

Міністерство освіти і науки України  
Харківський національний автомобільно-дорожній університет  
Кафедра інформаційних технологій та мехатроніки

**МЕТОДИЧНІ ВКАЗІВКИ**  
**до виконання курсового проекту**

**з навчальної дисципліни** Програмування  
(назва навчальної дисципліни згідно навчального плану)

**підготовки** Бакалавра, 1-й курс  
(назва освітньо-кваліфікаційного рівня)

**галузі знань** 0502 Автоматика та управління  
(шифр і назва галузі знань)

**напряму підготовки** 6.050201 Системна інженерія  
(шифр і назва напряму підготовки)

**спеціальності** 6.050201 Комп'ютерні системи управління  
(шифр і назва кваліфікації для бакалавра, спеціальності - для магістра)

рухомими об'єктами

розділ “Структурне програмування на мові C++  
у середовищі Microsoft Visual Studio 2010”

Харків, 2016

**Розроблені та внесені:** кафедрою Інформаційних технологій та мехатроніки

**Розробник:** доцент кафедри Інформаційних технологій та мехатроніки  
канд. техн. наук Симбірський Г. Д.

Обговорено та рекомендовано до затвердження на засіданні кафедри  
Інформаційних технологій та мехатроніки.

Протокол № 19 від 27 травня 2016 р.

Завідуючий кафедрою к.т.н. \_\_\_\_\_ Подоляка О.А.

**“Узгоджено”**

Декан факультету Комп’ютерних технологій та мехатроніки ТЗ  
Професор \_\_\_\_\_ Левтеров А.І.

27 травня 2016 р.

## ЗМІСТ

ВСТУП .....	4
1. ЦІЛІ ТА ЗАВДАННЯ КУРСОВОГО ПРОЕКТУ .....	5
2. ПОРЯДОК ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ .....	6
2.1. Формулювання вимог до програми .....	6
2.2. Проектування програми .....	6
2.3. Кодування програми .....	6
2.4. Тестування програми .....	7
2.5. Пояснювальна записка .....	7
3 ВМІСТ РОЗДІЛІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ ДО КУРСОВОГО ПРОЕКТУ .....	7
3.1. Перелік необхідних розділів пояснювальної записки.....	7
3.2. Рекомендації по створенню розділів пояснювальної записки.....	8
4. ВИМОГИ ДО СТВОРЮВАНОВОГО ПРОГРАМНОГО ПРОДУКТУ ..	12
5. ВИМОГИ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ ДО КУРСОВОГО ПРОЕКТУ .....	14
5.1. Загальні вимоги .....	14
5.2. Структурні елементи та розділи .....	15
5.3. Заголовки .....	15
5.4. Нумерація сторінок .....	15
5.5. Нумерація розділів та підрозділів .....	15
5.6. Переліки .....	16
5.7. Ілюстрації .....	16
5.8. Таблиці .....	16
5.9. Формули і рівняння .....	16
5.10. Посилання .....	17
5.11. Скорочення .....	17
5.12. Перелік посилань .....	17
5.13. Додатки .....	17
6. ЗАХИСТ КУРСОВОГО ПРОЕКТУ ТА ЙОГО ОЦІНКА .....	18
7. ПЕРЕЛІК ДЖЕРЕЛ, ЩО РЕКОМЕНДОВАНІ ДЛЯ ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ .....	19
ДОДАТОК А. Бланк титульного аркушу курсового проекту .....	20
ДОДАТОК Б. Зразок листа завдання на курсову роботу .....	21
ДОДАТОК В. Приклад змісту реферату .....	22
ДОДАТОК Д. Приклад оформлення пояснювальної записки до курсвого проекту .....	23

## ВСТУП

Зміни, що відбуваються сьогодні в різних сферах професійного і суспільного життя, пов'язані перш за все з широким впровадженням засобів інформаційних та комунікаційних технологій. Вони вимагають від фахівця у галузі комп'ютерних наук вміння розробляти програмне забезпечення для розв'язування задач науки, техніки, економіки і управління; створювати і використовувати інформаційні моделі процесів і явищ; використовувати інформаційні технології у проектно-конструкторській, управлінській та фінансовій діяльності.

Однією з дисциплін, що формують теоретичну і практичну базу у професійній підготовці майбутніх системних інженерів, є програмування.

Курсовий проект (КП) - це перша самостійна наукова праця майбутнього системного інженера. При її виконанні студент поглиблює знання з фундаментальних та професійно-орієнтованих дисциплін, засвоює методику експериментальних досліджень, оволодіває навичками співставлення результатів своїх досліджень із літературними даними, аналізу, узагальнення і літературного оформлення одержаних результатів з теми дослідження, набуває вміння вести науковий пошук, що розвиває в молодого інженера творчий підхід до роботи.

Головна виробнича мета проекту - створення якісного робочого програмного забезпечення. Педагогічна мета роботи над проектом - творче застосування всіх знань і умінь, отриманих студентом за час навчання, в процесі вирішення реальної завдання щодо створення ПЗ. Мета роботи над курсовим проектом, що має дослідницький характер, - поглиблення знань з програмування. Структура курсового проекту повинна відповідати стандартам і правилам, прийнятим для НДР (науково-дослідних робіт).

У КП відсутні такі найважливіші етапи створення програмного забезпечення, як економічне обґрунтування та складання бізнес-проектів, враховуючи обмеженість розв'язуваних завдань і невеликі розміри програмного продукту, створюваного в рамках курсового проектування.

Всі етапи створення програмного забезпечення в обов'язковому порядку документуються і складають основу пояснювальної записки (ПЗ) до курсового проекту. Проектні рішення, тобто пакет виконуваних програм, вихідний код та блок-схема алгоритма оформляються як додатки до ПЗ.

У цьому методичному посібнику описані елементи структури курсового проекту і дані рекомендації по його створенню:

- Загальний порядок роботи над проектом.
- Складові частини проекту і вимоги до них.
- Структура і зміст Пояснювальної записки і входять до неї документів.
- Стандарти на оформлення документації.
- Рекомендації щодо організації робіт над створенням програмного забезпечення і теоретичними дослідженнями.
- Рекомендації з використання засобів та інструментів створення ПЗ.

Методичний посібник є документом, який регламентує роботу викладачів і студентів над курсовим проектом з дисциплін "Програмування" (розділ "Структурне програмування на мові С++ у середовищі Microsoft Visual Studio 2010") для спеціальності 7.05020101 "Комп'ютеризовані системи управління та

автоматика” напряму підготовки 6.050201 “Системна інженерія”. Студент повинен при виконанні курсового проекту використати набуті теоретичні знання для вирішення конкретних прикладних задач. Основна мета виконання даного курсового проекту - закріплення на практиці вміння використовувати математичні методи розрахунків і методи проектування програм та на базі розроблених алгоритмів вміти розробляти код програми, використовуючи інтегроване середовище мови програмування (редагування, компілювання, виконання), та отримувати результати роботи програми.

## 1. ЦІЛІ ТА ЗАВДАННЯ КУРСОВОГО ПРОЕКТУВАННЯ

Курсовий проект з дисципліни “Програмування” (перший розділ “Структурне програмування на мові C++ у середовищі Microsoft Visual Studio 2010”) передбачає виконання комплексного завдання науково – дослідницького характеру.

Метою курсового проектування є закріплення, поглиблення та узагальнення знань, якими студент оволодів під час вивчення цієї частини курсу, в набутті навичок використання основ алгоритмізації та програмування на алгоритмічних мовах високого рівня з використанням принципів об’єктно-орієнтованого проектування та програмування, методики розробки програмних модулів та програмного забезпечення комп’ютерних систем управління, набуття студентами теоретичних знань та практичних навичок в області використання сучасних систем візуального проектування програмних засобів комп’ютерних систем управління та освоєння принципів та методів сучасних технологій програмування, набуття навичок професійних та практики наукових досліджень з оформленням відповідного текстового, програмного та ілюстративного матеріалу у формі наукового звіту.

В результаті вивчення дисципліни та роботи над курсовим проектом студенти повинні **знати**:

- засоби мови, синтаксис, семантичні і функціональні властивості C++;
- основи роботи з прикладними програмами;
- етапи підготовки і рішення задач на комп’ютері;
- типові конструкції алгоритмів;
- принципи структурного та процедурного програмування;
- правила використання функцій;
- принципи консольного та файлового вводу-виводу даних;
- основні засоби розробки програм з використанням зовнішніх файлів;

### **вміти:**

- на базі опису умови задачі, використовуючи математичні методи розрахунків та методи проектування програм розробляти алгоритми на сучасній мові в умовах комп’ютеризованого робочого місця;

- на базі розроблених алгоритмів вміти розробляти код програми, використовуючи інтегроване середовище мови програмування (редагування, компілювання, виконання) та отримувати результати роботи програми.

Під час виконання курсового проекту студент повинен продемонструвати:

- вміння збирати і аналізувати відповідні матеріали про об’єкт дослідження, використовуючи сучасні джерела інформації, включаючи Інтернет ресурси;

- спроможність проводити необхідні обґрунтування для розробки програмних засобів систем управління різного призначення, тощо;
- здатність доводити розв'язання поставленої задачі до логічного кінця;
- вміння аналізувати отримані результати і робити з них висновки.

Курсовий проект є самостійною роботою студента.

Відповідальність за правильність аналітичних висновків, результатів розрахунків і моделювання, а також оформлення несе студент, що є автором КП.

## **2. ПОРЯДОК ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ**

Основні етапи виконання курсового проекту:

- а) формулювання вимог до створюваної програми;
- б) проектування програми;
- в) кодування програми;
- г) тестування програми;
- д) оформлення пояснювальної записки;
- е) захист роботи.

### **2.1. Формулювання вимог до створюваної програми.**

Формулювання вимог є самостійним етапом роботи і завершується створенням двох документів: індивідуального плану проекту та розділу пояснювальної записки “Специфікація програми”.

Індивідуальний план складає студент на початку роботи над проектом и копію плану у той же час надає викладачу. За суттєве відставання від календарного плану викладач повинен знижувати оцінку курсової роботи.

На етапі формулювання вимог студент повинен:

- 1) сформулювати призначення програми;
- 2) визначити перелік функцій, які виконує програма;
- 3) запропонувати інтерфейс користувача, який надає можливість останньому дістатися цих функцій.

