



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

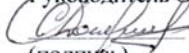
Согласовано

«УТВЕРЖДАЮ»

Школа естественных наук

Заведующий кафедрой
компьютерных систем

Руководитель ОП

 Должииков С.В.
(подпись) (Ф.И.О. рук. ОП)
« 18 » июня 2015 г.



 Кулешов Е.Л.
(подпись)
« 18 » июня 2015 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Информатика и основы программирования

Направление подготовки 09.03.02 Информационные системы и технологии

Профиль подготовки: «Информационные системы и технологии в связи»

Форма подготовки очная

курс 1, 2 семестр 1-3
лекции 54 час.
практические занятия 18 час.
лабораторные работы 72 час.
в том числе с использованием МАО лек. 0 / пр. 0 / лаб. 0 час.
всего часов аудиторной нагрузки 144 час.
в том числе с использованием МАО 0 час.
самостоятельная работа 180 час.
в том числе на подготовку к экзамену 81 час.
контрольные работы не предусмотрены
курсовая работа/курсовой проект не предусмотрен
зачет не предусмотрен
экзамен 1-3 семестр

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования, утвержденного приказом Министерства образования и науки РФ от 12.03.2015 № 219

Рабочая программа обсуждена на заседании кафедры компьютерных систем, протокол № 14 от «18» июня 2015 г.

Заведующий кафедрой Кулешов Е.Л.

Составитель: доцент Пустовалов Е.В.

I. Рабочая программа пересмотрена на заседании кафедры:

Протокол от «_____» _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании кафедры:

Протокол от «_____» _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

Аннотация к рабочей программе дисциплины «Информатика и основы программирования»

Дисциплина «Информатика и основы программирования» предназначена для студентов, обучающихся по направлению подготовки 09.03.02 Информационные системы и технологии, профиль «Информационные системы и технологии в связи», и входит в базовую часть блока «Дисциплины (модули)» учебного плана (индекс Б1.Б.14).

Общая трудоемкость освоения дисциплины составляет 8 зачетных единиц, 288 часа. Учебным планом предусмотрены лекционные занятия (54 часа), практические занятия (18 часов), лабораторные работы (72 часа), самостоятельная работа студентов (144 часа, в том числе 81 час на подготовку к экзаменам). Дисциплина реализуется на 1 и 2 курсах в 1-3 семестрах. Форма промежуточной аттестации – экзамены.

Дисциплина «Информатика и основы программирования» опирается на уже изученные дисциплины, такие как «Информатика» в средней школе. В свою очередь она является базой для освоения других дисциплин учебного плана, таких как «Технологии программирования», «Технологии обработки информации», «Моделирование систем», «Основы высокопроизводительных вычислений», «Языки ассемблера».

Содержание дисциплины охватывает следующий круг вопросов: системы счисления, представление информации в ЭВМ, линейные, разветвляющиеся и циклические алгоритмы, структура программы на языке С++, переменные, операции, выражения, ввод/вывод информации, ветвление, циклы, массивы и указатели, алгоритмы обработки массивов, динамические массивы, функции, обработка символьной информации, структуры данных, динамические структуры, классы, потоковые классы ввода/вывода информации, строковый класс, контейнерные классы и алгоритмы.

Целью изучения дисциплины «Информатика и основы программирования» является формирование у обучающихся теоретических

знаний основных понятий в области информатики и программирования и практических навыков составления алгоритмов и написания программ.

Задачи дисциплины:

- изучение языка программирования C++;
- знакомство с методами структурного и объектно-ориентированного программирования;
- знакомство с основными структурами данных и типовыми алгоритмами их обработки;
- развитие навыков алгоритмизации и программирования;
- развитие готовности создавать программные продукты для решения прикладных задач в различных областях.

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы, характеризуют этапы формирования следующих компетенций:

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК-3 способность применять основные приемы и законы создания и чтения чертежей и документации по аппаратным и программным компонентам информационных систем	Знает	основные подходы к документированию программ
	Умеет	документировать программные компоненты информационных систем
	Владеет	навыками сопровождения программных продуктов
ОПК-5 способность использовать современные компьютерные технологии поиска информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению	Знает	основные структуры данных и методы структурного и объектно-ориентированного программирования
	Умеет	осуществлять поиск информации, необходимой для профессиональной деятельности
	Владеет	навыками критического анализа и обоснования идей и подходов при создании программных средств
ПК-11 способность к проектированию базовых и прикладных информационных технологий	Знает	основные алгоритмические структуры, структуры данных и типовые алгоритмы обработки данных
	Умеет	применять методы алгоритмизации и программирования при проектировании информационных систем
	Владеет	навыками проектирования, реализации и внедрения проекта информационной системы

ПК-12 способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные)	Знает	один из основных языков программирования
	Умеет	реализовывать алгоритмы в виде программного кода
	Владеет	навыками алгоритмизации и программирования для разработки программных средств
ПК-25 способность использовать математические методы обработки, анализа и синтеза результатов профессиональных исследований	Знает	основные приемы алгоритмизации математических методов
	Умеет	реализовать в виде программного кода математические методы обработки данных
	Владеет	навыками использования математических методов в профессиональной деятельности

Для формирования вышеуказанных компетенций в рамках дисциплины «Информатика и основы программирования» применяются следующие методы активного обучения: лекция-беседа, экспресс-опрос, анализ конкретных ситуаций.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Раздел I. Информация и алгоритмы (6 час.)

