

Государственное бюджетное образовательное учреждение высшего профессионального образования Московской области

МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ ПРИРОДЫ, ОБЩЕСТВА  
И ЧЕЛОВЕКА «ДУБНА»  
(университет «Дубна»)

Факультет естественных и инженерных наук

Кафедра «Энергия и окружающая среда»

УТВЕРЖДАЮ

И.о. проректора по учебно-методической и научной работе

\_\_\_\_\_ С. В. Моржухина

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

## **ПРОГРАММА ДИСЦИПЛИНЫ**

### **«Языки программирования»**

Направление подготовки  
160100.62 «Авиастроение»

Профиль подготовки  
«Самолетостроение»

Квалификация выпускника  
Бакалавр

Форма обучения  
очная

г. Дубна, 2014 г.

Программа составлена в соответствии с ФГОС ВПО с учетом рекомендациями ПрООП ВПО по направлению и профилю подготовки бакалавров 160100 «Авиастроение».

Программа рассмотрена на заседании кафедры Энергия и окружающая среда

Протокол заседания № \_\_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

И.о. заведующего кафедрой \_\_\_\_\_ /Деникин А.С./

СОГЛАСОВАНО:

Декан факультета \_\_\_\_\_ /Деникин А.С./

Рецензент: \_\_\_\_\_  
(ученая степень, ученое звание, Ф.И.О., место работы, должность)

Руководитель библиотечной системы \_\_\_\_\_ /В.Г. Черепанова/

## СОДЕРЖАНИЕ

<b>1. Цели и задачи освоения дисциплины.....</b>	<b>4</b>
<b>2. Место дисциплины в структуре ООП бакалавриата.....</b>	<b>4</b>
<b>3. Требования к уровню освоения содержания дисциплины.....</b>	<b>4</b>
<b>4. Содержание и структура дисциплины .....</b>	<b>5</b>
4.1. Структура дисциплины .....	5
4.2. Содержание разделов дисциплины .....	6
4.3. Практические занятия (семинары).....	7
4.4. Домашние работы.....	9
4.5. Контрольные работы .....	9
<b>5. Образовательные технологии .....</b>	<b>9</b>
5.1. Методические рекомендации для студентов: .....	9
5.2. Методические рекомендации для преподавателей: .....	10
<b>6. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации.....</b>	<b>11</b>
6.1. Контрольные вопросы по отдельным темам для проведения контрольных опросов, домашние задания, темы курсовых работ, вопросы к экзамену .....	12
6.2. Задание для самостоятельной работы в Интернет .....	14
<b>7. Учебно-методическое обеспечение дисциплины.....</b>	<b>15</b>
7.1. Основная литература .....	15
7.2. Дополнительная литература .....	15
7.3. Интернет-ресурсы .....	15
<b>8. Материально-техническое обеспечение дисциплины.....</b>	<b>15</b>

## **1. Цели и задачи освоения дисциплины**

В процессе развития информатики как прикладной науки появились разные подходы к программированию. Курс «Языки программирования» призван содействовать знакомству студентов с различными парадигмами проектирования и разработки программного обеспечения. Он способствует развитию алгоритмического мышления, навыков программирования студентов.

Цель дисциплины: изучение методов программирования для овладения знаниями в области технологии программирования; подготовка к осознанному использованию, как языков программирования, так и методов программирования.

Воспитательной целью дисциплины является формирование у студентов научного, творческого подхода к освоению технологий, методов и средств производства программного обеспечения.

Основные задачи курса программирования на основе структурного и объектно-ориентированного подхода:

- знакомство с методами структурного и объектно-ориентированного программирования как наиболее распространенными и эффективными методами разработки программных продуктов;
- обучение разработке алгоритмов на основе структурного и объектно-ориентированного подхода;
- закрепление навыков алгоритмизации и программирования на основе изучения языка программирования C++;
- знакомство с основными структурами данных и типовыми методами обработки этих структур;

Концепция дисциплины основана на том, что она имеет общеобразовательный и в определенной степени мировоззренческий характер и предназначена для формирования специалиста с широким научным кругозором.

