
Ц	Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Ч	Ш	Щ	Ь	Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц
Ш	Щ	Ь	Ы	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
Щ	Ь	Ы	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
Ь	Ы	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
Й	Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь
Є	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й
Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є
Я	А	Б	В	Г	Д	Е	Ж	З	И	І	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Й	Є	Ю

Додаток Б
Зразок оформлення титульного аркуша

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

КУРСОВА РОБОТА
з дисципліни «Програмування»
на тему:
«Розробка програмного застосунку для реалізації шифрування»

Виконав: студент II курсу групи 1 БС-23 б
спеціальності F5 Кібербезпека та захист інформації
Студентов С. С.

Керівник: ст. викл. кафедри ЗІ
Каплун В. А.

Члени комісії: Каплун В. А.
Радченко Є. В.
Баришев Ю. В.

Вінниця 2026

Додаток В
Зразок оформлення індивідуального завдання

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Рівень вищої освіти перший (бакалаврський)
Спеціальність F5 – Кібербезпека та захист інформації
Освітня програма «Безпека інформаційних і комунікаційних систем» (або «Етичний хакінг і кібербезпека»)

ЗАТВЕРДЖУЮ
Зав. кафедри ЗІ, д. т. н., проф.
_____ Володимир ЛУЖЕЦЬКИЙ
_____ _____ 2026 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу з дисципліни "Програмування"
студенту групи 1 БС-25 б Студентову С. С.

Тема: «Розробка програмного застосунку для реалізації шифрування»

Вхідні дані:
- об'єкт шифрування – повідомлення з клавіатури або текстові файли;
- метод шифрування – пряма перестановка символів;
- засоби для реалізації програми: мова – Java, Java FX; IntelliJ IDEA;
- інформаційні (текстові і графічні) матеріали до вебсторінки;
- засоби для реалізації вебсторінки: HTML, CSS, JavaScript;
- операційна система – сімейство Windows (Linux, IoS тощо).

Вихідні дані:
- програмний застосунок для зашифрування/розшифрування повідомлень;
- вебзастосунок.

Текстова частина: Вступ. Розробка і реалізація програмного застосунку. Розробка і реалізація вебзастосунку. Висновки. Список використаних джерел. Додатки.

Ілюстративна частина: Загальна схема роботи програмного засобу. Схеми алгоритмів зашифрування і розшифрування повідомлень. UML-діаграма класів проекту. Фрагменти інтерфейсу програми. Фрагменти вебсторінки.

Дата видачі: 7 вересня 2026 р.

Керівник _____ Валентина КАПЛУН

Завдання отримав _____ Сергій СТУДЕНТОВ

Додаток Г
Зразок оформленя анотацій

АНОТАЦІЯ

Студентов С. С. Розробка програмного застосунку для шифрування файлів: курсова робота з дисципліни «Програмування».

У курсовій роботі здійснено розробку десктопного застосунку для шифрування повідомлень методом транслітерації. Для досягнення результату використано …

Додатково розроблено вебсторінку, яка супроводжує програмний засіб і надає інформацію про …

ABSTRACT

Studentov S.S. Development of a software application for file encryption: course work on the discipline "Programing".

In the course work, a desktop application for encrypting messages using the transliteration method was developed. To achieve the result, the main capabilities of the …..

In addition, a web page was developed that accompanies the software tool and provides information about …

Додаток Д

Зразок оформлення змісту

ЗМІСТ

Додаток Ж

Приклад оформлення списку використаних джерел різного характеру

Посилання на книги (один автор, два автори, три автори):

1. Бородкіна І. Л. WEB-технології та WEB-дизайн: застосування мови HTML для створення електронних ресурсів. Київ : Ліра-К, 2022. 212 с.
2. Мамаєв М., Петренко С. Технологія захисту інформації в інтернеті : спеціальний довідник. Київ : Дніпро, 2022. 848 с.
3. Страуструп Б. Програмування: принципи і практика з використанням C++ : пер. з англ. 2-ге вид. Київ : Діалектика, 2020. 1328 с.

Посилання на книги (чотири автори і більше):

4. Клименко М. І., Панасенко Є. В., Стреляєв Ю. М., Ткаченко І. Г. Варіаційне числення та методи оптимізації : навч. посіб. Запоріжжя : ЗНУ, 2023. 84 с.
5. Операційне числення : навч. посіб. / С. М. Гребенюк та ін. Запоріжжя : ЗНУ, 2022. 88 с.

Посилання на журнали:

6. Єршов А. А., Петров Б. Б. Стабільні методи оцінювання параметрів. *Автоматика и телемеханіка*. 2023. № 8. С. 86-91.
7. Bletskan D. I., Glukhov K. E. Electronic structure of 2H-SnSe2: ab initio modeling and comparison with experiment. *Semiconductor Physics Quantum Electronics & Optoelectronics*. 2024. Vol. 19, No 1. P. 98-108.

