

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС  
"ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ"  
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ  
"КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО"  
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

## **ПРОГРАМУВАННЯ ТА АЛГОРИТМІЧНІ МОВИ ПРОГРАМУВАННЯ**

### **МЕТОДИЧНІ ВКАЗІВКИ**

**до виконання курсового проекту**

галузь знань 12 інформаційні технології  
122 «Комп'ютерні науки та інформаційні технології»  
124 «Системний аналіз»

*Затверджено Методичною комісією ННК «ІПСА» НТУУ «КПІ»  
Протокол № 7 від 17.05.2017*

Укладачі: Назарчук І.В.  
Швачко Г.Г.

Київ  
ННК «ІПСА» НТУУ «КПІ»  
2017

Програмування та алгоритмічні мови 2. Програмування:Методичні вказівки до виконання курсового проекту для студентів галузі знань 12 «Інформаційні технології» спеціальність 122 «Комп'ютерні науки та інформаційні технології», 124 «Системний аналіз»/Уклад.:І.В.Назарчук, Г.Г.Швачко – К.НТУУ «КПІ», 2017 – 32с.

*Затверджено Методичною комісією  
ННК «ІПСА» НТУУ «КПІ»  
Протокол № 7 від 17.05.2017*

## ПРОГРАМУВАННЯ ТА АЛГОРИТМІЧНІ МОВИ ПРОГРАМУВАННЯ

МЕТОДИЧНІ ВКАЗІВКИ  
до виконання курсового проекту

для студентів галузі знань  
12 «Інформаційні технології»  
спеціальність  
122 «Комп'ютерні науки та інформаційні технології»,  
124 «Системний аналіз»

Укладачі: Назарчук Ірина Василівна  
Швачко Галина Георгіївна

Відповідальний  
редактор О.Л. Тимошук, к.т.н., доцент

Рецензент А.А. Гагарін, к.т.н.

## ЗМІСТ

ВСТУП.....	4
1. ОРГАНІЗАЦІЯ ПІДГОТОВКИ КУРСОВОЇ РОБОТИ.....	5
1.1 Загальні положення.....	5
1.2 Вибір теми курсової роботи.....	5
1.3 Тематика курсових робіт.....	5
1.4 Етапи виконання роботи.....	9
1.5 Графік виконання роботи.....	11
2. СТРУКТУРА ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....	13
2.1 Титульний аркуш.....	13
2.2 Зміст.....	13
2.3 Перелік умовних позначень.....	14
2.4 Вступ.....	15
2.5 Основна частина.....	15
2.6 Висновки.....	17
2.7 Додатки.....	17
2.8 Список використаних джерел.....	17
3. ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....	18
3.1 Загальні вимоги до оформлення роботи.....	18
3.2 Поради щодо оформлення тексту.....	22
4. ЗАХИСТ РОБОТИ.....	23
4.1 Документи, які мають бути підготовлені перед захистом.....	23
4.2 Процедура захисту.....	23
4.3 Критерії оцінювання.....	24
5. ПЕРЕЛІК НАВЧАЛЬНО-МЕТОДИЧНОЇ ЛІТЕРАТУРИ.....	27
ДОДАТКИ.....	28
Додаток А Бланк титульної сторінки курсової роботи.....	28
Додаток Б. Зразок оформлення бланку завдання на курсову роботу.....	29
Додаток В. Зразки таблиць, малюнків, схем.....	30

# 1. ОРГАНІЗАЦІЯ ПІДГОТОВКИ КУРСОВОЇ РОБОТИ

## 1.1. Загальні положення

Курсова робота — це індивідуальне завдання студенту з дисципліни «Програмування», в якому комплексно використовуються знання та вміння, набуті в процесі вивчення дисципліни.

Мета курсової роботи — підвищення та закріплення набутих під час вивчення дисципліни «Програмування» теоретичних знань і практичних вмінь студентів.

Завдання курсової роботи:

- закріплення вмінь самостійно спроектувати загальний алгоритм роботи складної системи, провести необхідні обґрунтування для розробки програмних засобів, підібрати оптимальні алгоритми для реалізації окремих складових, скористатись для цього сучасними джерелами інформації, у тому числі Інтернет-ресурсами;
- закріплення вмінь самостійно вибрати середовище розробки;
- закріплення вмінь самостійно реалізувати побудовані алгоритми програмно, довести роботу до логічного кінця, проаналізувати отримані результати;
- ознайомлення з основними етапами розробки закінченого програмного продукту та основними вимогами до оформлення текстового, програмного та ілюстративного матеріалу звіту до нього.

Даний курсовий проект – це перша самостійна навчально-виробнича розробка студентів першого курсу, яка має бути виконана й оформлена відповідно до сучасних технологій дослідження, практичної реалізації та вимог оформлення супровідної документації.

Знання та вміння, здобуті в результаті виконання курсової роботи, в подальшому застосовуватимуться студентом при виконання наступних курсових та дипломних робіт, індивідуальних навчально-дослідних та навчально-виробничих завдань, наукових досліджень.

## **1.2. Вибір теми роботи**

Студенти самостійно вибирають тему роботи із запропонованих або можуть запропонувати свою власну після обговорення її з керівником, виходячи із власних схильностей, інтересів та знань.

Тематика (перелік можливих тем) курсових робіт затверджується рішенням кафедри і оприлюднюється не пізніше як за 3 місяці до планового терміну їх захисту.

Затвердження тем курсових робіт відбувається на підставі оформлення (разом з керівником) листа Завдання, який подається на кафедру не пізніше як за три місяці до терміну захисту роботи. У разі, якщо студент не оформив в означений термін лист Завдання без поважної причини, тема роботи закріплюється на розсуд кафедри.

Не допускається виконання курсових робіт на однакову тему різними студентами.

У випадку виникнення потреби змінити тему курсової роботи студент подає на кафедру відповідну мотивовану заяву з візою наукового керівника на ім'я завідувача кафедри. Рішення приймається не пізніше ніж за місяць до планового терміну захисту курсових робіт.

## **1.3 Тематика курсових робіт**

Тематика курсових робіт може бути розділена на окремі групи, кожна із яких має свої особливості при реалізації і орієнтована на застосування спеціальних знань з інших дисциплін, що викладаються або вже були викладені на час виконання курсової роботи. Нижче надано приблизний перелік тем курсових робіт по групам і визначено їх особливості. Цей список може бути розширений за узгодженням з керівником.

