

## **МЕТОДИЧНІ ВКАЗІВКИ**

**до виконання курсової роботи з дисципліни**

**"ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ"**

**для студентів напряму підготовки 6.170101**

**“Безпека інформаційних і комунікаційних систем ”**

## ЗМІСТ

|  |    |
|--|----|
| 1 ТЕМАТИКА ТА ЗМІСТ КУРСОВОЇ РОБОТИ.....                     | 3  |
| 2 ВИМОГИ ДО РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ .....                | 4  |
| 3 ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....                      | 6  |
| 3.1 Загальні правила оформлення.....                         | 6  |
| 3.2 Структура пояснювальної записки .....                    | 8  |
| 3.3 Вміст вступної частини пояснювальної записки .....       | 8  |
| 3.3.1 Титульний аркуш .....                                  | 8  |
| 3.3.2 Індивідуальне завдання .....                           | 9  |
| 3.3.3 Анотація .....   | 9  |
| 3.3.4 Зміст.....   | 10 |
| 3.4 Вміст і оформлення основної частини .....                | 10 |
| 3.4.1 Вступ .....  | 10 |
| 3.4.2 Розробка і обґрунтування структури програми .....      | 11 |
| 3.4.3 Використання схем .....                                | 11 |
| 3.4.4 Програмна реалізація задачі.....                       | 14 |
| 3.4.5 Тестування програми і розробка інструкцій .....        | 15 |
| 3.4.6 Висновки .....   | 17 |
| 3.4.7 Оформлення переліку використаних джерел.....           | 17 |
| 3.5 Оформлення додатків.....                                 | 18 |
| 4 ГРАФІК ВИКОНАННЯ КУРСОВОЇ РОБОТИ І ПОРЯДОК ЇЇ ЗАХИСТУ .... | 19 |
| ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ.....                       | 20 |
| Додаток А. Варіанти завдань на курсову роботу .....          | 21 |
| Додаток Б. Приклад оформлення титульного аркуша .....        | 22 |
| Додаток В. Приклад оформлення індивідуального завдання ..... | 23 |
| Додаток Г. Символи даних, процесів і ліній.....              | 24 |

# 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 «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”: робота з файлами, робота з масивами, різноманітні види операторів (умовні оператори, оператори вибору, оператори циклу і т.д.), застосування принципів об'єктно-орієнтованого програмування, 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 семестра. 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 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 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 операційної системи Windows. Розроблена програма обов'язково повинна мати такі складові.

1. *Застосування основних принципів об'єктно-орієнтованого програмування:* абстрактні типи даних, інкапсуляція, успадкування класів, поліморфізм. Студент повинен добре розуміти, з яких причин він використав у програмі той чи інший принцип, 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 найсучаснішою і найуживанішою, і знання основних її концепцій обумовлює розуміння побудови інших технологій програмування.

2. *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.

3. *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). Реалізація інтерфейсу та використання елементів керування повинні базуватися на API-функціях, оскільки саме ці функції є складовою ядра операційної системи і основою для побудови програмних засобів багатьох мов програмування і використання більшістю сучасних програмних середовищ.

4. *Використання файлів для зберігання та зчитування інформації.* Це може бути або введення початкової інформації 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.

5. *Перевірка цілісності даних* на рівні перевірки правильності введеної інформації (числової, символної, 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.
6. *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, з поясненнями, допоміжними вікнами повідомлень.
7. *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 КР за ДСТУ 3008-95. Текст пояснювальної записки повинен бути набраний на комп'ютері та роздрукований на принтері.

**Шрифт і відступи.** Текст пояснювальної записки повинен бути набраний у будь-якому текстовому редакторі шрифтом Times New Roman розміром 14 з інтервалом між рядками 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 – 2.0 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 (ДСТУ 3008-95) 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, 3.2.2.1 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 – Основні типи даних

| Номер | Тип даних | Опис                            |
|-------|-----------|---------------------------------|
| 1     | Float     | Дійсні числа з плаваючою точкою |
| 2     | Integer   | Цілі числа                      |
| ...   | ... ..    | ... ..                          |

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

Продовження таблиці 1

| 1   | 2      | 3                |
|-----|--------|------------------|
| 25  | String | Рядок символів   |
| 26  | Char   | Символьні літери |
| ... | ... .. | ... ..           |

**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. 3–5”, або в дужках по тексту (рис. 3.6). Посилання на раніше наведений рисунок дають зі скороченим словом ”дивись” (див. рис. 4) за ходом чи в кінці речення.