## **2. Место дисциплины в структуре ООП бакалавриата**

Дисциплина «Языки программирования» изучается на 3 курсе в 5 семестре.

Изучение дисциплины «Языки программирования» базируется на знании математических дисциплин, курсов «Информатики», «Основы информационных технологий».

Материалы курса являются важной составляющей при работе над курсовыми работами в ряде специальных курсов, среди которых «Сопротивление материалов», «Основы автоматизации проектно-конструкторских работ», «Теория упругости и метод конечных элементов», «Динамика полета», «Прочность летательных аппаратов», «Теория автоматического регулирования», а также при подготовке дипломного проекта.

## **3. Требования к уровню освоения содержания дисциплины.**

Изучение дисциплины «Языки программирования» направлено на формирование у студентов следующих компетенций:

- владеть культурой мышления, способностью обобщать, анализировать и воспринимать информацию, ставить цели и выбирать пути их достижения (ОК-1);
- осознавать сущность и значение информации в развитии современного общества; владеть основными методами, способами и средствами получения, хранения, переработки информации (ОК-10);
- иметь навыки работы с компьютером как средством управления информацией (ОК-11);
- готовностью к решению сложных инженерных задач с использованием базы знаний математических и естественнонаучных дисциплин (ПК-1);

- владеть методами и иметь навыки моделирования и создания авиационных конструкций на основе современных информационных технологий с использованием средств автоматизации проектно-конструкторских работ (ПК-6);
- иметь навыки математического моделирования процессов и объектов на базе стандартных пакетов исследований (ЭИ-1).

В результате изучения дисциплины «Языки программирования» студент должен:

<i>Результат обучения</i>	<i>Компетенция</i>	<i>Образовательная технология</i>	<i>Вид задания</i>
<b>Знать и иметь представление:</b> о конструировании алгоритмов, о методах структурного и модульного программирования, об абстракциях основных структур данных (списки, множества и т.п.) и методах их обработки и способах реализации, о методах и технологиях программирования;	ОК-1,10 ПК-1	Л1-4	Д1-18 КрР-1,2,3
<b>Уметь:</b> разрабатывать алгоритмы, реализовывать алгоритмы на языке программирования высокого уровня, описывать основные структуры данных, реализовывать методы обработки данных, работать в средах программирования;	ОК-1,10,11 ПК-1 ЭИ-1	Л1-4, С1-18	Д1-18 КрР-1,2,3
<b>Приобрести опыт деятельности:</b> структурного программирования, алгоритмизации, работы в среде программирования (составление, отладка и тестирование программ; разработка и использование интерфейсных объектов).	ОК-10,11 ПК-1,6 ЭИ-1	Л1-4, С1-18	Д1-18 КрР-1,2,3
<b>Владеть, иметь опыт:</b> разработки алгоритмов, описания структур данных, описания основных базовых конструкций, программирования на языке высокого уровня, работы в различных средах программирования.	ОК-10,11 ПК-1,6 ЭИ-1	Л1-4, С1-18	Д1-18 КрР-1,2,3

#### 4. Содержание и структура дисциплины

##### 4.1. Структура дисциплины

Общая трудоемкость дисциплины составляет 2 ЗЕТ, 72 часов.

Вид работы	Трудоемкость, часов	
	3 семестр	Всего
<b>Общая трудоемкость</b>	<b>72</b>	<b>72</b>
<b>Аудиторная работа:</b>	<b>54</b>	<b>54</b>
<i>Лекции (Л)</i>	18	18
<i>Практические семинарские занятия (ПЗ)</i>	36	36
<i>Лабораторные работы (ЛР)</i>	–	–
<b>Самостоятельная работа</b>	<b>18</b>	<b>18</b>
<i>Самоподготовка (проработка и повторение лекционного материала и материала учебников, учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам, рубежному контролю и т.п.)</i>	18	18
Курсовой проект, курсовая работа	–	–

Расчетно-графическое задание	–	–
Реферат	–	–
Эссе	–	–
Самостоятельное изучение разделов	–	–
Подготовка и сдача экзамена	–	–
<b>Вид промежуточного контроля</b>	Зачет с оценкой	Зачет с оценкой