Тема 1. Представление информации в ЭВМ (4 час.)

Системы счисления. Формы представления целых и дробных чисел, правила выполнения действий с числами. Представление текстовой и символьной информации. Представление графической информации. Количество информации.

Тема 2. Алгоритмы (2 час.)

Алгоритмы и их свойства. Базовые алгоритмические конструкции: линейные, разветвляющиеся, циклические.

Раздел II. Структурное программирование (30 час.)

Тема 1. Базовые средства языка C/C++ (4 час.)

Типы данных. Переменные. Константы. Операции. Выражения. Структура программы на Си.

Тема 2. Организация ввода/вывода (4 час.)

Вывод на экран и ввод с клавиатуры в стиле Си (функции printf и scanf). Поточковый ввод/вывод в стиле C++.

Тема 3. Базовые конструкции структурного программирования (8 час.)

Оператор присваивания. Логические операции. Условный оператор if. Вложенные условные операторы. Сложные условия. Оператор выбора switch. Циклические операторы for, while, do while. Вложенные циклы.

Тема 4. Массивы и указатели (8 час.)

Понятие массива. Базовые алгоритмы обработки массивов (ввод, вывод, суммирование, подсчет количества элементов, поиск, сортировка). Двумерные массивы. Указатели и операции с ними. Связь массивов и указателей. Динамическое выделение памяти.

Тема 5. Функции (6 час.)

Понятие функции. Объявление, определение и вызов функции. Оператор return. Параметры (аргументы) функции. Использование массивов, указателей и ссылок в качестве параметров функции. Рекурсия. Функция main().

Раздел III. Структуры данных (10 час.)

Тема 1. Символьные строки (2 час.)

Понятие строки. Ввод/вывод строк. Библиотечные функции для работы со строками.

Тема 2. Структуры (2 час.)

Понятие структуры и её объявление. Доступ к полям структуры. Перегрузка операций для структурных типов данных.

Тема 3. Динамические структуры (6 час.)

Понятие о динамических структурах. Организация основных типов динамических структур (стек, очередь, линейный список, бинарное дерево).

Раздел IV. Основы объектно-ориентированного программирования (8 час.)

Тема 1. Классы (4 час.)

Описание класса. Данные и методы класса. Описание объектов. Вызов методов класса. Указатель this. Конструкторы и деструкторы. Наследование.

Тема 2. Контейнерные классы (4 час.)

Векторы. Очереди. Списки. Стеки. Словари. Примеры использования контейнерных классов.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Практические занятия (18 час.)

Занятие 1. Использование символов русского языка в консольных приложениях (2 час.)

1. Проблема кодирования символов русского языка.
2. Достоинства и недостатки различных способов русификации консольных приложений.

Занятие 2. Определение и использование структур (4 час.)

1. Определение структуры для представления геометрических векторов. Перегрузка операций для векторов.
2. Определение структуры для хранения информации о студентах. Создание массива структур и передача его в функцию. Поиск в массиве структур.

Занятие 3. Динамические структуры (4 час.)

1. Организация стека.
2. Организация очереди с чтением данных из файла.

Занятие 4. Описание и использование классов (4 час.)

1. Описание класса для работы с датами.
2. Описание класса для хранения информации о человеке. Наследование классов.

**Занятие 5. Использование стандартных библиотечных классов
(4 час.)**

1. Использование строкового класса.
2. Использование контейнерных классов.

Лабораторные работы (72 час.)

Лабораторная работа № 1. Ввод/вывод информации (2 час.)

Лабораторная работа № 2. Условный оператор (2 час.)

Лабораторная работа № 3. Циклы (2 час.)

Лабораторная работа № 4. Массивы (2 час.)

Лабораторная работа № 5. Двумерные массивы (2 час.)

Лабораторная работа № 6. Функции (2 час.)

Лабораторная работа № 7. Обработка символьной и строковой информации (2 час.)

Лабораторная работа № 8. Структуры. Перегрузка операций (2 час.)

Лабораторная работа № 9. Массивы структур (2 час.)

Лабораторная работа № 10. Реализация стека (4 час.)

Лабораторная работа № 11. Реализация очереди (10 час.)

Лабораторная работа № 12. Реализация линейного списка (10 час.)

Лабораторная работа № 13. Описание класса (10 час.)

Лабораторная работа № 14. Использование строкового класса (6 час.)

Лабораторная работа № 15. Использование контейнерных классов (14 час.)