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 12 – Вигляд екрана при обробці пункту меню “Файл”

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.

**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;
  - 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 ДСТУ 3008-95 титульний аркуш виконується за встановленим зразком. Зразок титульного аркушу пропонується у додатка 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:

**XX-XX.XXX.XXX.XX.XXX XX**  
          1      2      3      4      5      6

- де
- 1 (XX-XX) – числовий шифр кафедри, прийнятий у ВДТУ (08-20);
  - 2 (XXX) – умовне скорочення для дисципліни (ЗПЗ, ТП, ПРГ і т.д.);
  - 3 (XXX) – перша цифра 0, якщо це проект або 1, якщо робота, друга і третя цифри означають рік, наприклад, 08 – 2008 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 другою сторінкою після титульного аркуша. Зразок індивідуального завдання до курсової роботи наведено в додатку В. Обов'язковим в індивідуальному завданні є наведення вхідних і вихідних даних.

Керівник роботи пропонує зміст пояснювальної записки, як правило, в розроблених методичних вказівках. У навчальних цілях зміст може висвітлюватись в індивідуальному завданні. В залежності від специфіки індивідуального завдання керівник курсової роботи може пропонувати тему, яка підлягає конкретному обґрунтуванню та розробці індивідуального завдання, яке дещо відрізняється від типового.

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 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  $\frac{1}{3}$  частину сторінки (не перевищувати  $\frac{1}{2}$  сторінки).

Анотацію розміщують безпосередньо за аркушем з індивідуальним завданням, починаючи з нової сторінки (третьої), нумерація якої не зазначається і в зміст не входить.

### 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 четвертою сторінкою. 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

СПИСОК 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 (ДСТУ 3008-95) 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 загальноживане поняття у вигляді аббревіатури, 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 аббревіатури.

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 Розробка і обґрунтування структури програми

В даному розділі повинна бути детально описана мета роботи, алгоритм досягнення результату, передбачена розробка загальної схеми функціонування програми.

В цьому розділі 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.

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 даному розділі. В результаті повинна бути розроблена схема роботи програми.

Кожна програма повинна мати певні вхідні і вихідні дані, як вони будуть готуватись, звідки вводиться у програму, де зберігатись, що буде вихідними даними і у якому вигляді – це також повинно бути описано у цьому розділі або подано у вигляді схеми даних.

Крім того, в даному розділі слід розробити 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.

### 3.4.3 Використання схем

Окремими підпунктами основного розділу або завершенням певних підрозділів бажано розробити і описати різні схеми:

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

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

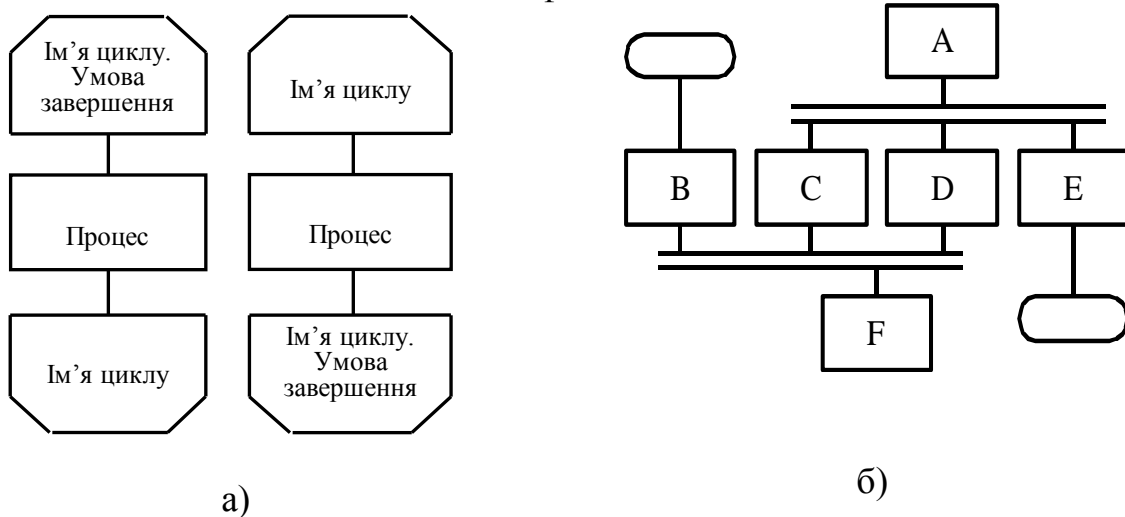


Рисунок 1 – Приклад застосування символів процесу (а – використання символів меж циклу; б – використання символів паралельних дій)

