МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи з дисципліни

"ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ"

для студентів спеціальності 125 «Кібербезпека»
спеціалізації «Безпека інформаційних і комунікаційних систем»

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсової роботи з дисципліни

"ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ"

для студентів спеціальності 125 «Кібербезпека»
спеціалізації «Безпека інформаційних і комунікаційних систем»

Вінниця
ВНТУ
2018

Рекомендовано до друку Методичною радою Вінницького національного технічного університету Міністерства освіти і науки України (Протокол № 7 від 28.03.2018 р.)

Рецензенти:

Д. І. Катєльніков, кандидат технічних наук, доцент

Ю. В. Булига, кандидат технічних наук, доцент

Методичні вказівки до виконання курсової роботи з дисципліни «Технологія програмування» / Уклад. В. А. Каплун, А. В. Остапенко-Боженова, В. В. Лукічов. – Вінниця: ВНТУ, 2018. – 42 с.

Методичні вказівки призначені для надання допомоги при виконанні курсової роботи з дисципліни "Технологія програмування" і оформленні пояснювальної записки до неї. У роботі зроблено акцент на застосуванні об’єктно-орієнтованого програмування, як нової технології програмування, на необхідності використання API-функцій, знання яких має неабияке значення при захисті програмного забезпечення, та на тих базових конструкціях, які доцільно використовувати при реалізації завдання курсової роботи. Крім того, у методичних вказівках дано рекомендації щодо коректного оформлення пояснювальної записки до курсової роботи, наведено ряд прикладів подання текстової інформації та графічної частини.

Навчальне самостійне електронне мережне видання

Методичні вказівки
до виконання курсової роботи
з дисципліни «Технологія програмування»

Укладачі:
Каплун Валентина Аполінаріївна
Остапенко-Боженова Аліна Василівна
Лукічов Віталій Володимирович

Електронний ресурс PDF.
Підписано до видання 25.07.2018 р. Зам. № Р2018-013
Видавець та виготовлювач - Вінницький національний технічний університет,
Інформаційний редакційно-видавничий центр. ВНТУ, ГНК, к.114,
Хмельницьке шосе, 95, м. Вінниця, 21021,
тел. (0432) 65-18-06.
press.vntu.edu.ua;
Email: irvc.vntu@gmail.com.
Свідоцтво суб’єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

ЗМІСТ

1 ТЕМАТИКА ТА ЗМІСТ КУРСОВОЇ РОБОТИ

Курсова робота (КР) з дисципліни «Технологія програмування» – це самостійна робота, яка охоплює весь матеріал, викладений під час вивчення дисципліни «Технологія програмування», і містить елементи (задачі) навчального, аналітично-розрахункового та науково-дослідницького характеру.

В курсовій роботі з дисципліни «Технологія програмування» студент повинен показати знання мов програмування, розуміння основних концепцій нових технологій програмування, вміння самостійно розробити схему роботи програми в цілому та алгоритми складових поставленої задачі, підібрати засоби його програмної реалізації та подати розробку у вигляді, зручному для його використання сторонніми користувачами.

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

Під час виконання курсової роботи студенти повинні використати всі знання, отримані ними під час вивчення дисципліни “ Технологія програмування”: різноманітні види операторів, робота з файлами, робота з масивами, застосування принципів об’єктно-орієнтованого програмування, розробка графічного інтерфейсу, робота у візуальних середовищах програмування.

Зміст курсової роботи визначається завданням, яке видається на консультації викладачем кожному студенту. Завдання видається не пізніше 6 днів з початку семестру. Курсове проектування включає декілька послідовних етапів, які, в загальному випадку, пов'язані зі змістовною постановкою задачі, розробкою індивідуального технічного завдання, 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. 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 студента, його вміння самостійно здійснити попередній аналіз предметної області і розробити технічне завдання.

2 ВИМОГИ ДО РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ

Програма, яка є результатом виконання курсової роботи, повинна бути повноцінним додатком операційних систем Windows або Unix. Розроблена програма обов'язково повинна мати такі складові.

1. *Застосування основних принципів об'єктно-орієнтованого програмування(ООП)*: абстрактні типи даних, інкапсуляція, успадкування класів, поліморфізм. Студент повинен добре розуміти, з яких причин він використав у програмі той чи інший принцип ООП, довести доцільність його використання, вміти пояснити це під час захисту роботи.

   Дана вимога пояснюється тим, що саме технологія об'єктно-орієнтованого програмування є на даному етапі однією з найсучаснішою і найуживанішою, і знання основних її концепцій обумовлює розуміння побудови інших технологій програмування.
2. *Використання для реалізації курсової роботи будь-якої сучасної мови програмування*: C++, Java, C# або інших. При цьому у пояснювальній записці необхідно обґрунтувати вибір тієї або іншої мови і програмного середовища.
3. *Дотримання технології модульного програмування.* Бажано фрагменти коду, що мають певне самостійне значення, оформляти у вигляді процедур та функцій, з яких формуються відповідні файли заголовків. Структура головної програми та додаткових програмних модулів повинні бути зрозумілими, змістовними. Програми повинні бути читабельними (розташування операторних дужок, структурність вкладених операторів тощо) і мати відповідні коментарі, що пояснюють певні фрагменти коду. Це полегшить розуміння програми, продемонструє вміння студентів коректно і грамотно користуватись основними прийомами програмування, допоможе при захисті курсової роботи.
4. *Реалізація дружнього інтерфейсу*: використання багаторівневого меню, діалогових вікон, різноманітних елементів керування роботою програми, можлива графічна інтерпретація результатів, попередження про можливі помилки при введенні інформації, підказки під час інтерактивного режиму роботи і т.д. Меню програми обов'язково повинно містити пункти з інформацією про розробника програми, короткі теоретичні відомості (наприклад, пояснення принципу шифрування, формалізований опис алгоритму тощо), перегляд вихідного тексту програми (можливо з розділенням на підпункти, які відповідають окремим підпрограмам).
5. *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. Причому, якщо задача передбачає різні варіанти вхідних даних, для кожного з випадків необхідно підготувати свій набір

вхідних даних. Це повинно знайти своє відображення і при розробці математичної моделі, і при програмуванні, і при аналізі результатів.

Дана вимога при виконанні курсової роботи зумовлена тим, що робота з файлами є надзвичайно необхідною у будь-якій галузі програмування: при зберіганні і використанні інформації різноманітного характеру (числової, текстової, графічної тощо), при розробці і супроводженні баз даних, при передачі і отриманні повідомлень і т.д.

6. *Перевірка цілісності даних* на рівні перевірки правильності введеної інформації (числової, символьної, великі літери, малі літери, належність до алфавіту тощо), перевірка існування потрібних файлів і т.д. Ця вимога є необхідною, оскільки програми, в основному, створюються для пересічних користувачів, які не обізнані з тонкощами програмування і особливостями комп'ютерних систем. І тому при розробці програм необхідно враховувати можливість неправильного введення даних; допомогу при появі помилок та при відсутності необхідної для програми інформації.
7. *Подання інформації* (як вхідної, так і результуючої) повинно буди зрозумілим, мати необхідні пояснення. Всі результати вхідних, проміжних, результуючих дій повинні бути виведені на екран у вигляді, зручному для розуміння та аналізу стороннім користувачем, з поясненнями, допоміжними вікнами повідомлень.
8. *Розробка супроводжувальної документації* до програмної розробки. Необхідність виконання цієї вимоги пояснюється тим, що кожен товар (а програма також є товаром) повинен супроводжуватись детальним описом, що включає наведення його характеристик і параметрів, розробників, правила користування і т.д. Для курсової роботи супроводжувальна документація являє собою пояснювальну записку, яка має повністю описувати процес виконання задачі: побудову схем і алгоритмів, послідовність розробки процедур і функцій, їх опис та підключення до основної програми, перевірку правильності роботи, тестування на предмет ефективності роботи, інструкції для користування нею. Про правила оформлення пояснювальної записки йтиметься далі.

3 ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

3.1 Загальні правила оформлення

При оформленні пояснювальної записки необхідно дотримуватись вимог до оформлення курсових робіт (ДСТУ 3008:2015). Текст пояснювальної записки повинен бути набраний на комп'ютері та роздрукований на принтері.

Обсяг пояснювальної записки. Обсяг пояснювальної записки повинен бути в межах від 20 до 25 аркушів. При цьому аркуші додатків не враховуються. Разом з тим, нумерація аркушів додатків продовжує нумерацію сторінок основної частини пояснювальної записки.

Шрифт і відступи. Текст пояснювальної записки повинен бути набраний у будь-якому текстовому редакторі шрифтом TimesNewRoman розміром 14 з інтервалом між рядками від 1.15 до 1.5. Шрифт та міжрядковий інтервал у додатках можуть бути довільними, але такими, щоб можна було прочитати і зрозуміти. Відступи: зліва – 2.5 см, справа – 1 см, решта – 1.5 см.

Нумерація сторінок. Сторінки повинні бути пронумеровані, починаючи з третьої (зміст), у правому верхньому кутку сторінки. Нумерація додатків продовжує основну нумерацію.

Оформлення розділів і підрозділів. Структурними елементами основної частини ПЗ є розділи, підрозділи, пункти, підпункти, переліки.

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 заголовок. Заголовки структурних елементів необхідно нумерувати тільки арабськими числами.

Кожен розділ рекомендується починати з нової сторінки. Заголовок розділу записують посередині великими літерами з більш високою насиченістю.

