

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ**

УТВЕРЖДАЮ

Декан ФВТ



Л.Р. Фионова

« 16 » _____ 2015 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.1.14 Программирование на языках высокого уровня

Направление подготовки – *09.03.03 Прикладная информатика*

Профиль подготовки – *Прикладная информатика в экономике*

Квалификация (степень) выпускника – *бакалавр*

Форма обучения – *очная*

г. Пенза, 2015 г.

1. Цели освоения дисциплины

Целями освоения дисциплины «**Программирование на языках высокого уровня**» являются: подготовка студента к самостоятельному выполнению работ по созданию программных систем прикладной информатики, соответствующих общекультурным и профессиональным компетенциям; овладение студентами знаниями и навыками в области программирования приложений.

2. Место дисциплины в структуре ОПОП бакалавриата

Учебная дисциплина «**Программирование на языках высокого уровня**» относится к базовой части блока Б1, шифр дисциплины Б1.1.14.

Дисциплина опирается на знания, полученные студентами в ходе изучения следующих дисциплин: «Основы компьютерной обработки информации», «Математика», «Основы алгоритмизации и программирования».

Компетенции, приобретенные в ходе изучения дисциплины «**Программирование на языках высокого уровня**», готовят студента к освоению профессиональных компетенций.

Знания, умения и навыки, полученные в результате изучения дисциплины, найдут применение при изучении следующих дисциплин: «Программирование Интернет-приложений», «Программная инженерия», «Программирование в компьютерных сетях», а также при выполнении курсовых проектов и выпускной бакалаврской работы.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины «Программирование на языках высокого уровня»

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО по данному направлению:

Коды компетенции	Наименование компетенции	Структурные элементы компетенции (в результате освоения дисциплины обучающийся должен знать, уметь, владеть)
ПК-2	способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение	Уметь: адаптировать прикладное программное обеспечение для решения поставленных задач
ПК-8	способность программировать приложения и создавать программные прототипы решения прикладных задач	Знать: методы структурного и объектно-ориентированного программирования
		Уметь: разрабатывать и отлаживать эффективные алгоритмы с использованием современных технологий программирования
		Владеть: навыками программирования в современных средах
ПК-12	способность проводить тестирование компонентов программного обеспечения ИС	Знать: методы тестирования программ
		Уметь: разрабатывать тесты
		Владеть: навыками тестирования компонентов программного обеспечения

4. Структура и содержание дисциплины «Программирование на языках высокого уровня»

4.1. Структура дисциплины

Общая трудоемкость дисциплины составляет 7 зачетных единиц, 252 часа.

№ п/п	Наименование разделов и тем дисциплины (модуля)	Семестр	Недели семестра	Виды учебной работы, включая самостоятельную работу студентов и трудо- емкость (в часах)						Формы текущего контроля успеваемости (по неделям семестра)		
				Аудиторная работа			Самостоятельная работа			Защита лабораторных работ	Курсовой проект	
				Всего	Лекции	Лабораторные занятия	Всего	Подготовка к лабораторным занятиям	Курсовой проект			Подготовка к экзамену
1	Введение	3	1	1	1							
2	Раздел 1. Основы структурного программирования на языках C/C++	3	1-10	60	20	40	61	21	20	20		
3	Тема 1.1. Переменные и базовые типы языка	3	1	1	1		2		1	1		
4	Тема 1.2. Принятие решений. Условные операторы	3	2	1	1		2		1	1		
5	Тема 1.3. Организация циклов	3	2	1	1		2		1	1		
6	Тема 1.4. Числовые массивы	3	3-4	13	3	10	13	5	4	4	4	
7	Тема 1.5. Символьные массивы. Работа со строками	3	4-5	13	3	10	11	5	3	3	6	
8	Тема 1.6. Указатели	3	6	1	1		2		1	1		
9	Тема 1.7. Динамическое распределение памяти	3	6-7	2	2		4		2	2		
10	Тема 1.8. Функции	3	7-8	13	3	10	11	5	3	3	9	

4.2. Содержание дисциплины

4.2.1. Содержание лекционного курса

Введение

Краткая история возникновения языка C++. Цель и задачи дисциплины и ее место в подготовке бакалавра.