**III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ
САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Информатика и основы программирования» представлено в Приложении 1 и включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристику заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства		
			текущий контроль	промежуточная аттестация	
1	Раздел I. Информация и алгоритмы	ОПК-5	Знает	ПР-1	вопросы к экзамену 1-5
		ПК-11	Умеет	ПР-13	практические задания
		ПК-25	Владеет	ПР-6	лабораторные работы
2	Раздел II. Структурное программирование	ОПК-3	Знает	ПР-1	вопросы к экзамену 6-27
		ПК-12	Умеет	ПР-13	практические задания 1-2
		ПК-25	Владеет	ПР-6	лабораторные работы 1-6
3	Раздел III. Структуры данных	ПК-11	Знает	ПР-1	вопросы к экзамену 28-36
		ПК-12	Умеет	ПР-6	практические задания 3-4
			Владеет	ПР-6	лабораторные работы 7-9
4	Раздел IV. Основы объектно-ориентированного программирования	ОПК-3	Знает	ПР-1	вопросы к экзамену 28-36
		ПК-12	Умеет	ПР-6	практическое задание 5
			Владеет	ПР-6	лабораторные работы 10-15

ПР-1 – тестирование

ПР-6 – лабораторная работа

ПР-13 – индивидуальные задания

Тестовые задания, индивидуальные задания, задания для лабораторных работ, вопросы к экзамену, методические материалы, определяющие процедуры оценивая знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений,

навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

(электронные и печатные издания)

№	Название	Ссылка в ЭК НБ ДВФУ	Внешняя ссылка
1	Кирнос, В.Н. Информатика 2. Основы алгоритмизации и программирования на языке C++ [Электронный ресурс] : учебно-методическое пособие / В.Н. Кирнос. – Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2013. – 160 с.	http://lib.dvfu.ru:8080/lib/item?id=IPRbooks:IPRbooks-14011&theme=FEFU	http://www.iprbookshop.ru/14011.html
2	Колдаев, В.Д. Структуры и алгоритмы обработки данных: Учебное пособие / В.Д. Колдаев. – М.: ИЦ РИОР, НИЦ ИНФРА-М, 2014. – 296 с.	http://lib.dvfu.ru:8080/lib/item?id=Znanium:Znanium-418290&theme=FEFU	http://znanium.com/bookread2.php?book=418290
3	Комлев, Н.Ю. Объектно-ориентированное программирование / Н.Ю. Комлев. – М.: СОЛОН-Пр., 2015. – 298 с.	http://lib.dvfu.ru:8080/lib/item?id=Znanium:Znanium-884394&theme=FEFU	http://znanium.com/bookread2.php?book=884394
4	Кузин, А.В. Программирование на языке Си / А.В. Кузин, Е.В. Чумакова. – М.: Форум, НИЦ ИНФРА-М, 2015. – 144 с.	http://lib.dvfu.ru:8080/lib/item?id=Znanium:Znanium-505194&theme=FEFU	http://znanium.com/bookread2.php?book=505194
5	Царев, Р. Ю. Программирование на языке Си: учеб. пособие / Р. Ю. Царев. – Красноярск: Сиб. федер. ун-т, 2014. – 108 с.	http://lib.dvfu.ru:8080/lib/item?id=Znanium:Znanium-510946&theme=FEFU	http://znanium.com/bookread2.php?book=510946

Дополнительная литература

(электронные и печатные издания)

№	Название	Ссылка в ЭК НБ ДВФУ	Внешняя ссылка
1	Дейл, Н. Программирование на C++ [Электронный ресурс] : самоучитель / Н. Дейл, Ч. Уимз, М. Хедингтон. – М.: ДМК Пресс,	http://lib.dvfu.ru:8080/lib/item?id=Lan:Lan-1219&theme=FEFU	https://e.lanbook.com/reader/book/1219

	2007. – 672 с.		
2	Кивран, В.К. Программирование в среде Visual C++ 6 [Электронный ресурс] : учебное пособие / В.К. Кивран. – Самара: Самарский государственный архитектурно-строительный университет, ЭБС АСВ, 2014. – 118 с.	http://lib.dvfu.ru:8080/lib/item?id=IPRbooks:IPRbooks-43185&theme=FEFU	http://www.iprbookshop.ru/43185.html
3	Курипта, О.В. Основы программирования и алгоритмизации [Электронный ресурс] : практикум / О.В. Курипта, О.В. Минакова, Д.К. Проскурин. – Воронеж: Воронежский государственный архитектурно-строительный университет, ЭБС АСВ, 2015. – 133 с.	http://lib.dvfu.ru:8080/lib/item?id=IPRbooks:IPRbooks-59123&theme=FEFU	http://www.iprbookshop.ru/59123.html
4	Основы алгоритмизации и программирования [Электронный ресурс] : лабораторный практикум. – Ставрополь: Северо-Кавказский федеральный университет, 2015. – 211 с.	http://lib.dvfu.ru:8080/lib/item?id=IPRbooks:IPRbooks-63112&theme=FEFU	http://www.iprbookshop.ru/63112.html
5	Сорокин, А.А. Объектно-ориентированное программирование [Электронный ресурс] : учебное пособие. Курс лекций / А.А. Сорокин. – Ставрополь: Северо-Кавказский федеральный университет, 2014. – 174 с.	http://lib.dvfu.ru:8080/lib/item?id=IPRbooks:IPRbooks-63110&theme=FEFU	http://www.iprbookshop.ru/63110.html
6	Страуструп, Б. Дизайн и эволюция C++ [Электронный ресурс] / Б. Страуструп. – М.: ДМК Пресс, 2007. – 448 с.	http://lib.dvfu.ru:8080/lib/item?id=Lan:Lan-1222&theme=FEFU	https://e.lanbook.com/reader/book/1222

Перечень ресурсов информационно-телекоммуникационной сети

Интернет

1. Основы программирования на языках Си и C++ для начинающих <http://cppstudio.com/>
2. Портал о программировании <https://code-live.ru/>
3. C++ для начинающих <http://proginfo.ru/>
4. Справка по C++ <https://ru.cppreference.com>