#### 4.2. Содержание разделов дисциплины

№ п/п	Наименование разделов и подразделов	Неделя				Формы текущего контроля успеваемости (по неделям семестра) <sup>1</sup>
			Л	ПЗ	СР	Форма промежуточной аттестации (по семестрам)
			18	36	18	Зачет с оценкой – 5 семестр
<b>5 семестр</b>						
1.	Введение в программирование	1-3	2	6	4	КО
2.	Язык программирования С++	4-8	6	10	4	КО
3.	Модульное программирование. Программирование абстрактных типов данных	9-13	4	10	4	КО
4.	Объектно-ориентированное программирование	14-17	6	10	6	КО

#### 5 семестр:

##### 1. Введение в программирование

- 1.1. Методологии программирования. Программирование как раздел информатики. История и эволюция информатики.
- 1.2. Алгоритмизация и программирование. Этапы решения задач на ЭВМ. Понятие алгоритма. Способы записи алгоритмов.
- 1.3. Принципы структурного программирования. Языки программирования высокого уровня. Программное обеспечение и технологии программирования.

##### 2. Язык программирования С++

- 2.1. Общая характеристика языка С++. Структуры данных. Определение констант. Описание переменных.
- 2.2. Стандартные типы данных. Целые типы. Символьный и булевский типы данных. Эквивалентность и совместимость типов.
- 2.3. Типы, определяемые программистом: перечисляемый, интервальный. Тип дата-время.
- 2.4. Основные операторы языка. Перечень операторов С++. Оператор присваивания.
- 2.5. Операторы (процедуры) ввода-вывода. Управление выводом данных в консольном режиме (простейшее форматирование).

<sup>1</sup> Формы текущего контроля: защита лабораторной работы (ЛР), выполнение курсового проекта (КП), курсовой работы (КР), расчетно-графического задания (РГЗ), домашнего задания (ДЗ), написание реферата (Р), эссе (Э), коллоквиум (К), рубежный контроль (РК), тестирование (Т), контрольная работа (КрР), контрольный опрос (КО) и др.

- 2.6. Условный оператор. Логические выражения. Оператор множественного ветвления.
- 2.7. Операторы цикла: с предусловием, с постусловием, с параметром.
- 2.8. Структурированные типы языка программирования высокого уровня. Массивы. Примеры задач с численными, символьными, булевыми массивами.
- 2.9. Строковый тип данных. Записи. Оператор присоединения. Операции отношения.
- 2.10. Файлы. Понятие логического и физического файлов. Файловые типы.
- 2.11. Общие процедуры для работы с файлами. Типизированные файлы. Текстовые файлы.
- 2.12. Нетипизированные файлы и процедуры ввода-вывода. Прямой и последовательный доступ к компонентам файлов.
- 2.13. Алгоритмы поиска и сортировки. Простой и бинарный поиск.
- 2.14. Сортировки: выбором, обменом, вставкой. Анализ сложности алгоритмов на примере сортировок.

- 3. Модульное программирование. Программирование абстрактных типов данных
  - 3.1. Процедуры и функции. Подпрограммы. Формальные параметры. Параметры-значения, параметры-переменные, параметры-константы.
  - 3.2. Локальные и глобальные идентификаторы подпрограмм.
  - 3.3. Рекурсия. Внешние подпрограммы.
  - 3.4. Модули. Общая структура модуля. Подпрограммы в модулях. Компиляция и использование модулей.
  - 3.5. Организация динамических структур данных (абстрактных типов данных). Динамические структуры. Динамическое распределение памяти.
  - 3.6. Виды списков. Примеры использования списков. Организация динамических структур данных: стек, очередь, двоичное дерево поиска.