### **1.3.1. Інформаційно-пошукова система**

Утворити інформаційно-пошукову систему як файл (або зв'язані файли) з записами фіксованої структури будь-якого наповнення, наприклад:

1. Бібліотека.

2. Деканат.
3. Записник.
4. Бухгалтерський облік.
5. Складське господарство.
6. Тлумачний словник з наведенням синонімів та/або перекладу.

Передбачити можливість створення нового запису, редагування вмісту існуючого, видалення, сортування та пошуку записів за однією або декількома ознаками, виведення потрібної інформації у вигляді звіту певного формату.

### **1.3.2. Навчальні програми та програми тестування**

Програми повинні містити частини: теоретичні відомості, власне навчання, тестування з окремого розділу або теми. Приклади тем:

1. Системи числення та робота в них.
2. Навчальна програма роботи з клавіатурою.
3. Навчальна програма роботи в обраному програмному середовищі.
4. Програма тестування знань з обраної області.
5. Психологічні тести.

Психологічні тести повинні містити реалізацію декількох різних тестів, бажано за різними принципами (відповідь на запитання, вибір із можливих варіантів).

### **1.3.3. Спеціалізовані математичні пакети**

Програми повинні містити частини: теоретичні відомості, можливість вибору варіантів розв'язання задачі. Приклади тем:

1. Пакет роботи з векторами та матрицями
2. Пакет розв'язку систем лінійних рівнянь (різними методами)
3. Пакет обчислення визначеного інтегралу (різними методами)
4. Пакет пошуку коренів нелінійного рівняння (різними методами)
5. Розв'язок систем рівнянь з одним невідомим в алгебрі множин
6. Мінімізація булевих виразів
7. Робота з кінцевими решітками
8. Перевірка на функціональну повноту набору булевих функцій

9. Побудова замикання наданого набору булевих функцій
10. Побудова графіку функції із наданого набору із можливістю зміни окремих параметрів
11. Побудова графіку функції за наданим користувачем рядком, що містить припустимий опис функції (синтаксичний аналізатор)
12. Пошук похідної за наданим користувачем рядком, що містить припустимий опис функції (синтаксичний аналізатор з подальшим перетворенням отриманого результату)
13. Калькулятор. За заданим символьним рядком з математичною формулою обчислити значення функції в заданій точці (з використанням польського інверсного запису)
14. Автоматичне доведення теорем (метод семантичних дерев, метод резолюцій з модифікаціями)

#### **1.3.4. Спеціальні структури даних**

Програми повинні містити можливість побудови та візуалізації усїєї структури, а також пошуку, перестановки, видалення окремих елементів або частин. Для деяких програм варто передбачити реалізацію класичних алгоритмів, зв'язаних з даною структурою. Приклади тем:

1. Однозв'язні списки та робота з ними
2. Двобічно зв'язані списки та робота з ними
3. Бінарні дерева та робота з ними
4. Орієнтовані та не орієнтовані графи. Реалізація передбачає визначення, відображення графу та розв'язок однієї або декількох задач із переліку
  - a. пошук всіх можливих шляхів, що зв'язують дві вершини
  - b. пошук найкоротшого шляху, що зв'язує дві вершини
  - c. перевірка графу на зв'язність
  - d. визначення найкоротших відстаней від наданої вершини до інших (алгоритм Дейкстри)
  - e. розмальовка вершин неорієнтованого графа
  - f. алгоритм пошуку компонент зв'язності графу

- g. алгоритм Флері побудови Ейлерова циклу
- h. задача комівояжера

### **1.3.5. Ігрові програми, статичні ігри в графічному або текстовому режимі**

Приклади таких ігор:

1. „П’ятнадцять” – в полі  $4 \times 4$  розставляється 15 чисел без повторень, одне положення пусте. Використовуючи це пусте положення, рухаючи на нього сусідні поля, що містять числа, розмістити числа за зростанням по рядках. В загальному випадку задачу можна реалізовувати для поля  $N \times N$ .
2. „Ханойська вежа” – є три піраміди, на першу із яких вдягнуто кільця (в загальному випадку  $N$ ), причому діаметр кілець зростає згори вниз, а дві інші – порожні. Використовуючи одну із пірамід як допоміжну, перемістити кільця з першої піраміди на останню, при цьому будь-яке кільце можна перемістити на кільце з більшим діаметром. Відобразити послідовність розв’язку задачі на екрані. Задача може розв’язуватись як комп’ютером, так і користувачем.
3. „Вгадати слово” – комп’ютером випадково із наданого словника обирається слово і виводиться його тлумачення та послідовність порожніх клітинок, кількість яких співпадає з кількістю букв у даному слові. За наданий час або визначену кількість кроків гравцю потрібно вгадати слово. Літери, що вже були названі, мають бути помічені. (Гра може бути реалізовано у вигляді „Поля чудес”).
4. „Бики та корови” – комп’ютером випадково задане чотиризначне число без повторень, яке гравцю потрібно вгадати за надану кількість кроків. На кожному кроці комп’ютер повертає інформацію про кількість угаданих цифр, і кількість таких з них, що стоять на своєму місці. (Комп’ютер може одночасно розгадувати число користувача відповідно певної стратегії, яку потрібно розробити самостійно).
5. „Життя” – моделюється життя гіпотетичної колонії клітинок. Клітинка залишається живою, якщо має двох або трьох сусідів, При іншому числі сусідів клітинка гине. Якщо у будь-якої порожньої клітинки рівно три сусіди, то в ній



з'являється нове життя. На полі  $n \times n$  визначити початковий стан колонії та дослідити її розвиток по крокам.

6. „Вірус” – на ігровому полі  $14 \times 14$  клітинок, колір кожної з яких задано випадково з 6 можливих варіантів, потрібно за 24 кроки розповсюдити вірус на усе поле. Гра починається з лівого верхнього кута. За один хід можна перейти на одну сусідню клітинку по горизонталі або по вертикалі. Клітинка, на яку було здійснено хід, стає „зараженою” разом з усіма сполученими з нею однокольоровими клітинками.

7. „Морський бій” – гра реалізується за класичними правилами між гравцем та комп'ютером, який робить свої кроки випадково або використовуючи певну стратегію (можуть бути різні рівні складності).