Заголовки розділів, підрозділів, пунктів та підпунктів (при наявності заголовка) записують з абзацу малими літерами, починаючи з великої. Перед заголовком і після нього пропускають один рядок.

Розділи нумерують порядковими номерами в межах всього документа (1, 2, і т. д.). 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.

Допускається розміщувати текст між заголовками розділу і підрозділу, між заголовками підрозділу і пункту. Посилання в тексті на розділи виконується за формою: “...наведено в розділі 3”.

Оформлення таблиць. Таблицю розміщують симетрично до тексту після першого посилання на даній сторінці або на наступній, якщо на даній вона не уміщується 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. Підписи таблиць розташовуються над таблицею з вказанням її номера і назви, вирівнявши по лівому краю таблиці. Наприклад,

Таблиця 1.1 – Основні типи даних

Тип даних	Опис
1 Float	Дійсні числа з плаваючою точкою
2 Integer	Цілі числа
3 Double	Дійсні числа з плаваючою точкоюподвійної точності
… … … …	**… … …**

На всі таблиці мають бути посилання за формою “ ... в табл. 1 або в дужках по тексту (табл. 1.1). Посилання на раніше наведену таблицю дають зі скороченим словом ”дивись” (див. табл. 1.1) за ходом чи в кінці речення.

При перенесенні частин таблиці на інші сторінки, повторюють або продовжують найменування граф. 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
… … …	**… … …**

Оформлення рисунків. Розміщують рисунки в тексті або в додатках. В тексті ілюстрацію розміщують симетрично до тексту після першого посилання на неї або на наступній сторінці, якщо на даній вона не уміщується без повороту. На всі рисунки мають бути посилання за формою: “ ... на рис. 3.3–3.5”, або в дужках по тексту (рис. 3.6). Посилання на раніше наведений рисунок дають зі скороченим словом ”дивись” (див. рис.3.4) за ходом чи в кінці речення.

Між ілюстрацією і текстом пропускають один рядок.

Нумерують ілюстрації в межах розділів, вказуючи номер розділу і порядковий номер ілюстрації в розділі, розділяючи крапкою. Дозволяється нумерувати рисунки в межах всього документа.

Кожен рисунок повинен мати номер і підпис, розташовані під рисунком по центру. Крапку в кінці не ставлять, знак переносу не використовують. Якщо найменування рисунка довге, то його продовжують у наступному рядку, починаючи від найменування. Наприклад,

Рисунок 1.2 – Вигляд екрана при обробці пункту меню “Файл”

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 зразу ж 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 знака після нього.

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 граматики, 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.

3.2 Структура пояснювальної записки

Пояснювальна записка повинна відповідати індивідуальному завданню, а її оформлення – чинним державним стандартам, які слід враховувати на момент виконання розробки з врахуванням всіх офіційних змін, введених в дію.

Пояснювальна записка повинна мати таку структуру:

1. *Вступна частину,* яка містить:
   - титульний аркуш;
   - індивідуальне завдання;
   - анотацію;
   - зміст.

2. *Основна частина*, яка складається із:
   - вступу;
   - суті курсової роботи (технічної частини пояснювальної записки);
   - висновків;
   - список використаних джерел.
3. *Додатки*, які розміщуються після основної частини пояснювальної записки курсової роботи.

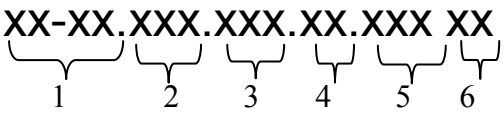
3.3 Вміст вступної частини пояснювальної записки

3.3.1 Титульний аркуш

Титульний аркуш є першою сторінкою КР, яка не нумерується. Згідно з діючим стандартом титульний аркуш виконується за встановленим зразком. Зразок титульного аркушу пропонується у додатку Б. Для курсової роботи титульний аркуш виконується без рамки.

На титульному аркуші для курсових робіт подаються: тема курсової роботи; запис „Пояснювальна записка ...” із зазначенням спеціальності, умовне позначення згідно з прийнятою системою (див. далі); перераховується науковий ступінь та звання керівника. Підписи керівника та студента із зазначенням термінів обов’язкові.

Для курсових робіт доцільною є предметна система умовних позначень, яка має таку структуру:



де 1 (XX-XX) – числовий шифр кафедри, прийнятий у ВДТУ (08-20);
2 (XXX) – умовне скорочення для дисципліни (ЗПЗ, ТП, ПБД і т. д.);
3 (XXX) – перша цифра 0, якщо це проект або 1, якщо робота, друга і третя цифри означають рік, наприклад, 18 – 2018 рік);
4 (XX) – варіант завдання на курсову роботу (наприклад, 01, 02, ...);
5 (XXX) – перший символ – номер групи (1 або 2), наступні два символи задають номер студента за списком у журналі академічної групи;
6 (XX) – код документа (ПЗ – пояснювальна записка).

Робота, яка подається у вигляді копії, до захисту не приймається.

3.3.2 Індивідуальне завдання

Конкретний зміст кожної КР, етапи виконання визначає керівник на підставі індивідуального завдання, затвердженого завідувачем кафедри і затвердженого на засіданні кафедри.

Попередньо керівник видає індивідуальне завдання до курсової роботи. Індивідуальне завдання в перелік змісту не вноситься і має бути

наступною сторінкою після титульного аркуша. Зразок індивідуального завдання до курсової роботи наведено в додаткуВ.Обов'язковим в індивідуальному завданні є наведення вхідних і вихідних даних, приблизний перелік частин пояснювальної записки і перелік документів ілюстративної частини.

Індивідуальне завдання до курсової роботи має містити термін видачі, підписи керівника та студента.

Завдання на курсову роботу повинно бути підготовлено студентом не пізніше другого тижня з початку навчального семестру, підписано викладачем, що видав завдання і студентом, що прийняв його до виконання.

3.3.3 Анотація

Анотація (двома мовами: АНОТАЦІЯ та ABSTRACT) 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).

Анотацію 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.

3.3.4 Зміст

Зміст розташовують безпосередньо після анотації, починаючи з нової сторінки. До змісту включають: вступ; послідовно перелічені назви всіх розділів, підрозділів, пунктів і підпунктів (якщо вони мають заголовки) суті роботи; висновки; перелік використаних джерел; назви додатків і номери сторінок, які містять початок матеріалу. Зміст не включає титульний лист, індивідуальне завдання на курсову роботу та анотацію. Нумерація у змісті починається зі ВСТУПУ (відповідно до нумерації у пояснювальній записці). Сам зміст за нумерацією пояснювальної записки є третьою або четвертою сторінкою в залежності від того, розміщені анотації на одній або на двох сторінках.

Сторінки додатків також входять до змісту. Нумерація сторінок повинна бути наскрізною, включаючи додатки.

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

Приклад оформлення змісту:

ВСТУП
1 РОЗРОБКА ...
1.1 Варіанти ...
1.1.1 ...
…….
2 ЗАГОЛОВОК ДРУГОГО РОЗДІЛУ ….
2.1 Заголовки підрозділів
2.1.1 ...
…..
3 ЗАГОЛОВОК ТРЕТЬОГО РОЗДІЛУ
3.1 Заголовки підрозділів
3.1.1 ...
…..
ВИСНОВКИ
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ
ДОДАТКИ
Додаток А. Назва першого додатка
Додаток Б. Назва другого додатка
…..

3.4 Вміст і оформлення основної частини

3.4.1 Вступ

Вступ пишуть з нової пронумерованої сторінки із заголовком ВСТУП посередині великими літерами з більш високою насиченістю (жирністю) шрифту.

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.

Вступ висвітлює:

- стан розвитку проблеми в даній галузі, до якої має відношення розробка (важливість нових технологій програмування, програмних середовищ, API-програмування, мов програмування тощо);
- галузь використання та призначення даної розробки;
- мету та загальну постановку задачі;
- актуальність, яка повинна подаватись в останньому абзаці вступу з метою стислого викладання суті обраної розробки.

Обсяг вступу не повинен перевищувати 1-2 сторінок.

3.4.2 Технічні розділи пояснювальної записки

Таких розділів може бути декілька. Це основні розділи пояснювальної записки, в яких викладено весь процес розробки програмного забезпечення. В даних розділах повинна бути детально описана мета роботи, алгоритми досягнення результату, передбачена розробка загальної схеми функціонування програми.

В цих розділах формуються і обґрунтовуються основні вимоги до технічних і програмних показників розробки, які надалі у стислому вигляді будуть подані у вигляді інструкцій, вимоги до характеристик програми, яка розробляється для розв'язання поставленої задачі. Тут наводиться математична модель (якщо необхідно), проводяться дослідження використовуваних методів, аналіз роботи методів і алгоритмів на простих прикладах. Результатом такого аналізу повинні бути розроблені блок-схеми алгоритмів, схеми взаємозв’язків між підпрограмами.

Обов’язковим підрозділом пояснювальної записки повинно бути обґрунтування вибору програмних засобів для реалізації: мови програмування, технології програмування, програмного середовища.

Підготовка інтерфейсу програми, що розробляється, також повинна знайти відображення у даному розділі, які види вікон треба використати і чому, які пункти меню доцільно включити і для чого, які елементи керування слід передбачити і для чого, – все це повинно знайти відображення в технічній частині пояснювальної записки. Кожна програма повинна мати певні вхідні і вихідні дані, як вони будуть готуватись, звідки вводитись у програму, де зберігатись, що буде вихідними даними і у якому вигляді – це також повинно бути описано і подано у вигляді схеми даних.