- 4. Объектно-ориентированное программирование
  - 4.1. Введение в объектно-ориентированное программирование. Инкапсуляция, наследование, полиморфизм.
  - 4.2. Разбор примеров задач на объектно-ориентированное программирование.
  - 4.3. Реализация абстракций данных методами объектно-ориентированного программирования.
  - 4.4. Математические объекты: рациональные и комплексные числа, вектора, матрицы. Библиотеки объектов.
  - 4.5. Объектно-событийное и объектно-ориентированное программирование.
  - 4.6. Идеология программирования под Windows. Событие и сообщение.
  - 4.7. Виды событий. События от мыши и клавиатуры.
  - 4.8. Программирование управления событиями. Обработка исключительных событий.
  - 4.9. Основы визуального программирования. Компонент. Иерархия компонентов.

4.3. Практические занятия (семинары).

№	№ раздела дисциплины	Наименование практических занятий (семинаров)	Кол. часов
<b>5 семестр</b>			
1.	1	Вводное занятие. Методология программирования. Программирование как раздел информатики.	2

2.	1	Алгоритмизация и программирование. Этапы решения задач на ЭВМ. Понятие алгоритма. Способы записи алгоритмов.	2
3.	1	Принципы структурного программирования. Языки программирования высокого уровня. Программное обеспечение и технологии программирования.	2
4.	2	Общая характеристика языка C++. Структуры данных. Определение констант. Описание переменных. Стандартные типы данных. Целые типы. Символьный и булевский типы данных. Эквивалентность и совместимость типов.	2
5.	2	Указатели и массивы. Примеры задач с численными, символьными, булевскими массивами. Основные операторы языка. Перечень операторов C++. Оператор присваивания. Операторы (процедуры) ввода-вывода. Управление выводом данных в консольном режиме (простейшее форматирование).	2
6.	2	Контрольная работа по проеденному материалу. Условный оператор. Логические выражения. Оператор множественного ветвления.	2
7.	2	Операторы цикла: с предусловием, с постусловием, с параметром. Структурированные типы языка программирования высокого уровня. Типы, определяемые программистом: перечисляемый, интервальный. Строковый тип данных. Записи. Оператор присоединения. Операции отношения.	2
8.	2	Файлы. Понятие логического и физического файлов. Файловые типы. Нетипизированные файлы и процедуры ввода-вывода. Прямой и последовательный доступ к компонентам файлов. Алгоритмы поиска и сортировки. Простой и бинарный поиск. Сортировки: выбором, обменом, вставкой. Анализ сложности алгоритмов на примере сортировок.	2
9.	3	Процедуры и функции. Подпрограммы. Формальные параметры. Параметры-значения, параметры-переменные, параметры-константы.	2
10.	3	Локальные и глобальные идентификаторы подпрограмм. Рекурсия. Внешние подпрограммы.	2
11.	3	Модули. Общая структура модуля. Подпрограммы в модулях. Компиляция и использование модулей.	2
12.	3	Организация динамических структур данных (абстрактных типов данных). Динамические структуры. Динамическое распределение памяти.	2
13.	3	Виды списков. Примеры использования списков. Организация динамических структур данных: стек, очередь, двоичное дерево поиска.	2
14.	4	Контрольная работа по проеденному материалу. Введение в объектно-ориентированное программирование. Инкапсуляция, наследование, полиморфизм. Разбор примеров задач на объектно-ориентированное программирование.	2
15.	4	Реализация абстракций данных методами объектно-ориентированного программирования.	2
16.	4	Математические объекты: рациональные и комплексные числа, вектора, матрицы. Библиотеки объектов. Объектно-событийное и объектно-ориентированное программирование.	2



17.	4	Контрольная работа по проеденному материалу. Идеология программирования под Windows. Событие и сообщение.	2
18.	4	Виды событий. События от мыши и клавиатуры. Программирование управления событиями. Обработка исключительных событий. Основы визуального программирования. Компонент. Иерархия компонентов.	2

