

Міністерство освіти і науки України

Харківський національний автомобільно-дорожній університет

Кафедра інформаційних технологій та мехатроніки

## МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсового проекту

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

Для іноземних студентів

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

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

**Напряму** 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

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.

## СОДЕРЖАНИЕ

|  |    |
|--|----|
| ВСТУПЛЕНИЕ .....   | 4  |
| 1. ЦЕЛИ И ЗАДАНИЯ КУРСОВОГО ПРОЕКТА .....  | 5  |
| 2. ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОГО ПРОЕКТА .....                                      | 6  |
| 2.1. Формулировка требований к программе .....                                     | 6  |
| 2.2. Проектирование программы .....  | 6  |
| 2.3. Кодировка программы .....   | 6  |
| 2.4. Тестирование программы .....  | 7  |
| 2.5. Объяснительная записка .....  | 7  |
| 3 СОДЕРЖАНИЕ РАЗДЕЛОВ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ<br>К КУРСОВОМУ ПРОЕКТУ .....           | 7  |
| 3.1. Перечень необходимых разделов объяснительной записки.....                     | 7  |
| 3.2. Рекомендации по созданию разделов объяснительной записки.....                 | 8  |
| 4. ТРЕБОВАНИЯ К СОЗДАВАЕМОМУ ПРОГРАММНОМУ<br>ПРОДУКТУ .....                        | 12 |
| 5. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ<br>ЗАПИСКИ К КУРСОВОМУ ПРОЕКТУ .....      | 14 |
| 5.1. Общие требования .....  | 14 |
| 5.2. Структурные элементы и разделы .....  | 15 |
| 5.3. Заглавия .....  | 15 |
| 5.4. Нумерация страниц .....   | 15 |
| 5.5. Нумерация разделов и подразделений .....                                      | 15 |
| 5.6. Перечни .....   | 16 |
| 5.7. Иллюстрации .....   | 16 |
| 5.8. Таблицы .....   | 16 |
| 5.9. Формулы и уравнения .....   | 16 |
| 5.10. Ссылка .....   | 17 |
| 5.11. Сокращение .....   | 17 |
| 5.12. Перечень ссылок .....  | 17 |
| 5.13. Дополнения .....   | 17 |
| 6. ЗАЩИТА КУРСОВОГО ПРОЕКТА И ЕЕ ОЦЕНКА .....                                      | 18 |
| 7. ПЕРЕЧЕНЬ ИСТОЧНИКОВ, РЕКОМЕНДОВАННЫХ ДЛЯ<br>ВЫПОЛНЕНИЯ КУРСОВОГО ПРОЕКТА .....  | 19 |
| ПРИЛОЖЕНИЕ А. Бланк титульной страницы записки .....                               | 20 |
| ПРИЛОЖЕНИЕ Б. Образец страницы задания для курсового<br>проектирования .....       | 21 |
| ПРИЛОЖЕНИЕ В. Пример содержания реферата .....                                     | 22 |
| ПРИЛОЖЕНИЕ Д. Пример оформления пояснительной записки к<br>курсовому проекту ..... | 23 |

## ВСТУПЛЕНИЕ

Изменения, происходящие сегодня в разных сферах профессиональной и общественной жизни, связаны с широким внедрением средств информационных и коммуникационных технологий. Они требуют от специалиста в отрасли компьютерных наук умения разрабатывать программное обеспечение для решения задач науки, техники, экономики и управления; создавать и использовать информационные модели процессов и явлений; использовать информационные технологии в проектно-конструкторской, управленческой и финансовой деятельности.

Одной из дисциплин, которые формируют теоретическую и практическую базу в профессиональной подготовке будущих системных инженеров, является программирование.

Курсовой проект (КП) - это первый самостоятельный научный труд будущего системного инженера. При его выполнении студенты углубляют знания из фундаментальных и профессионально-ориентированных дисциплин, усваивают методику экспериментальных исследований, овладевают навыками сопоставления результатов своих исследований с литературными данными, анализа, обобщения и литературного оформления полученных результатов из темы исследования, приобретают умение вести научный поиск, который развивает у молодого инженера творческий подход к работе.

Главная производственная цель проекта - создание качественного рабочего программного обеспечения (ПО). Педагогическая цель работы над проектом - творческое применение всех знаний и умений, полученных студентом за время учебы, в процессе решения реальной задачи относительно создания ПО. Цель работы над курсовым проектом, который имеет исследовательский характер, - углубление знаний из программирования. Структура курсового проекта должна отвечать стандартам и правилам, принятым для НИР (научно-исследовательских работ).

В КП отсутствуют такие важнейшие этапы создания программного обеспечения, как экономическое обоснование и составление бизнес-проектов, учитывая ограниченность решаемых заданий и небольшие размеры программного продукта, создаваемого в рамках курсового проектирования.

Все этапы создания программного обеспечения в обязательном порядке документируются и составляют основу пояснительной записки (ПЗ) к курсовому проекту. Проектные решения, то есть пакет выполняемых программ, исходный код и блок-схема алгоритма оформляются как приложения к ПЗ.

В настоящем методическом руководстве описанные элементы структуры курсового проекта и данные рекомендации по его созданию:

- общий порядок работы над проектом;
- составные части проекта и требования к ним;
- структура и содержание Пояснительной записки и входящих и нее документов;
- стандарты на оформление документации;

- рекомендации относительно организации работ над созданием программного обеспечения и теоретическими исследованиями;

- рекомендации из использования средств и инструментов создания ПО.

Методическое пособие является документом, который регламентирует работу преподавателей и студентов над курсовым проектом по дисциплине "Программирование" (раздел "Структурное программирование на языке C++ в среде Microsoft Visual Studio 2010") для специальности 7.05020101 "Компьютеризованные системы управления и автоматика" направление подготовки 6.050201 "Системная инженерия". Студент должен при выполнении курсового проекта использовать приобретенные теоретические знания для решения конкретных прикладных задач. Основная цель выполнения данного курсового проекта - закрепление на практике умения использовать математические методы расчетов и методы проектирования программ и на базе разработанных алгоритмов уметь разрабатывать код программы, используя интегрированную среду языка программирования (редактирование, компилирование, выполнение), и получать результаты работы программы.

## 1. ЦЕЛИ И ЗАДАНИЯ КУРСОВОГО ПРОЕКТИРОВАНИЯ

Курсовой проект из дисциплины "Программирования" (первый раздел "Структурное программирование на языке C++ в среде Microsoft Visual Studio 2010") предусматривает выполнение комплексного задания научно - исследовательского характера.

Целью курсового проектирования является закрепление, углубление и обобщение знаний, которыми студент овладел во время изучения этой части курса, в приобретении навыков использования основ алгоритмизации и программирования на алгоритмических языках высокого уровня с использованием принципов объектно-ориентированного проектирования и программирования, методики разработки программных модулей и программного обеспечения компьютерных систем управления, приобретения студентами теоретических знаний и практических навыков в области использования современных систем визуального проектирования программных средств компьютерных систем управления и освоения принципов и методов современных технологий программирования, приобретения профессиональных навыков и практики научных исследований с оформлением соответствующего текстового, программного и иллюстративного материала в форме научного отчета.

В результате изучения дисциплины и работы над курсовым проектом студенты должны **знать:**

- средства языка, синтаксис, семантические и функциональные свойства C++;
- основы работы с прикладными программами;
- этапы подготовки и решения задач на компьютере;
- типичные конструкции алгоритмов;
- принципы структурного и процедурного программирования;
- правила использования функций;
- принципы консольного и файлового ввода и вывода данных;
- основные средства разработки программ с использованием внешних файлов;

**уметь:**

- на базе описания условия задачи, используя математические методы расчетов и методы проектирования программ разрабатывать алгоритмы на современном языке программирования в условиях компьютеризированного рабочего места;
- на базе разработанных алгоритмов уметь разрабатывать код программы, используя интегрированную среду языка программирования (редактирование, компилирование и выполнение) и получать результаты работы программы.

Во время выполнения курсового проекта студент должен продемонстрировать:

- умение собирать и анализировать соответствующие материалы об объекте исследования, используя современные источники информации, включая Интернет ресурсы;
- возможность разрабатывать необходимые обоснования для разработки программных средств систем управления разного назначения;
- способность доводить решение поставленной задачи до логического конца;
- умение анализировать полученные результаты и делать из них выводы.

Курсовой проект является самостоятельной работой студента.

Ответственность за правильность аналитических выводов, результатов расчетов и моделирования, а также оформление несет студент, который является автором КП.

## **2. ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОГО ПРОЕКТА**

Основные этапы выполнения курсового проекта:

- а) формулировка требований к создаваемой программе;
- б) проектирование программы;
- в) кодировка программы;
- г) тестирование программы;
- д) оформление пояснительной записки;
- е) защита работы.

### **2.1. Формулировка требований к создаваемой программе**

Формулировка требований является самостоятельным этапом работы и завершается созданием двух документов: индивидуального плана проекта и раздела объяснительной записки "Спецификация программы".

Индивидуальный план складывает студент в начале работы над проектом и копию плана в то же время предоставляет преподавателю. За существенное отставание от календарного плана преподаватель должен снижать оценку курсовой работы.

На этапе формулировки требований студент должен:

- 1) сформулировать назначение программы;
- 2) определить перечень функций, которые выполняет программа;
- 3) предложить интерфейс пользователя, который предоставляет возможность последнему использовать эти функции.