Специфікація програми складається з послідовного опису усіх виділених функцій, та пов'язаного з ними графічного інтерфейсу користувача. У специфікації програми треба обов'язково надати зображення усіх вікон програми.

### **2.2. Проектування програми.**

На етапі проектування визначається архітектурний стиль, структура програми, а також формат похідних, вихідних та внутрішніх даних. Співвідношення між блоками та частинами програми відображується за допомогою блок-схеми алгоритму вирішення задачі. Під час проектування можна розробити план функціонального тестування, та необхідні тести.

Завершується проектування створенням проектної специфікації програми.

### **2.3. Кодування програми.**

На етапі кодування створюється код програми. Він повинен бути гідним чином прокоментований і повністю задовольняти вимогам до програмного коду, наведеним у додатку цих вказівок.

Увесь код програми має бути наведений у додатку до пояснювальної записки. У тексті основних розділах записки можливо наводити лише невеликі за розміром окремі фрагменти коду.

#### **2.4. Тестування програми.**

Розрізняють два основних різновиду тестування: функціональне та модульне. Функціональне тестування перевіряє, як програма виконує свої функції в умовах коректних та некоректних дій користувача. Воно планується на етапі проектування і виконується у міру того, як кодуються окремі функції програми і перед захистом програми.

Модульне тестування робиться одночасно з кодуванням і має за мету оперативну перевірку окремих частин програми. Виконання модульного тестування вельми бажано, але не є обов'язковим.

#### **2.5. Пояснювальна записка.**

Оформлення пояснювальної записки є останнім етапом виконання курсової роботи і має надати студентові навичок документування програмного продукту.

### **З ВМІСТ РОЗДІЛІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ ДО КУРСОВОГО ПРОЕКТУ**

#### **3.1. Перелік необхідних розділів пояснювальної записки**

Пояснювальна записка до курсового проекту повинна містити наступні складові елементи:

**Титульний лист.**

**Завдання на курсову роботу.**

**Реферат.**

**Зміст.**

**Вступ.**

**Розділ 1. Обґрунтування алгоритму й структури програми.**

**1.1. Постановка і математичний опис задачі.**

**1.2. Опис програми, визначення та опис функцій.**

**1.3. Розробка та обґрунтування алгоритму й структури програми.**

**Розділ 2. Розробка програми.**

**2.1. Специфікація програми.**

**2.2. Розробка програми.**

**2.3. Проектування і опис інтерфейсу користувача.**

**2.4. Файлова структура проекту. Опис файлів.**

**Розділ 3. Тестування програми і результати її виконання.**

**Висновки.**

**Література.**

**Додаток А. Блок-схема алгоритму програми.**

**Додаток Б. Лістинг (код з коментарями) програми.**

**Додаток В. Графічна інтерпретація результатів.**

## **3.2. Рекомендації по створенню розділів пояснювальної записки**

### **3.2.1. Титульний аркуш**

Титульний аркуш оформляється відповідно до загальних університетських вимог і містить назви міністерства, університету, факультету і кафедри, тему проекту, групу і прізвище студента, посаду і прізвище керівника, рік виконання проекту.

Приклад оформлення титульного листа наведено у додатку А. Титульний лист має бути підписаний студентом та керівником курсового проектування.

### **3.2.2. Завдання**

Завдання оформляється на стандартному бланку або друкується за тією ж формою (див. додаток Б).

### **3.2.3. Реферат**

Реферат містить коротку характеристику обсягу проекту (кількість сторінок, таблиць, рисунків, літературних джерел, додатків), перелік ключових слів і анотацію до проекту (див. додаток В).

У тексті реферату коротко описують прийняті в проекті технічні, організаційні, економічні рішення, а також пристрої, їх ефективність і міра впровадження. У лаконічній формі має бути представлений опис основних частин проекту. У формі, що стверджує, наголошується, що було виконано, розраховано, сконструйовано, запропоновано, модернізовано і так далі. Реферат рекомендується складати після написання пояснювальної записки проекту.

Перелік ключових слів повинен характеризувати зміст реферованої записки і включати від 5 до 15 ключових слів в називному відмінку, викладених в рядки через коми.

Ключові слова в сукупності повинні поза текстом давати достатньо повне уявлення про зміст записки і забезпечувати можливість інформаційного пошуку.

Обсяг реферату не більше 2000 знаків і не повинний перевищувати одного листа. Приклад змісту реферату наведено у додатку В.

Реферат є першим текстовим листом пояснювальної записки виконується на листі формату А4 з основним написом згідно ДСТ 2.104-68 ЄСКД чи за формою 5 ДСТ 21.101 СПДС і вважається третім листом пояснювальної записки.

### **3.2.4. Зміст**

Зміст є обов'язковим розділом пояснювальної записки курсового проекту, який включає назву розділів і підрозділів записки (починаючи зі вступу), список використаної літератури, додатки і їх найменування, а також номер сторінки, на якій розміщений або починається розділ або додаток. Зміст записки розміщують на першому (заголовному) листі розділу, що має основний напис, згідно ДСТ 2.104-68 (додаток Г). Зміст поміщають після титульного листа і реферату і включають в загальну чисельність листів пояснювальної записки.

### **3.2.5. Вступ**

Вступ характеризує актуальність теми, сучасний стан області знань, що досліджується та коротку характеристику проекту (1-2 с.).



### **3.2.6. Постановка і математичний опис задачі**

Постановка задачі – це словесне її формулювання, що береться з індивідуального завдання. Тут можна вказати припущення, при яких вирішується задача, початкові дані, порядок обробки результатів тощо.

Математичний опис задачі – це перелік математичних виразів, формул, рівнянь, констант та математичних позначок, за допомогою яких можуть бути отримані результати рішення задачі. Це математичні співвідношення, що приведені в індивідуальному завданні, а також ті, що виконавець проекту використав з технічної літератури.

Основною метою даного розділу пояснювальної записки є аналіз вимог завдання на курсову роботу й формулювання додаткових технічних вимог, які безпосередньо впливають з нього, та мети дослідження.

На основі аналізу потрібно конкретизувати основні параметри програми, її можливості та функції, засоби їх реалізації, інтерфейс із користувачем, середовище, в якому програма може працювати.

Якщо у попередньому розділі лише формулюють загальну методику розв'язування задач, то тут визначають конкретний план роботи. Просте переписування вимог завдання на курсову роботу не допускається, потрібно творчо осмислити його та сформулювати конкретні технічні вимоги і шляхи їх досягнення.

Обсяг розділу — 1..2 сторінки.

### **3.2.7. Опис програми, визначення та опис функцій**

Опис програми – це відомості про логічну структуру і функціонування програми. Опис програми містить в собі:

1) загальні відомості (позначення і найменування програми; програмне забезпечення, необхідне для функціонування програми; мова програмування, на якому написана програма);

2) функціональне призначення (призначення програми; відомості про функціональні обмеження);

3) опис логічної структури (алгоритм рішення задачі; методи, що використовуються; структура програми з описом функцій складових частин і зв'язку між ними; зв'язки програми з іншими програмами). Опис логічної структури програми виконуються з урахуванням тексту програми на початковій мові;

4) технічні засоби (типи ЕОМ і пристроїв), які використовуються при роботі програми;

5) виклик і завантаження програми (способи виклику програми, послідовність роботи; повідомлення, які видаються програмою);

6) початкові дані (характер, організація і попередня підготовка початкових даних; формат, опис і спосіб кодування даних);

7) проміжні дані і результати (характер і організація даних; формат, опис і спосіб їх кодування).

При описі програми допускається зміст підрозділу ілюструвати прикладами пояснень, таблицями, схемами, графіками.

Крім того, в цьому підрозділі повинен бути приведений перелік розроблених виконавцем курсового проекту функцій, що використані для декомпозиції задачі

при її розв'язанні. В цьому переліку повинні бути вказані задачі, що вирішуються за допомогою цих функцій.

### 3.2.8. Розробка та обґрунтування алгоритму й структури програми

У цьому підрозділі звіту робляться визначення послідовності дій, визначаються етапи рішення задачі, які потребують розгалужень, циклів, оформлення процедур або функцій.

При використанні функцій обов'язково вказується призначення функції і її математичне зображення. Наприклад, функція обчислювання факторіалу  $Y=X!$

Крім того, вказується метод рішення задачі, рекурентні формули тощо. Наприклад, обчислювання факторіалу зводиться до обчислювання добутку:

$$Y = X! = 1*2*3*...*X = \prod_{j=1}^X j;$$

Обчислювання добутку виконується за наступними рекурентними співвідношеннями:

1.  $Y = 1;$
2.  $Y = Y*j;$

Розробляти алгоритм буде набагато легше, коли вже є визначений зміст і структура файлів початкових даних і результатів.

Файли початкових даних і результатів повинні бути наявними і прості за структурою. Їх логічна побудова повинна відповідати змісту задачі, структурам і типам даних тощо.

Після цього розробляється блок-схема алгоритму рішення задачі з використанням символів, визначених ДЕСТ 19.002-90, 19.003-90. Схеми алгоритмів розробляються для кожного додаткового блоку (функції) і для основної програми.

При розробці алгоритму треба використовувати покроковий спосіб побудови алгоритму.

На першому кроці розробляється загальна схема вирішення задачі, тобто на першому кроці треба відповісти на запитання, що зробити, щоб поставлена задача була вирішена. На другому кроці переходять до деталізації кожного блоку загальної схеми, тобто з'ясувати і вирішити, як реалізувати кожний блок загальної схеми, щоб задача була вирішена.

Результатом другого (третього, четвертого тощо) кроку буде детальний алгоритм (один блок-одна дія) вирішення задачі.

У пояснювальній записці на аркушах форматом А4 відображається схема алгоритму, яка розроблена на останньому кроці.

Обсяг розділу - 2-8 сторінок.

### 3.2.9. Специфікація програми

Цей підрозділ важливе значення для розробки програми вирішення задачі. У ньому повинно бути здійснено присвоєння усім об'єктам задачі імен згідно правилам алгоритмічної мови C++. Крім того, необхідно визначити, які дані будуть початковими, які проміжними, а які – результатами.