Отже, тут викладається послідовність основних кроків, необхідних для створення конкретного програмного продукту. Якщо використовуються якісь особливі методи або логічні розв’язування поставленої задачі, їх теж потрібно описати. Таким чином, даний розділ повинен бути підготовкою для наступного етапу – етапу програмування.

Пропонується приблизно такий перелік розділів:

- розробка і реалізація основних алгоритмів і функцій;
- розробка і реалізація інтерфейсної частини програмного засобу;
- тестування розробленого засобу.

3.4.3 Опис програмної реалізації задачі

Необхідно дати обґрунтування того, чому ті або інші засоби програмування доцільно використати для реалізації певних цілей завдання. Проводиться покроковий опис програмної реалізації алгоритму поставленої задачі, додержуючись принципів структурного програмування. Описуються основні структури мови програмування, які використані в даній роботі. Наводяться фрагменти (не більше в 5-10 операторів, а не програма цілком!). Описуються всі основні змінні, а також допоміжні

масиви, якщо вони використовуються при розв'язанні задачі. Всі допоміжні модулі, класи, процедури і функції, основні і заголовні файли, які використовуються при розробці програмного забезпечення, також повинні бути описані коротко, якщо вони є стандартними, і детально, якщо вони є продуктом роботи студента.

Фрагменти коду, наведені у пояснювальній записці, треба наводити іншим шрифтом (наприклад, Courier New) і розміром 10-12 пт.

Якщо розроблено власні підпрограми або функції, вони повинні бути описані і дано посилання на ті додатки, де знаходиться лістинг програми. Якщо використовуються стандартні функції, навести їх доцільність та мету використання і внутрішнє наповнення цих функцій (також з посиланням на відповідні додатки).

Даний розділ також має бути дуже змістовним, конкретним, зрозумілим, оскільки саме він демонструє знання та навички у галузі програмування.

3.4.4 Тестування програми і розробка інструкцій

Цей розділ повинен бути присвячений тестуванню розробленої програми для формування інструкцій та рекомендацій для роботи з нею. В ньому необхідно продемонструвати весь хід виконання програми на всіх режимах її роботи. Для цього слід підготувати різні комплекти вхідних даних, правильні і такі, що призводять до помилок. При здійсненні і описі аналізу роботи програми можна наводити ілюстрації (вигляд екрана), що демонструють основні режими і можливості функціонування програми. Ілюстрації можна виносити у додатки, а в тексті пояснювальної записки посилатись на ці додатки.

3.4.5 Використання схем і діаграм

Окремими підпунктами основного розділу або завершенням певних підрозділів бажано розробити і описати різні схеми (додаток Г):

- схеми даних;
- схеми програм;
- схеми роботи системи;
- схеми взаємодії програм;
- схеми ресурсів системи.

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

Схеми можуть бути виконані у вигляді UML-діаграм, серед яких можуть бути такі (додаток Є):

- діаграма варіантів використання;
- діаграма класів;
- діаграма станів;
- діаграма діяльності;

– діаграма послідовності;
– діаграма компонентів тощо.

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.

3.4.6 Розробка інструкцій по роботі з програмою

Оскільки темою даної курсової роботи є розробка програмного продукту, то необхідно передбачити розробку конкретних інструктивних матеріалів для роботи з програмою. Інструкції необхідно винести у додаток, який може мати назву: “Інструкції для роботи з програмою ...”, або “Інструктивні документи з ...”, або “Рекомендації для ...”.

Таким чином, необхідно передбачити розробку рекомендацій для роботи з програмою: інструкцію програмісту, інструкцію системному програмісту, інструкцію оператору (користувачу), інструкцію з технічного обслуговування.

Інструкція з технічного обслуговування. Інструкція з технічного обслуговування повинна містити такі підпункти:

- *вступ* (призначення інструкцій, перелік експлуатаційних документів, якими повинні додатково до інструкцій користуватися при технічному обслуговуванні і експлуатації);
- *загальні вказівки* (порядок технічного обслуговування, вказівки щодо організації і особливостей його проведення);
- *вимоги до технічних засобів* (вказують мінімальний склад технічних засобів, що забезпечують роботу програми);
- *опис функцій* (максимальний перелік функцій, що здійснюються цією програмою; опис сумісного функціонування технічних засобів і програми з вказанням методу обробки помилок; опис організації вхідних і вихідних даних для перевірки роботоздатності; опис взаємодій пристроїв з програмою, результатів взаємодій, висновки із результатів роботи програми).

Інструкція системного програміста. Інструкція системного програміста повинна відповідати ГОСТ 19.503-79 і містити наступні підпункти:

- *загальні відомості про програму* (призначення і функції програми і зведення про технічні і програмні засоби, що забезпечують виконання даної програми);
- *структура програми* (відомості про структуру програми, її складові частини, про зв'язки між складовими частинами і про зв'язки з іншими програмами);
- *настроювання програми* (опис дій для налаштування програми на

умови конкретного застосування – налагодження на склад технічних засобів, вибір функцій і ін. При необхідності наводять пояснювальні приклади.);

- *перевірка програми* (опис способів перевірки, що дозволяють дати загальний висновок про роботоздатність програми – контрольні приклади, методи прогону, результати);
- *додаткові можливості* (опис можливостей програми і способів їх вибору);
- *повідомлення системному програмісту* (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).

Інструкція програміста. Інструкція програміста повинне містити такі підрозділи (відповідно до ГОСТ 19.504-79):

- *призначення і умови застосування програми* (тут слід вказати призначення і функції, виконувані програмою, умови, необхідні для виконання програми – об'єм оперативної пам'яті, вимоги до складу і параметрів пристроїв, вимоги до програмного забезпечення і т. п.);
- *характеристика програми* (опис основних характеристик і особливостей програми – тимчасові характеристики, режими роботи, засоби контролю правильності виконання і самовідновлення програми і т. п.);
- *звернення до програми* (опис процедур виклику програми (способи передачі управління, параметрів і ін.);
- *вхідні і вихідні дані* (опис організації використовуваної вхідної і вихідної інформації і, при необхідності, її кодування);
- *повідомлення* (тексти повідомлень, видаваних програмісту або оператору в ході виконання програми, опис їх змісту і дій, які слід виконати після аналізу цих повідомлень).

Інструкція оператора. Інструкція оператора повинне містити такі підрозділи (відповідно до ГОСТ 19.505-79):

- *призначення програми* (відомості про призначення програми і інформація, достатня для розуміння функцій програми і її експлуатації);
- *умови виконання програми* (умови, необхідні для роботи програми: мінімальний і/або максимальний склад апаратних і програмних засобів і т. п.);
- *виконання програми* (послідовність дій оператора, 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 оператору* (тексти повідомлень в ході виконання програми, опис їх змісту і відповідні дії оператора: у разі збою, можливості повторного запуску програми і т. п.).

3.4.7 Висновки

Висновки оформляють з нової пронумерованої сторінки великими літерами більш високої насиченості.

У висновках наводяться основні результати роботи над курсовою роботою. На основі проведених досліджень результатів 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.

3.4.8 Оформлення переліку використаних джерел

Список містить перелік літературних джерел, на які повинні бути обов'язкові посилання в тексті пояснювальної записки. Література (книги, статті, патенти, журнали) в загальний список записується в порядку посилання на неї в тексті. В даному переліку дається оформлений відповідно до вимог державних стандартів, список тих джерел (книги, підручники, журнали, електронні адреси), які було використано в процесі виконання роботи, і на яку є посилання в тексті пояснювальної записки. Кожне джерело повинно бути вказано разом з видавництвом, роком видання, кількістю сторінок. Літературу записують мовою оригіналу, записують з абзацу і нумерують арабськими цифрами, починаючи з одиниці.

По ходу викладення матеріалу у пояснювальній записці повинні бути посилання на джерела, які наводяться в кінці відповідного речення або частини речення (не у довільному місці!) у квадратних дужках (після пробілу, перед крапкою чи комою).

Приклад оформлення переліку використаних джерел різного характеру.

Приклад посилання на книги:

1. Мамаев М., Технология защиты информации в интернете : [Специальный справочник] / Максим Мамаев, Сергей Петренко. - СПб. : Питер, 2002. – 848 с. – ISBN 5-318-00244-7.
2. Лужецький В. А. Інформаційна безпека : навчальний посібник / В. А. Лужецький, О. П. Войтович, А. В. Дудатьєв – Вінниця : УНІВЕРСУМ-Вінниця, 2009. – 240 с. – ISBN 978-966-641-265-5

Приклад посилання на ГОСТ і ДСТУ:

3. Единая система конструкторской документации. Общие требования к текстовым документам : ГОСТ 2.105-95. – [Чинний від 1996—01—07]. – Мінськ: Межгосударственный совет по стандартизации, метрологии и сертификации, 1996. – 29 с. – (Міждержавний стандарт).

4. Захист інформації. Технічний захист інформації. Основні поняття : ДСТУ 3396.0-96. – [Чинний від 1997—01—01]. – К. : Держспоживстандарт України, 1996. – 20 с. – (Національні стандарти України).

Приклад посилання на патенти:

5. Адаптивний метод ущільнення даних : патент 14709 : МКЗ 07-07 / Горін О. М., Волощенко О. В., Чуріп О. О. ; власник патенту Вінницький національний технічний університет. — № 200601173 ; заявл. 26.07.06 ; опубл. 10.08.07, Бюл. № 12 (кн. 2). — 2 с. : іл.