Спецификация программы состоит из последовательного описания всех выделенных функций, и связанного с ними графического интерфейса пользователя.

В спецификации программы надо обязательно предоставить изображение всех окон программы.

## **2.2. Проектирование программы**

На этапе проектирования определяется архитектура и структура программы, а также формат производных, выходных и внутренних данных. Соотношение между блоками и частями программы отображается с помощью блок-схемы алгоритма решения задачи. Во время проектирования нужно разработать план функционального тестирования и необходимые тесты.

Завершается проектирование созданием проектной спецификации программы.

## **2.3. Кодировка программы**

На этапе кодировки создается код программы. Он должен быть надлежащим образом прокомментирован и полностью удовлетворять требованиям к программному коду, приведенным в этих указаниях.

Весь код программы должен быть приведен в приложении к объяснительной записке. В тексте основных разделов записки допустимо приводить лишь небольшие по размеру отдельные фрагменты кода.

## **2.4. Тестирование программы**

Различают две основных разновидности тестирования: функциональное и модульное.

Функциональное тестирование проверяет, как программа выполняет свои функции в условиях корректных и некорректных действий пользователя. Оно планируется на этапе проектирования и выполняется по мере того, как кодируются отдельные функции программы и перед защитой программы.

Модульное тестирование выполняется одновременно с кодировкой и имеет целью оперативную проверку отдельных частей программы. Выполнение модульного тестирования весьма желательно, но не является обязательным.

## **2.5. Пояснительная записка**

Оформление пояснительной записки является последним этапом выполнения курсового проекта и должно предоставить студенту навыков документирования программного продукта.

# **3. СОДЕРЖАНИЕ РАЗДЕЛОВ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОМУ ПРОЕКТУ**

## **3.1. Перечень необходимых разделов объяснительной записки**

Пояснительная записка к курсовому проекту должна содержать следующие составные элементы:

**Титульный лист.**

**Задание на курсовое проектирование.**

**Реферат.**

**Содержание.**

**Вступление.**

**Раздел 1. Обоснование алгоритма и структуры программы.**

**1.1. Постановка и математическое описание задачи.**

**1.2. Описание программы; определения и описание функций.**

**1.3. Разработка и обоснование алгоритма и структуры программы.**

**Раздел 2. Разработка программы.**

**2.1. Спецификация программы.**

**2.2. Разработка программы.**

**2.3. Проектирование и описание интерфейса пользователя.**

**2.4. Файловая структура проекта. Описание файлов.**

**Раздел 3. Тестирование программы и результаты ее выполнения.**

**Выводы.**

**Литература.**

**Приложение А. Блок-схема алгоритма программы.**

**Приложение Б. Листинг (код с комментариями) программы.**

**Приложение В. Графическая интерпретация результатов.**

### **3.2. Рекомендации по созданию разделов Пояснительной записки**

#### **3.2.1. Титульный лист**

Титульный лист оформляется в соответствии с общими университетскими требованиями и содержит названия министерства, университета, факультета и кафедры, тему проекта, группу и фамилию студента, должность и фамилию руководителя, год выполнения проекта.

Пример оформления титульного листа приведен в Приложении А. Титульный лист и Задание должны быть подписаны студентом и руководителем курсового проектирования.

#### **3.2.2. Задание**

Задание оформляется на стандартном бланке или печатается по той же форме (см. Приложение Б).

#### **3.2.3. Реферат**

Реферат содержит краткую характеристику объема проекта (количество страниц, таблиц, рисунков, литературных источников, дополнений), перечень ключевых слов и аннотацию к проекту (см. Приложение В).

В тексте реферата кратко описывают разработанные в проекте технические, организационные, экономические решения, а также устройства, их эффективность и мера внедрения. В лаконичной форме должно быть представлено описание основных частей проекта. В утвердительной форме отмечается, что было выполнено, рассчитано, сконструировано, предложено, модернизировано и так далее. Реферат рекомендуется составлять после написания пояснительной записки проекта.



Перечень ключевых слов должен характеризовать содержимое реферируемой Пояснительной записки и включать от 5 до 15 ключевых слов в именительном падеже, изложенных в строки через запятые.

Ключевые слова в совокупности должны вне текста давать достаточно полное представление о содержимом записки и обеспечивать возможность информационного поиска.

Объем реферата не больше 2000 знаков и не должен превышать одну страницу. Пример содержания реферата приведен в Приложении В.

Реферат является первой текстовой страницей пояснительной записки и выполняется на листе формата А4 с основной надписью согласно ГОСТ 2.104-68 ЕСКД или по форме 5 ГОСТ 21.101 СПДС и считается третьей страницей Пояснительной записки.

#### **3.2.4. Содержание**

Содержание является обязательным разделом пояснительной записки к курсовому проекту, который включает названия разделов и подразделов записки (начиная со вступления), список использованной литературы, Приложения и их наименования, а также номер страницы, на которой размещен или начинается раздел или дополнение. Содержание Пояснительной записки размещают после титульной листа и реферата и включают в общую численность страниц объяснительной записки.

#### **3.2.5. Вступление**

Вступление характеризует актуальность темы курсового проекта, современное состояние области знаний, которая исследуется и краткую характеристику курсового проекта (1-2 с.).

#### **3.2.6. Постановка и математическое описание задачи**

Постановка задачи - это словесная ее формулировка, которая берется из индивидуального задания. Здесь можно указать допущения, при которых решается задача, исходные данные, порядок обработки результатов и тому подобное.

Математическое описание задачи - это перечень математических выражений, формул, уравнений, констант и математических отметок, с помощью которых могут быть получены результаты решения задачи. Это математические соотношения, которые приведены в индивидуальном задании, а также те, которые исполнитель проекта использовал из технической литературы.

Основной целью данного раздела Пояснительной записки является анализ требований задания на курсовую работу и формулировка дополнительных технических требований, непосредственно вытекающих из него, и цели исследования.

На основе анализа нужно конкретизировать основные параметры программы, ее возможности и функции, средства их реализации, интерфейс с пользователем, среду, в которой программа может работать.

Если в предыдущем разделе лишь формулируют общую методику решения задач, то здесь определяют конкретный план работы. Простое переписывание требований Задания к курсовому проекту не допускается, нужно творчески

осмыслить Задание и сформулировать конкретные технические требования и пути их достижения.

### 3.2.7. Описание программы, определение и описание функций

Описание программы - это сведения о логической структуре и характере функционирования программы. Описание программы включает в себе:

1) общие сведения (обозначение и наименование программы; программное обеспечение, необходимое для функционирования программы; язык программирования, на котором написана программа);

2) функциональное назначение (назначение программы; сведения о функциональных ограничениях);

3) описание логической структуры (алгоритм решения задачи; методы, которые используются; структура программы с описанием функций составных частей и связей между ними; связи программы с другими программами). Описание логической структуры программы выполняется с учетом текста программы на языке программирования;

4) технические средства (типы ЭВМ и устройств), которые используются при работе программы;

5) вызов и загрузка программы (способы вызова программы; последовательность работы; сообщения, которые кажутся программой);

б) исходные данные (характер, организация и предыдущая подготовка начальных данных; формат, описание и способ кодировки данных);

7) промежуточные данные и результаты (характер и организация данных; формат, описание и способ их кодировки).

При описании программы допускается содержание подраздела сопроводить объяснениями, таблицами, схемами и графикой.

Кроме того, в данном подразделе должен быть приведен перечень разработанных исполнителем курсового проекта функций, используемых для декомпозиции задачи при ее решении. В этом перечне должны быть указаны задачи, которые решаются с помощью этих функций.

### 3.2.8. Разработка и обоснование алгоритма и структуры программы

В данном подразделе отчета даются определения последовательности действий, определяются этапы решения задачи, которые требуют разветвления, циклы, оформления процедур или функций.

При использовании функций обязательно указывается назначение функции и ее математическое отображение. Например, функция вычисления факториала  $Y=X!$ . Кроме того, указывается метод решения задачи, рекуррентные формулы и тому подобное. Например, вычисление факториала сводится к вычислению следующего произведения:

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

Вычисление произведения выполняется по следующим рекуррентным соотношениям:

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

Разрабатывать алгоритм будет намного легче, если есть определенное содержание и структура файлов исходных данных и результатов.

Файлы начальных данных и результатов должны иметься в наличии и быть простыми по структуре. Их логическое построение должно отвечать содержанию задачи, структурам и типам данных.

После этого разрабатывается блок-схема алгоритма решения задачи с использованием символов, определенных стандартами ДСТУ 19.002-90 и 19.003-90. Схемы алгоритмов разрабатываются для каждого дополнительного блока (функции) и для основной программы.

При разработке алгоритма надо использовать пошаговый способ построения алгоритма.

На первом шаге разрабатывается общая схема решения задачи, то есть на первом шаге надо ответить на вопрос, что необходимо выполнить для решения поставленной задачи.

На втором шаге переходят к детализации каждого блока общей схемы, то есть необходимо выяснить и решить, как реализовать каждый блок общей схемы для решения задачи.

Результатом второго (третьего, четвертого и далее) шага будет детальный алгоритм (один блок - одно действие) решения задачи.

В пояснительной записке на листах формата А4 отображается схема алгоритма, которая разработана на последнем шаге.

Объем раздела – 3-8 страниц.