8. „Реверсі” – ігрове поле має розмір  $8 \times 8$ , в центрі якого на початку розміщуються по діагоналі дві чорних та дві білих фішки двох гравців. На кожному черговому кроці гравець виставляє нову фішку свого кольору, намагаючись „оточити” якомога більше фішок супротивника. Оточені фішки супротивника змінюють колір. Фішки можуть бути оточені у будь-яких горизонтальних та вертикальних напрямках. Гра закінчується, коли не залишається вільних клітин. Виграє той гравець, фішок кольору якого більше. Принципово гра може бути реалізована на полі  $n \times n$ .

9. „Японський кросворд” – завдання може складатись із двох частин: побудови кросворду за наданим малюнком та розв'язку кросворду, запропонованого комп'ютером із відповідної бібліотеки.

10. „Шашки” – реалізує 8-клітинні шашки для двох гравців. Програма повинна перевіряти припустимість ходу.

11. „П'ять в рядок” – хрестики - нулики на „нескінченному” полі. Гравці намагаються поставити в неперервний рядок п'ять хрестиків або нуликів. Гра реалізується або між двома гравцями, тоді комп'ютер повинен перевіряти коректність ходу, або між гравцем та комп'ютером, в цьому випадку комп'ютер повинен реалізувати певну стратегію гри.

12. „Сокобан”. Гру реалізувати за класичними правилами між комп'ютером та гравцем або у будь-якій її інтерпретації.

### **1.3.6. Ігрові програми на випадковий рух графічного об'єкту за наявності перешкод або генерацію випадкових ігрових ситуацій**

15. „Тетріс”. Гру реалізувати за класичними правилами між комп'ютером та гравцем або у будь-якій її інтерпретації.
16. „Діггер”. Реалізувати керований рух графічного об'єкту в лабіринті або іншій ігровій ситуації з можливістю врахування додаткових умов (призи, переслідування, перешкоди і т.ін.)
17. „Теніс” – обмежений рух м'яча та керування ракеткою.
18. „Іподром” – програма реалізує одночасний рух декількох об'єктів з випадковою швидкістю. Можна передбачити випадкову появу інших подій (неподолана перешкода, травма і т.ін.). На початку гри гравець обирає об'єкт, та, якщо він виграє, отримує призові бали.

### **1.4. Етапи виконання роботи**

Кожний студент отримує своє власне індивідуальне завдання, яке повинен виконати протягом семестру.

Виконання студентом курсової роботи з дисципліни «Програмування» включає в себе такі послідовні етапи:

1. Вибір теми роботи.
2. Обговорення з керівником і оформлення бланка завдання з підписами керівника і студента – як закріплення теми індивідуального завдання.
3. Аналіз постановки задачі і вибір методів з використанням відповідних літературних джерел та інтернет-ресурсів.
4. Побудова алгоритму розв'язку задачі. Рекомендований підхід при побудові алгоритму – спадний для побудови загального алгоритму роботи програми як складної структурної системи. Окремо потрібно пропрацювати алгоритми роботи окремих складових системи – підсистем. Алгоритми можуть бути оригінальні – авторські, можна скористатись стандартними класичними алгоритмами. Коректність обраного алгоритму має бути доведена. При побудові загального алгоритму потрібно враховувати можливість легко його удосконалити для збільшення або покращення функціональності.

5. Проектування інтерфейсу. Інтерфейс має бути дружнім до того користувача, на якого розрахований даний програмний продукт. Він має бути добре прокоментований та інтуїтивно зрозумілий для нього. Для будь-якого програмного продукту має бути передбачене основне меню та декілька підменю. В основному меню передбачити пункти допомоги, інформації про розробника, пункт для виходу із програми та інші – відповідно темі.

6. Програмна реалізація. Програма повинна бути реалізована як консольний додаток або як WINDOWS-додаток (з дозволу керівника), представляти собою проект, зібраний з декількох модулів, у тому числі авторських, поданих з використанням заголовкових файлів. Функції, що входять до складу одного файлу, мають бути логічно пов'язані. Окрему функцію бажано будувати таким чином, щоб вона відповідала за якусь одну дію. Користуватися глобальними даними не бажано, оскільки це збільшує контекстну залежність даної функції.

7. Тестування програми і аналіз результатів. Тестування програми варто проводити як на стандартних даних (які апріорі дають очікувані результати), так і на нестандартних даних для забезпечення коректної роботи програми на "крайніх" значеннях (наприклад, усі нулі або свідомо завелике число). Програма повинна бути захищена і забезпечувати стале виконання, тобто не повинна реагувати на натиснення будь-яких клавіш, крім допустимих на даному етапі виконання.

8. Демонстрація програми.

9. Оформлення пояснювальної записки.

10. Здача курсової роботи науковому керівнику на перевірку.

11. захист курсової роботи.

Університетська освіта передбачає оволодіння навичками самостійно ставити і розв'язувати завдання, самостійно працювати з літературою та оволодівати потрібним інструментарієм для реалізації конкретної задачі. Тому більшу частину роботи студент повинен виконувати самостійно, але не втрачати при цьому зв'язку з керівником, головне завдання якого – спрямувати студента на розв'язок конкретних питань з обраної теми, допомогти у пошуку шляхів та методів вирішення проблеми.

Обов'язковими компонентами керівництва виконанням студентами курсових робіт є:

– вступна настанова (групова співбесіда) тривалістю до однієї пари з усіма студентами, які виконують курсову роботу під керівництвом цього викладача;

– щотижнево на консультаціях та лабораторних заняттях студент повинен згідно з графіком звітуватись про виконану роботу, і, в разі потреби, отримати консультацію щодо подальшого виконання курсової роботи.

### **1.5. Графік виконання роботи**

Графік виконання курсової роботи представлений у наведеній нижче таблиці 1.1 і є обов'язковим для виконання усіма студентами.

Таблиця 1.1. Графік виконання курсової роботи з дисципліни «Програмування»

<b>№ п/п</b>	<b>Назва етапів курсового проекту (роботи)</b>	<b>Строк виконання етапів роботи</b>
1.	Вибір теми курсової роботи. Опрацювання відповідної літератури. Оформлення бланка завдання.	1-й, 2-й тижні
2.	Аналіз постановки задачі.	3-й тиждень
3.	Вибір та дослідження методів, вибір відповідних структур даних, розробка алгоритму. Перше узгодження з керівником.	4-й тиждень
4.	Проектування інтерфейсу.	5-й тиждень
5.	Друге узгодження з керівником.	6-й тиждень
6.	Програмна реалізація.	7, 8, 9-й тижні
7.	Демонстрація першого варіанту. Третє узгодження з керівником.	10-й тиждень
8.	Заключне тестування програми.	11-й тиждень
9.	Аналіз результатів. Оформлення звіту.	12, 13-й тиждень
10.	Захист та демонстрація курсової роботи.	14, 15-й тижні

## 2. СТРУКТУРА ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Пояснювальна записка до курсової роботи складається з таких структурних елементів:

- титульний аркуш;
- заповнений бланк завдання;
- зміст;
- перелік умовних позначень (за необхідності);
- вступ;
- основна частина (розділи роботи);
- висновки;
- список використаної літератури;
- додатки.