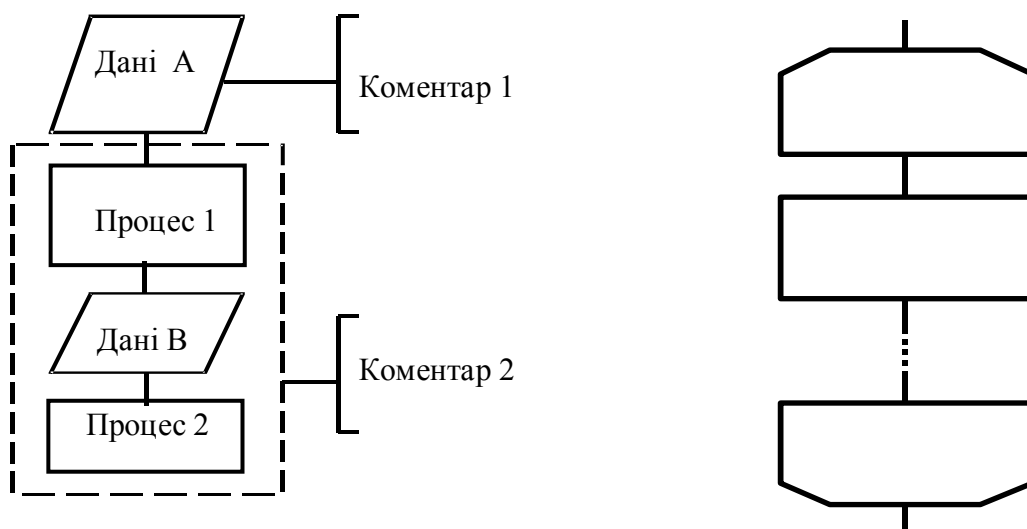


Рисунок 2 – Приклади використання спеціальних символів

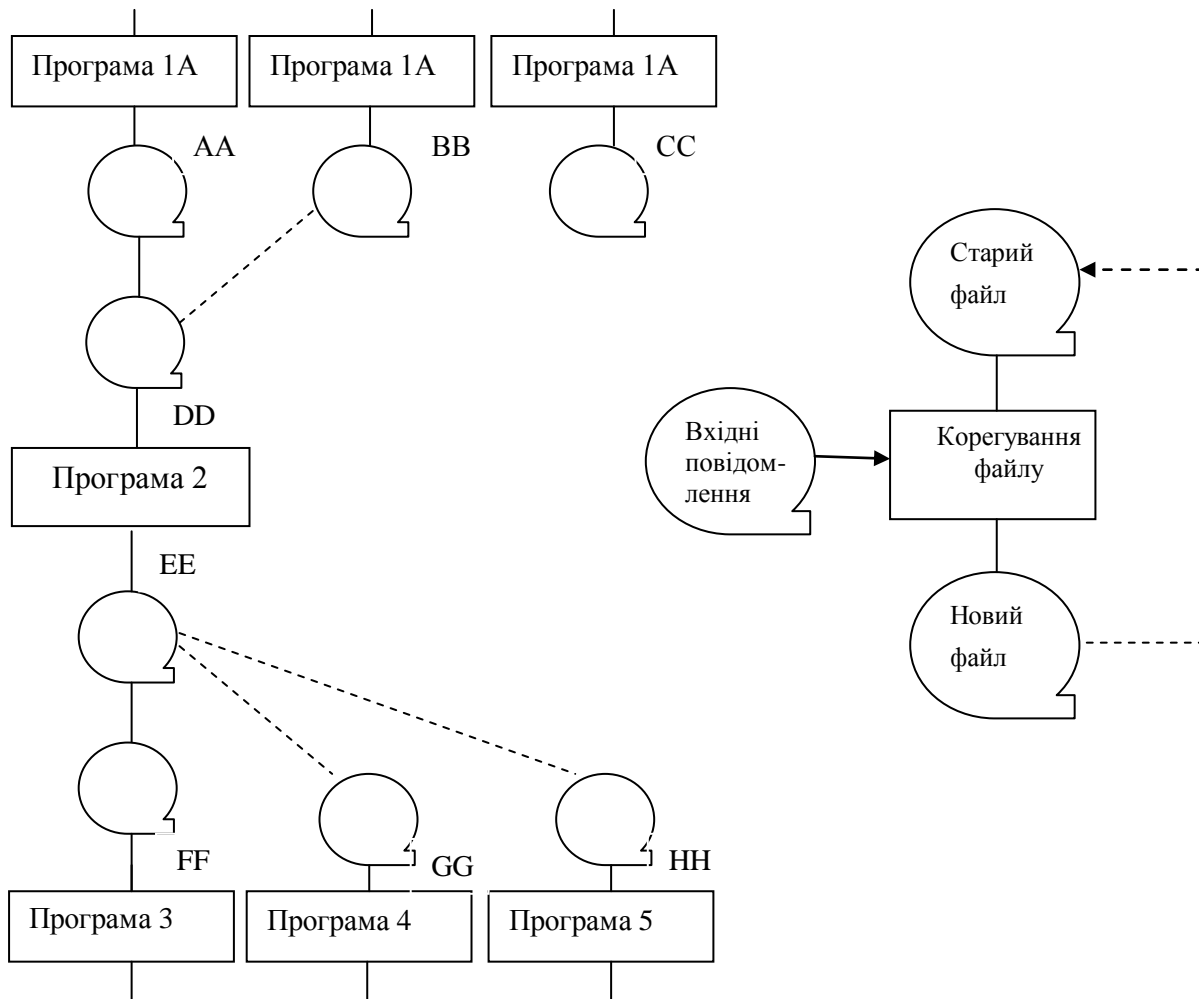


Рисунок 3 – Приклади використання символів ліній

**Схеми даних** відображають шлях даних при розв'язанні задач і визначають етапи обробки, а також різні використовувані носії даних. Схема даних складається з:

- 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 (або) 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 системи** відображають 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 програм** відображає шлях активацій програм і взаємодій з відповідними даними. Кожна програма в схемі взаємодії програм показується тільки один раз (у схемі роботи системи програма може зображатися 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, використовуваних для полегшення написання і читання схеми.

**Схема ресурсів системи** відображає 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 (центральні 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.

#### 3.4.4 Програмна реалізація 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 заголовочні файли, які використовуються при розробці програмного забезпечення, також повинні бути описані коротко, якщо вони є стандартними, і детально, якщо вони є продуктом роботи студента.

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).

Оскільки програма розробляється для її використання в операційному середовищі Windows, то слід звернути особливу увагу на повідомлення від

операційної системи та їх обробку і на використання API-функцій.

Крім того, оскільки розроблена програма має бути повноцінним додатком Windows, то необхідно докладно описати програмні засоби для реалізації користувацького інтерфейсу: які діалогові вікна використані, які елементи керування за що відповідають, які засоби використані для реалізації підказок, надання допомоги, виведення повідомлень.