### **3.2.9. Спецификация программы**

Данный подраздел имеет большое значение для разработки программы решения задачи. В нем должно быть осуществлено присваивание всем объектам задачи имен согласно правилам алгоритмического языка C++. Кроме того, необходимо определить, какие данные будут исходными, какие - промежуточными, а какие - результатами.

При решении задач разработчик должен определить, какие данные, приведенные в задаче, целесообразно использовать в качестве константы, а какие данные - как переменные. Надо хорошо уяснить, что для того, чтобы программа была универсальной, то есть позволяла производить вычисления для различных исходных данных, необходимо использовать переменные, а не константы. Как правило, в качестве константы используются физические, математические и логические константы. При этом если имя константы не может быть представлено латинской буквой, то надо найти в латинском алфавите или близкую по звучанию букву, или соединение букв, которое бы несло в себе смысл этой константы.

Далее необходимо определить, нужны ли при решении задачи нестандартные типы данных. Если нужны, то требуется определить имена этих типов и их структуру.

После этого следует перейти к описанию переменных, то есть разработать спецификацию на программу (функцию). Описание переменных целесообразно

распределить на описание начальных, промежуточных и переменных, которые характеризуют результаты решения задачи. Пример спецификации программы приведен в таблице 1.

Если при решении задачи применяются функции, их также надо перечислить и специфицировать:

- формальные параметры и их форма применения (параметр-значение или параметр-переменная);
- способ применения переменных, которые используются в процедуре или функции (локальные или глобальные).

В таблице 1 для примера приведена спецификация некоторой программы.

Таблица 1.  
программы

Спецификация

| Назначение данных | Математическое обозначение | Имя в программе | Структура данных | Тип данных | Смысл переменной     |
|-------------------|----------------------------|-----------------|------------------|------------|----------------------|
| Начальная         | $\alpha$                   | Alfa            | простая          | float      | Аргумент функции     |
| Начальная         | $\pi$                      | PI              | простая          | float      | Константа            |
| Промежуточная     | $i$                        | I               | простая          | int        | Параметр цикла       |
| Результат         | $x(i)$                     | ArrayX          | массив           | float      | Сопротивление в цепи |

### 3.2.10. Разработка программы

**Это подразделение является основным в курсовой работе. Его объем - 5-10 страниц.**

В зависимости от избранного стиля программирования, возможные разные подходы к разработке программы. В самых простых случаях она идет последовательно, алгоритм программы реализуется с помощью последовательности операторов, включительно с условными и операторами цикла.

В структурном программировании применяется иерархический подход, в котором отдельные структурные единицы превращаются в функции, которые могут принимать параметры и возвращать результат. Описание функции в таком случае выполняется один раз в определенном месте программы, а используется она многократно, при этом лишь изменяются параметры, и не нужно каждый раз повторять одни и те же операторы, которые включены в функцию.

### 3.2.11. Описание файлов данных и интерфейса программы

Если программа использует файлы как источник входных данных или для хранения промежуточных или конечных результатов работы, то в данном подразделении следует привести описание формата этих файлов. Он может быть выполнен в текстовом, табличном или графическом виде.

Можно добавлять к нему примеры реальных файлов с данными программы.

### **3.2.12. Тестирование программы и результаты ее выполнения**

В данном разделе надо описать методику тестирования программы, тесту данные и привести результаты работы программы. Если программа работает в графическом режиме, то следует распечатать копию графического окна программы. Если результатом работы программы является текстовый файл, то необходимо вывести содержимое этого файла и содержимое неосновных диалоговых окон можно подать в текстовом виде.

Если для отражения работы программы необходимо значительное количество распечаток, то их можно подать в дополнениях.

Объем раздела — 1.5 страниц. В разделе нужно сделать вывод, который подтверждает (или отрицает) работоспособность программы.

### **3.2.13. Выводы**

В выводах по КП в реферативной форме должны быть описанные результаты, полученные студентом на каждом из этапов выполнения работы (аналитическому, этапу проектирования ПО, экспериментальному исследованию, анализу полученных результатов), а также выводов относительно достижения цели курсового проектирования, перспективы развития данной отрасли и тому подобное. Выводы пишутся на 1-2 страницы и как раздел не нумеруются.

### **3.2.14. Литература.**

В этом разделе приводится перечень источников, на которые были ссылки в тексте.

Список должен формироваться в порядке ссылок за текстом и вмещать библиографические сведения официально изданных книг, статей, патентов, депонируемых рукописей и тому подобное.

Как раздел, перечень литературы не нумеруется. Формат описания литературных источников должен отвечать ДСТ 7.1.-84.

### **3.2.15. Дополнения**

В этот раздел включают блок-схемы программ и их листинги, а также другие документы. Кроме этого, в дополнения помещиваются таблицы, графика и методики, которые по каким-то причинам не вошли в объяснительную записку, но нужны для объяснений.

## 4. ТРЕБОВАНИЯ К СОЗДАВАЕМОМУ ПРОГРАММНОМУ ПРОДУКТУ

### 4.1. Общие требования к программе

Независимо от темы курсового проекта разрабатываемая программа должна удовлетворять следующим общим требованиям.

**1. Устойчивость программы.** Программа не должна терять работоспособность ни при каких, даже некорректных, действиях пользователя. Любые действия, угрожающие потерей информации, должны быть подтверждены пользователем. Информация, которая вводится, должна проходить логический контроль.

**2. Обеспечение целостности данных.** При любых действиях пользователя не должны теряться данные или их целостность (некорректность индексов, потеря ссылок в связках после удаления-добавления записей и т. д.).

**3. Функциональная полнота.** Должны быть реализованы все функции, указанные в спецификации программы.

**4. Термины и интерфейс.** В диалоговых окнах употребляются только термины, понятные пользователю, и не употребляются термины разработчика («запись», «индексация» и т. д.). Появление служебных англоязычных сообщений недопустимо. В сообщениях пользователя следует придерживаться норм вежливости, цветовая гамма должна подражать общепринятым рекомендациям.

**5. Использование клавиатуры.** На каждом этапе нажатие любой клавиши должно или игнорироваться, или вызывать предусмотренные действия, описанные в средствах помощи. Привязка действий к клавишам должна быть общепринятой: **F1** - помощь; **Enter** - согласие, завершение ввода; **Esc** - отказ, возвращение к предыдущему узлу ветки алгоритма (с возобновлением экранной формы); **Tab** - переход к следующему полю, окну и т. д.; **Shift + Tab** - возвращение к предыдущему полю и т., д.

### 4.2. Требования к программному коду

Программа, подлежащая разработке, должна быть компактной и «читаемой». Такие программы удобны при изучении и эксплуатации.

Программа должна составляться так, чтобы ее статичная форма максимально точно отражала динамику ее выполнения.

Программа представляет собой последовательность операторов алгоритмического языка программирования. Каждый из них предназначен для детализации определенного действия. Например, с помощью оператора присваивания осуществляется перевычисление выражения и присваивание полученного результата переменной; с помощью условного оператора осуществляется разветвление, и тому подобное. Значит, программа будет простой для изучения, если последовательность выполнения операторов близка к последовательности размещения этих операторов в ее тексте.

В этом случае проверка программы (особенно ее логики) не вызывает затруднений, так как операторы выполняются в порядке записи. Это облегчает отладку программы и поиск ошибок.

Какие условия надо выполнять для этого?

Во-первых, для написания программы надо использовать операторы, с помощью которых реализуются стандартные блоки алгоритма. Например:

- Оператор присваивания - для реализации функциональных блоков.
- Условный оператор - для реализации блоков "Выбор".
- Оператор цикла - для реализации блоков "Повторение".

Во-вторых, при составлении программы надо стремиться к тому, чтобы как можно реже использовать оператор безусловного перехода - **goto**, поскольку его применение ведет к усложнению логики и структуры программы.

Таким образом, для того, чтобы программа была простою в изучении, она должна сочетать представление ее текста в пространстве (на бланке, в памяти ПК) и выполнение операторов в реальном времени в том порядке, в котором они записаны.

Требования к читаемости являются одними из важнейших, потому что характеризуют качество оформления программы.

Как добиться этого?

1. Вид программы должен полностью совпадать с алгоритмом. Это достигается, если каждому стандартному блоку алгоритма отвечает свой оператор в программе и по месту размещения, и по времени выполнения. Например, блоку повторения должен отвечать оператор цикла, а не группа операторов, в которой цикл организован с помощью условного оператора и оператора безусловного перехода.

2. Все обозначения в алгоритме и программе должны совпадать. Например, переменные  $x$ ,  $x_0$ ,  $\pi$  алгоритму целесообразно обозначить в программе  $x$ ,  $x_0$ ,  $\pi$ .

3. При записи программы на бланке надо руководствоваться рациональным стилем отображения. То есть в каждой строке необходимо размещать только один оператор. Если оператор содержит внутренние операторы, например, операторы цикла, то их следует записывать в отдельных строках со сдвигом вправо на одну или несколько позиций.

4. Логические участки программы, а также операторы цикла, условия и подобные нужно выделять с помощью табулятора, чтобы с первого взгляда на программный код было понятно структуру программы.

5. Все операторы целесообразно сопровождать комментариями.

## **5. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

### **5.1. Общие требования**

Общими требованиями к пояснительной записке являются:

- логическая последовательность подачи материала;
- сжатость, четкость и конкретность подачи теоретических и практических результатов работы
- доказательность выводов и обоснованность рекомендаций.