### 2.1. Титульний аркуш

Титульний аркуш оформлюється виключно згідно наведеного у додатку А зразку.

Тема роботи повинна зазначатись ідентично темі, затвердженій кафедрою. У разі невідповідності робота до захисту не приймається. Прізвище, ім'я та по батькові студента та керівника роботи повинні зазначатись повністю.

Бланк завдання складається з двох сторінок і оформлюється на одному аркуші з двох сторін. Зразок надано у додатку Б.

Поля, заповнені у зразку, можуть бути надруковані. Поля, що передбачають індивідуальну інформацію, потрібно заповнити разом з керівником і поставити свій підпис, а керівник має поставити свій підпис. Заповнити бланк завдання необхідно відповідно графіку виконання роботи.

### 2.2. Зміст

Зміст пояснювальної записки до курсової роботи повинен послідовно містити назви всіх структурних елементів роботи 1-го та 2-го рівнів (окрім

титульного аркуша та самого змісту) та посилання на номери сторінок, на яких починається даний структурний елемент [6].

Візуально зміст роботи повинен відображати ієрархію структурних елементів роботи (перелік умовних позначень, вступ, розділи, висновки, додатки — елементи 1 рівня; підрозділи — елементи 2 рівня).

На початку по центру розміщується текст "ЗМІСТ" (без лапок), оформлений напівжирним шрифтом. Приклад оформлення змісту пояснювальної записки надано в Додатку Г.

### **2.3. Перелік умовних позначень**

Перелік умовних позначень є необов'язковим елементом роботи. Він складається у випадку, коли робота містить маловідомі скорочення, аббревіатури, символи, специфічні терміни.

Перелік друкується двома колонками, в яких ліворуч за абеткою наводять позначення чи терміни, праворуч — їх детальне розшифрування (тлумачення).

Якщо в роботі певний термін, скорочення чи позначення повторюється менше трьох разів, його у перелік не включають, а його розшифрування наводять у тексті при першому згадуванні.

### **2.4. Вступ**

Вступ до курсової роботи повинен містити такі обов'язкові елементи (у такому ж порядку):

- актуальність: чому було важливо та актуально обрати саме таку тему;
- мета ("розробити та реалізувати закінчений програмний продукт для ...");
- завдання ("на основі аналізу літературних джерел [визначити основні риси майбутньої розробки, оцінити існуючі методи та алгоритми реалізації її складових], складено структурний алгоритм роботи майбутньої програми ... та реалізовано його у вигляді програмного продукту ..., що складається з ... модулів, у тому числі стандартних ..., з використанням процедурного програмування");

- практичне значення одержаних результатів (можливо, місце застосування програмного продукту або його частини);
- апробація результатів дослідження (необов'язковий елемент; можливо, результати застосування програмного продукту або його частини);
- використане програмне забезпечення ("При виконанні роботи було використане таке програмне забезпечення: стандартне середовище розробки ..., операційна система ...; веб-браузер ... для роботи з веб-сайтами ...; текстовий редактор ... для підготовки та оформлення курсової роботи");
- структура роботи ("Робота складається зі вступу, трьох розділів, висновків, додатків та списку використаних джерел").

Кожен елемент вступу повинен починатись з нового абзацу. Назви елементів повинні виділятися напівжирним шрифтом.

Загальний обсяг вступу не повинен перевищувати 2 сторінок.

## **2.5. Основна частина**

Основна частина роботи складається з трьох послідовних розділів, які присвячені наступним питанням.

### **2.5.1. Перший розділ**

Перший розділ має назву "ПОСТАНОВКА ЗАДАЧІ". Містить два підрозділи.

Перший підрозділ присвячений огляду існуючих підходів до розв'язання поставленої задачі, аналізу існуючих алгоритмів, варіантів візуалізації (для ігрових програм), тощо.

Другий присвячений обґрунтуванню вибраного варіанту реалізації і детальному опису постановки задачі. Постановка задачі може бути сформульована математично або на змістовному рівні (умови гри для ігрових програм, перелік методів і функцій, що треба реалізувати для інформаційних систем, тощо). Крім того, описуються вимоги до програмних та технічних засобів.



### **2.5.2. Другий розділ**

Другий розділ має назву "РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ" і включає два підрозділи.

Перший підрозділ має назву "Метод розв'язку задачі" і містить виклад рішення, що пропонується. Залежно від теми роботи, цей підрозділ може включати опис необхідного математичного апарату, фізичного явища, обґрунтування вибору обраного алгоритму розв'язку, опис основних структур даних. У цьому підрозділі можуть бути використані такі елементи як формули, рисунки, таблиці і графіки.

Другий підрозділ має назву "Алгоритм розв'язку задачі", містить опис загального алгоритму роботи всієї програми. Може містити описи одного або декількох самостійно розроблених оригінальних або модифікованих стандартних алгоритмів, що реалізують окремі частини загального алгоритму.

У цьому підрозділі обов'язково має бути присутній рисунок зі структурною блок-схемою узагальненого алгоритму роботи програми (1-2 сторінки), у якій виділяються головні функціональні блоки програми та відображається зв'язок між ними. За необхідності, можуть бути представлені також блок-схеми описаних вище оригінальних алгоритмів [9].

### **2.5.3. Третій розділ**

Третій розділ "ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ" присвячений реалізації програмного продукту. Містить опис основної програми і користувацьких програмних модулів у порядку їх оголошення. Складається з таких підрозділів: "Опис головних структур і змінних програми", "Опис головних функцій програми", "Опис інтерфейсу" та "Результати роботи програмного продукту".

Два перших підрозділи можуть бути побудовані у вигляді тексту або таблиць, що містять відповідні описи, згруповані за належністю до певного заголовкового файлу. Обов'язково потрібно надавати опис тих структур та змінних, що винесені у заголовкові файли та таких, розуміння призначення яких може вплинути на розуміння функціонування важливої частини або програми загалом. Опис даних містить тип та семантику.