При розв'язанні задач розробник повинен визначити, які дані, що приведені в задачі, доцільно використовувати як константи, а які дані – як змінні. Треба добре з'ясувати собі, що для того, щоб програма була універсальною, тобто дозволяла обчислювати різні ситуації, необхідно використати змінні, а не константи. Як правило, у якості константи використовуються фізичні, математичні, логічні константи. При цьому, якщо призначення константи не може бути представлено латинською літерою, то треба знайти у латинському алфавіті або близьку по звучанню літеру, або сполучення літер, яке б несло у собі сенс цієї константи.

Далі треба визначити, чи потребується при вирішенні задачі застосування нестандартних типів даних. Якщо потребується, то треба визначити імена цих типів і їх структуру.

Після цього слід перейти до опису змінних, тобто розробити специфікацію на програму (функцію). Опис змінних доцільно розподілити на опис початкових, проміжних і змінних, які характеризують результати вирішення задачі. Приклад специфікації програми наведений у таблиці 1.

Коли при вирішенні задачі застосовуються функції, то їх також треба перелічити і пояснити:

- формальні параметри і їх форма застосування (параметр-значення або параметр-змінна);
- спосіб застосування змінних, які використовуються у процедурі або функції (локальні або глобальні).

У таблиці 1 для прикладу наведено специфікацію деякої програми.

Таблиця 1. Специфікація програми

Призначення даних	Математичне позначення	Ім'я у програмі	Структура даних	Тип даних	Сенс змінної
Початкова	$\alpha$	Alfa	проста	float	Аргумент функції
Початкова	$\pi$	PI	проста	float	Константа
Проміжна	$i$	I	проста	int	Параметр циклу
Результат	$x(i)$	ArrayX	масив	float	Опір в ланцюзі

### 3.2.10. Розробка програми

Цей підрозділ є основним у курсовій роботі. Його обсяг – 5-10 сторінок.

У залежності від обраного стилю програмування, можливі різні підходи до розробки програми. У найпростіших випадках вона йде послідовно, алгоритм програми реалізується за допомогою послідовності операторів, включно з умовними та операторами циклу.

У структурному програмуванні застосовується ієрархічний підхід, у якому окремі структурні одиниці перетворюються на функції, що можуть приймати параметри та повертати результат. Опис функції в такому випадку виконується один раз у певному місці програми, а використовується вона багато разів, при цьому

лише змінюються параметри, і не потрібно щоразу повторювати одні й ті ж оператори, які включені у функцію.

### **3.2.11. Опис файлів даних та інтерфейсу програми**

Якщо програма використовує файли як джерело вхідних даних або для зберігання проміжних чи кінцевих результатів роботи, то в даному підрозділі слід навести опис формату цих файлів. Він може бути виконаний у текстовому, табличному чи графічному вигляді.

Можна додавати до нього приклади реальних файлів із даними програми.

### **3.2.12. Тестування програми і результати її виконання**

У даному розділі треба описати методику тестування програми, тестові дані та навести результати роботи програми. Якщо програма працює в графічному режимі, то слід роздрукувати копію графічного вікна програми. Якщо результатом роботи програми є текстовий файл, то необхідно вивести вміст цього файлу та вміст неосновних діалогових вікон можна подати в текстовому вигляді.

Якщо для відображення роботи програми необхідна значна кількість роздруківок, то їх можна подати в додатках.

Обсяг розділу — 1..5 сторінок. У розділі потрібно зробити висновок, який підтверджує (або заперечує) працездатність програми.

### **3.2.13. Висновки**

У висновках по КП в реферативній формі повинні бути описані результати, отримані студентом на кожному з етапів виконання роботи (аналітичному, етапу проектування ПЗ, експериментальному дослідженню, аналізу отриманих результатів), а також висновків щодо досягнення мети курсового проектування, перспективи розвитку даної галузі тощо. Висновки пишуться на 1-2 сторінки і як розділ не нумеруються.

### **3.2.14. Література.**

В цьому розділі наводиться перелік джерел, на які були посилання в тексті.

Список повинен формуватися в порядку посилань за текстом і вміщувати бібліографічні відомості офіційно виданих книжок, статей, патентів, депонованих рукописів тощо.

Як розділ перелік літератури не нумерується. Формат опису літературних джерел повинен відповідати ДСТ 7.1.-84.

### **3.2.15. Додатки**

В цей розділ включають блок-схеми програм та їх лістинги, а також інші документи. Крім цього, в додатки поміщуються таблиці, графіки та методики, які з якихось причин не увійшли до пояснювальної записки, але потрібні для пояснень.

## 4. ВИМОГИ ДО СТВОРЮВАНОВОГО ПРОГРАМНОГО ПРОДУКТУ

Незалежно від теми курсового проекту програма, що розробляється, повинна задовольняти наступним загальним вимогам.

**Стійкість програми.** Програма не повинна втрачати працездатності ні при яких, навіть некоректних, діях користувача. Всякі дії, що загрожують втратою інформації, мають бути підтверджені користувачем. Інформація, що вводиться, скрізь, де це можливо, піддається логічному контролю.

**Забезпечення цілісності даних.** При будь-яких діях користувача не повинні втрачатися дані або їх цілісність (некоректність індексів, втрата посилань в зв'язках після видалення-додавання записів і т. д.).

**Функціональна повнота.** Мають бути реалізовані усі функції, вказані в специфікації програми.

**Терміни і інтерфейс.** У діалогових вікнах використовуються тільки терміни, зрозумілі користувачеві, і не використовуються терміни розробника («запис», «індексація» і т. д.). Поява службових англійських повідомлень неприпустимо. У повідомленнях користувача слід дотримуватися норм ввічливості, колірна гамма повинна наслідувати загальноприйнятій рекомендації.

**Використання клавіатури.** На будь-якому етапі натиснення будь-якої клавіші повинне ігноруватися або викликати передбачені дії, описані в засобах допомоги. Прив'язка дій до клавіш має бути загальноприйнятною: **F1** - допомога; **Enter** - згода, завершення введення; **Esc** - відмова, повернення до попереднього вузла гілки алгоритму (з відновленням екранної форми); **Tab** - перехід до наступного поля, вікна і т. д.; **Shift + Tab** - повернення до попереднього поля і т., д.

**Вимоги до програмного коду.** Програма, що підлягає розробці, повинна бути як можна менш громіздкою, наявною і «читаємою». Такі програми зручні при їх вивченні та експлуатації.

Програма повинна складатися так, щоб її статична форма максимально точно відбивала динаміку її виконання.

Оскільки програма представляє собою послідовність операторів алгоритмічної мови програмування, кожен з котрих призначено для деталізації певної дії(наприклад, за допомогою оператора присвоєння здійснюється обчислювання виразу і присвоєння отриманого результату змінній; за допомогою умовного оператора здійснюється розгалуження, тощо), то програма буде простою для її вивчення, коли послідовність виконання операторів близька до послідовності розміщення цих операторів у тексті програм.

У цьому випадку перевірка програми (особливо її логіки) не викликає утруднень, так як оператори виконуються у порядку запису. Це полегшує владження програми та пошук помилок.

Які умови треба виконувати для цього? По-перше, для написання програми треба використовувати оператори, за допомогою яких реалізуються стандартні блоки алгоритму. Наприклад:

- Оператор присвоєння – для реалізації функціональних блоків.
- Умовний оператор – для реалізації блоку „Вибір”.
- Оператор циклу-для реалізації блоків „Повторювання”.

По-друге, при складанні програми треба прагнути до того, щоб як можна менш використовувати оператор безумовного переходу - **goto**, оскільки це веде до ускладнення логіки і структури програми.

Таким чином, для того щоб програма була простою при її вивченні, вона повинна поєднувати представлення її тексту у просторі (на бланку і у пам'яті ПК) і виконання операторів за часом (у тому порядку, як вони занесені у програмі).

Вимоги до наявності та «читаності» є одними з найважливіших, бо характеризують якість оформлення програми.

Як зробити програми наявною та «читаною»?

1. Форма програми повинна повністю збігатися з алгоритмом. Це досягається тоді, коли кожному стандартному блоку алгоритму відповідає свій оператор у програмі як по місцю розміщення, так і за часом виконання. Наприклад, блоку повторювання повинен відповідати оператор циклу, а не група операторів, в якій цикл організовано за допомогою умовного оператора і оператора безумовного переходу.

2. Всі позначення у алгоритмі і програмі повинні збігатися. Наприклад, змінні **x**, **x<sub>0</sub>**, **π** алгоритму доцільно позначити у програмі **x**, **x<sub>0</sub>**, **pi**.

3. При запису програми на бланку треба керуватися раціональним стилем зображення. Тобто у кожному рядку необхідно розміщувати тільки один оператор. Коли оператор вміщує внутрішні оператори, наприклад, оператори циклу, то внутрішні оператори слід записувати на окремих рядках із зсувом праворуч на одно або декілька позицій.

4. Логічні ділянки програми, а також оператори циклу, умови та подібні потрібно виділяти за допомогою табулятора, щоб з першого погляду на програмний код було зрозуміло структуру програми.

5. Всі оператори доцільно супроводжувати коментарями.

## **5. ВИМОГИ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ**

### **5.1. Загальні вимоги**

Загальними вимогами до пояснювальної записки є логічна послідовність викладання матеріалу, стислість, чіткість і конкретність викладання теоретичних і практичних результатів роботи, доказовість висновків і обґрунтованість рекомендацій.

Пояснювальна записка виконується машинним способом на одній стороні аркушу білого паперу формату А4 (210 x 297 мм) відповідно до [1]. Наведемо деякі положення цих документів.

Поля: верхнє, праве – 1 см, ліве – 2,5 см, нижнє – 2,5 см. Розмір аркушів – А4 (21×29,7 см), орієнтація – книжкова, положення палітурки – зліва. Форматування колонтитулів: від краю до верхнього колонтитулу – 0 см, від краю до нижнього колонтитулу – 0,7 см. Висота верхнього колонтитулу – 1 см. Вертикальне вирівнювання тексту – по верхньому краю сторінки. Потрібно також намагатись раціонально розташовувати текст в межах сторінки таким чином, щоб під час вкладання ілюстрацій та таблиць до тексту документу внизу таблиці не залишалось багато вільного місця (більше 3 см).

Прізвища, назви додатків, організацій, фірм та інші імена наводяться мовою

оригіналу.

Пояснювальна записка оформлюється студентом самостійно на стандартних аркушах паперу (формат А4).