Пояснительная записка оформляется студентом самостоятельно и печатается на принтере на одной стороне листа белой бумаги формата А4 (210 x 297 мм) в соответствии с [1]. Приведем некоторые положения этих документов.

Поля: верхнее и правое – 1 см, левое – 2,5 см, нижнее – 2,5 см. Размер листов – А4 (21×29,7 см), ориентация – книжная, положение переплета – слева.

При создании электронной версии пояснительной записки необходимо руководствоваться следующим:

- рукопись должна быть подготовлена в текстовом редакторе MS Word с шрифтом Times New Roman, размер шрифта для основного текста - 14;
- рисунки (схемы алгоритмов) необходимо строить в текстовых или графических редакторах;
- единичный интервал между строками;
- форматирование текста – по ширине;
- абзацный отступ – 1,25 см.

Форматирование колонтитулов: от края к верхнему колонтитулу - 0 см, от края к нижнему колонтитулу – 0,7 см. Высота верхнего колонтитула – 1 см.

Вертикальное выравнивание текста – по верхнему краю страницы. Нужно также пытаться рационально располагать текст в пределах страницы таким образом, чтобы во время вкладывания иллюстраций и таблиц к тексту документа внизу таблицы не оставалось много свободного места (больше 3 см).

Фамилии, названия программных приложений, организаций, фирм и другие имена собственные приводятся языком оригинала.

Пояснительная записка должна быть грамотно написана и аккуратно оформлена. Проверка правописания производится во вкладке Рецензирование меню Word.

Отдельные листы должны быть скреплены между собой (прошиты). Кроме того, отчет предоставляется студентами и в электронном виде (на дискете или флэш-памяти).

## **5.2. Структурные элементы и разделы**

Надо отличать структурные элементы записки, такие как "РЕФЕРАТ", "СОДЕРЖАНИЕ", "ВСТУПЛЕНИЕ" и разделы записки. Некоторые элементы, а именно "РЕФЕРАТ", "СОДЕРЖАНИЕ", "ВСТУПЛЕНИЕ", "ВЫВОДЫ", "ПЕРЕЧЕНЬ ССЫЛОК" создают разделы записки и их наименования служат заглавиями отдельных разделов. Такие разделы не имеют нумерации.

Основные структурные элементы записки могут разделяться на разделы, которые нумеруются и имеют собственные заголовки.

Разделы могут разделяться на подразделы, пункты и подпункты.

Шрифт заголовков Times New Roman, размер шрифта для - 14.

## **5.3. Заголовки**

Разделы и подразделы должны иметь заголовки, пункты и подпункты могут иметь заголовки. Заголовки структурных элементов записки и заголовки разделов необходимо располагать посередине строки и печатать прописными буквами без точки в конце, не подчеркивая. Заголовки подразделов, пунктов и подпунктов записки необходимо начинать с абзацного отступа и печатать малыми буквами, кроме первой прописной, не подчеркивая, без точки в конце. Абзацный отступ должен быть одинаковым по всему тексту записки и равняться 1,25 см. Если заголовок состоит из двух или более предложений, их разделяют точкой.

Переносы слов в заголовках разделов не допускаются. Расстояние между заголовком и следующим или предыдущим текстом должно быть не менее двух строк.

Не допускается размещать наименования раздела и подраздела, а также пункта



и подпункта в нижней части страницы, если после него расположена только одна строка текста.

Шрифт заголовков Times New Roman, размер шрифта для - 14.

#### **5.4. Нумерация страниц**

Нумеровать страницы необходимо арабскими цифрами, соблюдая сквозную нумерацию по всему тексту. Номер страницы проставляют в правом верхнем углу без точки в конце.

#### **5.5. Нумерация разделов и подразделов**

Разделы, подразделы, пункты и подпункты пояснительной записки надлежит нумеровать арабскими цифрами. Разделы пояснительной записки должны иметь порядковую нумерацию в пределах содержания пояснительной записки и обозначаться арабскими цифрами без точки, например, 1, 2, 3. Подразделы должны иметь порядковую нумерацию в пределах каждого раздела. Номера подразделов должны состоять из номеров разделов и подразделов, которые разделяются точкой, например, 1.1, 1.2, 1.3.

#### **5.6. Перечни**

При необходимости могут быть использованы перечни. Перед перечнем ставят двоеточие. Перед каждой позицией перечня надо ставить малую букву алфавита со скобкой или дефис (первый уровень детализации). Для дальнейшей детализации перечня надо использовать арабские цифры со скобкой (второй уровень детализации).

#### **5.7. Иллюстрации**

Иллюстрации надо располагать в пояснительной записке непосредственно после текста, в котором они впервые упоминаются, или на следующей странице. На все иллюстрации должны быть ссылки в объяснительной записке.

Иллюстрации могут иметь название, которое располагают под иллюстрацией. При необходимости под иллюстрацией еще располагают пояснительные данные. Иллюстрация обозначается словом "Рисунок", которое вместе с названием помещают после пояснительных данных, например, "Рисунок 2.1 - Схема алгоритма".

Иллюстрации необходимо нумеровать арабскими цифрами в пределах раздела, за исключением иллюстраций, приведенных в дополнениях. Номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой. Если иллюстрация не вмещается на одной странице, можно переносить ее на другие страницы, при этом название иллюстрации располагают на первой странице, объяснительные данные - на каждой странице, и под ними указывают: "Рисунок, лист ".

#### **5.8. Таблицы**

Таблицы необходимо располагать в объяснительной записке непосредственно после текста, в котором они впервые упоминаются, или на следующей странице. На все таблицы должны быть ссылки в пояснительной записке. Таблица может иметь

название, которое печатают строчными буквами (кроме первой прописной) и размещают над таблицей. Если строки или графы таблицы выходят за формат страницы, таблицу разделяют на части, располагая одну часть под другой, или рядом, или переносят части таблицы на следующую страницу. Слово "Таблица", указывают один раз слева над первой частью таблицы, над другими частями пишут: "Продолжение таблицы" с указанием номера таблицы. Таблицы надо нумеровать арабскими цифрами в пределах раздела, за исключением таблиц, приведенных в дополнениях. Номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой.

### **5.9. Формулы и уравнения**

Формулы и уравнения располагают в пояснительной записке непосредственно после текста, в котором они впервые упоминаются, на середине страницы. Выше и ниже каждой формулы должно быть оставлено не меньше одной свободной строки.

Формулы и уравнения надо нумеровать арабскими цифрами порядковой нумерацией в пределах раздела, за исключением формул и уравнений, приведенных в приложениях. Номер формулы или уравнения состоит из номера раздела и порядкового номера формулы или уравнения, разделенных точкой.

Номер формулы или уравнения указывают на уровне формулы или уравнения в скобках в крайнем правом положении на строке.

Объяснение символов и числовые коэффициенты, которые входят в формулу или уравнение, необходимо делать непосредственно после формулы или уравнения в той последовательности, в которой они предоставлены в формуле или уравнении.

Объяснение значения каждого символа или числового коэффициента необходимо давать с новой строки. Первую строку объяснения начинают с абзаца словом "где" без двоеточия.

Переносить формулы или уравнения на другую строку допускается только на знаках выполняемых операций, причем знак операции в начале такой строки повторяют. При переносе формулы или уравнения на знаке операции умножения применяют знак "×". Формулы, которые располагаются одна за другой, и не разделены текстом, отделяют запятой.

### **5.10. Ссылки**

Ссылки в тексте на источники необходимо приводить в последовательности их упоминания в записке, указывая порядковый номер, выделенный двумя квадратными скобками, например: [5], [4,7] или [2-8].

### **5.11. Сокращения**

В тексте пояснительной записки допускается делать только общеупотребляемые сокращения (например, "и т.д., и т.п., и др.") или распространенные аббревиатуры.

Если в пояснительной записке принята специфическая терминология или употребляются малораспространенные сокращения, новые символы, обозначения и др., то их перечень необходимо предоставить в отдельном списке. Перечень должен располагаться столбиком, в котором слова по алфавиту приводятся сокращенно, а рядом - их подробное толкование.

Если специальные символы, термины, сокращения и обозначения приводятся менее трех раз, перечень их не составляется, а их толкование делают в тексте при первом упоминании.

### **5.12. Перечень ссылок**

Перечень ссылок оформляется как раздел объяснительной записки из новой страницы в соответствии с требованиями ДСТУ 7. 1-84 "Библиографическое описание документа. Общие требования и правила составления".

Для оформления библиографических ссылок и правильной записи общепринятых сокращений рекомендуется использовать методические указания [3], которые составлены с учетом соответствующих стандартов.

### **5.13. Приложения**

Приложения необходимо оформлять как продолжение пояснительной записки на следующих ее страницах или в отдельной части, располагая приложения в порядке появления ссылок на них в тексте пояснительной записки. Каждое приложение должно начинаться с новой страницы и иметь заголовок, напечатанный сверху строчными буквами (первая – прописная), симметрично относительно текста страницы.

На середине строки над заголовком строчными буквами с первой прописной должно быть напечатано слово "Приложение " и прописная буква, которая помечает приложение. Приложения следует помечать последовательно прописными буквами русского алфавита, за исключением букв Й, О, Ч, Ъ. Одно приложение обозначается как А.

Приложения должны иметь общую с другой частью пояснительной записки сквозную нумерацию. Текст каждого приложения при необходимости может быть разделен на подразделы и пункты, которые нумеруются арабскими цифрами в пределах каждого приложения.