Обов'язково потрібно надавати опис усіх розроблених функцій, незалежно від розміру та призначення. Опис кожної функції містить її прототип та призначення, а також семантику кожного параметра під час виклику.

Приклади відповідних таблиць представлені у Додатку В.

Підрозділ "Опис інтерфейсу" представляє собою фактично керівництво користувача даного програмного продукту і містить описи усіх реалізованих меню та підменю, ілюстровані рисунками з відповідними екранними формами (можуть бути чорно-білі).

Підрозділ "Результати роботи програмного продукту" містить інформацію про результати тестування програми та порівняльну характеристику з існуючими продуктами такого плану, виявлені недоліки та рекомендації щодо використання. Може містити ілюстрації певних ігрових ситуацій, станів інформаційно-пошукової системи або інших результатів роботи програми.

## **2.6. Висновки**

Висновки та пропозиції є стислим викладенням підсумків дослідження. У даному випадку – висновки по розробленому програмному продукту: що і як вдалось реалізувати, переваги та недоліки розробки, можливі шляхи подальшого покращення. Обсяг висновків становить 1–2 сторінки.

## **2.7. Додатки**

Додаток повинен містити текст основної програми та тексти підключених модулів власної розробки (як заголовкові файли, так і файли реалізації) у порядку їх оголошення. Тексти функцій можуть містити стислі коментарі щодо їх призначення.

## **2.8. Список використаної літератури**

Список використаної літератури повинен містити перелік в алфавітному порядку усіх літературних джерел, які були використані при написанні курсової роботи.

Джерела оформлюються згідно вимог стандарту ДСТУ ГОСТ 7.1:2006 «Система стандартів з інформації, бібліотечної та видавничої справи.

Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання» [8].

### 3. ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Пояснювальна записка (ПЗ) оформлюється згідно ДСТУ 3008-95 (ГОСТ 7.32-91) Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.

#### 3.1. Загальні вимоги до оформлення роботи

Пояснювальна записка друкується у текстовому редакторі на аркушах білого паперу формату А4 лише з одного боку аркушу.

##### 3.1.1. Параметри сторінок

Ліве поле — 3 см, верхнє, нижнє та праве — 2 см.

Всі сторінки роботи нумеруються послідовно, починаючи з номера 1. Номер розміщується у правому нижньому куті аркуша. На першій сторінці (титульному аркуші) та листі завдання номер не ставиться. Тому перша нумерована сторінка матиме номер 4.

##### 3.1.2. Оформлення тексту

Всі текстові елементи записки оформлюються наступним чином: шрифт *Times New Roman*, розмір 14 пунктів, полуторний міжрядковий інтервал, колір — чорний, фон — білий. Інші шрифти дозволяється використовувати лише в ілюстраціях.

Для абзаців тексту, нумерованих та маркірованих списків встановлюється вирівнювання тексту по ширині, абзацний відступ — 5 знаків ( $\approx 1,25$  см).

##### 3.1.3. Зміст

Зміст роботи створюється виключно засобами автоматичного генерування змісту текстового редактора і повинен містити посилання на структурні елементи роботи, включаючи номери сторінок.

Вимоги щодо оформлення елементів змісту:

- номери сторінок вирівнюються вправо, проміжки між назвами елементів та номерами заповнюються крапками;
- для елементів 1 рівня: абзацний відступ відсутній, шрифт нормальний, всі літери прописні;
- для елементів 2 рівня: відступ зліва 0,5–1,0 см, шрифт нормальний;

– не можна використовувати напівжирний, нахилений та інші засоби шрифтового оформлення елементів змісту.

#### **3.1.4. Заголовки**

Всі структурні елементи роботи 1-го рівня (зміст, перелік умовних позначень, вступ, розділи, висновки, додатки, список використаних джерел) повинні починатись з нової сторінки. Структурні елементи нижчих рівнів починаються відразу після завершення попереднього елемента.

Кожен структурний елемент роботи повинен починатись із заголовка — спеціального окремого абзацу, який не може відриватись від наступного за ним абзацу тексту (розташовуватись на іншій сторінці).

Вимоги щодо оформлення заголовків:

– Заголовки 1-го рівня: вирівнювання по центру, текст напівжирний, всі прописні. Абзац починається з нової сторінки, абзацний відступ відсутній. Інтервал після абзацу — не менше 30 пунктів.

– Заголовки 2-го та нижчих рівнів: вирівнювання по ширині, текст напівжирний. Абзацний відступ — ідентичний відступу абзаців тексту роботи. Інтервали перед та після абзацу — по 6 пунктів.

– Крапки в кінці заголовків не ставляться.

Розділи роботи мають бути пронумеровані таким чином: після слова "РОЗДІЛ" ставиться його номер арабською цифрою, після номера крапка не ставиться, потім з нового рядка друкується заголовок розділу.

Підрозділи роботи 2-го рівня нумеруються таким чином: "X.Y. Заголовок підрозділу", де X — номер розділу, Y — номер підрозділу в межах розділу. Після номеру ставиться крапка.

Заголовки 1-го та 2-го рівнів обов'язково повинні бути відображені у змісті роботи.

#### **3.1.5. Таблиці**

Кожна таблиця повинна мати номер та назву, які розміщуються перед таблицею і оформлюються таким чином: спочатку з вирівнюванням вправо вказується номер у форматі "Таблиця X.Y.", де X — номер розділу (дodatку), Y

— номер таблиці в межах розділу (додатку); далі з нового рядка вказується назва, яка вирівнюється по центру (крапка в кінці назви не ставиться).

Після назви з нового рядка розміщується сама таблиця, яка не може відриватись від назви та номеру (розміщуватись на іншій сторінці). Таблиця обов'язково повинна мати заголовки колонок, які описуються у першому рядку таблиці ("шапка" таблиці).

Якщо таблиця не може бути розташована на одній сторінці, то її поділяють на частини, кожна з яких розміщена на окремій сторінці та починається з рядка заголовку з назвами стовпців. Кожній частині таблиці (на початку чергової сторінки) передуює запис "Продовж. табл. Х.У."