#### 4.4. Домашние работы

№	Тема задания	неделя
<b>5 семестр</b>		
Д1.	Завершение выполнения аудиторного практического задания.	1
Д2.	Завершение выполнения аудиторного практического задания. Разработка алгоритма «Вычисление геометрической прогрессии», составление блок-схемы.	2
Д3.	Завершение выполнения аудиторного практического задания. Установка компилятора языка C++ на рабочий домашний компьютер.	3
Д4.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 1	4
Д5.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 1 [2] гл. 1.11	5
Д6.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 1, [2] гл. 2.10	6
Д7.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 2, [2] гл. 1.7	7
Д8.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 2, [2] гл. 2.12	8
Д9.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 4, [2] гл. 2.14	9
Д10.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 4, [2] гл. 2.14	10
Д11.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 4, [2] гл. 3.2	11
Д12.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 7, [2] гл. 1.12-1.14	12
Д13.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 7, [2] гл. 1.12-1.14	13
Д14.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 7, [2] гл. 1.12-1.14	14
Д15.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 7, [2] гл. 3.1	15
Д16.	Завершение выполнения аудиторного практического задания. Изучение литературы [1] гл. 7, [2] гл. 3.1	16
Д17.	Завершение выполнения аудиторного практического задания. Изучение литературы [2] гл. 3.1	17
Д18.	Подготовка к зачету	17

#### 4.5. Контрольные работы

№	Тема работы	Неделя
КР1	Типы данных и синтаксис операторов C++.	6 неделя, 5 семестр
КР2	Процедуры, функции, динамические типы данных, структуры.	14 неделя, 5 семестр
КР3	Объектно-ориентированное программирование. Классы, наследование, инкапсуляция.	17 неделя, 5 семестр

## 5. Образовательные технологии

### 5.1. Методические рекомендации для студентов:

Рабочей программой настоящей дисциплины предусмотрена самостоятельная работа студентов в объеме, определяемом учебным планом. Самостоятельная работа проводится с целью углубления знаний по дисциплине и предусматривает:

- чтение студентами рекомендованной литературы и усвоение теоретического материала дисциплины;
- подготовку к практическим занятиям;
- работу с Интернет-источниками;
- подготовку к сдаче коллоквиумов, выполнению тестовых заданий и сдаче зачетов и экзаменов.

Планирование времени на самостоятельную работу, необходимого на изучение настоящей дисциплины, студентам лучше всего осуществлять на весь семестр, предусматривая при этом регулярное повторение пройденного материала. Материал, законспектированный на лекциях, необходимо регулярно дополнять сведениями из литературных источников, представленных в рабочей настоящей программе дисциплины. По каждой из тем для самостоятельного изучения, приведенных в рабочей программе дисциплины следует сначала прочитать рекомендованную литературу и при необходимости составить краткий конспект основных положений, терминов, сведений, требующих запоминания и являющихся основополагающими в этой теме и для освоения последующих разделов курса.

Для расширения знаний по дисциплине рекомендуется использовать Интернет-ресурсы: проводить поиск в различных системах, сайтах и обучающих программах, рекомендованных преподавателем на лекционных занятиях.

Перечень обязательных видов работы студентов:

1. Посещение лекционных и практических занятий;
2. Выполнение курсового проекта;
3. Выполнение домашних практических работ;
4. Работа с Интернет-источниками;
5. Чтение рекомендованной литературы.

Преподаватель оценивает работу студента на основании контрольного опроса в начале каждого практического занятия, а также по результатам выполнения домашних заданий и защиты курсового проекта.

#### *5.2. Методические рекомендации для преподавателей:*

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

**Информационные технологии:** использование электронных образовательных ресурсов при самостоятельной работе с содержанием дисциплины, а также при подготовке к лекциям и практическим занятиям.

**Работа в команде:** совместная работа студентов в группе при коллективном решении задач на практических занятиях с коллективным обсуждением алгоритмов и результатов решений.

Одной из задач преподавателей, ведущих занятия по настоящей дисциплине является выработка у студентов осознания важности, необходимости и полезности знания дисциплины для дальнейшей работы их инженерами, специалистами. Методическая модель преподавания дисциплины основана на применении активных методов обучения.

Принципами организации учебного процесса являются:

- активное участие студентов в учебном процессе;

- проведение практических занятий, определяющих приобретение навыков решения проблемы;
- приведение примеров применения изучаемого теоретического материала к реальным практическим ситуациям.

Для более глубокого изучения предмета преподаватель предоставляет студентам информацию о возможности использования по разделам дисциплины Интернет-ресурсов, кафедральной библиотеки.