1. Основы структурного программирования на языке C/C++

1.1. Переменные и базовые типы языка

Переменные. Константы. Базовые типы данных. Операции языка. Выражения. Преобразование типов в выражениях. Приведение типов.

1.2. Принятие решений. Условные операторы.

Операторы **if**, **if – else**, **if – else if – else**, **switch – case – default**, оператор условия **?**, операторы перехода **break**, **continue**, безусловный оператор перехода **goto**. Вложенные условные операторы. Логические условия.

1.3. Организация циклов

Операторы цикла **while**, **for**, **do – while**. Составные операторы цикла и операторы отношения. Оператор **continue**.

1.4. Числовые массивы

Одномерные массивы, их объявление. Двумерные массивы. Многомерные массивы. Инициализация массивов.

1.5. Символьные массивы. Работа со строками

Задание и инициализация символьных массивов. Библиотечные функции для работы со строками.

1.6. Указатели

Указатель как средство доступа к данным. Использование указателей в элементарных задачах программирования. Взаимосвязь указателей и массивов. Допустимые операции с указателями и массивами. Массивы указателей. Указатели на указатели.

1.7. Динамическое распределение памяти

Динамически распределяемая область памяти («куча»). Библиотечные функции **calloc()**, **malloc()**, **realloc()** и **free()**. Операторы **new** и **delete**.

1.8. Функции

Объявления и определений функций. Способы задания формальных параметров и типов возвращаемых данных. Вызов функций, передача аргументов по значению и ссылке. Функции с переменным числом аргументов. Подставляемые (встраиваемые) функции. Перегрузка функций. Рекурсивные функции.

1.9. Файловый ввод-вывод

Двоичные и текстовые потоки. Библиотечные функции для работы с файлами. Обработка файлов в потоковом режиме. Прямой доступ к данным файла.

1.10. Структуры и объединения

Определение структуры. Инициализация структуры. Массив структур. Оператор **typedef**. Структуры и функции. Объединения. Перечислимые типы.

1.11. Препроцессор языка

Назначение препроцессора. Основные директивы препроцессора.

2. Основы объектно-ориентированного и визуального программирования на языке C++

2.1. Классы.

Понятие абстрактных типов данных. Общедоступные, защищенные и скрытые данные. Операции «.» и «->». Область видимости класса.

Функции-члены классов. Инлайн-функции. Инициализация данных. Конструкторы и деструкторы.

Статические члены класса. Указатель **this**. Указатели на члены класса.

Дружественные функции и дружественные классы.

2.2. Наследование.

Построение производного класса.

2.3. Полиморфизм

Раннее и позднее связывание. Виртуальные функции. Абстрактные классы.

2.4. Графика в С++

Использование графики в визуальных приложениях. Графический интерфейс. Рисование графически примитивов: точек, линий, прямоугольников, эллипсов. Отображение графиков функций с помощью графических примитивов.

Заключение

Основные тенденции развития языка С++. Стандарт С++11.

4.2.2. Перечень и содержание лабораторных занятий

Основные цели лабораторного практикума – привитие студентам навыков в проектировании программ, в пользовании справочной литературой и справочными электронными ресурсами, а также подготовке их к самостоятельной работе над курсовым проектом. Тематика задач, решаемых на лабораторных занятиях, определяется содержанием соответствующих разделов рабочей программы. Содержание задач увязывается со спецификой будущей специальности. Форма проведения лабораторного практикума – решение задач, связанных с выполнением отдельных этапов проектирования программного продукта.