Пояснювальна записка повинна бути грамотна написана і акуратно оформлена, окремі листи повинні бути скріплені між собою (прошиті). Крім того, звіт представляється студентами в електронному вигляді (на дискеті чи флешці).

При створенні електронної версії пояснювальної записки необхідно керуватися наступним:

- рукопис повинен бути підготовлений в текстовому редакторі MS Word з шрифтом Times New Roman, розмір шрифту для основного тексту – 14;
- обов'язково формувати по ширині тексту;
- малюнки (схеми алгоритмів) необхідно будувати в текстових або графічних редакторах;
- одиничний інтервал між рядками.

## **5.2. Структурні елементи та розділи**

Треба розрізняти структурні елементи записки, такі як “РЕФЕРАТ”, “ЗМІСТ”, “ВСТУП” і розділи записки. Деякі елементи, а саме “РЕФЕРАТ”, “ЗМІСТ”, “ВСТУП”, “ВИСНОВКИ”, “ПЕРЕЛІК ПОСИЛАНЬ”, створюють розділи записки і їхні найменування служать заголовками окремих розділів. Такі розділи не мають нумерації. Основні структурні елементи записки можуть поділятися на розділи, які нумеруються і мають власні заголовки.

Розділи можуть поділятися на підрозділи, пункти та підпункти.

Шрифт - звичайний.

## **5.3. Заголовки**

Розділи і підрозділи повинні мати заголовки, пункти і підпункти можуть мати заголовки. Заголовки структурних елементів записки і заголовки розділів необхідно розташовувати посередині рядка і друкувати прописними літерами без крапки наприкінці, не підкреслюючи. Заголовки підрозділів, пунктів та підпунктів записки необхідно починати з абзацного відступу і друкувати малими літерами, крім першої прописної, не підкреслюючи, без крапки наприкінці. Абзацний відступ повинен бути однаковим по всьому тексту записки і має дорівнювати п'яти знакам (1,2 см). Якщо заголовок складається з двох або більше речень, їх розділяють крапкою.

Переноси слів у заголовку розділу не припускаються. Відстань між заголовком і наступним або попереднім текстом повинна бути не менше двох рядків.

Не припускається розміщати найменування розділу, підрозділу, а також пункту і підпункту в нижній частині сторінки, якщо після нього розташований тільки один рядок тексту.

Шрифт - Times New Roman, розмір – 14.

## **5.4. Нумерація сторінок**

Нумерувати сторінки необхідно арабськими цифрами, дотримуючи наскрізну нумерацію по всьому тексті. Номер сторінки проставляють у правому верхньому кутку без крапки наприкінці.

## **5.5. Нумерація розділів та підрозділів**

Розділи, підрозділи, пункти, підпункти пояснювальної записки варто нумерувати арабськими цифрами. Розділи пояснювальної записки повинні мати порядкову нумерацію в межах суті пояснювальної записки і позначатися арабськими цифрами без крапки, наприклад, 1, 2, 3. Підрозділи повинні мати порядкову нумерацію в межах кожного розділу. Номери підрозділів складаються з номерів розділів і підрозділів, що розділяються крапкою, наприклад, 1.1, 1.2, 1.3.

## **5.6. Переліки**

За необхідністю можуть бути використані переліки. Перед переліком ставлять двокрапку. Перед кожною позицією переліку треба ставити малу літеру алфавіту з дужкою або дефіс (перший рівень деталізації). Для подальшої деталізації переліку треба використовувати арабські цифри з дужкою (другий рівень деталізації).

## **5.7. Ілюстрації**

Ілюстрації треба розташовувати в пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, або на наступній сторінці. На всі ілюстрації повинні бути посилання в пояснювальній записці. Ілюстрації можуть мати назву, що розташовується під ілюстрацією. При необхідності під ілюстрацією ще розташовують пояснювальні дані. Ілюстрація позначається словом “Рисунок”, що разом із назвою поміщають після пояснювальних даних, наприклад, “Рисунок 2.1 – Схема алгоритму”.

Ілюстрації необхідно нумерувати арабськими цифрами в межах розділу, за винятком ілюстрацій, наведених у додатках. Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, розділених крапкою. Якщо ілюстрація не вміщується на одній сторінці, можна переносити її на інші сторінки, при цьому назву ілюстрації розташовують на першій сторінці, пояснювальні дані – на кожній сторінці, і під ними вказують: “Рисунок , аркуш ”.

## **5.8. Таблиці**

Таблиці необхідно розташовувати в пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, або на наступній сторінці. На всі таблиці повинні бути посилання в пояснювальній записці. Таблиця може мати назву, яку друкують малими літерами (крім першої прописної) і розміщують над таблицею. Якщо рядки або графи таблиці виходять за формат сторінки, таблицю поділяють на частини, розташовуючи одну частину під іншою, або поруч, або переносяться частини таблиці на наступну сторінку. Слово “Таблиця ”, вказують один раз зліва над першою частиною таблиці, над іншими частинами пишуть: “Продовження таблиці ” з вказівкою номера таблиці. Таблиці треба нумерувати арабськими цифрами в межах розділу, за винятком таблиць, наведених у додатках. Номер таблиці складається з номера розділу і порядкового номера таблиці, розділених крапкою.

## **5.9. Формули і рівняння**

Формули і рівняння розташовують у пояснювальній записці безпосередньо після тексту, у якому вони вперше згадуються, посередині сторінки. Вище і нижче



кожної формули повинно бути залишено не менше одного вільного рядка. Формули і рівняння треба нумерувати арабськими цифрами порядковою нумерацією в межах розділу, за винятком формул і рівнянь, наведених у додатках. Номер формули або рівняння складається з номера розділу і порядкового номера формули або рівняння, розділених крапкою. Номер формули або рівняння вказують на рівні формули або рівняння в дужках у крайньому правому положенні на рядку. Пояснення символів і числові коефіцієнти, що входять до формули або рівняння, варто робити безпосередньо після формули або рівняння в тій послідовності, у якій вони надані у формулі або рівнянні. Пояснення значення кожного символу або числового коефіцієнта необхідно давати з нового рядка. Перший рядок пояснення починають з абзацу словом "де" без двокрапки. Переносити формули або рівняння на інший рядок припускається тільки на знаках виконуваних операцій, причому знак операції на початку такого рядка повторюють. При переносі формули або рівняння на знаку операції множення застосовують знак "х". Формули, що прямують одна за одною, і не розділені текстом, відокремлюють комою.

### **5.10. Посилання**

Посилання в тексті на джерела необхідно наводити в послідовності їх згадування в записці, вказуючи порядковий номер, виділений двома квадратними дужками, наприклад: [5], [4,7] або [2-8].

### **5.11. Скорочення**

У тексті пояснювальної записки припускається робити тільки загально-вживані скорочення (наприклад "і т.і., і т.п., та ін.") або поширені аббревіатури. Якщо в пояснювальній записці прийнята специфічна термінологія, або вживаються малопоширені скорочення, нові символи, позначення і ін., їх перелік необхідно надати в окремому списку. Перелік повинен розташовуватися стовпчиком, у якому слова за абеткою наводяться скорочено, а справа - їхнє докладне тлумачення. Якщо спеціальні символи, терміни, скорочення, і позначення наводяться менше трьох разів, перелік їх не складається, а їхнє тлумачення роблять у тексті при першому згадуванні.

### **5.12. Перелік посилань**

Перелік посилань оформлюється як розділ пояснювальної записки з нової сторінки відповідно до вимог ДСТ 7. 1-84 "Библиографическое описание документа. Общие требования и правила составления".

Для оформлення бібліографічних посилань, правильного запису загальноприйнятих скорочень рекомендується використовувати методичні вказівки [3], які складені з урахуванням відповідних стандартів.

### **5.13. Додатки**

Додатки необхідно оформляти як продовження пояснювальної записки на наступних її сторінках, або в окремій частині, розташовуючи додатки в порядку появи посилань на них у тексті пояснювальної записки. Кожний додаток повинен починатися з нової сторінки і мати заголовок, надрукований вгорі малими літерами з першої прописної, симетрично щодо тексту сторінки. Посередині рядка над

заголовком малими літерами з першої прописної повинно бути надруковано слово "Додаток " і прописна буква, що позначає додаток. Додатки слід позначати послідовно прописними буквами українського алфавіту, за винятком букв Г, Є, З, І, Ї, Й, О, Ч, Ї. Один додаток позначається як А. Додатки повинні мати загальну з іншою частиною пояснювальної записки наскрізну нумерацію. Текст кожного додатка, при необхідності, може бути розділений на підрозділи і пункти, що нумеруються арабськими цифрами в межах кожного додатка.

## **6. ОРГАНІЗАЦІЯ ПРОЕКТУВАННЯ І ЗАХИСТ ПРОЕКТУ**

Курсовий проект належить до форми самостійної роботи студентів. Видача завдань здійснюється у перші два тижні семестру. Кожний студент складає календарний план роботи над темою і затверджує його у викладача. Час консультацій узгоджується з викладачем. Студенти консультуються у керівника індивідуально. Основні етапи роботи:

- ознайомлення з літературними джерелами за тематикою поставленої задачі;
- розробка алгоритмів та вивчення засобів вирішення задачі курсового проекту;
- написання програми і налагоджування на персональному комп'ютері;
- з'ясування одержаних проміжних теоретичних і практичних результатів з керівником проекту.

Після показу керівникові готового проекту та його затвердження студент починає писати пояснювальну записку.

Пояснювальна записка здається керівнику для перевірки, після якої студент виправляє виявлені недоліки. Керівник допускає студента до захисту і призначає дату і час захисту. Курсовий проект повинен бути виконаний і захищений не пізніше, ніж за 1 тиждень до початку сесії.

Захист відбувається перед комісією кафедри. Студент повинен уміти виділити і чітко викласти основні моменти проекту, відповісти на поставлені запитання, довести свою компетентність в області знань, що досліджується.

Загальна оцінка залежить від теоретичної підготовки студента в області знань відповідно завданню, від якості розробленої програми та пояснення її особливостей, змісту пояснювальної записки, повноти опрацювання теоретичного матеріалу, а також від оформлення записки.