Приклад посилання на web-сторінки:

6. Windows Vista [Електронний ресурс]. – Режим доступу : URL : http://ru. wikipedia. org/wiki/Windows_Vista- Назва з екрану.
7. Новые технологии написания вирусов [Електронний ресурс]. – Режим доступу : URL : http://bezpeka.com/ru/news/2008/10/30/virus-injection.html - Назва з екрану.

3.5 Оформлення додатків

Додатки повинні містити матеріал, який не увійшов в основні розділи пояснювальної записки: лістинги програм, підпрограм та функцій, результати тестування програми у вигляді образів екранів, таблиць, графіків, схеми роботи програм, схеми алгоритмів, схеми ресурсів і даних, схеми взаємодії програм тощо.

Кожен додаток необхідно починати з нової сторінки, вказуючи зверху посередині рядка слово “Додаток” і через пропуск – його позначення. Додатки позначають послідовно великими українськими літерами, за винятком літер Ґ, Є, З, І, Ї, Й, О, Ч, Ь, наприклад, Додаток А, Додаток Б.

Кожен додаток повинен мати тематичний (змістовний) заголовок, який записують посередині рядка малими літерами з першої великої.

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.

Приклад оформлення додатків можна переглянути у додатках до даних методичних вказівок.

3.6 Оформлення ілюстративної частини

Ілюстративна частина призначена для того, щоб допомогти студенту вдало захистити свою роботу. В ілюстративний матеріал слід виносити усі ті матеріали, які допоможуть представити етапи розробки курсової роботи, основні схеми, алгоритми, математичну модель, фрагменти інтерфейсу розробленої програми.

Перед ілюстративною частиною у пояснювальну записку вставляється окремий аркуш, в центрі якого розташовується надпис «Ілюстративна частина», а далі слідують безпосередньо аркуші ілюстративного матеріалу. Кожний з них повинен бути оформлений у вигляді плакату: підпис матеріалу повинен бути вгорі по центру, далі по центру сутність плакату.

Аркуші ілюстративної частини не нумеруються і не вносяться у зміст пояснювальної записки.

4 ГРАФІК ВИКОНАННЯ КУРСОВОЇ РОБОТИ І ПОРЯДОК ЇЇ ЗАХИСТУ

4.1 Рекомендований графік виконання курсової роботи

Рекомендується такий графік виконання курсової роботи, який враховує самостійну роботу студентів під час 4-го триместру (14-16 тижнів).

Зміст розділу	Термін виконання
Отримання завдання на курсову роботу, розробка і оформлення індивідуального завдання	1-2 тижні
Розробка структури програмного забезпечення: 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 тижні
Розробка програмного забезпечення 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 тижні
Тестування розробленого програмного продукту та виправлення виявлених недоліків. Підготовка контрольних прикладів.	11 тиждень
Оформлення пояснювальної записки до курсової роботи, розробка рекомендацій для роботи з розробленою програмою	12-13 тижні
Здача курсової роботи на попередню перевірку: демонстрація роботи програми та чернетки пояснювальної записки (бажано тверда копія, але можливий електронний варіант)	13 тиждень
Корегування і доповнення (при необхідності) програми згідно із зауваженнями керівника курсової роботи, врахування і виправлення пояснювальної записки	13-14 тижні
Захист курсової роботи	13-14 тижні

Готовність до захисту курсової роботи визначає керівник за результатами попередньої перевірки якості пояснювальної записки та роботоздатності програми (згідно із графіком попереднього захисту). Записка повинна бути здана керівнику на перевірку не менше, як за тиждень до визначеного терміну захисту роботи. Якщо робота виконана в повному обсязі і не має принципових помилок, керівник допускає студента до захисту. В іншому випадку робота повертається студенту на доопрацювання протягом вказаного терміну.

4.2 Оцінювання виконання курсової роботи

Після позитивного висновку про готовність курсової роботи студент повинен захистити її перед комісією у складі двох викладачів, які призначені кафедрою.

Курсова робота оцінюються за 100-бальною шкалою:

Сума балів за всі види за КП	Оцінка ECTS	Оцінка за національною шкалою
90 - 100	A	відмінно
82 – 89	B	добре
74 – 81	C	
64 – 73	D	задовільно
60 – 63	E	
35 – 59	FX	незадовільно з можливістю повторного складання
0 – 34	F	незадовільно з обов'язковим повторним вивченням дисципліни

Розподіл бальної оцінки за виконання курсової роботи:

Пояснювальна записка	Практична реалізація	Ілюстративна частина	Інструкції	Захист роботи	Сума балів
до 25	до 40	до 10	10	до 15	100

Пояснювальна записка (ПЗ) оцінюється за такими параметрами:

- дотримання вимог щодо оформлення ПЗ згідно ДСТУ 3008:2015;
- відповідність ПЗ і програми індивідуальному завданню;
- відповідність ПЗ програмному забезпеченню;
- логічна пов’язаність розділів пояснювальної записки.

Програмна частина оцінюється за такими параметрами:

- виконання основної задачі курсової роботи;
- зрозумілість і зручність інтерфейсу програми;
- якість написання коду програми (наявність коментарів, читабельність, структура програми, використання функцій тощо);
- передбачення системи допомоги, обробки виключних ситуацій;
- наявність у додатках інструкцій по роботі з програмою і їх відповідність програмному забезпеченню.

Захист роботи оцінюється за такими параметрами:

- демонстрація працездатного програмного засобу або інших результатів розробки;
- вміння пояснити основні моменти розробки;
- обґрунтованість відповідей на запитання членів комісії;
- представлення розробки (презентація, ілюстративний матеріал тощо).

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Либерти Джесс. Освой самостоятельно C++ за 21 день. – Москва-Санкт-Петербург-Киев: Вильямс, 2001. - 815 с.
2. Герберт Шилдт, Полный стправочник по С. – Москва-Санкт-Петербург-Киев : Вільямс, 2003. – 800 с.
3. ГлушаковС.В., Коваль А.В., Смирнов С.В.. Язык прграммирования С++. – Харьков: Фолио, 2002. - 500 с.
4. Гради Буч. Объектно-ориентированное программирование с примерами применения.: Пер. с англ.. – М.: Конкорд, 1992. - 519 с.
5. Ганеев Р.М. Проектирование пользовательского интерфейса средствами Win32 API. – М.: Горячая линия - Телеком, 2001. - 334 с.
6. Румянцев П.В. Азбука программирования Win32API. – М.: Горячая линия - Телеком, 2001. - 310 с.
7. Семеренко В.П. Програмування мовами С та С++ в середовищі Windows. Навчальний посібник. – Вінниця: ВДТУ, 2002. – 128 с.
8. Румянцев П.В. Работа с файлами в Win32API. – М.: Горячая линия - Телеком, 2002. - 216 с.
9. Бабаш А.В., Шанкин Г.П. Историякриптографии. Ч.1. – М.: Гелтос АРВ, 2002. – 240 с.
10. Кухар В.М., Тадіян С. І.Математика. Множини. Логіка. Цілі числа. Практикум. – Київ: Вища школа, 1989. – 333 с.
11. Вебер Д., Технология JAVA в подлиннике: пер. с англ.. – СПб.: БХВ-Петербург, 2001. – 1104 с.
12. Хабибуллин И.Ш. Самоучитель Java. – СПб.: БХВ-Петербург, 2002. – 464 с.
13. Бишоп Д. Єффективная работа: Java 2. – СПБ.: Питер Л.: Издательская группа BHV, 2002. – 592 с.
14. Морган Майкл. Java 2. Руководство разработчика.: Пер. с англ..: Уч. пос. – М.: Издательский дом “Вільямс”, 2000. – 720 с.
15. Яворски Джим, Пол Дж. Перроун. Система безопасности Java. Руководство разработчика.: Пер. с англ..: Уч. пос. - М.: Издательский дом “Вільямс”, 2001. – 528 с.
16. Хейлсберг А., Торгерсен М., Вилтамут С., Голд П. Язык программирования C#. Классика Computers Science. – СПб.: Издательский дом Питер, 2011. – 784 с.
17. Стиллмен Э., Грин Д. Head First. Изучаем C#. 3-е изд. – СПб.: Издательский дом Питер, 2017. – 816 с.
18. Дубовой В.М., Москвіна С.М., Никитенко О.Д. Моделювання процесів і систем керування з використанням UML. – Вінниця: ВНТУ. – 2009. – 103 с.

1.1 Додаток А

1.2 Варіанти завдань на курсову роботу

Варіант 1.Реалізувати **шифрування Даніеля Дефо**[13, С.168]. 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".

Варіант 2. Реалізувати **шифрування "слободське письмо"** [13, С.46]. Ідея використання літер "чужого алфавіту" для засекречування повідомлень отримала розвиток у XVII-XVIII ст.. Вона полягає в написанні українських (російських) слів латинськими літерами.

Варіант 3. 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 розміром5×5 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.

Реалізуватитакож шифрування **методом прямої перестановки** для випадку, коли запис повідомлення у таблицю відбуватиметься по рядках.

Варіант 4. Реалізувати шифрування **методом поодинокої перестановки за ключем (система шифрування Фальконета)** [9, С.176]**.**

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.

Варіант 5.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.

Варіант 6. Реалізувати **шифр маршрутної перестановки (шифр "Сцитала")** [9, С.33]. 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 $t<n \cdot m$, то решта пустих клітинок заповнюється аршрутом" – шляхом, що проходить через усі клітинки таблиці. Ключем шифру є числа n, m та вказаний маршрут.