#### Виды и содержание учебных занятий

##### *Практические занятия.*

Практические занятия по дисциплине проводятся в интерактивном режиме по технологии работа в команде. Содержанием практических занятий является решение задач и проведение консультаций.

Организация занятий следующая:

- в начале занятия объявляется его тема, и ставятся познавательные цели;
- преподавателем демонстрируется вариант решения одной из типовых задач (при необходимости);
- студенты поочерёдно выполняют решение задачи у доски;
- в ходе демонстраций решений проводятся коллективные обсуждения, выявляются ошибки и недочеты;
- при решении задач на подбор сечений группа студентов разбивается на команды, каждая из которых проводит проверочный расчёт одного из сечений;
- преподаватель подводит итоги работы команд, оценивает степень достижения поставленных целей, объявляет тему следующего занятия.

##### *Управление самостоятельной работой студента.*

Содержанием внеаудиторной самостоятельной работы студентов является интерактивная работа с учебным материалом дисциплины по электронному комплексу лекций, по учебным пособиям, выполнения двух расчетно-графических работ.

Студент имеет полную свободу выбора траектории обучения. По каждой дидактической единице (параграфу) учебного материала предусмотрен самоконтроль. В конце каждой главы пособия приведены задания для самоконтроля и примеры решений задач, на каждый раздел курса предусмотрено домашнее задание.

Итогом самостоятельной работы студента по разделу дисциплины в совокупности с работой на аудиторных занятиях является оценка по разделу.

*Таблица: Интерактивные образовательные технологии*

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	Количество часов
5	семинары	Практические занятия по дисциплине проводятся в интерактивном режиме по технологии работа в команде	18
Итого:			18

#### **6. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации**

Текущий контроль осуществляется в виде защиты выполненных в ходе семинаров практических заданий и выполнения контрольных работ в конце изучения каждого раздела. По окончании курса проводится зачет (с оценкой) по результатам проделанной в ходе семестра работы. Для получения зачета необходимо выполнить все практические задания (написать компьютерные программы), заданные в течение семестра. Каждая выполненная работа, должна быть защищена.

В конце семестра оценка определяется исходя из числа выполненных и защищенных студентом практических работ. Каждая работа оценивается по 5-ти бальной системе. Общая оценка определяется усреднением оценки по итогам выполненных заданий в течение семестра.

Если выполненных заданий недостаточно для получения положительной оценки, студент имеет возможность добрать недостающее количество баллов в течение зачетной недели, отвечая на теоретические вопросы и/или выполняя практические задания из числа нереализованных задач. В этом случае допускается упрощение формулировки задания в сочетании с проведением контрольного опроса по разделам курса. Список таких вопросов с указанием количества баллов за каждый приведен в пункте 6.1.

*Методика формирования оценки «зачтено»:*

«зачтено» – в течение семестра студент посетил не менее 75% занятий, пропуски занятий по уважительной причине. В целом выполнены все домашние задания. Результаты выполнения контрольных работ или индивидуальных заданий (рефераты, типовые расчеты, курсовые работы и т.п.) удовлетворительные.

«не зачтено» – в течение семестра студент посетил менее 50% занятий, пропуски по неуважительной причине. Не выполнены более 50% домашних заданий. Результаты выполнения хотя бы одной контрольных работ неудовлетворительные. Неудовлетворительная оценка получена за выполнение индивидуальных заданий.

*6.1. Контрольные вопросы по отдельным темам для проведения контрольных опросов, домашние задания, темы курсовых работ, вопросы к экзамену*

Темы контрольных работ и варианты заданий для них:

1. Варианты заданий для контрольной работы и проведения зачета:

*Задание 1.* Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Написать такой вариант программы, который, по Вашему мнению, будет правильным.

```
void My_Error()
{
    int X = 10, Y = 150, Sum;
    double v = exp(Y);
    Sum = X + v;
    cout >> "Сумма равна " << Sum
}
```

*Задание 2.* Составить программу идентификации треугольника по сторонам a, b, c. Определяемое свойство: является прямоугольным или не является прямоугольным (удобно использовать сравнение квадрата одной из сторон с суммой квадратов двух других сторон).