№ п/п	№ раздела	Наименование лабораторных работ	Кол. часов
1	1	Разработка программы по обработке числовых массивов (среда Microsoft Visual С++ – консольное приложение)	10
2	1	Разработка программы по обработке текстовой информации (среда Microsoft Visual С++ – консольное приложение)	10
3	1	Разработка программы с собственными функциями (среда Microsoft Visual С++ – консольное приложение)	10
4	1	Разработка приложения, использующего структуры и файлы (среда Microsoft Visual С++ – консольное приложение)	10
5	2	Разработка приложения, работающего с объектами классов (среда Microsoft Visual С++ – приложение Windows Forms)	12
6	2	Разработка приложения, использующего иерархию классов (среда Microsoft Visual С++ – приложение Windows Forms)	12
7	2	Разработка приложения по работе с графикой (среда Borland С++ Builder)	8

4.2.3. Содержание курсового проектирования

Курсовое проектирование заключается в разработке и отладке многомодульного приложения. Разработка ведется в среде Microsoft Visual С++ или Borland С++ Builder.

Темы курсовых проектов

1. Медианный фильтр
2. Справочная система “Маршруты городского транспорта”
3. Программа тестирования студентов
4. Телефонный справочник
5. Расписание занятий студентов

6. Редактор схем алгоритмов
7. Мастер создания визитных карточек
8. Переводчик текстов
9. Игра «Пятнашки»
10. Электронные часы
11. Вычисление определенного интеграла
12. Решение нелинейного алгебраического уравнения
13. Клавиатурный тренажер
14. Учебный графический редактор
15. Органайзер
16. Решение системы линейных алгебраических уравнений
17. Вычисление определенного интеграла
18. Решение нелинейного алгебраического уравнения
19. Численное решение обыкновенных дифференциальных уравнений
20. Системный будильник
21. Имитатор инсталлятора приложений
22. Имитатор почтовой системы
23. “Фоторобот”
24. Работа с комплексными числами
25. Работа с векторами
26. Работа с шестнадцатеричными данными

5. Образовательные технологии

В процессе изучения дисциплины применяются следующие образовательные технологии:

- чтение лекций проводится с использованием мультимедийного компьютерного проектора;
- практически все лекции и часть лабораторных занятий проводятся в интерактивной форме с разбором конкретных фрагментов программ, что составляет примерно 30 – 40 % аудиторных занятий;
- мастер-классы по работе в различных средах программирования (MS Visual C++, Borland C++ Builder);
- при выполнении лабораторного практикума и во время самостоятельной работы используются обучающие программы с сайта кафедры ИВС (<http://ivs-pgy.nm.ru>) и Интернет-ресурсы.

6. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

6.1. План самостоятельной работы студентов

№ нед.	Тема	Вид самостоятельной работы	Задание	Рекомендуемая литература	Кол-во часов
3	Основы структурного программирования на языках C/C++	Подготовка к лабораторному занятию. Оформление отчета по лабо-	Изучение работы с одномерными и двумерными числовыми массивами. Разработка	/1, 2/	5

		ракторной работе	программы по обработке числовых массивов		
4	Основы структурного программирования на языках C/C++	Курсовое проектирование	Анализ предметной области. Разработка технического задания на проект	/1, 2/	4
5	Основы структурного программирования на языках C/C++	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение библиотечных функций для работы со строками. Разработка программы по обработке текстовой информации	/1, 2/	5
6	Основы структурного программирования на языках C/C++	Курсовое проектирование	Разработка алгоритмов работы программы	/1, 2/	6
7	Основы структурного программирования на языках C/C++	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение объявления, определения и вызова функций. Разработка программы с собственными функциями	/1, 2/	5
8	Основы структурного программирования на языках C/C++	Курсовое проектирование	Разработка консольного варианта программы. Разработка тестов.	/1, 2/	8
10	Основы структурного программирования на языках C/C++	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение библиотечных функций по работе с файлами. Разработка приложения, использующего структуры и файлы	/1, 2/	6
11	Основы структурного программирования на языках C/C++	Курсовое проектирование	Отладка консольного приложения. Оформление пояснительной записки.	/1, 2/	6
13	Основы объектно-ориентированного и визуального программирования на языке C++	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение структуры класса и ее программной реализации. Разработка приложения, работающего с объектами классов	/1, 2/	5