5. Справочник по языку С++ <https://msdn.microsoft.com/ru-ru/library/3bstk3k5.aspx>

6. Форум программистов С++ <http://www.cyberforum.ru/cpp/>

Перечень информационных технологий и программного обеспечения

Место расположения компьютерной техники, на котором установлено программное обеспечение	Перечень программного обеспечения
Владивосток, о. Русский, п. Аякс д.10, корпус L, ауд. L 450 специализированная лаборатория кафедры КС: Лаборатория администрирования информационных систем	Microsoft Office Professional Plus 2019, подтверждающий документ № ЭА-261-18, дата окончания лицензии 30.06.2020. Microsoft Windows 10 Enterprise LTSC 2019, подтверждающий документ № ЭА-261-18, дата окончания лицензии 30.06.2020. Microsoft SQL Server Standard Core 2017, подтверждающий документ № ЭА-261-18, дата окончания лицензии 30.06.2020. Microsoft Office Professional 2003, авторизационный номер лицензиата №18597359ZZE0701, бессрочная лицензия Academic, номер лицензии №18643295. Microsoft Windows 7 Professional, авторизационный номер лицензиата №65541663ZZE1106, бессрочная лицензия Academic, номер лицензии №46260303. Adobe Acrobat Pro DC, подтверждающий документ № ЭА-442-15 от 18.01.16 лот 1, дата окончания лицензии 20.01.2019.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

В соответствии с целями и задачами дисциплины студент на аудиторных занятиях последовательно осваивает разделы и темы курса, готовится дома к практическим и лабораторным занятиям, выполняет задания для самостоятельной работы, проходит в течение семестра текущий контроль, сдает по окончании семестра экзамен.

Учебный процесс по дисциплине «Информатика и основы программирования» строится в следующей последовательности:

- посещение лекционных занятий, изучение и конспектирование теоретического материала по разделам и темам курса;
- подготовка к практическим и лабораторным занятиям на основе лекционного материала, а также с использованием основной и дополнительной литературы;
- активная работа на практических занятиях, способствующая расширению, закреплению и более глубокому усвоению материала;
- выполнение лабораторных работ, в ходе которых формируются практические навыки составления алгоритма решения задачи и его реализации на языке программирования;
- выполнение индивидуальных заданий в рамках самостоятельной работы для формирования устойчивых навыков успешного применения полученных знаний и умений;
- посещение консультаций с целью выяснения вопросов, возникающих при изучении конспектов лекций, подготовке к практическим и лабораторным занятиям, выполнении лабораторных работ и индивидуальных заданий;
- подготовка к экзамену.

Большое значение имеет хорошо продуманная организация труда студента, и прежде всего правильная организация времени. Для сокращения затрат времени на изучение дисциплины в первую очередь, необходимо своевременно выяснить, какой объем информации следует усвоить, какие умения приобрести для успешного освоения дисциплины, какие задания

выполнить для того, чтобы получить достойную оценку. Сведения об этом (списки рекомендуемой и дополнительной литературы, темы практических занятий, лабораторных работ, тестовые задания, задания для самостоятельной работы, другие необходимые материалы) имеются в разработанной рабочей учебной программы дисциплины.

Регулярное посещение лекций, лабораторных и практических занятий не только способствует успешному овладению профессиональными знаниями, но и помогает наилучшим образом организовать время, т.к. все виды занятий распределены в течение семестра планомерно, с учетом необходимых временных затрат. Важная роль в планировании и организации времени на изучение дисциплины отводится знакомству с планом-графиком выполнения самостоятельной работы студентов по дисциплине. В нем содержится виды самостоятельной работы для всех разделов дисциплины, указаны примерные нормы времени на выполнение и сроки сдачи заданий.

Важнейшей составной частью освоения курса является посещение лекций и обязательное их конспектирование. Глубокому освоению лекционного материала способствует предварительная подготовка, включающая чтение конспекта предыдущей лекции, работу с основной и дополнительной литературой.

Конспектирование лекции – важный шаг в запоминании материала, поэтому конспект лекций необходимо иметь каждому студенту. Задача студента на лекции – одновременно слушать преподавателя, анализировать и конспектировать информацию. При этом как свидетельствует практика, не нужно стремиться вести дословную запись.

После окончания лекционного занятия следует провести дополнительную работу с текстом конспекта: внимательно прочесть и проанализировать его, при этом необходимо расшифровать все имеющиеся сокращения, выявить непонятные места с тем, чтобы в дальнейшем выяснить их на консультации у преподавателя. Важно уметь оформить конспект так, чтобы основные моменты были выделены каким-либо образом и сразу бросались в глаза.

Хороший конспект – залог четких ответов на занятиях, успешного прохождения контрольных мероприятий, выполнения самостоятельной работы. Значимость конспектирования на лекционных занятиях несомненна. Проверено, что составление эффективного конспекта лекций может сократить в четыре раза время, необходимое для полного восстановления нужной информации.

Также необходимо помнить, что именно конспект лекций играет первостепенную роль при подготовке к экзамену, так как в отличие от учебных пособий он, как правило, содержит более свежую, актуальную информацию в объеме, соответствующем целям и задачам дисциплины.

Целью практических занятий является закрепление, расширение, углубление теоретических знаний, полученных на лекциях и в ходе самостоятельной работы, развитие познавательных способностей. Посещение и активная работа на практических занятиях создает задел для успешного выполнения лабораторных работ.