## **6. ОРГАНИЗАЦИЯ ПРОЕКТИРОВАНИЯ И ЗАЩИТА ПРОЕКТА**

Курсовой проект принадлежит к форме самостоятельной работы студентов. Выдача заданий осуществляется в первые две недели семестра. Каждый студент составляет календарный план работы над темой и утверждает его у преподавателя. Время консультаций согласуется с преподавателем. Студенты консультируются у руководителя индивидуально. Основные этапы работы:

- ознакомление с литературными источниками по тематике поставленной задачи;
- разработка алгоритмов и изучение средств решения задачи курсового проекта;
- написание программы и ее отладка на персональном компьютере;
- получение результатов решения поставленной задачи;
- обсуждение полученных промежуточных теоретических и практических результатов с руководителем проекта.

После показа руководителю готовой программы и ее утверждения студент начинает писать объяснительную записку.

Пояснительная записка предъявляется руководителю для проверки, после которой студент исправляет выявленные недостатки. Руководитель допускает студента к защите и назначает дату и время защиты. Курсовой проект должен быть выполнен и защищен не позже, чем за 1 неделю до начала сессии.

Защита происходит перед комиссией кафедры. Студент должен уметь выделить и четко изложить основные моменты проекта, ответить на поставленные вопросы, довести свою компетентность в области знаний, которая исследуется.

Общая оценка зависит от теоретической подготовки студента в области знаний соответственно заданию, от качества разработанной программы и объяснения ее особенностей, содержания пояснительной записки, полноты проработки теоретического материала, а также от качества оформления записки.

## **7. ПЕРЕЧЕНЬ ИСТОЧНИКОВ, РЕКОМЕНДОВАННЫХ ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОГО ПРОЕКТА**

1. ДСТУ 3008-95 "Документация. Отчеты в сфере науки и техники. Структура и правила оформления" - Государственный стандарт Украины.

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. Танненбаум Э. Современные операционные системы. 3-ое изд. - СПб.: Питер, 2010. - 1120с.

6. Хортон А. Visual C++ 2010. Полный курс. - К.: Диалектика, 2011. - 1205 с.

7. Химмельблау Д. Анализ процессов статистическими методами / Д. Химмельблау. - М.: Мир, 1973. - 957 с.

### **Электронные ресурсы**

1. Страница лектора Симбирского Г.Д. для потока РК в социальной сети "ВК" <http://vc.com/id23791456>. (Материалы для самостоятельной работы, а также лекции, методические указания к лабораторным работам и др.).

2. Симбирский Г. Д. Конспект лекций по дисциплине "Программирование" для студентов направления 6.050201 Системна інженерія спеціальності 6.050201 Комп'ютерні системи управління рухомими об'єктами (раздел "Структурное программирование на C++"). Портал ХНАДУ <http://files.khadi.kharkov.ua/mekhatroniki-transportnikh-zasobiv.html>

3. Симбирский Г. Д. Методические указания к лабораторным работам по дисциплине "Программирование" для студентов направления 6.050201 Системна інженерія спеціальності 6.050201 Комп'ютерні системи управління рухомими об'єктами (раздел "Структурное программирование на C++").- Портал ХНАДУ, <http://files.khadi.kharkov.ua/mekhatroniki-transportnikh-zasobiv.html>

## Приложение А. Бланк титульной страницы курсового проекта

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ  
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ  
УНИВЕРСИТЕТ

Факультет компьютерных технологий и мехатроніки  
Кафедра информационных технологий и мехатроніки

### КУРСОВОЙ ПРОЕКТ

по дисциплине "Программирование"  
(раздел "Структурное программирование на языке C++  
в среде Microsoft Visual Studio 2010")

Тема: **"Разработка программного приложения в среде  
Microsoft Visual Studio 2010 для статистического  
анализа случайных процессов"**

Студента(ки) 1 курса группы РК- 11 факультета КТМ  
направление - 6.050201 "Системная инженерия"  
специальность - 6.050201 "Компьютерные системы  
управления подвижными объектами "

---

(фамилия и инициалы)

Руководитель: доцент, канд.техн.наук Симбирский Г.Д.

Национальная шкала \_\_\_\_\_  
Количество баллов: \_\_\_\_\_ Оценка:  
ЕКТС \_\_\_\_\_

Подпись руководителя

---

Харьков, 2016

## Приложение Б. Образец задания на курсовой проект и календарного плана работ по проектированию

Харьковский национальный автомобильно-дорожный университет  
Факультет компьютерных технологий и мехатроники  
Кафедра информационных технологий и мехатроники

|                        |  |        |        |         |   |
|------------------------|--|--------|--------|---------|---|
| Дисциплина             | Программирования   |        |        |         |   |
| Направление подготовки | 6.050201 "Системная инженерия"                                     |        |        |         |   |
| Специальности          | 6.050201 "Компьютерные системы<br>управления подвижными объектами" |        |        |         |   |
| Курс                   | 1  | группа | РК- 11 | семестр | 2 |

### ЗАДАНИЕ

на курсовое проектирование студента

---

(фамилия, имя, отчество)

1. Тема проекта(работы) **Разработка программного приложения на языке C++ в среде Microsoft Visual Studio 2010 для статистического анализа случайных процессов**
2. Срок сдачи студентом законченного проекта 16.05.2016р.
3. Перечень вопросов, которые подлежат разработке:
  - 3.1. Вычислить массив **Y** действительных чисел из 40 элементов по заданной формуле:

$$Y_i = f(x_i)$$

где  $K = \underline{\hspace{2cm}}$  - 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 их оформит как функции. В дальнейшем все логически завершенные обособленные блоки программы обязательно оформят 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++, оформил 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. Перечень графического материала (с точным указанием обязательных чертежей):
- 4.1. Блок-схема алгоритма решения поставленной задачи
- 4.2. Графики зависимостей  $Y = f(x)$  и  $Y_{\sigma} = f(x)$  в среде **Excel**.

Дата выдачи задания \_\_\_\_\_

Подпись студента \_\_\_\_\_

## Календарный план выполнения курсового проекта

| № | Название этапов курсового проекта  | Срок выполнения этапов проекта | Примечание |
|---|--|--------------------------------|------------|
| 1 | Вычислить массив $Y$ действительных чисел из 40 элементов по заданной формуле.<br>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 неделя                       |            |
| 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 неделя                       |            |
| 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.<br>Массив случайных действительных чисел $Y_{\sigma}$ также записать в соответствующие текстовый и бинарный файлы. | 8 неделя                       |            |
| 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 неделя                      |            |
| 5 | Вычислить среднее значение, дисперсию и среднее квадратичное отклонение случайного процесса $Y_{\sigma} = f(x)$ .  | 12 неделя                      |            |
| 6 | Построить графики зависимостей $Y = f(x)$ и $Y_{\sigma} = f(x)$ при помощи средств языка программирования $C++$ , оформил это построение в виде функции.   | 14 неделя                      |            |

Студент \_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

Руководитель \_\_\_\_\_  
(подпись) \_\_\_\_\_  
(фамилия, инициалы)

“ \_\_\_\_ ” \_\_\_\_\_ 2016 г.



## Приложение В. Пример содержания реферата

### РЕФЕРАТ

Объяснительная записка состоит из: 50 с., 4 рисунков, 3 таблиц, 8 источников.

Объект исследования - модель случайного измерительного процесса.

Цель работы - анализ методов, алгоритмов и средств исследования случайных процессов.

Методы исследования - методы теории алгоритмов и словно программирования, методы статистической обработки информации.

Рассмотрен процесс моделирования стохастической измерительной информации и разработана программа-функция с использованием генератора случайных чисел для превращения эталонного массива данных в случайную (реальную) измерительную информацию.

Разработаны средства вывода массивов данных на консоль и в текстовые и бинарные файлы с использованием динамического программирования, адресации переменных и указателей.

Проведен анализ алгоритмов обработки измерительной информации или другой стохастической информации. Обосновывается выбор методов для статистического анализа результатов исследования стохастических процессов.

Разработано дополнение для визуализации алгоритмов статистической обработки информации.

**Ключевые слова:** программирование, язык C++, среда Microsoft Visual Studio 2010, структурное программирование, стохастические процессы, модель стохастического процесса, статистическая обработка измерительной информации, статистический анализ.

## **Приложение Д. Пример оформления пояснительной записки к курсовому проекту**

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ  
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ  
УНИВЕРСИТЕТ**

**Факультет компьютерных технологий и мехатроники  
Кафедра информационных технологий и мехатроники**

### **КУРСОВОЙ ПРОЕКТ**

по дисциплине "Программирование"  
(раздел "Структурное программирование на языке C++  
в среде Microsoft Visual Studio 2010")

Тема: **"Подготовка и решение прикладных задач на ПК"**

Студента(ки) **1 курса группы РК- 11 факультета КТМ**  
направления **6.050201 "Системная инженерия"**  
специальности **7.05020101 "Компьютеризованные  
системы управления и автоматика"**

          Петренко І.П.  
(фамилия и инициалы)

Руководитель    доцент, канд. техн. наук Смирнов А.П.