Вимоги щодо оформлення тексту у таблицях:

- Абзацний відступ відсутній.
- Текст заголовків та підзаголовків — напівжирний.
- Заголовки потрібно починати з великих літер, підзаголовки — з маленьких, якщо вони складають одне речення із заголовком, і з великих, якщо вони незалежні. Крапки в кінці не ставлять.
- Вертикальне та горизонтальне вирівнювання комірок заголовків — по центру.
- Вертикальне вирівнювання усіх інших комірок — вгору, горизонтальне — вліво.
- Для оформлення великих таблиць у додатках дозволяється зменшувати розмір шрифту (але він не повинен бути меншим від 10 пунктів) та робити одинарний міжрядковий інтервал.

Всі комірки таблиці повинні мати границі у вигляді чорної суцільної лінії товщиною 0,5 пунктів.

Ширина таблиці не повинна бути меншою 50 % та більшою 100 % ширини робочого поля аркуша.

Горизонтальне вирівнювання всієї таблиці — по центру.

Приклади таблиць наведенні у Додатку В.

### **3.1.6. Ілюстрації**

До ілюстрацій належать рисунки, схеми, графіки, діаграми. Кожна ілюстрація повинна бути підписана як рисунок.

Кожен рисунок повинен мати підпис, що містить номер та назву. Він розміщується в окремому рядку відразу після рисунку і має формат "Рис. Х.У. Назва", де Х — номер розділу (додатку), У — номер рисунку в межах розділу (додатку). Підпис не може відриватись від самого рисунку (розміщуватись на іншій сторінці).

Вимоги щодо оформлення рисунків та підписів: вирівнювання — по центру, без абзацу; шрифт нормальний; крапка в кінці назви рисунку не ставиться.

Рисунки і таблиці варто наводити в роботі безпосередньо після тексту за першою згадкою або на наступній сторінці. Якщо вони розміщені на окремих сторінках роботи, включають до загальної нумерації.

У текст курсової роботи повинні бути вставлені як рисунки (див. п. 2.5.3):

- Структурна блок-схема алгоритму роботи програми (обов'язково).
- Блок-схема алгоритму (блок-схеми алгоритмів) функцій, що складають основну розробку автора роботи (не обов'язково).
- Рисунки, що відображують результати роботи програми (наприклад екранні форми з заставкою, головним меню, діалогом з користувачем, тощо).

Приклади рисунків наведені у Додатку В.

### **3.1.7. Додатки**

Додатки повинні мати порядковий номер та назву.

Номер потрібно вирівняти по центру і записати у форматі "Додаток Х", де Х — власне номер (додатки послідовно нумеруються прописними літерами українського алфавіту, починаючи з А, причому Г, Є, І, Ї, Й, О, Ч, Ь не використовують). Якщо робота має один додаток, його потрібно позначити як Додаток.

Назва додатку вирівнюється по центру, починається з нового рядка відразу після номера, містить довільний опис змісту додатку. Шрифт — напівжирний.

Кожен додаток потрібно починати з нової сторінки. Посилання на кожен додаток у зміст роботи включати не потрібно.

### 3.2. Поради щодо оформлення тексту

Пояснювальна записка має бути оформлена відповідно до описаних вище загальних вимог, які є скороченим переказом відповідних стандартів, їх дотримання є обов'язковим.

Описані нижче правила сприятимуть візуально акуратному оформленню тексту. Це:

- не допускати у тексті подвійних пропусків, пропусків перед початком та після абзацу, перед знаками пунктуації (крапками, комами, двокрапками тощо);
- не допускати так званих "висячих" рядків — коли на сторінці розміщується лише один рядок абзацу, а решта — на іншій;
- по всьому тексту роботи використовуються однакові типи маркерів для маркірованих списків, однакові типи лапок (наприклад, «текст», “текст”, „текст” або "текст" – але обов'язково подвійні);
- стиль "*курсив*" у тексті може використовуватись для позначення окремих фрагментів тексту, на які автор хоче звернути увагу. Крім того, курсив може використовуватись для виділення при першому згадуванні прізвищ осіб, що згадуються в роботі, назв організацій чи підприємств та специфічних термінів;
- стиль "підкреслений" у тексті зазвичай не використовується, зокрема, слід уникати його при позначенні посилань на інтернет-ресурси.



## 4. ЗАХИСТ РОБОТИ

### 4.1. Документи, які мають бути підготовлені перед захистом

Перелік матеріалів, які студент повинен подати на кафедру перед захистом курсової роботи:

- електронний варіант пояснювальної записки (у форматі *Microsoft Office Word doc* або *docx*);
- електронний варіант власне програмної розробки курсової роботи (у вигляді окремого каталогу з текстовими файлами модулів з розширенням *.h*, *.cpp*, або інших робочих файлів, необхідних для роботи, та виконуваний *.exe*-файл).
- зброшурований друкований примірник пояснювальної записки до курсової роботи, який повинен містити на титульному аркуші підпис студента, резолюцію "До захисту" та підпис наукового керівника;

Відсутність будь-яких з перелічених документів на кафедрі є підставою для недопущення студента до захисту курсової роботи.

### 4.2. Процедура захисту

Захист курсових робіт проводиться на відкритому засіданні спеціально створеної комісії.

Комісії із захисту курсових робіт подаються усі документи, підготовлені перед захистом (див. п. 4.1) та залікові книжки.

Дозвіл на захист курсової роботи без залікової книжки може бути надано деканом факультету на підставі мотивованої заяви студента.

Процедура захисту передбачає:

- доповідь студента про зміст роботи, який супроводжується демонстрацією розробленого програмного продукту;
- запитання до автора;
- відповіді студента на запитання членів комісії;
- оголошення відгуку наукового керівника;
- заключне слово студента;

– оголошення рішення комісії про оцінку роботи.

Доповідь студент повинен підготувати заздалегідь у формі виступу тривалістю до 10 хвилин, в якому повинні бути висвітлені такі питання:

- обґрунтування актуальності теми;
- мета, завдання, стисла характеристика особливостей реалізації;
- демонстрація роботи програми з усними коментарями;
- висновки по роботі;
- відповіді на основні зауваження наукового керівника;
- висновки щодо набутих і вдосконалених під час виконання роботи умінь.

### **4.3.Критерії оцінювання**

Рейтинг студента складається з балів, які він отримує за самостійне виконання курсової роботи у відповідності до теми, сформульованій у завданні.