Посилання на стандарти:

8. ДСТУ 7152:2010. Оформлення публікацій у журналах і збірниках. [Чинний від 2010-02-18]. Вид. офіц. Київ, 2010. 16 с. (Інформація та документація).
9. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання. [Чинний від 2017-07-01]. Вид. офіц. Київ : УкрНДНЦ, 2015. 26 с.

Посилання на електронні джерела:

10. Стандартні вбудовані об’єкти. URL: https://webdoky.org/uk/docs/Web/JavaScript/Reference/Global_Objects/#standartni-obiekty-za-katehoriiamy (дата звернення: 16.11.2025).
11. Сучасний підручник з Javascript. URL: https://uk.javascript.info/ (дата звернення: 12.01.2026).

Додаток К

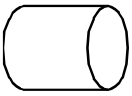
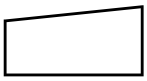
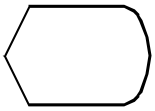
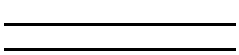
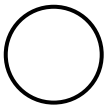
Зразок оформлення першої сторінки ілюстративної частини

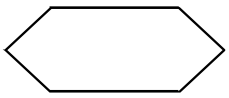
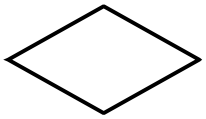
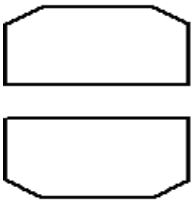
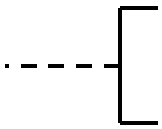
Додаток Х

Ілюстративна частина
до курсової роботи з дисципліни «Програмування»
на тему: «Розробка програмного застосунку для шифрування файлів»