Даний розділ також має бути дуже змістовним, конкретним, зрозумілим, оскільки саме він демонструє знання та навички у галузі програмування. Рекомендований обсяг даного розділу – 8-10 сторінок пояснювальної записки.

### 3.4.5 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.

Оскільки темою даної курсової роботи є розробка програмного продукту, то необхідно передбачити розробку конкретних інструктивних матеріалів для роботи з програмою. Даний підрозділ так і може називатись: “Інструкції для роботи з програмою ...”, або “Розробка інструктивних документів з ...”, або “Розробка рекомендацій для ...”.

Таким чином, слід передбачити розробку рекомендацій для роботи з програмою: інструкцію програмісту, інструкцію системному програмісту, інструкцію оператору (користувачу), інструкцію з технічного обслуговування. Інструкції для роботи з програмою можна виносити в додатки.

При описі вказаних підпунктів рекомендується наводити безпосередньо по тексту або винести у додатки вигляд діалогових вікон, образи екранів і т.д., що пояснюють наведений текст.

**Інструкція з технічного обслуговування.** Інструкція з технічного обслуговування повинна містити такі підпункти (відповідно до ГОСТ 19.508-79):

- *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.).



### 3.4.6 Висновки

Висновки оформляють з нової пронумерованої сторінки посередині (ДСТУ 3008-95) 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.

### 3.4.7 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. В списку кожен літературу записують з абзацу, нумерують арабськими цифрами, починаючи з одиниці. Правильне оформлення певного джерела інформації можна переглянути у переліку літературних джерел у будь-якому навчальному посібнику. Якщо у списку використаних джерел є посилання на Інтернет-сторінки, слід наводити разом з назвою Інтернет-сторінки.

Приклад оформлення переліку використаних джерел різного характеру.

#### *Посилання на книги*

1. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы и исходные тексты на языке С. 2-е изд. - СПб.: Вильямс, 2000. - 789 С.
2. Бабаш А. В., Шанкин Г. П. Криптография. Под ред. Шерстюка В. П., Применко Э. А. / Шанкин Г. П.. - М.: СОЛОН-Р, 2002. - 512 С.
3. Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В. Основы криптографии. Учебное пособие. - М.: Гелиос АРВ, 2001. - 480 С.

#### *Посилання на журнали*

4. Ершов А. А. Стабильные методы оценки параметров // Автоматика и телемеханика. – 1978. – №8. – С. 86-91.

#### *Посилання на ГОСТ і ДСТУ*

5. ГОСТ 7.9-77. Реферат и аннотация. – М.: Издательство стандартов, 1981. – 6 с.

#### *Посилання на патенти*

6. Пат. 3818311, США, МКИ НОЗК 17/60. Схема защиты полупроводникового переключателя. – Опубл. 04.05.84.

#### *Посилання на web-сторінки*

7. Інформаційна безпека // <http://www.domarev.ru/>.
8. Научная библиотека // [www.abris.crimea.ua/index.php](http://www.abris.crimea.ua/index.php).

### **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, блок-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 Г, 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.

## 4 ГРАФІК ВИКОНАННЯ КУРСОВОЇ РОБОТИ І ПОРЯДОК ЇЇ ЗАХИСТУ

Рекомендується такий графік виконання курсової роботи, який враховує самостійну роботу студентів під час 4-го триместру (14 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  | 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. Після позитивного висновку про готовність курсової роботи студент повинен захистити її перед комісією у складі двох викладачів, які призначені кафедрою.

## ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Либерти ДжесС. Освой самостоятельно С++ за 21 день. – Москва-Санкт-Петербург-Киев: «Вильямс», 2001. - 815 с.
2. Павловская Т .А. Программирование на языке высокого уровня С/С++. – С.-П.: «Питер», 2002. - 460 с.
3. Круподьорова Л. М., Петух А. М. Технологія програмування мовою Сі. Частина 1. Навчальний посібник. – Вінниця: ВНТУ, 2006. – 206 С.
4. Герберт Шилдт, Полный справочник по С. – Москва-Санкт-Петербург-Киев.: «Вильямс», 2003. – 800 с.
5. Грегори К. Использование Visual С++ 6. – Москва-Санкт-Петербург-Киев: «Вильямс», 2000. – 850 с.
6. Глушаков С. В., Коваль А. В., Черепнин С. А. Программирование на Visual С++ 6.0. – Харьков: Фолио, 2002. – 726 с.
7. Глушаков С. В., Коваль А. В., Смирнов С.В.. Язык программирования С++. – Харьков: «Фолио», 2002. - 500 с.
8. Гради Буч. Объектно-ориентированное программирование с примерами применения.: Пер. с англ.. – М.: Конкорд, 1992. - 519 с.
9. Ганеев Р. М. Проектирование пользовательского интерфейса средствами Win32API. – М.: Горячая линия - Телеком, 2001. - 334 с.
10. Румянцев П. В. Азбука программирования Win32API. – М.: Горячая линия - Телеком, 2001. - 310 с.
11. Семеренко В. П. Програмування мовами С та С++ в середовищі Windows. Навчальний посібник. – Вінниця: ВДТУ, 2002. – 128 с.
12. Румянцев П. В. Работа с файлами в Win32API. – М.: Горячая линия - Телеком, 2002. - 216 с.
13. Бабаш А. В., Шанкин Г. П. История криптографии. Ч.1. – М.: Гелтос АРВ, 2002. – 240 с.
14. Кухар В. М., Тадіян С. І. Математика. Множини. Логіка. Цілі числа. Практикум. – Київ: Вища школа, 1989. – 333 с.

**Додаток А**  
**Варіанти завдань на курсову роботу**

## Додаток Б

### Приклад оформлення титульного аркуша

Міністерство освіти і науки, молоді та спорту України  
Вінницький національний технічний університет  
Інститут інформаційних технологій та комп'ютерної інженерії  
Кафедра захисту інформації

### **РОЗРОБКА ПРОГРАМИ ДЛЯ РЕАЛІЗАЦІЇ ШИФРУВАННЯ МЕТОДОМ ПРЯМОЇ ПЕРЕСТАНОВКИ**

Пояснювальна записка  
до курсової роботи з дисципліни "Технологія програмування"  
за напрямом підготовки  
6.170101 – "Безпека інформаційних і комунікаційних систем"  
08-20.ТП.110.13.107 ПЗ

Керівник курсової роботи

ст. викл. каф. ЗІ \_\_\_\_\_ Каплун В. А.

Розробив студент гр. 1БС-07

\_\_\_\_\_ Холодов І. В.

Курсову роботу захищено

з оцінкою \_\_\_\_\_

\_\_\_\_\_ 2012 р.  
(Підпис керівника) (Дата)

\_\_\_\_\_  
(Підпис асистента)

## Додаток В

### Приклад оформлення індивідуального завдання

Вінницький національний технічний університет  
Інститут інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ  
Зав. кафедри ЗІ, д. т. н., проф.  
\_\_\_\_\_ В. А. Лужецький  
” \_\_\_\_ ” \_\_\_\_\_ 2012 р.

### ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу з дисципліни ”Технологія програмування”  
студенту групи ІБС-11 факультету КСМ Холодову І. В.

Тема: «Розробка програми для реалізації шифрування методом прямої  
перестановки»

Вимоги до курсової роботи.

1. Дослідити і засвоїти метод шифрування повідомлень.
2. Розробити загальну структурну схему функціонування програми в цілому і алгоритмів шифрування і розшифрування.
3. Підібрати програмні засоби для реалізації розробленого алгоритму, підготувати складові користувацького інтерфейсу.
4. Програмно реалізувати поставлену задачу і підготувати детальний опис послідовності розробки програмного засобу.
5. Протестувати програму на предмет ефективності і правильності її роботи і розробити рекомендації для роботи з нею.

Вихідні дані: об'єкт шифрування – повідомлення з клавіатури або текстові файли; метод шифрування – пряма перестановка символів; мова програмування – С++; програмні засоби – АРІ-функції; програмне середовище – Visual С++; інтерфейс – віконно-графічний; операційна система – сімейство Windows.