Виконання курсової роботи передбачає:

- 1) Створення працюючого програмного продукту, який би
  - відповідав призначенню, визначеному обраною темою,
  - забезпечував потрібну функціональність та мав розвинений інтерфейс для доступу користувача до означеної функціональності на рівні вимог, перелічених у бланку завдання,
  - мав структуру, яка відповідає загальним вимогам до програмного продукту, визначених п.п. 1.3, 1.4 даних Методичних вказівок.
- 2) Оформлення пояснювальної записки до роботи відповідно стандартам ДСТУ 3008-95;
- 3) захист курсової роботи.

Усі етапи виконання роботи повинні виконуватись вчасно у відповідності до затвердженого графіка.

Загальна система балів включає оцінку змісту роботи – програмного продукту (до 60 балів), її оформлення (до 30 балів ) та захисту (до 10 балів).

1. Програмний продукт ( $r_1$ ):
  - «відмінно», відповідає усім вимогам, визначеним під час постановки задачі, повністю забезпечує потрібну функціональність та дружній до

користувача інтерфейс, характеризується стійкістю (захистом від некоректного уведення даних та натиснення непередбачуваних клавіш) – 54-60 балів;

- «добре», може бути допущена незначна неточність в реалізації одної з функцій або незначне зауваження до інтерфейсу (наприклад, недостатньо контрастні кольори переднього плану і фону в одному з віконець), або повна відповідь з незначними неточностями – 45-53 бали;
- «задовільно», загалом працююча програма з суттєвими зауваженнями до функціональності, або неврахування вимоги стійкості коду, або недостатньо розвинений інтерфейс (наприклад, такий, що не дає доступу до усіх функціональних можливостей, або використання термінології, не зрозумілої ймовірному користувачу, або інше подібне) – 36-44 бали;
- «незадовільно», програма не забезпечує роботу більше 40% потрібних функцій або не має інтерфейсу користувача, або виконана не самостійно – 0 балів (цей факт може бути з'ясований під час захисту або співбесіди під час демонстрації працездатності програми).
- за кожний тиждень запізнення з виконанням графіку нараховується (-1) бал до даного пункту.

## 2. Оформлення пояснювальної записки ( $r_2$ ):

- «відмінно», відповідає практично усім (90-100%) вимогам ДСТУ щодо текстового матеріалу, містить потрібний ілюстративний та графічний матеріал, лістинг програмних модулів повний та має змістовні коментарі – 27-30 балів;
- «добре», вимоги ДСТУ виконано на рівні 75-90% , або робота має зауваження щодо повноти забезпечення ілюстративним матеріалом або до якості виконання даного матеріалу – 22-26 балів;
- «задовільно», робота може мати значні зауваження щодо якості текстового матеріалу та повноти забезпечення ілюстративним матеріалом (крім обов'язкових складових) – 18-21 бал;
- «незадовільно», робота не має необхідних складових, або не відповідає більшості вимог стандарту ДСТУ, автору запропоновано переробити

пояснювальну записку і далі діяти відповідно загального графіку перездач заборгованостей;

- за кожний тиждень запізнення зі здачею готової пояснювальної записки оцінка знижується на 5 балів.

3. Захист курсової роботи ( $r_3$ ):

- «відмінно», студент демонструє вільне володіння теоретичним матеріалом при показі готового продукту, відповідає на усі додаткові теоретичні і практичні запитання стосовно своєї роботи – 9-10 балів;
- «добре», під час відповідей на теоретичні питання можливі незначні неточності (до 75%) – 7-8 балів;
- «задовільно», можливі неправильні відповіді, які свідчать про виконання роботи з використанням деяких готових алгоритмів без їх творчого опрацювання – 5-7 балів;
- «незадовільно», студент не може продемонструвати готовий програмний продукт, не може його коректно прокоментувати, що свідчить про роботу у співавторстві - 0 балів.

Максимальна сума балів складає 100.

Студент, що отримав 0 балів з першого або третього пунктів, отримує сумарний рейтинг 0 балів і нову тему курсової роботи.

Сума балів за кожний з трьох складових оцінювання курсової роботи переводиться до залікової оцінки  $R$  згідно формули

$$R = r_1 + r_2 + r_3.$$

## 5. ПЕРЕЛІК НАВЧАЛЬНО-МЕТОДИЧНОЇ ЛІТЕРАТУРИ

1. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы. – М.: Издат. дом «Вильямс». 2000.- 382с.
2. Андерсон Р. Доказательство правильности программ. – М.: Мир, 1982.
3. Фаронов В.В. TurboPascal 7.0 Учебное пособие. М.: «Нолидж», 2001.
4. Швачко Г.Г. Основи алгоритмізації та програмування. Конспект лекцій [Електронний ресурс]//Електронний кампус НТУУ «КПІ» [сайт]/Єдине інформаційне середовище НТУУ «КПІ», 2013.–Режим доступу: <http://campus.kpi.ua>
5. Програмування: Метод. вказівки до самостійної роботи студентів напряму підготовки 6.050101 «Комп'ютерні науки» / Уклад.: І.В. Назарчук, О.Л. Тимощук – К. ННК «ІПСА» НТУУ «КПІ», 2012. – 23 с.
6. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення: ДСТУ 3008-95 (ГОСТ 7.32-91). – Чинний від 2006-01-01. К.: Держспоживстандарт України, 2006.– 231с. – (Національний стандарт України).
7. ДСТУ ГОСТ 7.1.2006 «Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання».
8. ДСТУ ГОСТ 7.1:2006 библиографический список, примеры оформления//Известия вузов. Радиоэлектроника. — Режим доступа: <http://radio.kpi.ua/author/rules/11-author-new-biblio>. — Дата доступа: 25.04.10. — Название с экрана.
9. Схеми алгоритмів і програм. Правила виконання: ДСТ 19.002-80 ЕСПД.
10. Види програм і програмних документів: ДСТ 19.101-77 ЕСПД.