Выполнение лабораторных работ имеет целью сформировать у обучающегося практические навыки применения полученных знаний, заключающиеся в умении составить алгоритм решения задачи и эффективно реализовать его в программе.

При подготовке к лабораторной работе обучающийся должен заранее изучить материал по теме работы, используя прежде всего конспекты лекций. Необходимо не только рассмотреть теоретические вопросы, но и тщательно разобрать имеющиеся примеры. При наличии вопросов, непонятных мест следует обратиться к основной и дополнительной литературе, получить консультацию у преподавателя.

Без серьезной предварительной подготовки обучающийся может не успеть выполнить лабораторную работу за отведенное время или вообще не справиться с заданием. Невыполнение всех или большей части лабораторных работ ведет к недопуску студента к зачету.

Важной частью работы студента является знакомство с рекомендуемой основной и дополнительной литературой, поскольку лекционный материал,

при всей его важности для процесса изучения дисциплины, содержит лишь минимум необходимых теоретических сведений. Кроме того, получение образования предполагает не только усвоение информации, но и формирование навыков исследовательской работы.

Во время изучения литературы следует конспектировать и составлять рабочие записи прочитанного, которые могут быть сделаны и в виде простого и развернутого плана, цитирования, тезисов, резюме, аннотации, конспекта. Такие записи удлиняют процесс проработки, изучения книги, но способствуют ее лучшему осмыслению и усвоению, выработке навыков кратко и точно излагать материал.

Наиболее надежный способ – составить конспект, т.е. краткое письменное изложение основного содержания. Составление конспекта требует активной мыслительной работы. Конспектируемый материал содержит информацию трех видов: главную, второстепенную и вспомогательную. Главной является информация, имеющая основное значение для раскрытия сущности того или иного вопроса, темы. Второстепенная информация служит для пояснения, уточнения главной мысли. К этому типу информации относятся разного рода комментарии. Назначение вспомогательной информации – помочь читателю лучше понять данный материал. Это всякого рода напоминания о ранее изложенном материале, заголовки, вопросы.

Работая над текстом, следует избегать его механического переписывания. Важно выделять главные положения, вспомогательную информацию при конспектировании не записывают. Желательно оставлять поля для внесения дополнений, поправок или фиксации собственных мыслей по данной записи.

Экзамен – это этап изучения дисциплины, имеющий целью проверить теоретические знания студента, его навыки и умение применять полученные знания при решении практических задач.

Подготовка к экзамену должна начинаться с первого занятия по дисциплине, на котором студенты получают общую установку преподавателя

и перечень основных требований к текущей и итоговой отчетности. При этом важно с самого начала планомерно осваивать материал, руководствуясь перечнем вопросов к экзамену. В течение семестра происходят пополнение, систематизация и корректировка студенческих наработок, освоение нового и закрепление уже изученного материала. Подготовка к экзамену предполагает самостоятельное повторение ранее изученного материала.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для проведения исследований, связанных с выполнением задания по практике, а также для организации самостоятельной работы студентам доступно следующее лабораторное оборудование и специализированные кабинеты, соответствующие действующим санитарным и противопожарным нормам, а также требованиям техники безопасности при проведении учебных и научно-производственных работ:

Наименование оборудованных помещений и помещений для самостоятельной работы	Перечень основного оборудования
Владивосток, о. Русский, п. Аякс д.10, корпус L, ауд. L 502 учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Мультимедийное оборудование: ЖК-панель 47", Full HD, LG M4716 CCBA - 1 шт. Парты и стулья
Владивосток, о. Русский, п. Аякс д.10, корпус L, ауд. L 450 специализированная лаборатория кафедры КС: Лаборатория администрирования информационных систем	11 компьютеров (системный блок модель - 30AGCT01WW P3+монитором AOC 28" L12868POU)
Читальные залы Научной библиотеки ДВФУ с открытым доступом к фонду (корпус А - уровень 10) Учебная аудитория для проведения самостоятельной работы и	Моноблок HP ProOne 400 All-in-One 19,5 (1600x900), Core i3-4150T, 4GB DDR3-1600 (1x4GB), 1TB HDD 7200 SATA, DVD+/-RW, GigE, Wi-Fi, BT, usb kbd/mse, Win7Pro (64-bit)+Win8.1Pro(64-bit), 1-1-1 Wty

подготовки к экзамену	Скорость доступа в Интернет 500 Мбит/сек. Рабочие места для людей с ограниченными возможностями здоровья оснащены дисплеями и принтерами Брайля; оборудованы: портативными устройствами для чтения плоскочечатных текстов, сканирующими и читающими машинами видеоувелечителем с возможностью регуляции цветовых спектров; увеличивающими электронными лупами и ультразвуковыми маркировщиками
-----------------------	---

В целях обеспечения специальных условий обучения инвалидов и лиц с ограниченными возможностями здоровья в ДВФУ все здания оборудованы пандусами, лифтами, подъемниками, специализированными местами, оснащенными туалетными комнатами, табличками информационно-навигационной поддержки.