*Задание 3.* Решить задачу с использованием оператора выбора: по введенному числу грибов k напечатать фразу «Мы нашли в лесу k грибов» и согласовать при этом окончание слова «гриб» с числом k. (Количество грибов может быть любым целым числом: 1, 3, 34, 127 и т.п. Окончание фразы определяется значением последней цифры.)

*Задание 4.* Сформулировать условие задачи, которая решается в данной программе:

```
#include <stdio.h>
#include <conio.h>
main()
{
    int N,n;
```

```

double x,s;
printf("Введите N: "); scanf("%d", &N);
printf("\n");
printf("Введите x: "); scanf("%lf", &x);
printf("\n");
s = 0;
for(n=1;n<=N;n++)
{
s += x/(3*n+2);
x *= x;
}
printf("S = %e",s);
getch();
}

```

*Задание 5.* Решить задачу с использованием цикла: найти  $k$ -е простое число в арифметической прогрессии 11, 21, 31, 41, 51, 61, ...

*Задание 6.* Решить задачу: подсчитать количество элементов массива, каждый из которых не меньше всех своих предшественников (т.е. элементов с меньшими индексами). Например, для массива -2, 3, 0, 13, 13, 4, -3 ответ будет 4.

*Задание 7.* Решить задачу с использованием подпрограммы: вывести наибольшую из первых цифр трех заданных чисел. Например, если  $a = 25$ ,  $b = 730$ ,  $c = 1995$ , то надо вывести цифру 7.

*Задание 8.* В данной последовательности  $a_1, a_2, \dots, a_n$  определить максимальное число среди элементов с номерами, кратными числу  $k$ . Например, для последовательности -1, 0, 12, -77, 22, -6, 70, 11, 3 и  $k = 3$  получаем ответ 12.

*Задание 9.* Вычислить  $\sqrt{2 + \sqrt{2 + L \sqrt{2}}}$  для  $n$  слагаемых.

*Задание 10.* Заполнить файл целыми числами из отрезка  $[-1000; 1000]$  с помощью датчика случайных чисел. Переписать в один из новых файлов те из компонент исходного файла, модуль которых является простым числом, в другой — все остальные.

*Задание 11.* Удалить из файла, содержащего целые числа, неположительные компоненты. Дополнительных файлов и массивов не использовать.

Вопросы, для подготовки к зачету (3 семестр):

1. Этапы решения задач с использованием ЭВМ.
2. Понятие алгоритма. Подходы к определению алгоритма. Свойства алгоритма. Способы записи алгоритма.
3. Языки программирования. Алгоритмические языки (алфавит, синтаксис, семантика). Способы описания синтаксиса (язык металингвистических формул, синтаксические диаграммы).
4. Система программирования C++.
5. Структура программы, элементы языка C++ (алфавит). Понятие типа данных.
6. Операции (арифметические, логические) на типах. Стандартные функции. Выражения.
7. Процедуры консольного ввода и вывода, управление вводом-выводом. Оператор присваивания. Совместимость по присваиванию.