Национальная шкала \_\_\_\_\_

Количество баллов: \_\_\_\_\_                   Оценка: ЕКТС \_\_\_\_\_

Подпись руководителя \_\_\_\_\_

Харьков, 2016

|               |  |        |               |         |          |
|---------------|--|--------|---------------|---------|----------|
| Дисциплина    | <b>Программирования.</b>   |        |               |         |          |
| Направление   | <b>6.050201 Системная инженерия</b>                                  |        |               |         |          |
| Специальность | <b>6.050201 Компьютерные системы управления подвижными объектами</b> |        |               |         |          |
| Курс          | 1  | группа | <b>РК- 11</b> | семестр | <b>2</b> |

**ЗАДАНИЕ**  
на курсовой проект студента

**Петренко Иван Петрович**

1. Тема проекта **Разработка программного дополнения в среде Microsoft Visual Studio 2010 для статистического анализа стохастических процессов**
2. Срок сдачи студентом законченного проекта (работы) **12.05.2016 г.**
3. Выходные даны к проекту: Задан массив  $N$  вещественных чисел. Сформировать новый массив из элементов начального. Каждый элемент нового массива является дробной частью соответствующего элемента начального массива. Вычислить сумму элементов нового массива. Максимальный размер массива - 50 чисел.
4. Перечень вопросов, которые подлежат разработке:
  - 4.1. Исходные данные (заданный массив) размещаются в файле DatesIn.dat.
  - 4.2. Результат (сформированный массив и сумму) разместить в файле DatesOut.dat. В случае отсутствия файла с начальными данными, в программе предусмотреть возможность формирования начальных данных «вручную». Информация в файлах должна храниться в текстовом формате.
  - 4.3. Исходный модуль, выполняемый модуль, исходные данные и результат размещаются в одном каталоге на магнитном носителе.
5. Перечень графического материала (с точным указанием обязательных чертежей):
  - 5.1. – блок-схема алгоритма решения задачи.

Дата выдачи задания **10.02.2016 г.**

Дата выполнения п.п. 4.1 - 4.3 **11.05.2016 г.**

## КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ КУРСОВОГО ПРОЕКТА

| № | Название этапов курсового проектирования        | Срок выполнения этапов проекта | Примечание |
|---|---|--------------------------------|------------|
| 1 | Получение задания к КП                          | 2-я неделя                     | Выполнено  |
| 2 | Анализ задания и математическое описание задачи | 3-я неделя                     | Выполнено  |
| 3 | Формулировка требований к создаваемой программе | 4-я неделя                     | Выполнено  |
| 4 | Разработка блок-схемы алгоритма                 | 6-я неделя                     | Выполнено  |
| 5 | Создание программного кода                      | 10-я неделя                    | Выполнено  |
| 6 | Тестирование программы                          | 11-я неделя                    | Выполнено  |
| 7 | Оформление объяснительной записки               | 14-я неделя                    | Выполнено  |
| 8 | Защита курсового проекта                        | 16-я неделя                    |            |

Студент \_\_\_\_\_  
(подпись)

Петренко И.П.

Руководитель \_\_\_\_\_  
(подпись)

Смирнов А.П.

“ \_\_\_ ” \_\_\_\_\_ 2016 г.

## РЕФЕРАТ

Объяснительная записка состоит из: 50 с., 4 рисунков, 3 таблиц, 8 источников.

Объект исследования - модель стохастического процесса.

Цель работы - анализ методов, алгоритмов и средств исследования стохастических процессов.

Методы исследования - методы теории алгоритмов и словно программирования, методы статистической обработки информации.

Рассмотрен процесс моделирования стохастической измерительной информации. Проведен анализ алгоритмов обработки измерительной информации или другой стохастической информации. Обосновывается выбор методов для статистического анализа результатов исследования стохастических процессов. Разработано дополнение для визуализации алгоритмов статистической обработки информации.

**Ключевые слова:** программирование, язык C++, среда Microsoft Visual Studio 2010, структурное программирование, стохастические процессы, модель стохастического процесса, статистическая обработка измерительной информации, статистический анализ.

## СОДЕРЖАНИЕ

|  |    |
|--|----|
| Задание на курсовую работу .....                                       | 2  |
| Реферат .....  | 4  |
| Вступление .....   | 6  |
| Раздел 1. Обоснование алгоритма и структуры программы .....            | 7  |
| 1.1. Постановка и математическое описание задачи .....                 | 7  |
| 1.2. Описание программы, определения и описание функций ....           | 7  |
| 1.3. Разработка и обоснование алгоритма и<br>структуры программы ..... | 8  |
| Раздел 2. Разработка программного кода .....                           | 9  |
| 2.1. Спецификация программы .....                                      | 9  |
| 2.2. Разработка программы .....  | 10 |
| Раздел 3. Тестирование программы и результаты .....                    | 10 |
| 3.1. Тестирование программы .....                                      | 10 |
| 3.2. Результаты выполнения программы .....                             | 11 |
| Выводы .....   | 11 |
| Литература .....   | 11 |
| Дополнение А. Блок-схема алгоритма программы .....                     | 12 |
| Дополнение Б. Листинг (код с комментариями) программы .....            | 16 |
| Дополнение В. Графическая интерпретация результатов .....              | 18 |

## ВСТУПЛЕНИЕ

Изменения, которые происходят сегодня в разных сферах профессиональной и общественной жизни, связаны, прежде всего, с широким внедрением средств информационных и коммуникационных технологий. Они требуют от специалиста в области компьютерных наук умения разрабатывать программное обеспечение для решения различных задач науки и техники, экономики и управления; создавать и использовать информационные модели процессов и явлений; использовать информационные технологии в проектно-конструкторской, управленческой и финансовой деятельности.

Рост спроса со стороны государственных и частных организаций на сложные системы автоматизации предприятий, надежные операционные среды, комплексы глобального телекоммуникационного управления, широкое внедрение информационных технологий в повседневную жизнь значительно повысили значение и престиж профессии программиста, а также других профессий ИТ отрасли.

Данная работа посвящена разработке программного обеспечения обработки массивов информации. Подобная информация может появляться в результате разнообразных процессов в технике, научных исследованиях, общественных процессах и тому подобное. Например, это может быть информация, которая является результатом измерений при исследованиях или регулировании физических процессов.

Такая информация всегда имеет стохастический (случайный) характер. Поэтому при исследовании методов обработки измерительной информации можно пользоваться ее смоделированным заменителем - эталонными значениями, которые обработаны с помощью генератора случайных чисел. Для анализа стохастической информации применяют разнообразные статистические методы.

Разработанная программа позволяет исследовать эффективность методов статистического анализа стохастических процессов. Она дает возможность регулировать при исследованиях параметры стохастической информации, чтобы оценить степень их влияния на эффективность методов статистического анализа.

## 1. ОБОСНОВАНИЕ АЛГОРИТМА И СТРУКТУРЫ ПРОГРАММЫ

### 1.1. Постановка и математическое описание задачи

Задан массив из  $N$  вещественных чисел. Необходимо сформировать новый массив из элементов заданного. Каждый элемент нового массива является дробной частью соответствующего элемента начального массива.

Вычислить сумму элементов нового массива.

Максимальный размер массива - 50 чисел.

Исходные данные (заданный массив) размещаются в файле **DatesIn.dat**. Результат (сформированный массив и сумма его элементов) разместить в файле **DatesOut.dat**.

В случае отсутствия файла с начальными данными, в программе предусмотреть возможность формирования начальных данных с клавиатуры. Информация в файлах должна храниться в текстовом формате.

Начальный модуль, выполняемый модуль, начальные данные, результат размещаются в одном каталоге на магнитном носителе.

При разработке программы используется следующая формула:

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

где  $S$  - сумма дробных частей вещественных чисел начального массива;

$i$  – номер элемента заданного массива;

$y_i$  - дробная часть вещественного числа начального массива;

$N$  - количество вещественных чисел начального массива.

Вычисление суммы выполняется по следующим рекуррентным (последовательным) соотношениям:

$$S=0;$$

$$S=S+y_i;$$

(на формуле лучше не заканчивать).

### 1.2. Описание программы, определения и описания функций

Программа называется «Преобразование массива» и предназначена для преобразования массива вещественных чисел в массив дробных частей этих чисел, а также для расчета суммы элементов искомого массива.

Программа функционирует в ОС Windows как консольное приложение.



Программа разработана с помощью интегрированной среды Microsoft Visual Studio 2010 на языке программирования C++.

Программа позволит обрабатывать массивы, которые содержат не больше 50 элементов. Ввод и вывод данных осуществляется на экран дисплея ПК, а также записываются в соответствующие файлы.

Запуск программы осуществляется с помощью выполняемого файла (\*.exe).

При отсутствии массива начальных данных выдается сообщение:

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

При получении этого сообщения необходимо вручную сформировать массив начальных данных (ввести данные с клавиатуры).

### 1.3. Разработка и обоснование алгоритма и структуры программы

(Опишите алгоритм своими словами. Как разбили на функции, где цикл **for**, где другие операторы).

Блок-схема алгоритма решения задачи приведена в Приложении А.

## 2. РАЗРАБОТКА ПРОГРАММЫ

### 2.1. Спецификация программы

Начальные данные являются массивом вещественных чисел, которые хранятся в файле "DatesIn.dat" в виде текстовой информации (под запись каждого числа отводится 8 байт).

Результаты - это массив вещественных чисел, которые хранятся в файле "DatesOut.dat" в виде текстовой информации (под запись каждого числа отводится 8 байт). Кроме того, в этом файле хранится сумма всех элементов этого массива (последние 8 байт).