## **7. ПЕРЕЛІК ДЖЕРЕЛ, ЩО РЕКОМЕНДОВАНІ ДЛЯ ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ**

1. ДСТУ 3008-95 "Документація. Звіти в сфері науки і техніки. Структура і правила оформлення" – Державний стандарт України.
2. Брауде Э. Технология разработки программного обеспечения / Э. Брауде. – СПб. : Питер, 2004.
3. Основные концепции и механизмы объектно-ориентированного программирования/ Е. В. Пышкин. - СПб: БХВ-Петербург. - 2005 г.

4. Рихтер Дж. Windows для профессионалов. Создание эффективных WIN32-приложений с учетом специфики 64-разрядной версии Windows. - СПб.: Русская Редакция, 2001 г. - 752 с.
5. Симбірський Г. Д. Конспект лекцій з дисципліни “Інформаційні технології” за напрямом підготовки 6.050702 “Електромеханіка” (розділ “Мова програмування С++”). – Харків: ХНАДУ, 2014. – 149 с.
6. Симбірський Г. Д. Методичні вказівки до лабораторних робіт з дисципліни “Інформаційні технології” за напрямом підготовки 6.050702 “Електромеханіка” (розділ “Мова програмування С++”). – Портал ХНАДУ, 2014. – 149 с.
7. Танненбаум Э. Современные операционные системы. 3-е изд. – СПб.: Питер, 2010. – 1120с.
8. Хортон А. Visual С++ 2010. Полный курс. - К.: Диалектика, 2011. – 1205 с.
9. Химмельблау Д. Анализ процессов статистическими методами / Д. Химмельблау. – М.: Мир, 1973. – 957 с.

### Електронні ресурси

1. Сторінка лектора Симбірського Г.Д. для потоку РК у соціальній мережі “ВК” <http://vc.com/id23791456>. (Матеріали для самостійної роботи, а також лекції, методичні вказівки до лабораторних робіт, питання для підготовки до здачі тестів, екзаменаційні білети тощо).
2. ХНАДУ "Файловий архів": <http://files.khadi.kharkov.ua/mekhatroniki-transportnikh-zasobiv.html>. (Лекції, методичні вказівки до лабораторних робіт, питання для підготовки до здачі тестів, екзаменаційні білети тощо.)

**Додаток А. Бланк титульного аркушу курсового проекту**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНІЙ УНІВЕРСИТЕТ

Факультет комп'ютерних технологій та мехатроніки

Кафедра інформаційних технологій та мехатроніки

**КУРСОВИЙ ПРОЕКТ**

з навчальної дисципліни “Програмування”  
(розділ “Структурне програмування на мові С++  
у середовищі Microsoft Visual Studio 2010”)

Тема: **“Розробка програмного додатка у середовищі  
Microsoft Visual Studio 2010 для статистичного  
аналізу стохастичних процесів”**

Студента (ки) **1** курсу групи **РК-11** факультету **КТМ**  
напряму підготовки **6.050201 “Системна інженерія”**  
спеціальності **6.050201 “Комп'ютерні системи управління  
рухомими об'єктами\_”**

---

(прізвище та ініціали)

Керівник: доцент, канд.техн.наук Симбірський Г.Д.

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ЄКТС \_\_\_\_\_

Підпис керівника \_\_\_\_\_

Харків, 2016

**Додаток Б. Зразок листа завдання на курсовий проект**  
Харківський національний автомобільно-дорожній університет  
Факультет комп'ютерних технологій та мехатроніки  
кафедра інформаційних технологій та мехатроніки

Дисципліна	Програмування				
Напрямок підготовки	6.050201 “Системна інженерія”				
Спеціальності	6.050201 “Комп’ютерні системи управління рухомими об’єктами”				
Курс	1	група	РК-11	семестр	2

**ЗАВДАННЯ**  
на курсовий проект студента

---

(прізвище, ім’я, по батькові)

1. Тема проекту (роботи)    **Розробка програмного додатка на мові C++ у середовищі Microsoft Visual Studio 2010 для статистичного аналізу стохастичних процесів**
2. Строк здачі студентом закінченого проекту (роботи)    16.05.2016р.
3. Перелік питань, які підлягають розробці:
  - 3.1. Вычислить массив  $Y$  действительных чисел из 40 элементов по заданной формуле:
$$Y_i = f(x_i)$$
где  $K=$ \_\_\_\_\_ – коэффициент;  $x_i$  - номер элемента массива ( $i=1,2,\dots, 40$ ).
  - 3.2. В программе предусмотреть запись полученного массива действительных чисел  $Y$ , промежуточных и итоговых результатов работы над программным кодом в текстовые и бинарные файлы в папку на диске  $D$  с применением указателей на массивы.
  - 3.3. Вычисление массива, его запись в файлы и чтение из них оформить как функции. В дальнейшем все логически завершенные обособленные блоки программы обязательно оформлять в виде функций.  
Передавать данные в функции необходимо с использованием указателей и динамической памяти.

- 3.4. Массив  $Y$  и другие массивы после считывания из бинарного и текстового файлов вывести на экран с пометкой, из какого файла выведен массив.
- 3.5. Использовать генератор случайных чисел для создания из массива  $Y$  массива случайных действительных чисел  $Y_{\sigma}$  с заданным среднеквадратическим отклонением  $\sigma$ , которое вводится в программу в диалоговом режиме.  
Массив случайных действительных чисел  $Y_{\sigma}$  также записать в соответствующие текстовый и бинарный файлы.
- 3.6. При помощи редактора электронных таблиц **Excel** из программного пакета **Microsoft Office 2010** используя полученные ранее текстовые файлы построить графики зависимостей  $Y = f(x)$  и  $Y_{\sigma} = f(x)$  в соответствии с ДСТУ (на одном графике).
- 3.7. Вычислить среднее значение, дисперсию и среднее квадратичное отклонение случайного процесса  $Y_{\sigma} = f(x)$ .
- 3.8. Построить графики зависимостей  $Y = f(x)$  и  $Y_{\sigma} = f(x)$  при помощи средств языка программирования C++, оформив это построение в виде функции.

**Примечание.** Программный код должен сопровождаться подробными комментариями буквально для каждого оператора.

Файлы для записи массивов данных должны располагаться на диске D в файле группы с именами **DataY.txt**, **DataY.dat**, **DataYS.txt**, **DataYS.dat**.

Кроме этих файлов должен быть организован файл **Result Фамилия РК11.txt**. В этот текстовый файл должна записываться вся информация о выполнении программой предусмотренных операций.

4. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

- 4.1. Блок-схема алгоритма решения поставленной задачи
- 4.2. Графики зависимостей  $Y = f(x)$  и  $Y_{\sigma} = f(x)$  в среде **Excel**.

5. Дата видачі завдання \_\_\_\_\_

6. Підпис студента \_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ**

№	Назва етапів курсового проекту	Строк виконання етапів проекту	Примітка
1	Вычислить массив $Y$ действительных чисел из 40 элементов по заданной формуле. В программе предусмотреть запись полученного массива действительных чисел $Y$ , промежуточных и итоговых результатов работы над программным кодом в текстовые и бинарные файлы в папку на диске $D$ с применением указателей на массивы.	4 тиждень	
2	3.4. Массив $Y$ и другие массивы после считывания из бинарного и текстового файлов вывести на экран с пометкой, из какого файла выведен массив.	6 тиждень	
3	Использовать генератор случайных чисел для создания из массива $Y$ массива случайных действительных чисел $Y_{\sigma}$ с заданным среднеквадратическим отклонением $\sigma$ , которое вводится в программу в диалоговом режиме. Массив случайных действительных чисел $Y_{\sigma}$ также записать в соответствующие текстовый и бинарный файлы.	8 тиждень	
4	При помощи редактора электронных таблиц <b>Excel</b> из программного пакета <b>Microsoft Office 2010</b> используя полученные ранее текстовые файлы построить графики зависимостей $Y = f(x)$ и $Y_{\sigma} = f(x)$ в соответствии с ДСТУ	10 тиждень	
5	Вычислить среднее значение, дисперсию и среднее квадратичное отклонение случайного процесса $Y_{\sigma} = f(x)$ .	12 тиждень	
6	Построить графики зависимостей $Y = f(x)$ и $Y_{\sigma} = f(x)$ при помощи средств языка программирования <b>C++</b> , оформив это построение в виде функции.	14 тиждень	

Студент \_\_\_\_\_  
(підпис)

\_\_\_\_\_ (прізвище, ініціали)

Керівник \_\_\_\_\_  
(підпис)

\_\_\_\_\_ (прізвище, ініціали)

“ \_\_\_\_ ” \_\_\_\_\_ 2016 р.

## Додаток В. Приклад змісту реферату

### РЕФЕРАТ

Пояснювальна записка складається з: 50 с., 4 рисунків, 3 таблиць, 8 джерел.

Об'єкт дослідження – модель стохастичного процесу.

Мета роботи – аналіз методів, алгоритмів та засобів дослідження стохастичних процесів.

Методи дослідження – методи теорії алгоритмів та мов програмування, методи статистичної обробки інформації.

Розглянуто процес моделювання стохастичної вимірювальної інформації та розроблено програму-функцію з використанням генератора випадкових чисел задля перетворення еталонного масиву даних на випадкову вимірювальну інформацію.

Розроблені засоби виведення масивів даних на консоль та у текстові та бінарні файли з використанням динамічного програмування, адресації змінних та вказівників.

Проведено аналіз алгоритмів обробки вимірювальної інформації чи іншої стохастичної інформації. Обґрунтовується вибір методів для статистичного аналізу результатів дослідження стохастичних процесів.

Розроблено додаток для візуалізації алгоритмів статистичної обробки інформації.

**Ключові слова:** програмування, мова C++, середовище Microsoft Visual Studio 2010, структурне програмування, стохастичні процеси, модель стохастичного процесу, статистична обробка вимірювальної інформації, статистичний аналіз.



**ДОДАТОК Д. Приклад оформлення пояснювальної  
записки до курсового проекту**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНІЙ УНІВЕРСИТЕТ

Факультет комп'ютерних технологій та мехатроніки  
Кафедра інформаційних технологій та мехатроніки

**КУРСОВИЙ ПРОЕКТ**

з навчальної дисципліни “Програмування”  
(розділ “Структурне програмування на мові C++  
у середовищі Microsoft Visual Studio 2010”)

Тема: **“Підготовка і рішення прикладних задач на ПК”**

Студента (ки) **1** курсу групи **РК-11** факультету **КТМ**  
напряму підготовки 6.050201 “Системна інженерія”  
спеціальності 7.05020101 “Комп'ютеризовані системи  
управління та автоматика”

Петренко І.П.