8. Условный оператор. Оператор множественного ветвления (выбора).
9. Циклы в C++: с предусловием, с постусловием. Связь с другими циклами.
10. Циклы в C++: с параметром. Связь с другими циклами.
11. Структурированные типы данных. Линейные массивы. Примеры задач.
12. Структурированные типы данных. Двумерные массивы. Примеры задач.
13. Сортировка массивов. Метод выбора. Двоичный поиск в массиве.
14. Сортировка массивов. Метод обмена.
15. Сортировка массивов. Метод вставок.
16. Подпрограммы в C++. Основные способы передачи параметров в подпрограмму, их сравнение.
17. Подпрограммы в C++. Область видимости. Локальные и глобальные идентификаторы.
18. Процедуры. Организация и вызов. Примеры.
19. Функции. Организация и вызов. Примеры.
20. Простые типы данных в C++.
21. Структурированные типы данных. Строковый тип данных в C++: основные процедуры и функции, примеры.
22. Рекурсия. Механизм рекурсии. Примеры.
23. Сортировка массивов. Метод быстрой сортировки.
24. Комбинированный тип данных (записи). Оператор присоединения. Записи с вариантами. Программирование типовых алгоритмов обработки записей.
25. Файловые типы в C++. Общие процедуры для работы с файлами. Компонентные (типизированные) файлы.
26. Текстовые файлы. Текст-ориентированные процедуры и функции. Типовые задачи.
27. Прямой и последовательный доступ к компонентам файла. Процедуры и функции, ориентированные на прямой доступ к компонентам файла.
28. Поиск в типизированных файлах. Сортировка файлов (на примере одного из методов).
29. Типизированные файлы. Файлы записей. Типовые алгоритмы обработки.
30. Статическая и динамически распределяемая память. Пример использования указателей.
31. Динамические структуры данных. Однонаправленный список. Процедуры обработки списка.
32. Динамические структуры данных. Двухнаправленный список. Процедуры обработки списка.
33. Динамические структуры данных. Кольцевой список (однонаправленный или двухнаправленный). Процедуры обработки списка.
34. Стек. Процедуры обработки.
35. Очередь. Процедуры обработки.
36. Двоичное дерево. Добавление в дерево и поиск в дереве.
37. Двоичное дерево. Удаление элемента из дерева.
38. Модуль. Общая структура модуля. Компиляция и подключение модуля.
39. Объектно-ориентированное программирование.
40. Пример реализации задачи на ООП в Builder C++.

## 6.2. Задание для самостоятельной работы в Интернет

В ходе изучения курса «Языки программирования» предусмотрено использование компьютера для разработки компьютерных кодов, а также для поиска материалов по тематике разделов в сети Интернет, в частности, по методам и алгоритмам программирования, а также для изучения дополнительных материалов по темам лекций.

## **7. Учебно-методическое обеспечение дисциплины**

### *7.1. Основная литература<sup>2</sup>*

1. **Павловская Т.А.** С/С++. Программирование на языке высокого уровня: Для магистров и бакалавров: Учебник для вузов / Павловская Татьяна Александровна; Рец. Г.И.Ревунков и др. - СПб.: Питер, 2011. - 464с.
2. **Архангельский А.Я.** Язык С++ в С++ Builder: Справочное и методическое пособие / Архангельский А.Я. - М.: Бином-Пресс, 2008. - 944с.
3. Страуструп Б. Язык программирования С++ / Страуструп Бьерн ; Пер.с англ. под ред. Н.Н.Мартынова. - спец.изд. - СПб. ; М. : БИНОМ, 2011. - 1136с. : ил. - ISBN 9785951804259

### *7.2 Дополнительная литература*

1. **Культин Н.Б.** Самоучитель С++ Builder в задачах и примерах / Культин Никита Борисович. - СПб.: БХВ-Петербург, 2007. - 336с.
2. **Borland С++ Builder 6:** Руководство разработчика / Холингворт Джаррод, Сворт Боб, Кэшмэн Марк, Густавсон Поль; Пер.с англ.В.Н.Заики и др.; Под ред. И.В.Красикова. - М.: Вильямс, 2004. - 976с.
3. **Задачи по программированию** / Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. - М.: Наука, 1988. - 224с.

### *7.3. Интернет-ресурсы*

1. <http://citforum.ru> Аналитическая информация по языкам программирования высокого уровня;
2. <http://www.rushelp.com> Компьютерная документация от А – Z;
3. <http://ishodniki.ru> Исходные тексты компьютерных кодов;
4. <http://delcb.com> Электронные версии книг и статей по программированию на языках С++ и Delphi.

## **8. Материально-техническое обеспечение дисциплины**

1. Компьютерные презентации по материалам лекций
2. Учебная аудитория, оборудованная персональными компьютерами с набором программного обеспечения: среда разработки с возможностью программирования на С++, включающая средства визуализации.

---

<sup>2</sup> Список основной литературы должен включать только источники, имеющиеся в наличии в библиотечной системе университета и удовлетворяющие предъявляемым требованиям. Необходимо согласование с руководителем библиотечной системы.