Назначение данных и переменных, их математические обозначения, имена, типы и тому подобное приведены в таб. 1.

Таблица 1. Перечень данных и переменных, которые используются в программы

| Назначение данных и переменных | Математическое обозначение переменной | Имя переменной в программе | Структура данных | Тип данных | Смысл переменной                 |
|--------------------------------|---------------------------------------|----------------------------|------------------|------------|----------------------------------|
| Исходная                       | $x_i$                                 | X[]                        | массив           | float      | Массив начальных данных          |
| Исходная                       | M                                     | M                          | константа        | int        | Максимальный размер массива      |
| Промежуточная                  | N                                     | N                          | простая          | int        | Реальный размер массива          |
| Промежуточная                  | K                                     | K                          | константа        | int        | Размер буфера обмена             |
| Промежуточная                  | i                                     | i                          | простая          | int        | Параметр цикла                   |
| Промежуточная                  | j                                     | j                          | простая          | int        | Параметр цикла                   |
| Промежуточная                  |                                       | buffer[]                   | массив           | int        | Буфер для обмена с файлом        |
| Результат                      | $y_i$                                 | Y[]                        | массив           | float      | Массив данных результата         |
| Результат                      | S                                     | Summa                      | простая          | float      | Сумма элементов искомого массива |

## 2.2. Разработка программы

(Самостоятельно подробно описать процесс разработки программы).

Разработанный программный код приведен в приложении Б.

## 3. ТЕСТИРОВАНИЕ ПРОГРАММЫ И РЕЗУЛЬТАТЫ ЕЕ ВЫПОЛНЕНИЯ

### 3.1. Тестирование программы

При тестировании проверялась работоспособность программы в двух вариантах:

- 1) отсутствие массива с начальными данными;
- 2) наличие массива с начальными данными (основной режим). Как начальные данные использовался массив вещественных чисел {2.45, 0.32, 5.41, 6.78, 3.93}.

В первом варианте программа выдала сообщение:

"Файл DatesIn.dat не существует!"  
 "Сформируйте входной массив!"

После ввода с клавиатуры массива вещественных чисел выполнение программы было продолжено. Массив начальных данных был записан программой на диск D компьютера. После чего функционирование программы осуществлялось в основном режиме.

В результате решения программы с контрольными результатами был сформированный массив {0.45, 0.32, 0.41, 0.78, 0.93, 2.89}. Массив результата данных был записан на диск D компьютера.

Содержание массива с исходными данными и сформированным массивом можно прочитать с помощью текстового редактора.

В процессе решения программы на экране дисплея отображалась следующая информация:

(Лучше, чтобы была копия окна ДОС, как в Методических указаниях из лабораторных работ).

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 искомым элементов

0.45 0.32 0.41 0.78 0.93

Сумма элементов массива = 2.89

На основании анализа результатов выполнения программы можно сделать следующие выводы:

- 1) Программа логически построена правильно.
- 2) Результаты решения задачи достоверны.
- 3) Все массивы в необходимом формате хранятся на диске D компьютера;
- 4) Программа может быть принята в эксплуатацию.

### 3.2. Результаты выполнения программы

Данная программа использовалась при обработке массивов данных об остатках топлива. В связи с тем, что информация о количестве топлива на диспетчерский пункт поступает с точностью до 1 литра, а датчики уровня фиксируют до 0.1 литра, возникла необходимость в оценке точности учета затрат топлива.

Как массив начальных данных в программе использовались данные о ежедневной затрате топлива 13 автомобилями(таблица 2).

Таблица 2. Данные о расходах топлива

|                            |     |     |     |     |     |     |     |     |     |     |     |     |     |
|----------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Визуальные затраты         | 7   | 5   | 1   | 4   | 8   | 9   | 0   | 2   | 9   | 2   | 4   | 5   | 4   |
| Затраты по данным датчиков | 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 результате выполнения программы были получены данные о перерасходе топлива в виде массива:

{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}.

Последнее число в массиве - сумма всех элементов (общий перерасход топлива).

## ВЫВОДЫ

В результате выполнения курсового проекта была разработана программа обработки массива данных, которые хранятся на диске D компьютера в текстовом виде.

Разработанная при курсовом проектировании программа может быть использована при решении различных прикладных задач, связанных с обработкой массивов данных.

Поскольку данная программа обрабатывает одномерные массивы информации, то дальнейшее развитие программы должно быть связано с обработкой многомерных массивов данных.

(Написать подробнее по пунктам Задания)

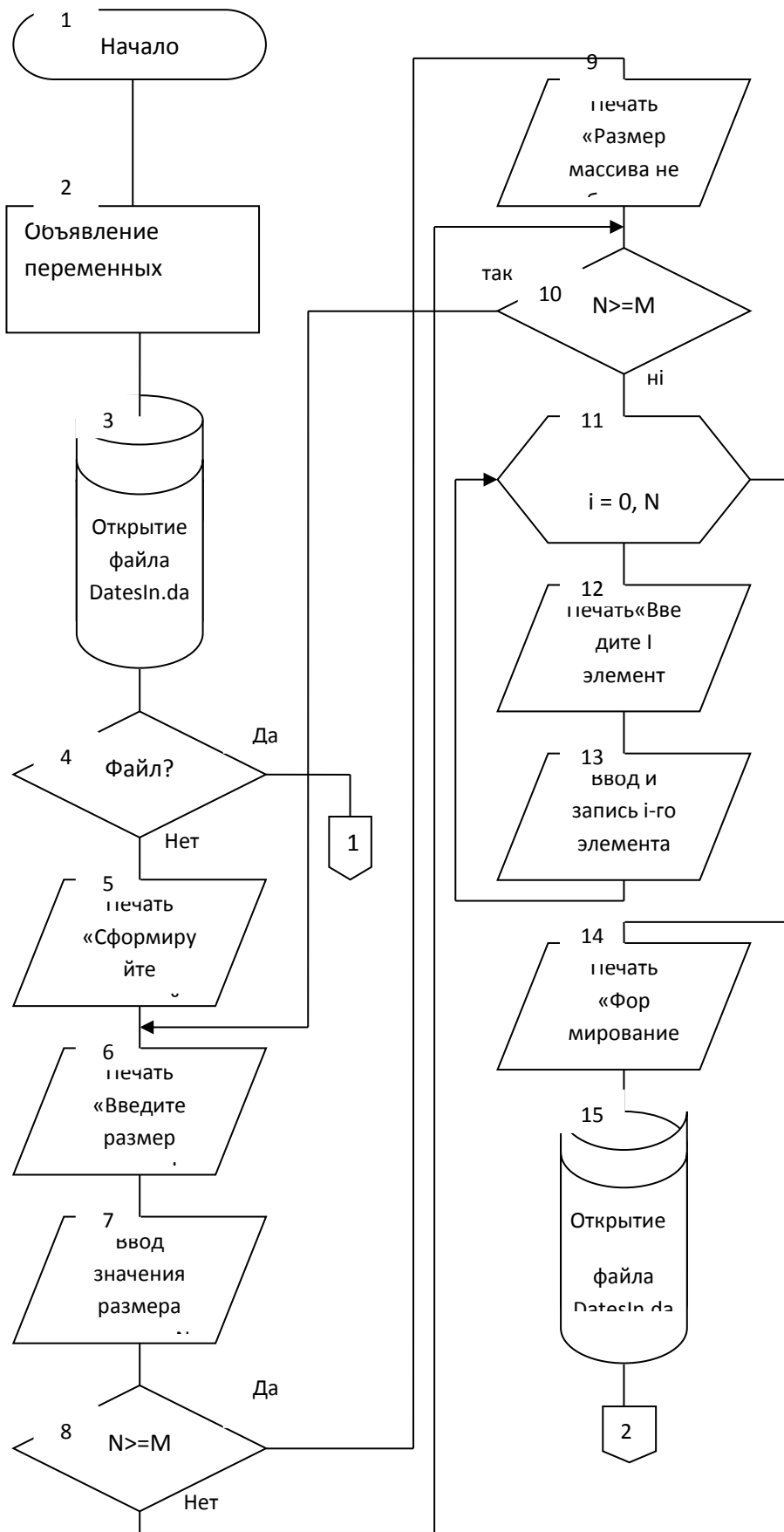
## ЛИТЕРАТУРА

1. ДСТУ 3008-95 "Документация. Отчеты в сфере науки и техники. Структура и правила оформления" - Государственный стандарт Украины.
2. Брауде Э. Технология разработки программного обеспечения / Э. Брауде. - СПб.: Питер, 2004.
3. Рихтер Дж. Windows для профессионалов. Создание эффективных WIN32-приложений с учетом специфики 64-разрядной версии Windows. - СПб.: Русская Редакция, 2001 г. - 752 с.
4. Танненбаум Э. Современные операционные системы. 3-ое изд. - СПб.: Питер, 2010. - 1120с.
5. Хортон А. Visual C++ 2010. Полный курс. - К.: Диалектика, 2011. - 1205 с.