14	Основы объектно-ориентированного и визуального программирования на языке C++	Курсовое проектирование	Разработка диаграммы классов приложения.	/1, 2/	2
16	Основы объектно-ориентированного и визуального программирования на языке C++	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение механизмов наследования и полиморфизма. Разработка приложения, использующего иерархию классов	/1, 2/	5
17	Основы объектно-ориентированного и визуального программирования на языке C++	Курсовое проектирование	Разработка и отладка варианта программы с графическим интерфейсом. Оформление пояснительной записки.	/1, 2/	10
18	Основы объектно-ориентированного и визуального программирования на языке C++	Подготовка к лабораторному занятию. Оформление отчета по лабораторной работе	Изучение библиотечных функций по работе с графикой. Разработка приложения по работе с графикой	/1, 2/	5

6.2. Методические указания по организации самостоятельной работы студентов

Планируются следующие виды самостоятельной работы:

- подготовка к лабораторным работам занятиям;
- оформление отчётов по лабораторным работам;
- работа по выполнению курсового проекта;
- подготовка к зачету;
- работа с конспектом лекций и изучение литературы при подготовке к экзаменам.

В качестве учебно-методического обеспечения по организации самостоятельной работы студентов при подготовке к аудиторным занятиям рекомендуется использовать основную и дополнительную литературу, а также электронные методические указания по данной дисциплине, размещенные в университетской локальной сети. Также во время самостоятельной работы используются материалы сайта «Интернет Университет Информационных Технологий» (<http://www.intuit.ru>).

6.3. Материалы для проведения текущего и промежуточного контроля знаний студентов

Контроль освоения компетенций

№ п\п	Вид контроля	Контролируемые темы (разделы)	Компетенции, компоненты которых контролируются
1	Текущий: собеседование при защите лабораторных работ	Разделы 1 – 2	ПК-2, ПК-8, ПК-12

2	Промежуточный: защита курсового проекта	Разделы 1 – 2	ПК-2, ПК-8, ПК-12
3	Промежуточный: зачет по результатам защиты лабораторных работ	Разделы 1 – 2	ПК-2, ПК-8, ПК-12
4	Промежуточный: экзамен (вопрос и задание)	Разделы 1 – 2	ПК-2, ПК-8, ПК-12

Примерный перечень вопросов и заданий к экзамену

Вопросы к экзамену

1. Структура программы на языке C/C++.
2. Понятия константы и переменной.
3. Статические и динамические переменные.
4. Глобальные и локальные данные.
5. Базовые типы данных языка C/C++: объявления, область значения, константы.
6. Логические тип данных в C++.
7. Стандартные целочисленные типы данных языка C/C++.
8. Представление вещественных чисел. Стандартные вещественные типы языка C/C++.
9. Представление символьных данных. Стандартный тип CHAR.
10. Что называется массивом в языках программирования?
11. Что называется структурой (записью) в языках программирования?
12. Логические операции в языках программирования.
13. Каким образом реализуется простое ветвление в программе?
14. Каким образом реализуется множественное ветвление в программе?
15. Каким образом реализуется цикл с предусловием в программе?
16. Каким образом реализуется цикл с постусловием в программе?
17. Каким образом реализуется цикл с заданным числом повторений?
18. Что такое составной оператор и как он реализуется?
19. Операторы досрочного завершения цикла.
20. Для чего и как выполняется связывание физического файла с логическим файлом?
21. Перечислите основные файловые типы.
22. Чем отличается последовательный доступ к компонентам файла от прямого?
23. Что такое указатель в программе?
24. Что такое разыменованье?
25. Что такое линейный односвязный список?
26. Принцип структурного программирования.
27. Чем функция main() отличается от других функций в C/C++?
28. Что такое рекурсия?
29. Структура модуля.
30. Концепции восходящего и нисходящего программирования.
31. Понятия объекта и класса в объектно-ориентированном программировании.
32. Принцип инкапсуляции в объектно-ориентированном программировании.
33. Наследование в объектно-ориентированном программировании.
34. Раннее и позднее связывание.
35. Понятие виртуальных методов в объектно-ориентированном программировании.
36. Какой метод класса называется конструктором?
37. Какой метод класса называется деструктором?
38. Понятие компонента и принцип компонентного программирования.