Додаток А Бланк титульної сторінки курсової роботи

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС  
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»  
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

## КУРСОВА РОБОТА

з дисципліни

Програмування

на тему: надрукувати свою тему

---

Студента (ки) 1 курсу групи *КА-ХХ*  
Спеціальність *124 Системний аналіз*.  
*Петренко Петро Петрович*

Керівник старший викладач *Іваненко І.І.*

Національна оцінка \_\_\_\_\_  
Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_  
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)  
\_\_\_\_\_  
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)  
\_\_\_\_\_  
(підпис) (вчене звання, науковий ступінь, прізвище та ініціал)

Київ – 20ХХрік

Додаток Б. Зразок оформлення бланку завдання на курсову роботу

<b>ННК „ІПСА” НТУУ „КПІ”</b>					
(назва вищого закладу освіти)					
Кафедра	<i>Математичних методів системного аналізу</i>				
Дисципліна	<i>Програмування</i>				
Галузь знань	<i>12 Інформаційні технології</i>				
Курс	<i>перший</i>	Група	<i>КА—ХХ</i>	Семестр	<i>другий</i>

**ЗАВДАННЯ  
на курсовий проект(роботу) студента**

(прізвище, ім'я, по батькові)	
1. Тема проекту(роботи)	
2. Строк здачі студентом закінченого проекту(роботи)	<i>ХХ.05.20ХХ р.</i>
3. Вихідні дані до проекту(роботи)	
4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)	
<i>1. Постановка задачі.</i>	
<i>2. Метод розв'язку задачі</i>	
<i>3. Загальна блок-схема алгоритму та опис алгоритму</i>	
<i>4. Опис програмного продукту.</i>	
<i>5. Результати роботи.</i>	
<i>6. Висновки.</i>	
<i>7. Список використаної літератури.</i>	
<i>Додаток А. Текст програми.</i>	
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)	
<i>1. Загальна блок-схема алгоритму.</i>	
<i>2. Ілюстрації роботи програми.</i>	
6. Дата видачі завдання	

## КАЛЕНДАРНИЙ ПЛАН

№/п	Назва етапів курсового проекту (роботи)	Строк виконання етапів роботи	Примітка
1.	Вибір теми курсової роботи. Опрацювання літератури. Оформлення листа Завдання.	2,3-й тиждень лютого	
2.	Аналіз постановки задачі. Узгодження з керівником попереднього плану роботи	3,4-й тиждень лютого	
3.	Вибір та дослідження методів та структур даних, Розробка загального алгоритму	1-й тиждень березня	
4.	Перше узгодження з керівником.	1-й тиждень березня	
5.	Проектування інтерфейсу. Розробка алгоритмів окремих блоків.	2-й тиждень березня	
6.	Друге узгодження з керівником.	2-й тиждень березня	
7.	Програмна реалізація.	3,4-й тижні березня	
8.	Демонстрація першого варіанту. Третє узгодження з керівником.	1-й тиждень квітня	
9.	Доопрацювання програми. Заключне тестування програми.	2,3-й тижні квітня	
10.	Аналіз результатів. Оформлення звіту.	до 2-го тижня травня	
11.	Захист та демонстрація курсової роботи.	до ХХ.05 – «А» до ХХ.05 – «В,С» до ХХ.05 – «D,E»	

Студент	
	(підпис)

Керівник		
	(підпис)	(прізвище, ім'я, по батькові)

	(дата)



## Додаток В. Зразки таблиць, малюнків, схем

Таблиця 6.1 Приклад оформлення таблиці з описом функцій <stdio.h>

№	Синтаксис	Опис
1.	<code>FILE * fopen(char * filename, char * type);</code>	Відкриває файл <i>filename</i> і пов'язує його з відповідним потоком <i>stream</i> . Функція як результат повертає адресний вказівник, який буде ідентифікувати потік <i>stream</i> в подальших операціях за умови успішного завершення, інакше - нуль ( <i>NULL</i> ).
2.	<code>int feof(FILE * stream);</code>	Це макрокоманда, яка проводить перевірку даного потоку <i>stream</i> на ознаку кінця файлу ( <i>EOF</i> ). Повертає ненульове значення, якщо при останній операції уведення було знайдено кінець файлу, інакше – 0.
3.	<code>int fseek(FILE * stream, long offset, int fromwhere);</code>	...

Деякий текст, де міститься посилання на рисунок 6.1.

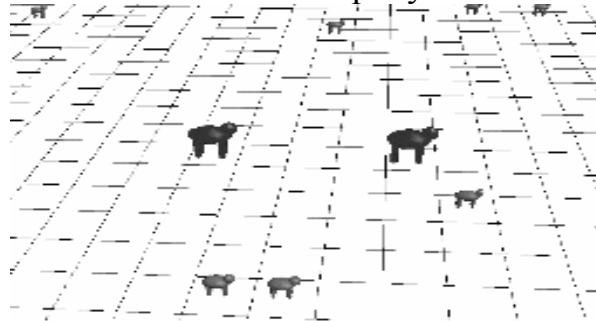


Рис. 6.1. Приклад життєвого простору у моделі «Хижак-Жертва»

Деякий текст, де міститься посилання на рисунок 6.2.

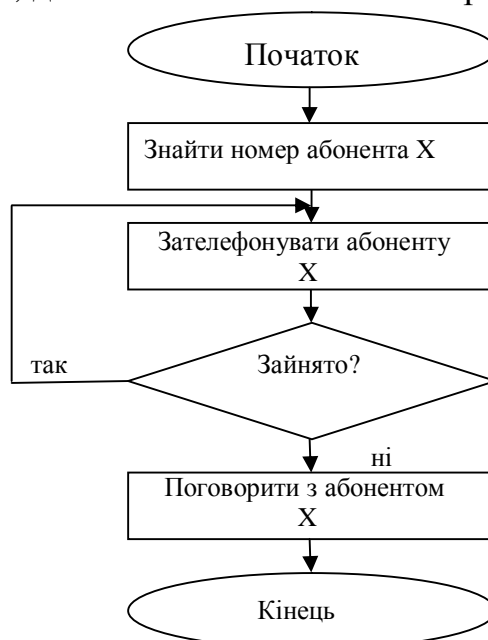


Рис. 6.2. Блок-схема алгоритму розмови з абонентом X

## Додаток Г. Приклад оформлення змісту

### ВСТУП

#### РОЗДІЛ 1 Постановка задачі

- 1.1 Огляд існуючих підходів до розв'язання поставленої задачі
- 1.2 Уточнена постановка задачі на розробку програмного забезпечення

#### РОЗДІЛ 2 Розробка програмного продукту

- 2.1 Метод розв'язку задачі
- 2.2 Алгоритм розв'язку задачі

#### РОЗДІЛ 3 Опис розробленого програмного продукту

- 3.1 Опис головних структур і змінних програми
- 3.2. Опис головних функцій програми
- 3.3. Опис інтерфейсу
- 3.4. Результати роботи програмного продукту.

### ВИСНОВКИ

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

#### Додаток А Текст програми