Приложение 1



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ
РАБОТЫ ОБУЧАЮЩИХСЯ**

по дисциплине «Информатика и основы программирования»

**Направление подготовки 09.03.02 Информационные системы и
технологии**

Профиль подготовки: «Информационные системы и технологии в связи»

Форма подготовки очная

Владивосток

2015

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение (час.)	Форма контроля
1 семестр				
1	3-16 недели семестра	Выполнение индивидуальных задания	9	Отчет по индивидуальному заданию
2	В течение семестра	Подготовка к экзамену	27	Экзамен
2 семестр				
3	В течение семестра	Подготовка к лабораторным работам	27	Отчет по лабораторной работе
4	В течение семестра	Подготовка к экзамену	27	Экзамен
3 семестр				
5	В течение семестра	Подготовка к лабораторным работам	27	Отчет по лабораторной работе
6	В течение семестра	Подготовка к экзамену	27	Экзамен
	Итого		144	

Методические указания по подготовке к лабораторным работам

Выполнение лабораторных работ имеет целью сформировать у обучающегося практические навыки применения полученных знаний, заключающиеся в умении составить алгоритм решения задачи и эффективно реализовать его в программе.

При подготовке к лабораторной работе обучающийся должен заранее изучить материал по теме работы, используя прежде всего конспекты лекций. Необходимо не только рассмотреть теоретические вопросы, но и тщательно разобрать имеющиеся примеры. При наличии вопросов, непонятных мест следует обратиться к основной и дополнительной литературе, получить консультацию у преподавателя.

Без серьезной предварительной подготовки обучающийся может не успеть выполнить лабораторную работу за отведенное время или вообще не справиться с заданием. Невыполнение всех или большей части лабораторных работ ведет к недопуску студента к экзамену.

По каждой лабораторной работе обучающийся должен сформировать отчет в электронном виде, включающий в себя описание алгоритма и тексты программ. Отчет по работе подлежит защите, т.е. студент должен быть готов дать устно ответы на вопросы, касающиеся особенностей алгоритма и его программной реализации.

Методические указания по выполнению индивидуальных заданий

Цель данного вида работы студента – закрепить знания, умения и навыки, полученные в ходе лекционных занятий.

Общими задачами самостоятельной работы студента являются:

- систематизация и закрепление полученных теоретических знаний;
- углубление и расширение теоретических знаний;
- формирование навыков работы с литературой;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений.

Данный вид работы осуществляется под руководством преподавателя, который выполняет функцию управления через контроль и коррекцию ошибок. Работа выполняется в удобное для обучающихся время и представляется преподавателю на проверку в указанные сроки.

Для правильного выполнения индивидуального задания по конкретной теме студент должен предварительно изучить теоретический материал по теме, используя конспекты лекций, основную и дополнительную литературу, иные источники.

По каждому заданию обучающийся должен сформировать отчет в электронном виде. Отчеты сдаются на проверку преподавателю. При наличии замечаний по выполнению задания их нужно исправить и сдать на повторную проверку.

Примерные индивидуальные задания

1. Представление чисел в ЭВМ

1. Представить числа, заданные в одной из систем счисления, в других системах счисления (двоичная, десятичная, шестнадцатеричная):

$$157_{10}, 2780_{10}, 11001001_2, 10010110_2, F2_{16}, 135_{16}$$

2. Записать 16-битовое представление целых чисел в памяти ЭВМ:

а) 285; б) -113

3. Записать 32-битовое представление дробных чисел в памяти ЭВМ:

а) 17; б) -3,25; в) 87,137

2. Вычисления в цикле

1. Составить алгоритм расчета по заданному натуральному значению n суммы $\sum_{i=1}^n \frac{x_i - y_i}{x_i + y_i}$, где $x_1 = 1, y_1 = 1, x_i = y_{i-1} + 1, y_i = 3x_{i-1} + 2y_{i-1}$.

2. Составить алгоритм вычисления суммы ряда для заданного значения x с заданной точностью 10^{-4} . Особое внимание обратить на эффективность алгоритма.

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + K, \quad \text{для } x \in R$$

3. Поиск и сортировка в массивах

1. Составить алгоритм для упорядочения заданной последовательности целых положительных чисел следующим образом: сначала должны располагаться все нечётные числа в порядке убывания, а затем все чётные в порядке возрастания. Например: из последовательности $\{2, 5, 3, 4, 7, 16, 1\}$ должно получиться $\{7, 5, 3, 1, 2, 4, 16\}$. Необходимо учесть, что:

- 1) последовательность может не содержать нечётных чисел;
- 2) последовательность может не содержать чётных чисел.

2. Составить алгоритм, который для двумерного массива целых чисел размера $m \times n$ находит значение величины $z = \min + \max$, где \min – наименьший элемент среди всех столбцов, содержащих больше положительных чисел, чем отрицательных, \max – наибольший элемент среди всех строк, содержащих хотя бы один ноль. Предусмотреть случаи:

1) отсутствия в массиве столбцов, содержащих больше положительных чисел, чем отрицательных; 2) отсутствия в массиве строк, содержащих хотя бы один ноль.

Методические рекомендации по подготовке к экзамену

Формой промежуточного контроля знаний студентов по дисциплине «Информатика и основы программирования» в 1-3 семестрах является экзамен. Экзамен – это этап изучения дисциплины, имеющий целью проверить теоретические знания студента, его навыки и умение применять полученные знания при решении практических задач.