Варіант 7. 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, отримуємо: ЯКООБРЕМЦВАОТАОРСУ. Ключем до шифру подвійної перестановки служить послідовність номерів стовпців і номерів рядків початкової таблиці (у нашому прикладі 312 та 316425, відповідно).

Варіант 8. Реалізувати **шифрування за допомогою магічних квадратів** [9, С.50]. 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.

Варіант 9.Реалізувати **систему шифрування Цезаря** [9, С.25]**.** 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	Д - В	И - Ж	Н - Л	С - П	Х - У	Щ - Ч	Э - Ы
Б - Я	Е - Г	К - З	О - М	Т - Р	Ц - Ф	Ъ - Ш	Ю - Ь
В - А	Ж - Д	Л - И	П - Н	У - С	Ч - Х	Ы - Щ	Я - Э
Г - Б	З - Е	М - К	Р - О	Ф - Т	Ш - Ц	Ь - Ъ	

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

Варіант 10.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	И	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	И	М	П	Т	Х	Ш	Ы	Ю	Б	Д	З	Л	О	С	Ф	Ч

Початковий алфавіт	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	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	Ъ	Э	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).

Варіант 11.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.

Варіант 12. 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≤k≤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	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Ы	Ь	Э	Ю	Я	**К**	**У**	**Р**	**С**	**1.2.1.1**	**О**	**В**	**А**	Б	Г	Д	Е	Ж	З	И	Л	М

21				25					30
Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	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 ЪРЫМСЖЗСЭОИЕЖГЭИ.

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

Варіант 13.Реалізувати шифрування за допомогою **шифруючої таблиці Трисемуса.** У 1508 р. абат з Германії ИоганнТрисемус 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 процеси шифрування і розшифрування. 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	И	Й	К
М	П	С	Т	У	Ф	Х	Ц
Ч	Ш	Щ	Ы	Ъ	Э	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 КУРСОВАЯ РАБОТА отримуємо шифротекст ЦЪИЩЙПВЛИВЬЙЪВ. Такі табличні шифри є монограмними, оскільки шифрування виконується по одній літері.

Варіант 14.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: $Y_i = (X_i + K_i) \; modm, \; (0 \le i < n)$, де K_i – i-й елемент ключової послідовності, m– кількість літер алфавіту. Процедура розшифрування описується співвідношенням: $Y_i = (X_i - K_i) \; modm, \; (0 \le i < n)$.

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

таблиця з випадковими 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.

Варіант 15.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. варіант 12). Для пояснення використаємо саме її. Процедура шифрування 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 біграмшифротексту 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 ТАЙНОЕ СТАНЕТ ЯВНЫМ. Розбиваємо його на біграми: ВС ЕТ АЙ НО ЕС ТА НЕ ТЯ ВН ЫМ. Дана послідовність біграм відкритого тексту перетворюється у таку послідовність: ГП ДУ ОВ ДЛ НУ ПД ДР ЦЫ ГА ЧТ.

   При розшифруванні застосовується зворотний порядок дій.

Варіант 16.Реалізувати **систему шифрування Віжинера**[9, С.97]**.** Система Віжинера (за іменем французького дипломата XVI ст. БлезаВіжинера) вперше була опублікована в 1586 р. і є однією з найстаріших та найбільш відомих багатоалфавітних систем. Даний шифр багатоалфавітної заміни можна описати таблицею шифрування, яка носить назву таблиці (квадрата) Віжинера. Для російського алфавіту вона має вигляд, показаний у додатка Д. 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 ОЦЕНКА. Зашифруємо повідомлення МОЯ КУРСОВАЯ РАБОТА.Випишемо наше повідомлення в рядок і запишемо під ним ключове слово з повторенням. В третій рядок будемо вписувати літери шифротекстуза таблицеюВіжинера.

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

Варіант 17.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 13). 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	Ы	П	Ч	
:	Д	У	О	К
З	Э	Ф	Г	Ш
Х	А	,	Л	Ъ

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

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 біграмишифротексту. 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 біграмушифротексту 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 біграмишифротексту беруть з лівої таблиці у стовпці, що відповідає другій літері біграми повідомлення. Друга літера біграмишифротексту береться з правої таблиці у стовпці, що відповідає першій літері біграми повідомлення. Тому біграма повідомлення ТО перетворюється у біграмушифротексту ЖБ.

Наприклад, зашифруємо повідомлення КУРСОВА РОБОТА. Розбиваємо повідомлення на біграми: КУ РС ОВ АР ОБ ОТ А. Отримуємо шифротекст: НЪ ГШ .. УМ ЪЛ ХЮ ЫД.

Варіант 18.Реалізувати **шифр“квадрат Полібія”** [9, С.36]. 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**

1.2.1.1.1.1 **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 використовується 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 запам'ятатискладно, користувачам шифру постійно приходиться тримати при собі ключ-квадрат. Тому 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.

Варіант 19.Реалізувати **шифр Чейза**[9, С.39]**.** 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 18) 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,Ъ	Ы	Э	Ю	Я

Ключем (другим) шифру є порядок розташування літер у таблиці. При шифруванні координати літер виписуються вертикально. Отже, 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, множиться на 9:

$$4927385 \times 9 = 44346465.$$

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.

Варіант 20.Реалізувати **шифрування за допомогою таблиць Тритемія**[9, С.57]**.** Реа-

лізація 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 Е). 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 ПРОГРАММА відкритого повідомлення перетвориться у послідовність ПСРЖФЕТЗшифротексту.

Варіант 21.Реалізувати **шифр Белазо**[9, С.87]. Даний спосіб шифрування винайдений італійцем ЖованомБелазо. 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 (див. варіант 20, додаток Е), 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

Варіант 22.Реалізувати **шифр "Атбаш"**[9, С.142].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:

А	Б	В	… … …	Э	Ю	Я
Я	Ю	Э	… … …	В	Б	А

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.

Варіант 23.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 ... Є 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.

Варіант 24.Реалізувати **книжковий шрифт Енея** [9, С.34-36, С.62]. 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.

Варіант 25.Реалізувати **шифр Порта**[9, С.87]**.** 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	ъ	ы	ь	э	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	ъ	ы	ь	э	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	ъ	ы	ь	э	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	ъ	ы	ь	э	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	Ъ	ы	ь	э	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	ъ	Ы	ь	э	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	ъ	ы	ь	э	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	ъ	ы	ь	э	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	ъ	ы	ь	э	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	ъ	ы	ь	э	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	ъ	ы	ь	э	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	ы	ь	э	ю	я	р	с	т	у	ф	х	ц	ч	ш	щ	ъ
13	**Ш**	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	Щ	ь	э	ю	я	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы
14	**Ъ**	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	Ы	э	ю	я	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь
15	**Ь**	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	Э	ю	я	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
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	ъ	ы	ь	э	ю

Шифруванняздійснюється 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 і інші літери (інтерваліи між словами не враховуються).Нариклад, зашифруємо за допомогою ключового слова «ШИФР» послідовність «КУРСОВАЯ РАБОТА».

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

Варіант 26.Реалізувати**омофоннузаміну**[9, С.204]**.** Омофонна замінааналогічнапростійзаміні, але маєєдинувідмінність: кожнійбуквівідкритого тексту ставиться у відповідністьдекількасимволівшифротексту. Наприклад, літера «А» заміняється на цифру 5, 13, 25 або 57; літера «Б» - на 7, 19, 12, 41 і т. д.

Варіант 28. **Генератори простих чисел.** Розробити програму, яка надає теоретичні відомості про генератори простих чисел і демонструє їх роботу (генерує вказану кількість простих чисел на вказаному інтервалі).

Варіант 29.**Решето Ератосфена.** Розробити програму, яка дозволяє скласти “решето

Ератосфена” на будь-якому інтервалі чисел. *([10],С.284-285).*

Варіант 30.**Подільність чисел.** Розробити програму, яка надає теоретичні відомості про загальну ознаку подільності (ознаку Паскаля) та демонструє визначення подільності чисел на 3, 5, 7, 9 і т.д. *([10], С. 277-280)*.

Варіант 31.**НСК і НСД.** Розробити програму, яка за допомогою алгоритму Евкліда дозволяє визначити НСК і НСД для будь-яких довільних чисел. *([10],С.288-290).*

Варіант 32.**Сортування.** Розробити програму, яка надає теоретичні відомості про основні методи сортування, демонструє їх застосування і наводить основні статистичні дані результатів сортування.

Варіант 33. **Генератори випадкових чисел.** Розробити програму, яка надає теоретичні відомості про генератори випадкових чисел і демонструє їх роботу (генерує вказану кількість випадкових чисел на вказаному інтервалі).

Варіант 34. **Ілюзії сприйняття.** Розробити програму, яка демонструє неадекватне відображення сприйманого предмета і його властивостей.

Варіант 35. **Графічний ключ.** Розробити програму, яка реалізує автентифікацію користувача за графічним зображенням.

Варіант 36. **Цифровий ключ.** Розробити програму, яка реалізує автентифікацію користувача за цифровим паролем.

Варіант 37. **Капча графічна.** Розробити програму, яка реалізує використання графічної капчі для захисту від емуляції реальних користувачів.

Варіант 38. **Капча текстова.** Розробити програму, яка реалізує використання текстової капчі для захисту від емуляції реальних користувачів.

Варіант 39. **Цифровий пароль.** Розробити програму, яка реалізує автентифікацію користувача з використаннямвіртуальної клавіатури.