(прізвище та ініціали)

Керівник      доцент, канд.техн.наук Смірнов А.П.

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_      Оцінка: ЄКТС \_\_\_\_\_

Підпис керівника \_\_\_\_\_

Харків, 2016

Харківський національний автомобільно-дорожній університет

Факультет комп'ютерних технологій та мехатроніки

кафедра інформаційних технологій та мехатроніки

Дисципліна            **Програмування**

Напрямок підготовки            **6.050201 Системна інженерія**

Спеціальність            **6.050201 Комп'ютерні системи управління  
рухомими об'єктами**

Курс    **1** група    **РК-11** семестр    **2**

### **ЗАВДАННЯ**

на курсовий проект студента

**Петренко Іван Петрович**

1. Тема проекту    **Розробка програмного додатка у середовищі Microsoft Visual Studio 2010 для статистичного аналізу стохастичних процесів**

2. Строк здачі студентом закінченого проекту (роботи)            **12.05.2016 р.**

3. Вихідні дані до проекту:            Заданий масив  $N$  дійсних чисел. Сформувати новий масив з елементів початкового. Кожний елемент нового масиву є дробовою частиною відповідного елемента початкового масиву. Обчислити суму елементів нового масиву.

Максимальний розмір масиву – 50 чисел.

4. Перелік питань, які підлягають розробці:

4.1. Початкові дані (заданий масив) розміщуються у файлі DatesIn.dat.

4.2. Результат (сформований масив і суму) розмістити у файлі DatesOut.dat. У разі відсутності файлу з початковими даними, в програмі передбачити можливість формування початкових даних «вручну». Інформація у файлах повинна зберігатися в текстовому форматі.

4.3. Початковий модуль, виконуваний модуль, початкові дані, результат розміщуються в одному каталозі на магнітному носії.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання            **10.02.2016 р.**

7. Дата виконання п.п. 4.1 – 4.6            **11.05.2016 р.**

## КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ

№	Назва етапів курсового проекту	Строк виконання етапів проекту	Примітка
1	Отримання завдання до КП	2-й тиждень	Виконано
2	Аналіз завдання та математичний опис задачі	3-й тиждень	Виконано
3	Формулювання вимог до створюваної програми	4-й тиждень	Виконано
4	Розробка блок-схеми алгоритму	6-й тиждень	Виконано
5	Створення програмного коду	10-й тиждень	Виконано
6	Тестування програми	11-й тиждень	Виконано
7	Оформлення пояснювальної записки	14-й тиждень	Виконано
8	Захист курсового проекту	16-й тиждень	

Студент \_\_\_\_\_  
(підпис)

Петренко І.П.

Керівник \_\_\_\_\_  
(підпис)

Смірнов А.П.

“ \_\_\_ ” \_\_\_\_\_ 2016 р.

## РЕФЕРАТ

Пояснювальна записка складається з: 50 с., 4 рисунків, 3 таблиць, 8 джерел.

Об'єкт дослідження – модель стохастичного процесу.

Мета роботи – аналіз методів, алгоритмів та засобів дослідження стохастичних процесів.

Методи дослідження – методи теорії алгоритмів та мов програмування, методи статистичної обробки інформації.

Розглянуто процес моделювання стохастичної вимірювальної інформації. Проведено аналіз алгоритмів обробки вимірювальної інформації чи іншої стохастичної інформації. Обґрунтовується вибір методів для статистичного аналізу результатів дослідження стохастичних процесів. Розроблено додаток для візуалізації алгоритмів статистичної обробки інформації.

**Ключові слова:** програмування, мова C++, середовище Microsoft Visual Studio 2010, структурне програмування, стохастичні процеси, модель стохастичного процесу, статистична обробка вимірювальної інформації, статистичний аналіз.

## ЗМІСТ

Завдання на курсову роботу .....	2
Реферат .....	4
Вступ .....	6
Розділ 1. Обґрунтування алгоритму й структури програми .....	7
1.1. Постановка і математичний опис задачі .....	
1.2. Опис програми, визначення та опис функцій .....	
1.3. Розробка та обґрунтування алгоритму й структури програми .....	
Розділ 2. Розробка програми	
2.1. Специфікація програми .....	
2.2. Розробка програми .....	
2.3. Проектування і опис інтерфейсу користувача .....	
Розділ 3. Тестування програми і результати її виконання .....	
3.1. Тестування програми .....	
3.2. Результати виконання програми .....	
Висновки .....	
Література .....	
Додаток А. Блок-схема алгоритму програми .....	
Додаток Б. Лістинг (код з коментарями) програми .....	
Додаток В. Графічна інтерпретація результатів .....	

## ВСТУП

Зміни, що відбуваються сьогодні в різних сферах професійного і суспільного життя, пов'язані перш за все з широким впровадженням засобів інформаційних та комунікаційних технологій. Вони вимагають від фахівця у галузі комп'ютерних наук вміння розробляти програмне забезпечення для розв'язування задач науки, техніки, економіки і управління; створювати і використовувати інформаційні моделі процесів і явищ; використовувати інформаційні технології у проектно-конструкторській, управлінській та фінансовій діяльності.

Зростання попиту з боку державних та приватних організацій на складні системи автоматизації підприємств, надійні операційні середовища, комплекси глобального телекомунікаційного управління, широке впровадження інформаційних технологій в повсякденне життя значно підвищили значення професії програміста, а також інших професій ІТ галузі.

Дана робота присвячена розробці програмного забезпечення обробки масивів інформації. Подібна інформація може з'являтися в результаті різноманітних процесів у техніці, наукових дослідженнях, суспільних процесів тощо. Наприклад, це може бути інформація, що є результатом вимірювань при дослідженнях чи регулюванні фізичних процесів.

Така інформація завжди має стохастичний (випадковий) характер. Тому при дослідженні методів обробки вимірювальної інформації можна користуватися її змодельованим замінником – еталонними значеннями, які оброблені за допомогою генератора випадкових чисел. Для аналізу стохастичної інформації застосовують різноманітні статистичні методи.

Розроблена програма дозволяє дослідити ефективність методів статистичного аналізу стохастичних процесів. Вона дає можливість регулювати при дослідженнях параметри стохастичної інформації, щоб оцінити ступінь їх впливу на ефективність методів статистичного аналізу.

## 1. ОБҐРУНТУВАННЯ АЛГОРИТМУ Й СТРУКТУРИ ПРОГРАМИ

### 1.1. Постановка і математичний опис задачі

1.1.2. Заданий масив  $N$  дійсних чисел. Сформувати новий масив з елементів початкового. Кожний елемент нового масиву є дробовою частиною відповідного елемента початкового масиву. Обчислити суму елементів нового масиву.

Максимальний розмір масиву – 50 чисел.

Початкові дані (заданий масив) розміщуються у файлі DatesIn.dat. Результат (сформований масив і суму його елементів) розмістити у файлі DatesOut.dat. У разі відсутності файлу з початковими даними, в програмі передбачити можливість формування початкових даних «вручну». Інформація у файлах повинна зберігатися в текстовому форматі.

Початковий модуль, виконуваний модуль, початкові дані, результат розміщуються в одному каталозі на магнітному носії.

При розробці програми використовується наступна формула:

$$S = \sum_{i=1}^N y_i,$$

де  $S$  – сума дробових частин дійсних чисел початкового масиву;

$i$  – номер елемента заданого масива;

$y_n$  – дробова частина дійсного числа початкового масиву;

$N$  – кількість дійсних чисел початкового масиву.

Обчислення суми виконується за наступними рекурентними співвідношеннями:

$$S=0;$$

$$S=S+y_n;$$

(на формулі краще не закінчувати).

### 1.2. Опис програми, визначення та опис функцій

Програма називається «Перетворення масиву» і призначена для перетворення масиву дійсних чисел в масив дробових частин цих чисел, а також для розрахунку суми елементів шуканого масиву.

Програма функціонує в ОС Windows як консольний додаток.

Програма розроблена за допомогою інтегрованого середовища Microsoft Visual Studio 2010 (Visual C++).

Програма дозволяє обробляти масиви, що містять не більше 50 елементів. Введення і виведення даних здійснюється на екран монітора ПК, а також записуються у відповідні файли.

Запуск програми здійснюється за допомогою виконуваного файлу (\*.exe).

8

За відсутністю масиву початкових даних видається повідомлення:

«Файл DatesIn.dat не существует! Сформируйте входной массив!»

При отриманні цього повідомлення необхідно вручну сформувати масив початкових даних (введення даних з клавіатури).

### 1.3. Розробка та обґрунтування алгоритму та структури програми

(Опишіть алгоритм своїми словами. Як розбили на функції, де цикл for, де ще щось).

Блок-схему алгоритму рішення задачі наведено у Додатку А.



## 2. РОЗРОБКА ПРОГРАМИ

### 2.1. Специфікація програми

Початкові дані є масивом дійсних чисел, що зберігаються у файлі „DatesIn.dat” у вигляді текстової інформації (під запис кожного числа відводиться 8 байт).

Результати – це масив дійсних чисел, що зберігаються у файлі „DatesOut.dat” у вигляді текстової інформації (під запис кожного числа відводиться 8 байт). Крім того, в цьому файлі зберігається сума всіх елементів цього масиву (останні 8 байт).

Призначення даних та змінних, їх математичні позначення, імена, типи тощо наведені у таб. 1.

Таблиця 1. Перелік даних та змінних, що використовуються у програмі

Призначення даних та змінних	Математичне позначення змінної	Ім'я змінної у програмі	Структура даних	Тип даних	Сенс змінної
Початкова	$x_i$	X[]	масив	float	Масив початкових даних
Початкова	M	M	константа	int	Максимальний розмір масиву
Проміжна	N	N	проста	int	Реальний розмір масиву
Проміжна	K	K	константа	int	Розмір буферу обміну
Проміжна	i	i	проста	int	Параметр циклу
Проміжна	j	j	проста	int	Параметр циклу
Проміжна		buffer[]	масив	int	Буфер для обміну з файлом
Результат	$y_i$	Y[]	масив	float	Масив даних результату
Результат	S	Summa	проста	float	Сума елементів шуканого масиву

### 2.2. Розробка програми

(Самостійно описати, як розробляли програму).