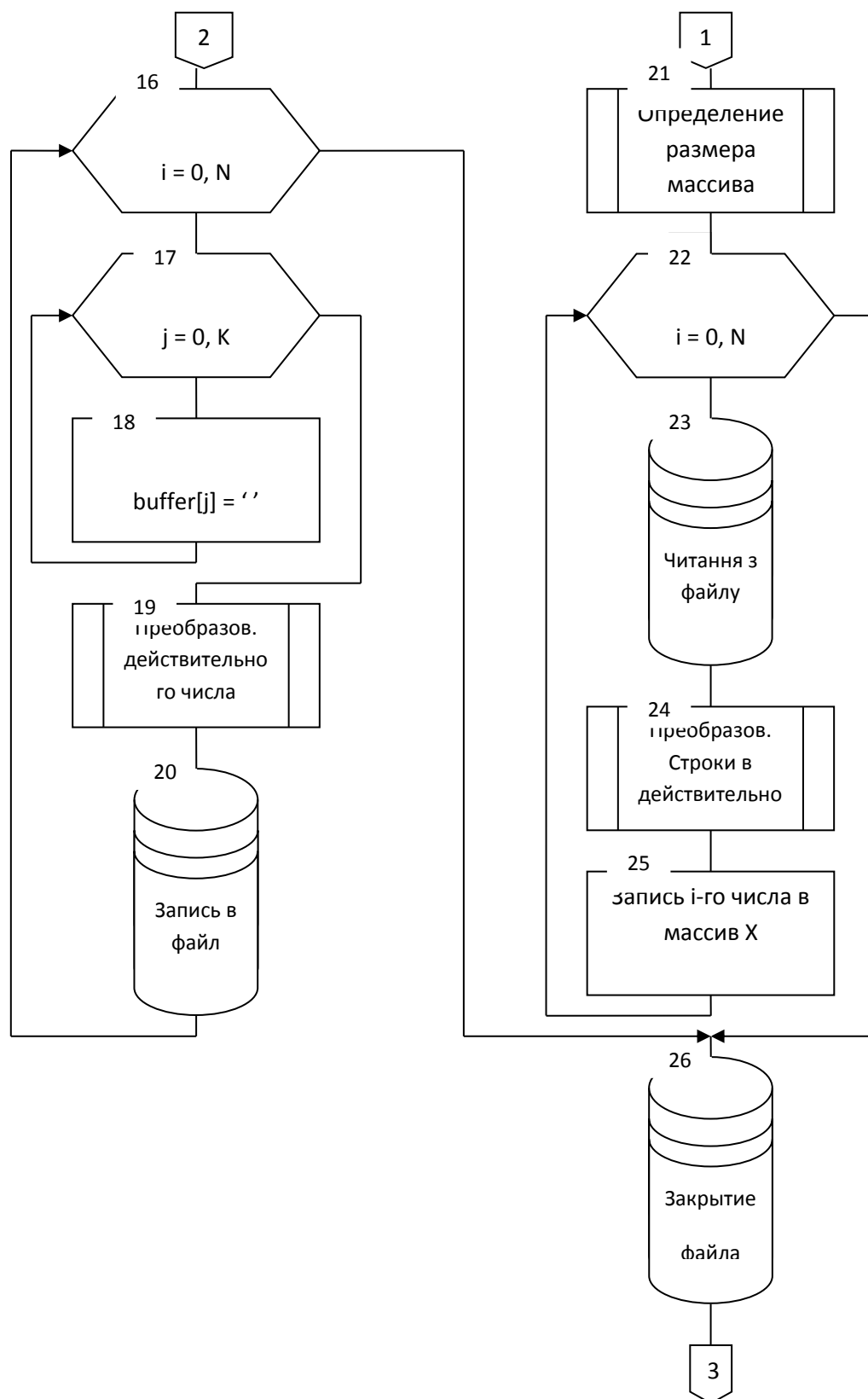
## Электронные ресурсы

1. Страница лектора Симбирского Г.Д. для потока РК в социальной сети "ВК" <http://vc.com/id23791456>. (Материалы для самостоятельной работы, а также лекции, методические указания к лабораторным работам, вопросы для подготовки к сдаче тестов, экзаменационные билеты и тому подобное).
2. Симбирский Г. Д. Конспект лекций по дисциплине "Программирование" для студентов направления 6.050201 Системна інженерія спеціальності 6.050201 Комп'ютерні системи управління рухомими об'єктами (раздел "Структурное программирование на C++"). Портал ХНАДУ <http://files.khadi.kharkov.ua/mekhatroniki-transportnikh-zasobiv.html>
3. Симбирский Г. Д. Методические указания к лабораторным работам по дисциплине "Программирование" для студентов направления 6.050201 Системна інженерія спеціальності 6.050201 Комп'ютерні системи управління рухомими об'єктами (раздел "Структурное программирование на C++").- Портал ХНАДУ, <http://files.khadi.kharkov.ua/mekhatroniki-transportnikh-zasobiv.html>

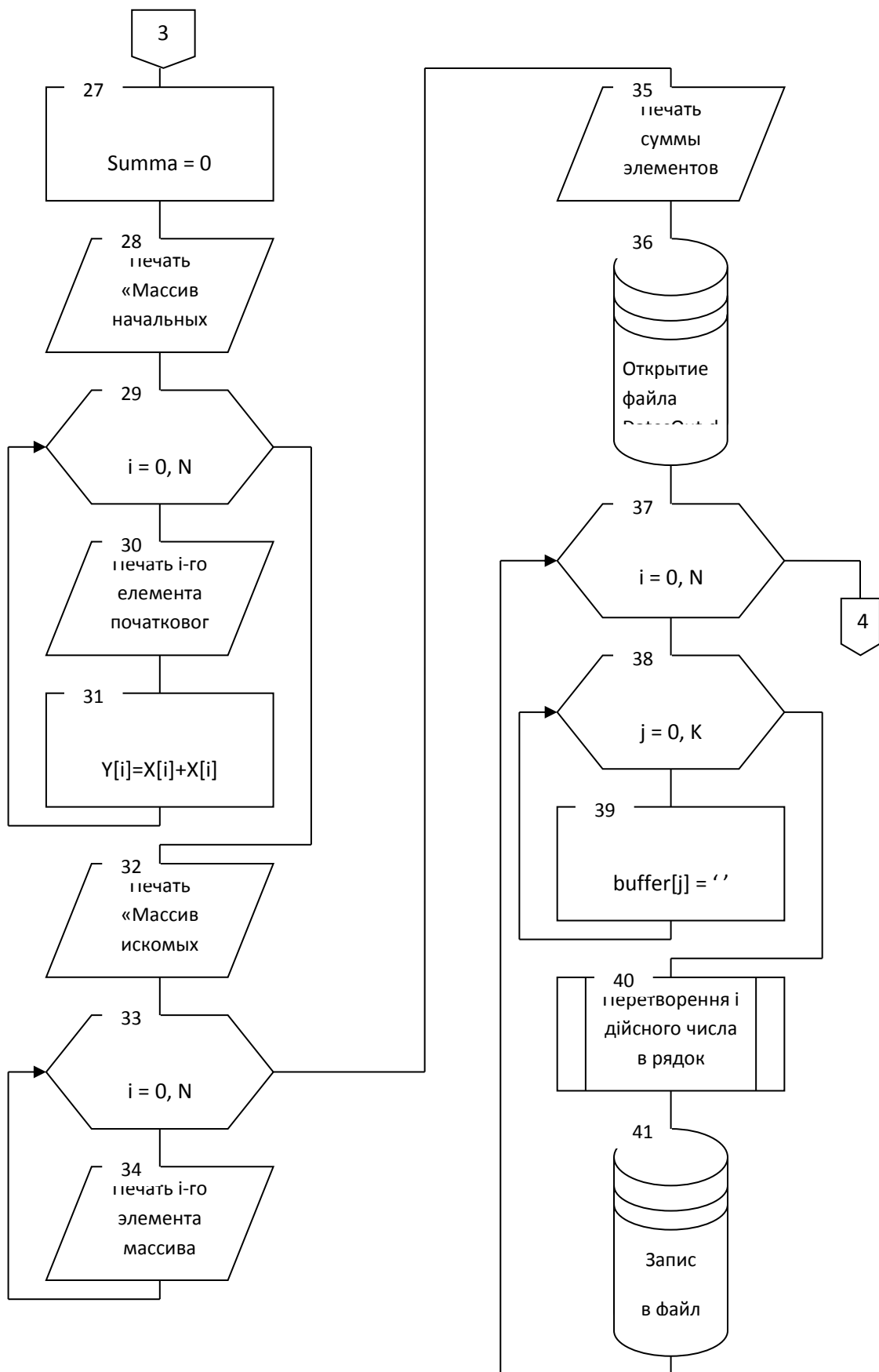
Приложение А. Блок-схема алгоритма программы



## Приложение А. Блок-схема алгоритма программы (продолжение)



## Приложение А. Блок-схема алгоритма программы (продолжение)





## Приложение А. Блок-схема алгоритма программы (продолжение)



## Приложение Б. Программный код с комментариями

**Внимание!!** Комментарии проставить к каждому оператору программы

```

//Программа преобразования заданного массива
#include <vcl.h>
#include <iostream.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int main()
{
    FILE *fFile; //Указатель на файловый поток
    const int M=51; //Максимальный размер массива
    const int K=8; //Размер буфера обмена (количество
//байтов)

    int N; //Реальный размер массива
    int i, j; //Счетчики
    float X[M]; //Начальный массив
    float Y[M]; //Сформированный массив
    float Summa; //Сумма элементов сформированного
//массива

    char buffer[K]; //Строка символов
    if((fFile = fopen("D:\\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("D:\\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("D:\\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;
}

```