Варіант 40. **Робота з числами великої розрядності.** Розробити програму, яка реалізує алгоритми роботи з довгими числами.

Варіант 41. **Розробка програм для реалізації різних ігор** (за власним алгоритмом розробника)**.**

Додаток Б

Приклад титульного аркуша

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

КУРСОВА РОБОТА

з дисципліни "Технологія програмування"
на тему: «**РОЗРОБКА ПРОГРАМИ ДЛЯ РЕАЛІЗАЦІЇ ШИФРУВАННЯ МЕТОДОМ ПРЯМОЇ ПЕРЕСТАНОВКИ»**

08-20.ТП.118.13.107 ПЗ

Студента(ки)2-го курсу групи 1 БС-17 б
спеціальності 125 «Кібербезпека»

Петрова П. П.

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

Національна шкала _____

Кількість балів: _____Оцінка ECTS _____

Члени комісії: _____ _____
(підпис) (прізвище та ініціали)

_____ _____
(підпис) (прізвище та ініціали)

м. Вінниця – 2018 рік

Додаток В
Приклад індивідуального завдання

Вінницький національний технічний університет
Факультет інформаційних технологій і комп'ютерної інженерії

ЗАТВЕРДЖУЮ
Зав. кафедри ЗІ, д. т. н., проф.

_____В. А. Лужецький

_____ _____ 2018 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу з дисципліни "Технологія програмування"
студенту групи 1БС-17Холодову І. В.

Тема: «Реалізація зашифрування/розшифрування»
Вхіднідані:
об'єкт шифрування: повідомлення з клавіатури або текстові файли;
метод шифрування: пряма перестановка символів;програмні засоби:мова програмування – C++, API-функції; програмне середовище – VisualStudio; інтерфейс: віконно-графічний; операційнесередовище – сімейство Windows.
Вихідні дані:
- програмний засіб для шифрування\розшифрування повідомлень;
- інструкції для роботи з програмою.

Текстовачастина
Вступ. 1. Розробка структури програмного засобу. 2. Реалізація основної задачі курсової роботи. 3. Аналіз роботи програмного засобу. Висновки. Перелік використаних джерел. Додатки.
Ілюстративна частина
Загальна схема роботи програмного засобу. Схема ресурсів і даних програми. Схема алгоритмів шифрування і розшифрування повідомлень. Структура проекту. Фрагменти інтерфейсу.

Дата видачі _____ 2018 р.

Завдання отримав _____ Холодов І. В.

Керівник курсової роботи _____ Каплун В. А.

Додаток Г

Використання схем алгоритмів, програм, даних і систем

Схеми алгоритмів, програм, даних і систем складаються з символів, що мають задане значення, короткого тексту пояснення і з'єднувальних ліній. Усі символи поділяються на такі підгрупи:

- символи даних (табл. Г.1);
- символи процесів (табл. Г.2). Приклад наведено на рис. Г.1.;
- спеціальні символи (табл. Г.3). Приклад наведено на рис. Г.2;
- символи ліній (табл. Г.4). Приклад наведено на рис. Г.3.

Символи поділяються на:

- основні, для випадків, коли точний вигляд процесу або носія даних невідомий або відсутня необхідність в описі фактичного носія даних;
- специфічні, використовувані тоді, коли відомий точний вигляд процесу або носія даних або коли необхідно описати фактичний носій даних.

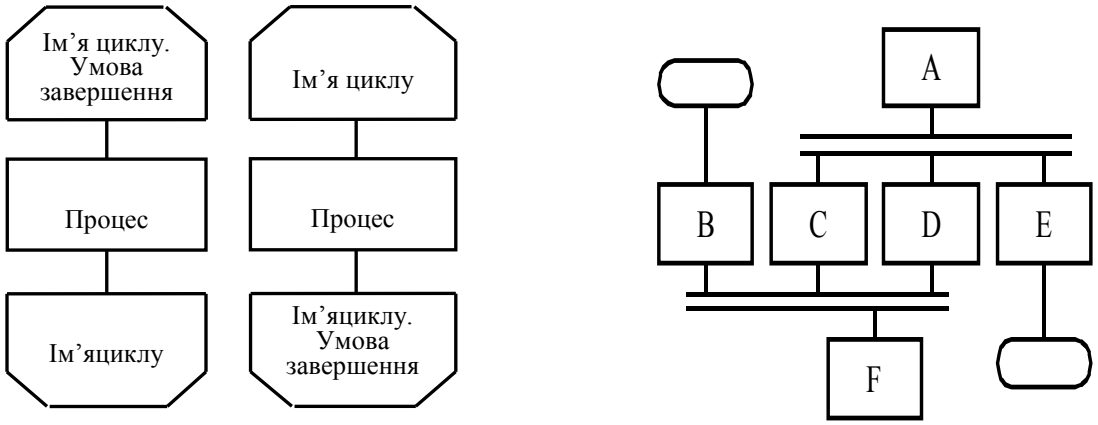


Рисунок Г.1 – Приклад застосування символів процесу (а – використання символів меж циклу; б – використання символів паралельних дій)