Розроблений програмний код наведено у додатку Б.

### 2.3. Проектування і опис інтерфейсу користувача

### 3. ТЕСТУВАННЯ ПРОГРАМИ І РЕЗУЛЬТАТИ ЇЇ ВИКОНАННЯ

#### 3.1. Тестування програми

При тестуванні перевірялася працездатність програми в двох варіантах:

- 1) відсутність масиву з початковими даними;
- 2) наявність масиву з початковими даними (основний режим). Як початкові дані використовувався масив дійсних чисел {2.45, 0.32, 5.41, 6.78, 3.93}.

В першому варіанті програма видала повідомлення:

"Файл DatesIn.dat не существует!"

"Сформируйте входной массив!"

Після введення з клавіатури масиву дійсних чисел виконання програми було продовжено. Масив початкових даних записався на магнітний носій. Після чого функціонування програми здійснюється в основному режимі.

В результаті рішення програми з контрольними результатами був сформований масив {0.45, 0.32, 0.41, 0.78, 0.93, 2.89}. Масив результату даних записався на магнітний носій.

Зміст масивів з початковими даними і сформованим масивом можна прочитати за допомогою текстового редактора.

В процесі рішення програми на екрані монітора відображалася інформація:

(Краще, щоб була копія вікна ДОС, як в Методичних вказівках з лабораторних робіт).

Файл DatesIn.dat не существует!

Сформируйте входной массив!

Введите размер массива!

Размер массива не больше 50!

5

Введите 1 элемент массива

2.45

Введите 2 элемент массива

0.32

Введите 3 элемент массива

5.41

Введите 4 элемент массива

6.78

Введите 5 элемент массива

3.93

Формирование массива начальных данных завершено!

Массив начальных данных

2.45 0.32 5.41 6.78 3.93

Массив искомых элементов

0.45 0.32 0.41 0.78 0.93

Сумма элементов массива = 2.89

На підставі аналізу результатів виконання програми можна зробити наступні висновки:

- 1) Програма логічно побудована правильно.
- 2) Результати рішення задачі вірні.
- 3) Усі масиви в необхідному форматі зберігаються на магнітному носії;
- 4) Програма може бути прийнята в експлуатацію.

### 3.2. Результати виконання програми

Дана програма використовувалася при обробці масивів даних про залишки палива. У зв'язку з тим, що інформація про кількість палива на диспетчерський пункт поступає з точністю до 1 літра, а датчики рівня фіксують до 0.1 літра, виникла необхідність в оцінці точності обліку витрати палива.

Як масив початкових даних в програмі використовувалися дані про щоденну витрату палива 13 автомобілями (таблиця 2).

Таблиця 2. Дані про витрати палива

Візуальна витрата	7	5	1	4	8	9	0	2	9	2	4	5	4
Витрата за даними датчиків	7.4	5.3	1.6	4.1	8.4	9.5	0.2	2.3	9.5	2.2	4.6	5.3	4.5

В результаті виконання програми були отримані дані про перевитрату палива у вигляді масиву:

{0.4 0.3 0.6 0.1 0.4 0.5 0.2 0.3 0.5 0.2 0.6 0.3 0.5 4.9}.

Останнє число в масиві – сума всіх елементів (загальна перевитрата палива).

## ВИСНОВКИ

В результаті виконання курсової роботи була розроблена програма обробки масиву даних, які зберігаються на магнітному носії в текстовому вигляді.

Розроблена в курсовій роботі програма може бути використаний при рішенні різних прикладних задач, пов'язаних з обробкою масивів даних.

Оскільки дана програма обробляє одновимірні масиви інформації, той подальший розвиток програми повинен бути пов'язаний з обробкою багатовимірних масивів даних.

(Написати розгорніше, по пунктам Завдання)

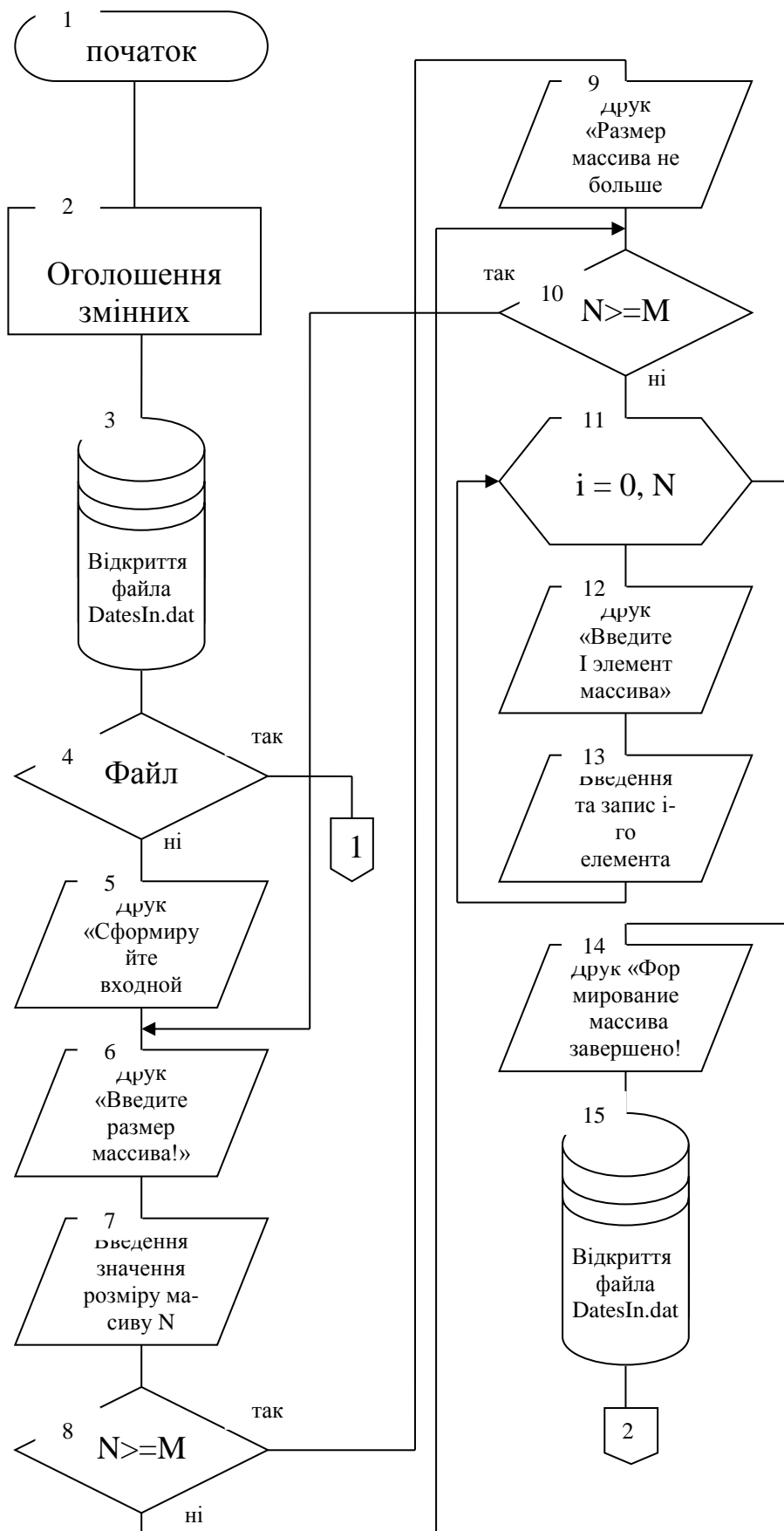
## ЛІТЕРАТУРА

1. ДСТУ 3008-95 "Документація. Звіти в сфері науки і техніки. Структура і правила оформлення" – Державний стандарт України.
2. Брауде Э. Технология разработки программного обеспечения / Э. Брауде. – СПб. : Питер, 2004.
3. Основные концепции и механизмы объектно-ориентированного программирования/ Е. В. Пышкин. - СПб: БХВ-Петербург. - 2005 г.
4. Рихтер Дж. Windows для профессионалов. Создание эффективных WIN32-приложений с учетом специфики 64-разрядной версии Windows. - СПб.: Русская Редакция, 2001 г. - 752 с.
5. Симбірський Г. Д. Конспект лекцій з дисципліни “Інформаційні технології” за напрямом підготовки 6.050702 “Електромеханіка” (розділ “Мова програмування C++”). – Харків: ХНАДУ, 2014. – 149 с.
6. Симбірський Г. Д. Методичні вказівки до лабораторних робіт з дисципліни “Інформаційні технології” за напрямом підготовки 6.050702 “Електромеханіка” (розділ “Мова програмування C++”). – Портал ХНАДУ, 2014. – 149 с.
7. Танненбаум Э. Современные операционные системы. 3-е изд. – СПб.: Питер, 2010. – 1120с.
8. Хортон А. Visual C++ 2010. Полный курс. - К.: Диалектика, 2011. – 1205 с.
9. Химмельблау Д. Анализ процессов статистическими методами / Д. Химмельблау. – М.: Мир, 1973. – 957 с.