Дата видачі \_\_\_\_\_ 2012 р.

Керівник \_\_\_\_\_ Каплун В. А.

Завдання отримав \_\_\_\_\_ Холодов І. В.

## Додаток Г


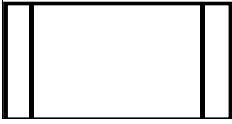


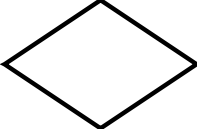

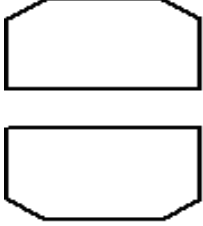
### Символи даних, процесів і ліній

Таблиця Г.1 – Символи даних

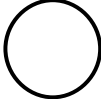

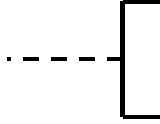

| <i>Основні символи даних</i>                       |   |   |
|--|---|---|
| Дані   |    | Символ відображає дані, носій даних невизначений  |
| Дані, що запам'ятовуються                          |    | Символ відображає дані, що зберігаються, у вигляді, придатному для обробки, носій даних невизначений  |
| <i>Специфічні символи даних</i>                    |   |   |
| Оперативний запам'ятовувальний пристрій            |    | Символ відображає дані, що зберігаються в оперативному запам'ятовувальному пристрої   |
| Запам'ятовувальний пристрій з послідовним доступом |    | Символ відображає дані, що зберігаються в запам'ятовувальному пристрої з послідовним доступом (магнітна стрічка, касета з магнітною стрічкою, магнітофонна касета)                            |
| Запам'ятовувальний пристрій 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)                         |
| Карта  |  | Символ відображає дані, подані на носії у вигляді карти (перфокарти, магнітні карти, карти зі зчитуваними мітками, карти з відривним ярликом, карти зі сканованими мітками).                  |
| Паперова стрічка                                   |  | Символ відображає дані, подані на носії у вигляді паперової стрічки   |
| Дисплей  |  | Символ відображає дані, подані у візуальній людиночитабельній формі на носії у вигляді пристрою відображення (екран для візуального спостереження, індикатори введення інформації)            |



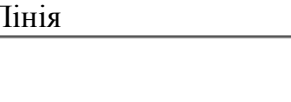
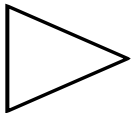


Таблиця Г. 2 – Символи процесу

| <i>Основні символи, процесу</i>   |   |  |
|-----------------------------------|---|--|
| Процес                            |    | Символ відображає функцію обробки даних будь-якого вигляду (виконання певної операції або їх групи, що приводить до зміни значення, форми інформації).   |
| <i>Специфічні символи процесу</i> |   |  |
| 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 реєстра 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

|   |   |
|---|---|
| <p>З'єднувач</p>   | <p>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</p>   |
| <p>Термінатор</p>  | <p>Символ відображає вихід в зовнішнє середовище і вхід із зовнішнього середовища (початок або кінець схеми програми, зовнішнє використання і джерело або пункт призначення даних)</p>  |
| <p>Коментар</p>    | <p>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</p>                           |
| <p>Пропуск</p>     | <p>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</p> |

Таблиця Г. 4 – Символи ліній

|  |  |
|--|--|
| <p><i>Основний символ ліній</i></p>  |  |
| <p>Лінія</p>                | <p>Символ відображає потік даних або управління. У разі необхідності або для підвищення легкості читання можуть бути додані стрілки-покажчики</p>  |
| <p><i>Специфічні символи ліній</i></p>   |  |
| <p>Передача управління</p>  | <p>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)</p> |
| <p>Канал зв'язку</p>        | <p>Символ відображає передачу даних по каналу зв'язку</p>  |
| <p>Пунктирна лінія</p>      | <p>Символ відображає альтернативний зв'язок між двома або більшою кількістю символів, а також використовується для обведення ділянки</p>   |