39. Что называется свойством компонента?
40. Понятие события компонента. Обработчик события.
41. Как организована иерархия визуальных компонентов в среде Borland C++ Builder?

Задания к экзамену

Среда разработки – Microsoft Visual C++ (консольное приложение).

1. Разработать программу, которая определяет, является ли введенная последовательность целых чисел произвольной длины монотонно возрастающей или нет.
2. Разработать программу, которая вводит текст на английском языке и выводит в алфавитном порядке все буквы, которые входят в этот текст по одному разу.
3. Разработать программу, которая вводит текст произвольной длины, определяет и выводит частоту распределения длин слов текста. (Знаками препинания в тексте пренебречь).
4. Разработать программу, которая вводит текст и выводит самое длинное слово (слова) этого текста.
5. Разработать программу, которая вводит количество точек, расположенных на плоскости, и находит пару самых удаленных друг от друга точек. Координаты точек формируются программой случайным образом.
6. Дан одномерный целочисленный массив из N элементов (N вводится), значения которых формируются случайным образом в диапазоне $[-99, 99]$. Найти количество и процентное соотношение положительных, отрицательных и нулевых элементов.
7. Разработать и испытать функцию для проверки баланса парных скобок в строке, то есть количество открытых скобок должно совпадать с количеством закрытых, причём необходимо отличать тип скобок – круглые, квадратные, фигурные. Функция возвращает значение типа `int` равное **0** – если баланс скобок нарушен, равно **1** – если баланс скобок соблюдается, равно **-1** – если скобок в строке нет.

```
// прототип функции проверки баланса парных скобок
int balanceBrackets(char *);
```
8. Дан одномерный целочисленный массив из N элементов, заданных случайными числами на промежутке $[a; b)$. Значения N, a, b вводятся с клавиатуры. Заменить все элементы массива, кратные 3, на сумму их цифр.
9. Дан одномерный целочисленный массив из N элементов, заданных случайными числами на промежутке $[a; b)$. Значения N, a, b вводятся с клавиатуры. Поменять местами первый минимальный и последний максимальный элементы массива.
10. Разработать программу выбора из заданного на плоскости множества точек N (N вводится) трех точек, не лежащих на одной прямой, составляющих треугольник наименьшей площади. Координаты точек формируются программой случайным образом.
11. Заданы два вектора с различным количеством элементов и натуральное число k (количество элементов векторов и k вводятся с клавиатуры). Объединить их в один вектор, включив второй вектор между k -м и $(k+1)$ -м элементами первого. Векторы должны храниться в памяти в виде динамических массивов, элементы их – случайные целые числа, распределенные по равномерному закону на интервале $[-99; 99]$.
12. Разработать программу, определяющую сумму накопления в банке по первоначальному вкладу, виду счета и количеству прошедших лет.
13. Разработать программу, вычисляющую
$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$$
 с точностью ϵ (значения x и ϵ вводятся с клавиатуры).
14. Известно, что в целочисленной матрице нет нулей. Заменить повторяющиеся в матрице элементы нулями. Элементы матрицы генерируются случайным образом в диапазоне $[-99, 99]$, размеры матрицы вводятся пользователем.

15. Разработать программу поиска максимального элемента среди минимальных элементов строк целочисленной матрицы размером $n \times m$. Значения элементов матрицы формируются случайным образом в диапазоне $[a, b]$ (значения n, m, a, b вводятся).

16. Разработать программу, которая формирует матрицу размером $m \times n$ (m и n вводятся с клавиатуры, элементы матрицы – случайные целые числа, распределенные по равномерному закону на интервале $[-99, 99]$), находит среднее арифметическое элементов матрицы, расположенных на главной диагонали и увеличивает каждый элемент матрицы на эту величину.