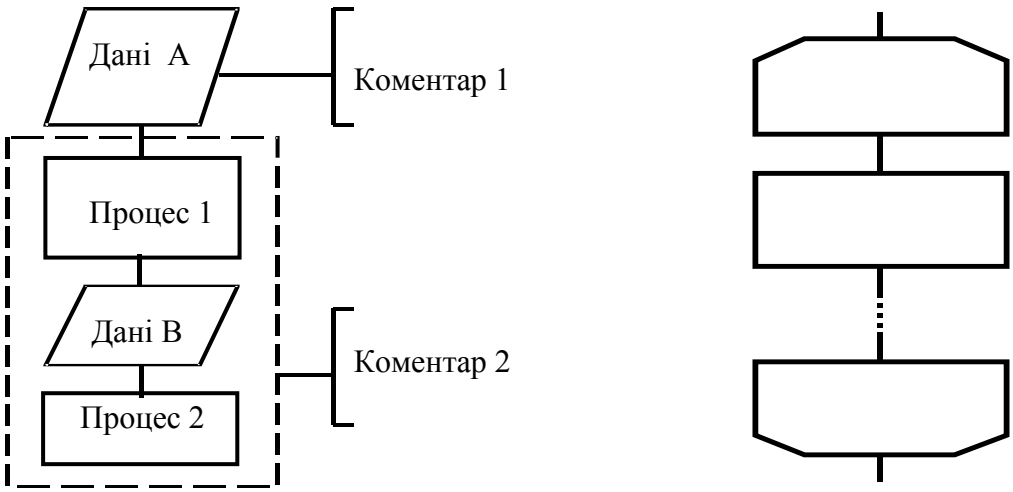


Рисунок Г.2 – Приклади використання спеціальних символів

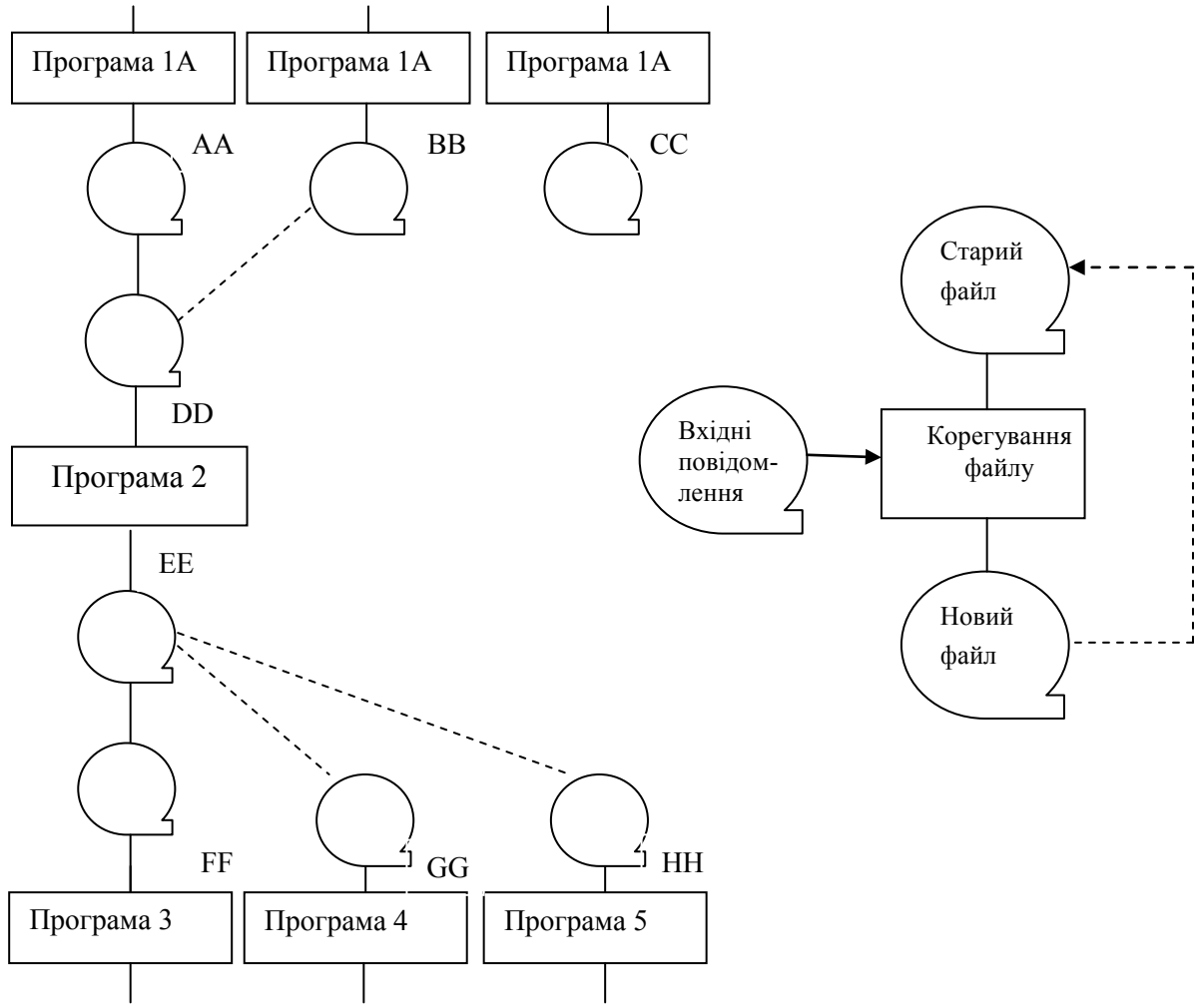


Рисунок Г.3 – Приклади використання символів ліній

Схеми даних відображають шлях даних при розв'язанні задач і визначають етапи обробки, а також різні використовувані носії даних. Схема даних складається з таких символів:символів даних, які можуть також вказувати вид носія даних;символів процесу, який виконується над даними (можуть також вказувати функції, виконувані обчислювальною машиною);символів ліній, які вказують потоки даних між процесами і носіями даних;спеціальних символів для полегшення написання і читання схеми.

Символи даних чергуються з символами процесу. Схема даних починається і закінчується символами даних (за винятком спец. символів).

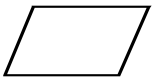
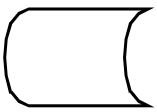
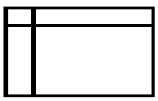
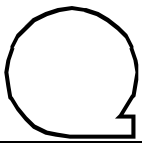
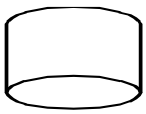
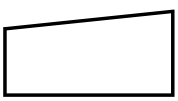
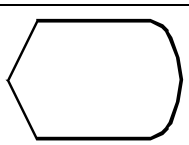
Схеми програм відображають послідовність операцій в програмі. Схема програми складається з:символів процесу, що вказують фактичні операції обробки даних (включаючи символи, що визначають шлях, якого слід дотримуватися з урахуванням логічних умов);лінійних символів, що вказують потік управління;спеціальних символів, використовуваних для полегшення написання і читання схеми.

Схеми роботи системивідображають управління операціями і потік даних в системі.Схема роботи системи складається з:символів даних, що вказують на наявність даних (символи даних можуть також вказувати на вид носія даних);символів процесу, що вказують операції, які слід виконати над даними, а також визначають логічний шлях, якого слід дотримуватися;лінійних символів, що вказують на потоки даних між процесами і (або) носіями даних, а також потік управління між процесами;спеціальних символів, використовуваних для полегшення написання і читання блок-схеми.

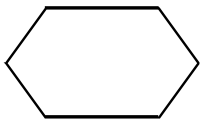
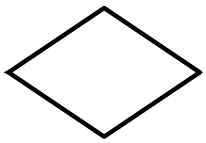
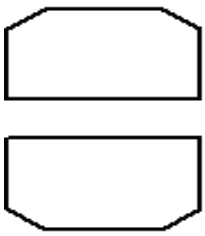
Схема взаємодії програмвідображає шлях активацій програм і взаємодій з відповідними даними. Кожна програма в схемі взаємодії програм показується тільки один раз (у схемі роботи системи програма може відображатися більше, ніж в одному потоці управління).Схема взаємодії програм складається з:символів даних, що вказують на наявність даних;символів процесу, що вказують на операції, які виконують над даними;лінійних символів, що відображають потік між процесами і даними, а також ініціації процесів;спеціальних символів – для полегшення написання і читання схеми.

Схема ресурсів системивідображає конфігурацію блоків даних, блоків обробки цих даних, яка потрібна для розв'язання задачі або набору задач.Схема ресурсів системи складається з:символів даних, що відображають вхідні, вихідні і запам'ятовують пристрої обчислювальної машини;символів процесу, що відображають процесори, канали і т. д.); лінійних символів, що відображають передачу даних між пристроями введення-виведення і процесорами, а також передачу управління між процесорами;спеціальних символів, використовуваних для полегшення написання і читання схеми.

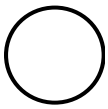
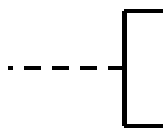
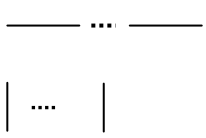
Таблиця Г.1 – Символи даних

Основні символи даних		
Дані		Символ відображає дані, носій даних невизначений
Дані, що запам'ятовуються		Символ відображає дані, що зберігаються, у вигляді, придатному для обробки, носій даних невизначений
Специфічні символи даних		
Оперативний запам'ятовуваль-ний пристрій		Символ відображає дані, що зберігаються в оперативному запам'ятовувальному пристрої
Запам'ятовуваль-нийпристрій з послідовним доступом		Символ відображає дані, що зберігаються в запам'ятовувальному пристрої з послідовним доступом (магнітна стрічка, касета з магнітною стрічкою, магнітофонна касета)
Запам'ятовуваль-нийпристрій з прямим доступом		Символ відображає дані, що зберігаються в запам'ятовувальному пристрої з прямим доступом (магнітний диск, магнітний барабан, гнучкий магнітний диск)
Документ		Символ відображає дані, подані на носії в легкій для читання формі (документ для оптичного або магнітного зчитування, мікрофільм, рулон стрічки з підсумковими даними, бланки введення даних)
Ручне введення		Символ відображає дані, що вводяться вручну під час оброблення з пристроїв будь-якого типу (клавіатура, перемикачі, кнопки, світлове перо, смужки зі штриховим кодом)
Дисплей		Символ відображає дані, поданіу візуальнійлюдиночитабельній формі на носії у вигляді пристроювідображення (екран для візуального спостереження, індикатори введення інформації)

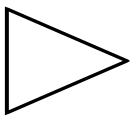
Таблиця Г. 2 – Символи процесу

Основні символи, процесу		
Процес		Символ відображає функцію обробки даних будь-якого вигляду (виконання певної операції або їх групи, що приводить до зміни значення, форми інформації).
Специфічні символи процесу		
Підпорядкований процес		Символ відображає підпорядкований процес, що складається з однієї або декількох операцій або кроків програми, які визначені у іншому місці
Ручна операція		Символ відображає будь-який процес, виконуваний людиною
Підготовка		Символ відображає модифікацію команди або групи команд з метою дії на деяку подальшу функцію (установлення перемикача, модифікація індексного регістра або ініціалізація програми)
Умова або вибір		Символ відображає умову, вибір або функцію типу перемикача, що має один вхід і ряд альтернативних виходів, один і лише один з яких може бути активізований після обчислення умов, визначених усередині цього символу
Паралельні дії		Символ відображає синхронізацію двох або більше паралельних операцій
Межа 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

Таблиця Г. 3 – Спеціальні символи

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
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)
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
Пропуск 	Символ (три крапки) використовують в схемах для відображення пропуску символу або групи символів, в яких не визначені ні тип, ні число символів. Символ використовують тільки в символах лінії або між ними. Він застосовується головним чином в схемах, що зображають загальні результати вибору з невідомим числом повторень

Таблиця Г. 4 – Символи ліній

Основний символ ліній	
Лінія	Символ відображає потік даних або управління. У разі необхідності або для підвищення легкості читання можуть бути додані стрілки-покажчики
Специфічні символи ліній	
Передача управління 	Символ відображає безпосередню передачу управління від одного процесу до іншого, іноді з можливістю прямого повернення до ініціювального процесу після того, як ініційований процес завершить свої функції. Тип передачі управління повинен бути названий усередині символу (наприклад, запит, виклик, подія)
Канал зв'язку 	Символ відображає передачу даних по каналу зв'язку
Пунктирна лінія	Символ відображає альтернативний зв'язок між двома або більшою кількістю символів, а також використовується для обведення ділянки