Додаток Л

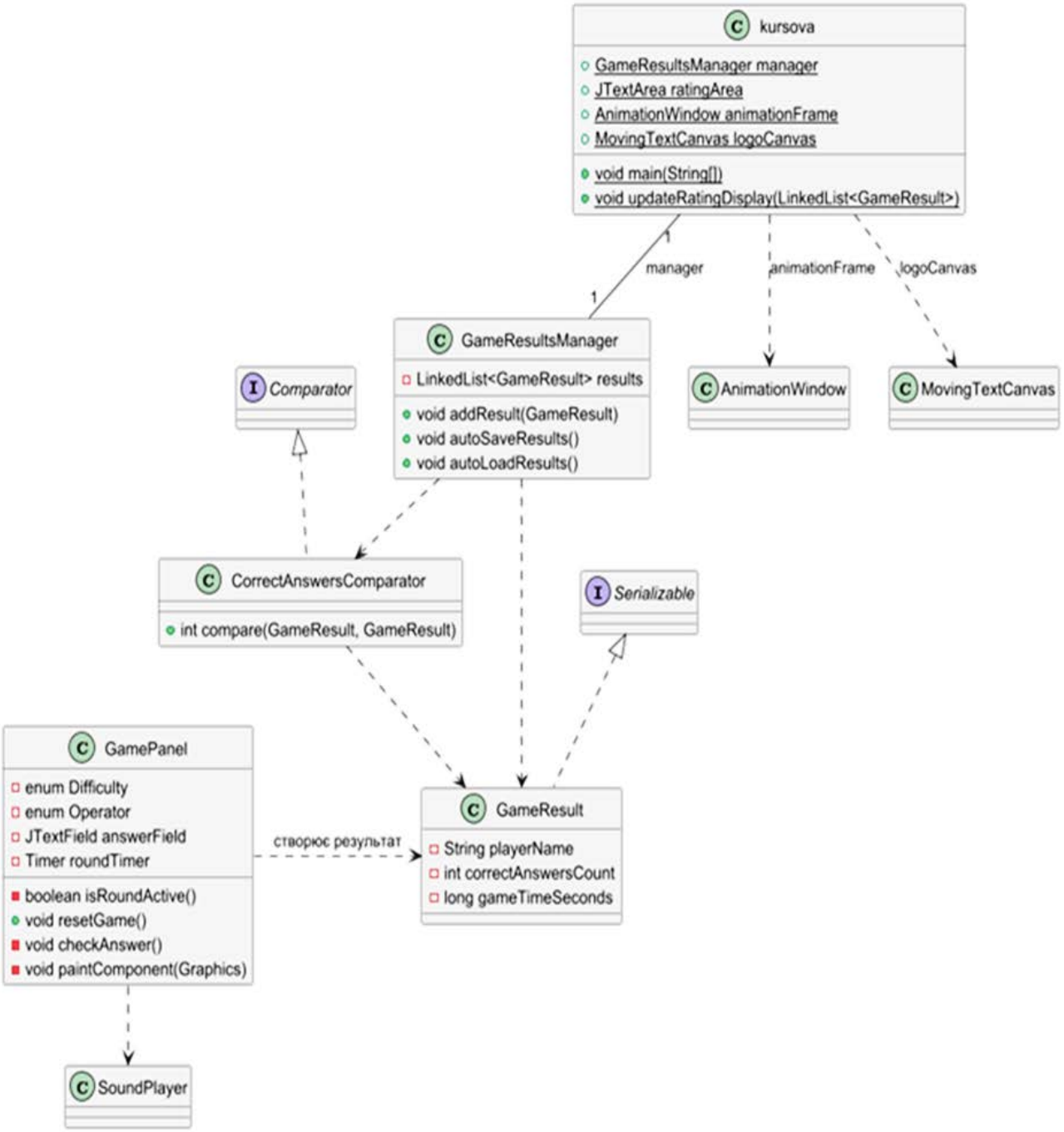
Символи даних, процесів і ліній

Термінатор		Символ відображає вихід в зовнішнє середовище і вхід із зовнішнього середовища (початок або кінець схеми програми, зовнішнє використовування і джерело або пункт призначення даних)
Дані		Символ відображає дані, носій даних невизначений
Оперативний запам'ятовувальний пристрій		Символ відображає дані, що зберігаються в оперативному запам'ятовувальному пристрої
Запам'ятовуючий пристрій прямого доступу		Символ відображає дані, що зберігаються в запам'ятовувальному пристрої з прямим доступом (наприклад, магнітний диск)
Документ		Символ відображає дані, подані на носії в легкій для читання формі (документ для оптичного або магнітного зчитування, мікрофільм, бланки даних)
Ручне введення		Символ відображає дані, що вводяться вручну під час оброблення з пристроїв будь-якого типу (клавіатура, перемикачі, кнопки, світлове перо тощо)
Дисплей		Символ відображає дані, подані у візуальній людино-читабельній формі на носії у вигляді пристрою відображення (екран, індикатори введення інформації)
Лінія		Символ відображає потік даних або управління. У разі необхідності або для підвищення легкості читання можуть бути додані стрілки-покажчики
Пунктирна лінія		Символ відображає альтернативний зв'язок між двома або більшою кількістю символів, а також використовується для обведення ділянки
Паралельні дії		Символ відображає синхронізацію двох або більше паралельних операцій
З'єднувач		Символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми і використовується для обривання лінії і продовження її у іншому місці. Відповідні символи-з'єднувачі повинні містити одне і те ж унікальне позначення
Процес		Символ відображає функцію обробки даних будь-якого вигляду (виконання певної операції або їх групи, що приводить до зміни значення, форми інформації).
Підпорядкований процес		Символ відображає підпорядкований процес, що складається з однієї або декількох операцій або кроків програми, які визначені у іншому місці

Підготовка для повторювань		Символ відображає модифікацію команди або групи команд з метою дії на деяку подальшу функцію (цикли з параметрами)
Умова або вибір		Символ відображає умову, вибір або функцію типу перемикача, що має один вхід і ряд виходів, і лише один з них може бути активований після обчислення умов усередині цього символу
Межі циклу		Символ, що складається з двох частин, відображає початок і кінець циклу. Обидві частини символу мають один і той самий ідентифікатор. Умови для ініціалізації, прирости, завершення поміщаються усередині символу на початку або в кінці залежно від розташування операції, що перевіряє умову
Коментар		Символ використовують для додавання описових коментарів або записів пояснень з метою пояснення або приміток. Пунктирні лінії в символі коментаря пов’язані з відповідним символом або можуть окреслювати групу символів. Текст коментарів або приміток повинен бути поміщений біля обмежуючої фігури

Додаток М

Зразок UML-діаграми класів



Електронне навчальне видання

Валентина Аполінаріївна Каплун
Євгеній Валентинович Радченко

Методичні вказівки до виконання курсових робіт з дисципліни «Програмування» зі спеціальності «Кібербезпека та захист інформації» (освітні програми «Безпека інформаційних і комунікаційних систем» та «Етичний хакінг і кібербезпека»)

Рукопис оформила В. Каплун

Редактор Н. Кравчук

Оригінал-макет виготовлено в РВВ ВНТУ

Підписано до видання 11.05.2026 р.
Гарнітура Times New Roman.
Зам. № 2026-053

Видавець та виготовлювач
Вінницький національний технічний університет,
Редакційно-видавничий відділ.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95, м. Вінниця, 21021.
press.vntu.edu.ua;
E-mail: rvv.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.