17. Разработать и испытать функцию "сжатия" исходной последовательности символов: каждая подпоследовательность, состоящая из нескольких вхождений одного и того же символа заменяется на текст $x(k)$, где x – символ, а k – строка, являющаяся записью числа вхождений в исходную последовательность.

18. Разработать программу, которая в вещественной квадратной матрице сортирует по возрастанию элементы каждой строки верхнего треугольника, по убыванию – нижнего, оставляет без изменения диагональные элементы. Элементы матрицы генерируются случайным образом в диапазоне $[-99.99, 99.99]$, размер матрицы вводится с клавиатуры.

7. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература

1. Конова, Е.А. Алгоритмы и программы. Язык С++ [Электронный ресурс]: учебное пособие для вузов [Гриф УМО] / Е.А. Конова, Г.А. Поллак. – СПб.: Лань, 2016. – 384 с. – Режим доступа: <http://e.lanbook.com/reader/book/90158/>. – Электрон. версия печ. публикации. – ЭБС Лань: требуется авторизация пользователя.

2. Подбельский, В.В. Курс программирования на языке Си [Электронный ресурс]: учебник для вузов [Гриф УМО] / В.В. Подбельский, С.С. Фомин. – М.: ДМК Пресс, 2013. – 384 с. – Режим доступа: <http://e.lanbook.com/reader/book/4148/>. – Электрон. версия печ. публикации. – ЭБС Лань: требуется авторизация пользователя.

б) дополнительная литература

3. Александров, Э.Э. Программирование на языке С в Microsoft Visual Studio 2010: учеб. пособие/ Э.Э. Александров, В.В. Афонин. – Саранск: Изд-во Мордов. ун-та, 2010. – 428 с. (Интернет-Университет Информационных Технологий (INTUIT.ru) – <http://www.intuit.ru/department/pl/prcmsvs2010/>).

4. Липачев, Е.К. Технология программирования. Базовые конструкции С/С++: учебно-справочное пособие/ Е.К. Липачев. – Казань: Казан. ун-т, 2012. – 142 с. (http://window.edu.ru/resource/139/78139/files/prog_C.pdf).

в) программное обеспечение и Интернет – ресурсы

1) Программные продукты Borland С++ Builder версии 6 или выше, Microsoft Visual С++ версии 2010 или выше.

2) Интернет-ресурсы

– Материалы раздела «Программирование» сайта «Интернет – Университет Информационных Технологий», <http://www.intuit.ru/>

– ЭБС Лань

– ЭБС biblio-online.ru

– Глухих М.И. Основы программирования на языках С и С++: Курс лекций «Программирование» (презентация). – Сайт кафедры компьютерных систем и программных технологий СПбГПУ (<http://kspt.ftk.spbstu.ru/course/cpp14>).

8. Материально-техническое обеспечение дисциплины

Лекционные занятия проводятся в аудитории, оснащенной компьютерным проектором, проекционным экраном, шторами, сетью электропитания 220 В.

Лабораторные занятия проводятся в классе, оснащенный персональными компьютерами с операционной системой Windows XP/Windows Vista/Windows 7/8/10 или Linux.

Рабочая программа дисциплины «Программирование на языках высокого уровня» составлена в соответствии с требованиями ФГОС ВО по направлению 09.03.03 «Прикладная информатика»,

Программу составил:

1. доцент каф. ИВС



(подпись)

Г.Ф. Убиенных

Настоящая программа не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.

Программа одобрена на заседании кафедры «Информационно-вычислительные системы»

Протокол № 12 от 15.06.2015 года

Зав. кафедрой ИВС



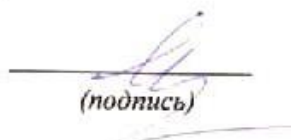
(подпись)

Ю.Н. Косников

Программа одобрена методической комиссией ФВТ

Протокол № 6 от «15» 06 2015 года

Председатель методической комиссии ФВТ



(подпись)

Н.Н. Коннов