Додаток Є
Приклади UML-діаграм

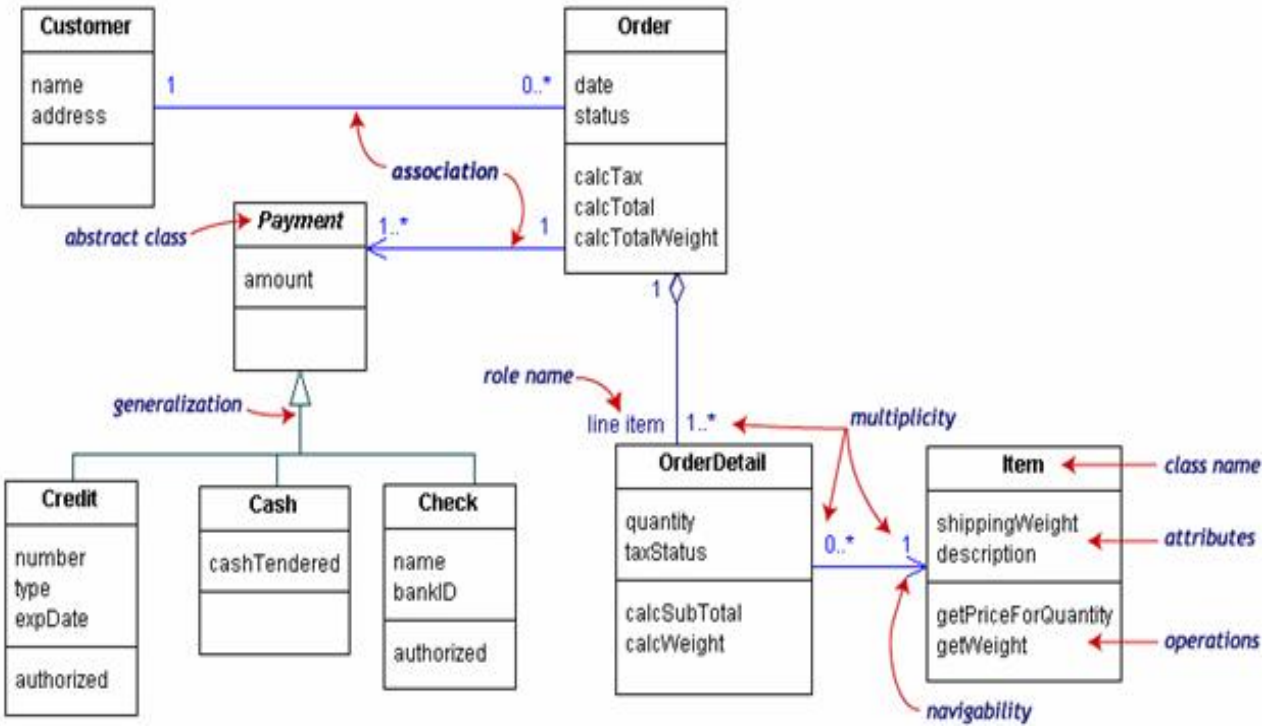


Рисунок Є.1 – Приклад діаграми класів

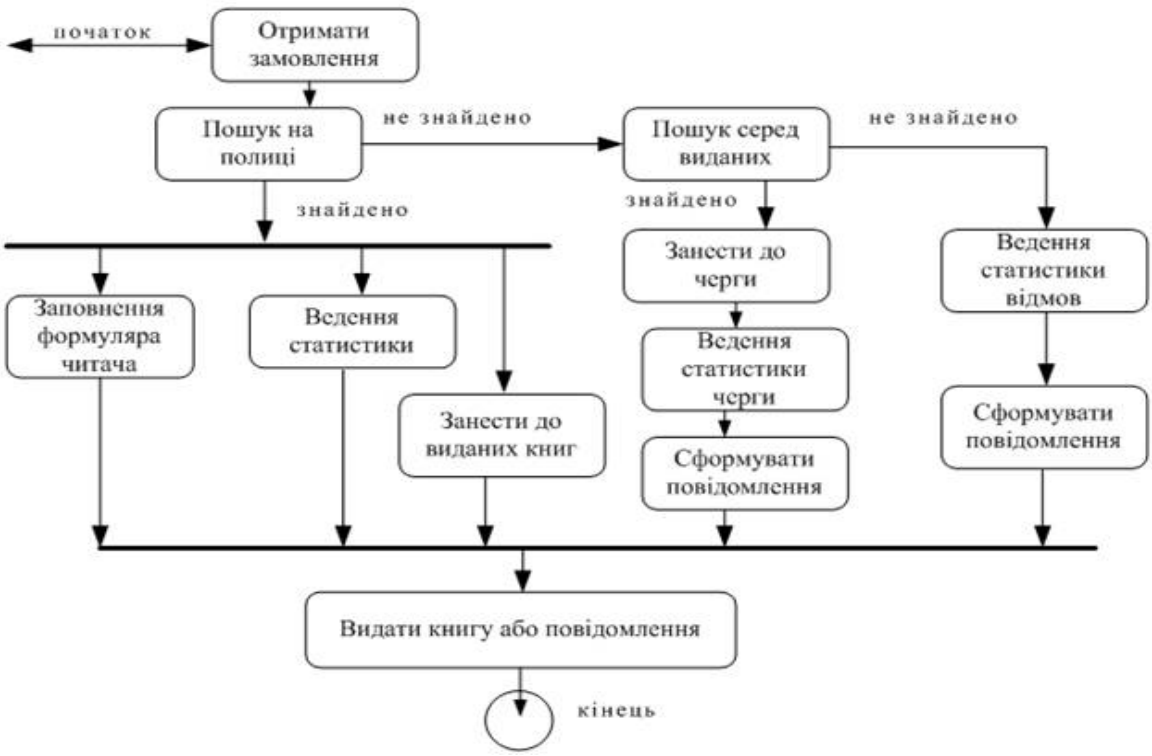


Рисунок Є.2 – Приклад діаграми діяльності

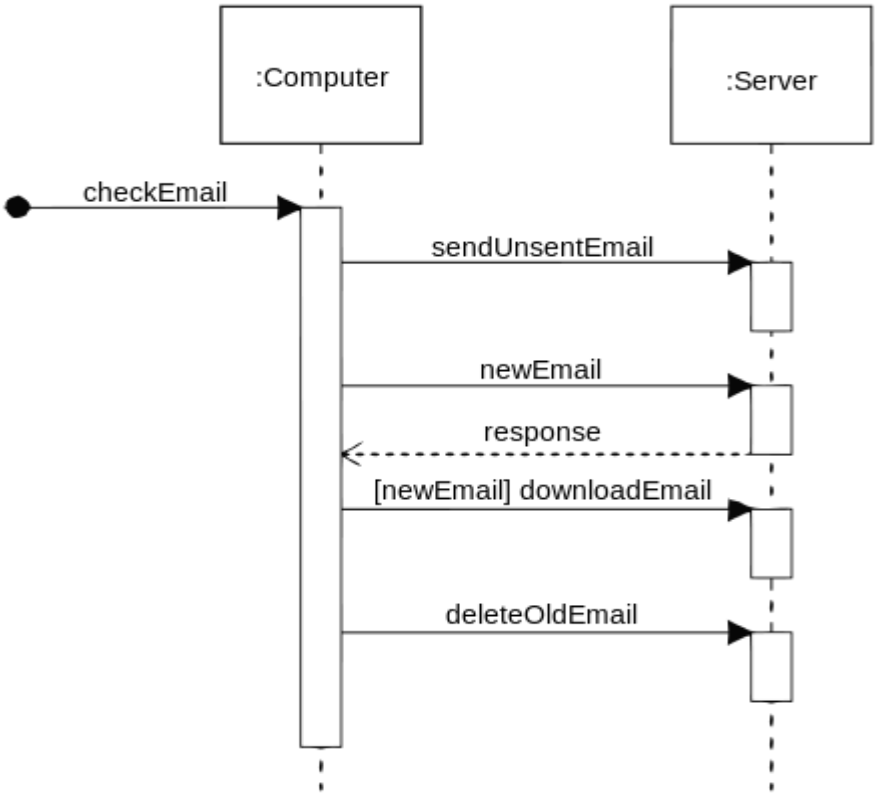


Рисунок Є.3 –Приклад діаграми послідовності

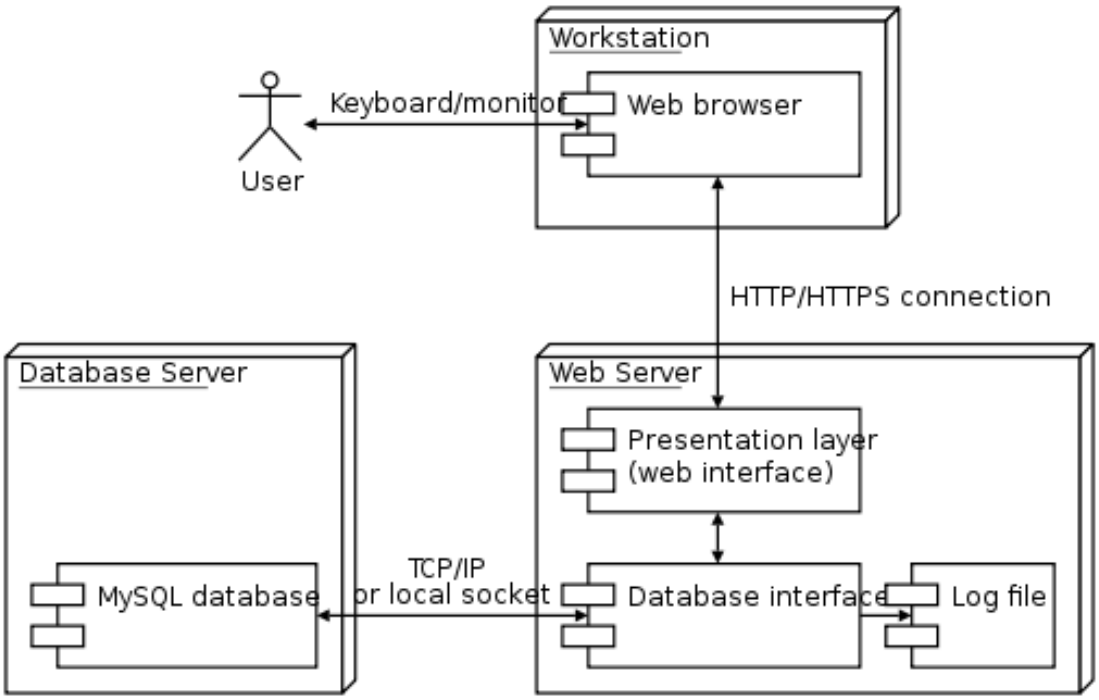


Рисунок Є.4 – Приклад діаграми розгортання