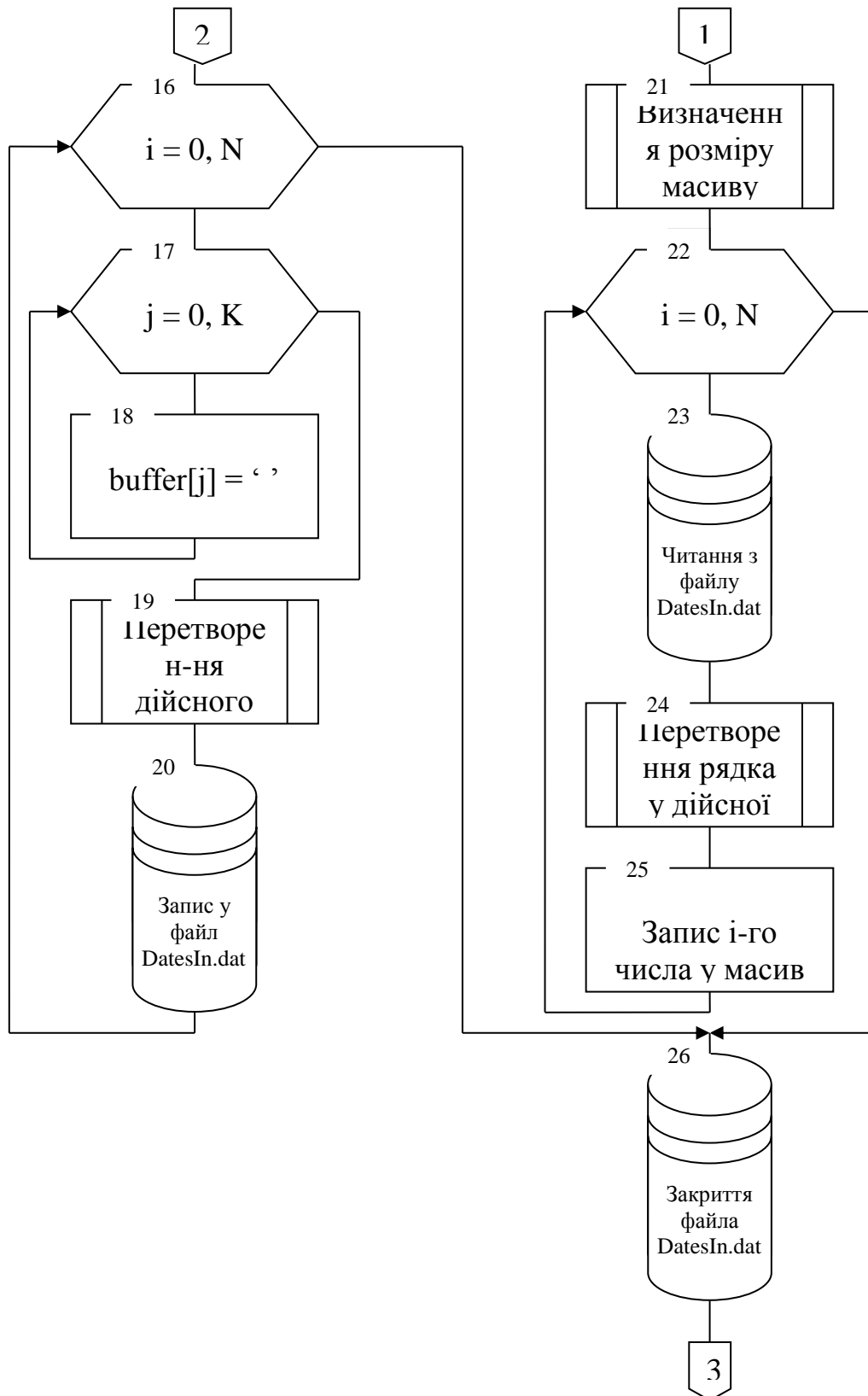
## Електронні ресурси

1. Сторінка лектора Симбірського Г.Д. для потоку РК у соціальній мережі “ВК” <http://vc.com/id23791456>. (Матеріали для самостійної роботи, а також лекції, методичні вказівки до лабораторних робіт, питання для підготовки до здачі тестів, екзаменаційні білети тощо).
2. ХНАДУ "Файловий архів": <http://files.khadi.kharkov.ua/mekhatroniki-transportnikh-zasobiv.html>. (Лекції, методичні вказівки до лабораторних робіт, питання для підготовки до здачі тестів, екзаменаційні білети тощо.)

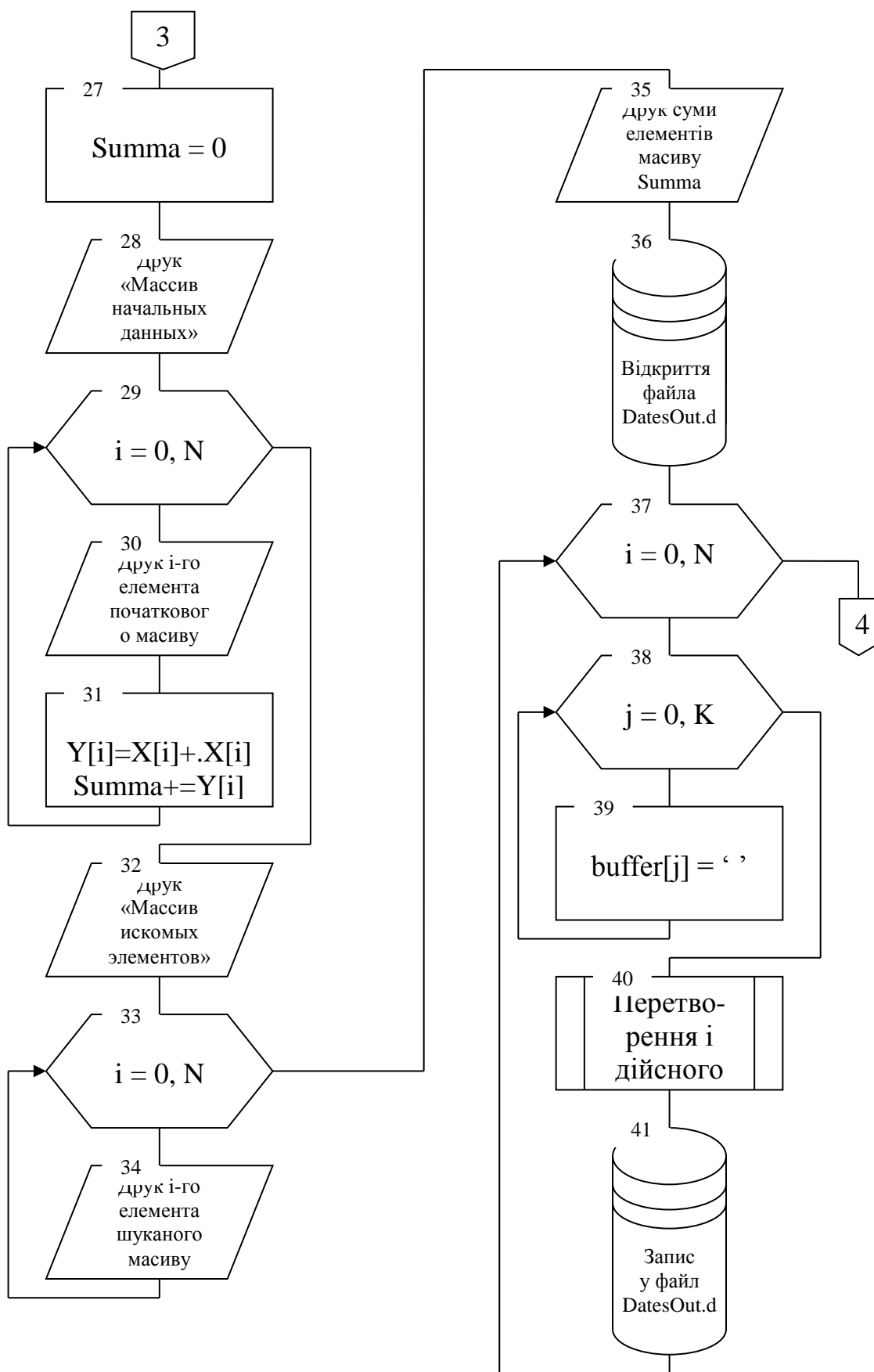
## Додаток А. Блок-схема алгоритму програми



## Додаток А. Блок-схема алгоритму програми (продовження)



## Додаток А. Блок-схема алгоритму програми (продовження)





## Додаток А. Блок-схема алгоритму програми (продовження)



## Додаток Б. Програмний код з коментарями

**Увага!!** Коментарі проставьте майже до кожного оператора програми

//Перетворення масиву....

```

#include <vcl.h>
#include <iostream.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int main()
{
    FILE *fFile;                //Показчик на файловий потік
    const int M=51;             //Максимальний розмір масиву
    const int K=8;              //Розмір буферу обміну(кількість байтів)
    int N;                       //Реальний розмір масиву
    int i, j;                    //Лічильники
    float X[M];                  //Початковий масив
    float Y[M];                  //Сформований масив
    float Summa;                 //Сума елементів сформованого масиву
    char buffer[K];              //Рядок символів
    if ((fFile = fopen ("DatesIn.dat", "rt")) == NULL)
    {
        cout<<"Файл DatesIn.dat не существует!"<<endl;
        cout<<"Сформируйте входной массив!"<<endl;
        do
        {
            cout<<"Введите размер массива!"<<endl;
            cout<<"Размер массива не больше 50!"<<endl;
            cin>>N;
            if (N>=M) cout<<" Размер массива больше 50! Повторите ввод"<<endl;
        } while (N>=M);
        for (i=0;i<N;i++)
        {
            cout<<"Введите "<<i+1<<"элемент массива"<<endl;
            cin>>X[i];
        }
        cout<<"Формирование массива начальных данных завершено!"<<endl;
        fFile = fopen ("DatesIn.dat", "wt");
        for (i=0;i<N;i++)
        {
            for (j=0;j<K;j++) buffer[j]=' ';
            gcvt(X[i],4,buffer);
            fwrite(&buffer, sizeof(buffer), 1, fFile);
        }
    }
    else
    {

```

```

    N = filelength(fileno(fFile))/sizeof(buffer);
    for (i=0;i<N;i++)
    {
fread(&buffer, sizeof(buffer), 1, fFile);
X[i]=atof(buffer);
    }
}
fclose(fFile);
Summa = 0;
cout<<"Массив начальных данных"<<endl;
{
cout<<X[i]<<" ";
Y[i] = X[i] - floor(X[i]);
Summa = Summa + Y[i];
}
cout<<endl<<"Массив искомых элементов"<<endl;
    for (i=0;i<N;i++) cout<<Y[i]<<" ";
        cout<<endl<<"Сумма элементов массива = "<<Summa<<endl;
fFile = fopen ("DatesOut.dat", "wt");
    for (i=0;i<N;i++)
    {
        for (j=0;j<K;j++) buffer[j]=' ';
            gcvt(X[i],4,buffer);
        fwrite(&buffer, sizeof(buffer), 1, fFile);
    }
gcvt(Summa,4,buffer);
fwrite(&buffer, sizeof(buffer), 1, fFile);
fclose(fFile);
cout<<"Для выхода из программы нажмите любую клавишу"<<endl;
getch();
return 0;
}

```

Додаток В. Графічна інтерпретація результатів