Подготовку к экзамену необходимо начинать с первого занятия по дисциплине, на котором студенты получают общую установку преподавателя и перечень основных требований к текущей и промежуточной аттестации. При этом важно с самого начала планомерно осваивать материал. В течение семестра происходят пополнение, систематизация и корректировка знаний, освоение нового и закрепление уже изученного материала.

Успешное освоение материала дисциплины требует от студента систематической работы:

- 1) не пропускать лекционные занятия;
- 2) не пренебрегать самостоятельной работой по дисциплине, своевременно выполнять индивидуальные задания;
- 3) регулярно обращаться к материалам лекционных занятий, повторять, систематизировать и дополнять их, в том числе с использованием самостоятельно найденной информации.

В экзаменационный билет по дисциплине «Информатика и основы программирования» включаются два теоретических вопроса (1 семестр) или теоретический вопрос и практическое задание (2 и 3 семестры)

Перечень вопросов к экзамену, а также критерии оценки на экзамене приведены в Приложении 2.

Приложение 2



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»
(ДФУ)

ШКОЛА ЕСТЕСТЕННЫХ НАУК

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Информатика и основы программирования»

Направление подготовки

09.03.02 Информационные системы и технологии

Профиль подготовки: «Информационные системы и технологии в связи»

Форма подготовки очная

Владивосток

2015

Паспорт
фонда оценочных средств по дисциплине
Информатика и основы программирования

Код и формулировка компетенции	Этапы формирования компетенции	
	ОПК-3 способность применять основные приемы и законы создания и чтения чертежей и документации по аппаратным и программным компонентам информационных систем	Знает
Умеет		документировать программные компоненты информационных систем
Владеет		навыками сопровождения программных продуктов
ОПК-5 способность использовать современные компьютерные технологии поиска информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению	Знает	основные структуры данных и методы структурного и объектно-ориентированного программирования
	Умеет	осуществлять поиск информации, необходимой для профессиональной деятельности
	Владеет	навыками критического анализа и обоснования идей и подходов при создании программных средств
ПК-11 способность к проектированию базовых и прикладных информационных технологий	Знает	основные алгоритмические структуры, структуры данных и типовые алгоритмы обработки данных
	Умеет	применять методы алгоритмизации и программирования при проектировании информационных систем
	Владеет	навыками проектирования, реализации и внедрения проекта информационной системы
ПК-12 способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные)	Знает	один из основных языков программирования
	Умеет	реализовывать алгоритмы в виде программного кода
	Владеет	навыками алгоритмизации и программирования для разработки программных средств
ПК-25 способность использовать математические методы обработки, анализа и синтеза результатов профессиональных исследований	Знает	основные приемы алгоритмизации математических методов
	Умеет	реализовать в виде программного кода математические методы обработки данных
	Владеет	навыками использования математических методов в профессиональной деятельности

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства		
			текущий контроль	промежуточная аттестация	
1	Раздел I. Информация и алгоритмы	ОПК-5	Знает	ПР-1	вопросы к экзамену 1-5 практические задания лабораторные работы
		ПК-11	Умеет	ПР-13	
		ПК-25	Владеет	ПР-6	

2	Раздел II. Структурное программирование	ОПК-3 ПК-12 ПК-25	Знает	ПР-1	вопросы к экзамену 6-27
			Умеет	ПР-13	практические задания 1-2
			Владеет	ПР-6	лабораторные работы 1-6
3	Раздел III. Структуры данных	ПК-11 ПК-12	Знает	ПР-1	вопросы к экзамену 28-36
			Умеет	ПР-6	практические задания 3-4
			Владеет	ПР-6	лабораторные работы 7-9
4	Раздел IV. Основы объектно-ориентированного программирования	ОПК-3 ПК-12	Знает	ПР-1	вопросы к экзамену 28-36
			Умеет	ПР-6	практическое задание 5
			Владеет	ПР-6	лабораторные работы 10-15

ПР-1 – тестирование

ПР-6 – лабораторная работа

ПР-13 – индивидуальные задания

Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		Критерии	Показатели
	Этап	Этап		
ОПК-3 способность применять основные приемы и законы создания и чтения чертежей и документации и по аппаратным и программным компонентам информационных систем	знает (пороговый уровень)	основные подходы к документированию программ	знание методики работы с документацией по программным компонентам информационных систем	способность охарактеризовать разновидности документации по программным компонентам информационных систем
	умеет (продвинутый)	документировать программные компоненты информационных систем	умение составлять документацию во время всех этапов жизненного цикла информационной системы	способность в совершенстве создавать и читать и документацию по программным компонентам информационных систем
	владеет (высокий)	навыками сопровождения программных продуктов	владение навыками работы с документацией по программным компонентам информационных систем и их сопровождению	способность грамотно документировать и сопровождать программный продукт
ОПК-5 способность использовать современные компьютерные технологии поиска	знает (пороговый уровень)	основные структуры данных и методы структурного и объектно-ориентированного программирования	знание основных структур данных; знание основных принципов структурного программирования; знание	способность перечислить и охарактеризовать основные способы представления данных,

информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению			идеологии объектно-ориентированного программирования	структуры данных; способность перечислить основные принципы структурного программирования; способность перечислить основные принципы объектно-ориентированного программирования
	умеет (продвинутый)	осуществлять поиск информации, необходимой для профессиональной деятельности	умение находить информацию о различных способах представления данных; умение находить информацию о методах, способах, приемах обработки данных	способность найти и применить информацию, требующуюся в ходе разработки, написания и отладки программы
	владеет (высокий)	навыками критического анализа и обоснования идей и подходов при создании программных средств	владение навыками анализа и выбора структур данных и алгоритмов при написании программ	способность осознанно выбрать представление данных и алгоритмы для решения задачи; способность обосновать свой выбор на основе различных подходов к понятию эффективности
ПК-11 способность к проектированию базовых и прикладных информационных технологий	знает (пороговый уровень)	основные алгоритмические структуры, структуры данных и типовые алгоритмы обработки данных	знание основных алгоритмических структур, структур данных и типовых алгоритмов обработки данных	способность охарактеризовать основные алгоритмические структуры; способность описать основные структуры данных; способность описать типовые

				алгоритмы обработки данных; способность перечислить основные принципы логического программирования
	умеет (продвинутый)	применять методы алгоритмизации и программирования при проектировании информационных систем	умение алгоритмизировать прикладные задачи	способность проанализировать задачу в предметной области и предложить подходы к ее решению на основе создания программного продукта
	владеет (высокий)	навыками проектирования, реализации и внедрения проекта информационной системы	владение навыками написания, отладки и тестирования программ на языке C++	способность спроектировать, создать и сопровождать программный продукт для решения прикладной задачи в предметной области
ПК-12 способность разрабатывать средства реализации информационных технологий (методические, информационные, математическое, алгоритмические, технические и программные)	знает (пороговый уровень)	один из основных языков программирования	знание языка программирования C++	способность перечислить основные типы данных; способность перечислить основные операции над данными; способность охарактеризовать операторы языка C++
	умеет (продвинутый)	реализовывать алгоритмы в виде программного кода	умение использовать стандартные алгоритмы; умение реализовывать алгоритмы в виде программного	способность выбрать алгоритм для решения задачи; способность написать и отладить программу на

			кода	языке С++, реализующую выбранный алгоритм
	владеет (высокий)	навыками алгоритмизации и программирования для разработки программных средств	владение навыками составления алгоритмов для реализации информационных технологий; владение навыками разработки и создания программных средств	способность выбрать и обосновать способ представления данных; способность составить алгоритм для решения задачи; способность написать, отладить и протестировать программу; способность сопровождать и модифицировать программный продукт
ПК-25 способность использовать математические методы обработки, анализа и синтеза результатов профессиональных исследований	знает (пороговый уровень)	основные приемы алгоритмизации математических методов	знание основных методов численного моделирования	способность перечислить основные численные методы; способность охарактеризовать область применения методов численного моделирования
	умеет (продвинутый)	реализовать в виде программного кода математические методы обработки данных	умение применять методы математического анализа и моделирования, решать прикладные математические задачи с использованием современных инструментальных средств	способность разрабатывать и реализовывать алгоритмы для решения научно-исследовательских задач; решать прикладные математические задачи с использованием современных инструментальных средств
	владеет (высокий)	навыками использования математических	владение навыками решения	способность применить методы

		методов в профессиональной деятельности	профессиональных задач на основе применения методов математического анализа и численного моделирования	математического анализа и моделирования для решения профессиональных задач; способность моделировать процессы и анализировать модели с использованием информационных технологий
--	--	---	--	---

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Промежуточная аттестация студентов. Промежуточная аттестация студентов по дисциплине «Информатика и основы программирования» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Промежуточная аттестация проводится в виде экзамена в 1-3 семестрах. Экзаменационный билет включает в себя два теоретических вопроса (1 семестр) или теоретический вопрос и практическое задание (2 и 3 семестры).

Список вопросов к экзамену

1. Позиционные системы счисления. Двоичная и шестнадцатеричная системы счисления.
2. Представление числовой, текстовой, графической информации в компьютере.
3. Алгоритмы и их свойства. Линейная алгоритмическая конструкция.
4. Разветвляющиеся алгоритмические конструкции.
5. Циклические алгоритмические конструкции.
6. Основные типы данных языка Си. Переменные.
7. Арифметические операции и выражения в языке Си.

8. Форматированный вывод на экран в стиле Си (функция printf).
9. Форматированный ввод с клавиатуры в стиле Си (функция scanf).
10. Поточковый ввод/вывод в C++.
11. Условный оператор.
12. Операции сравнения. Сложные условия.
13. Оператор выбора switch.
14. Оператор цикла for.
15. Оператор цикла while.
16. Оператор цикла do while.
17. Одномерные массивы.
18. Многомерные массивы.
19. Основные алгоритмы для работы с массивами (суммирование, подсчет элементов, поиск, сортировка).
20. Функции (объявление, определение, вызов).
21. Указатели и операции с ними. Использование указателей в качестве аргументов функции.
22. Ссылки. Использование ссылок в качестве аргументов функции.
23. Передача в функцию одномерного и двумерного массива.
24. Указатели на функцию и их использование в качестве аргументов функции.
25. Динамическое выделение памяти в C++.
26. Рекурсия.
27. Ввод и вывод информации с использованием файлов.
28. Символьные строки в Си.
29. Основные библиотечные функции для работы с символьными строками.
30. Структуры. Передача структур в функцию.
31. Перегрузка операций для структур.
32. Динамические структуры данных. Стек.
33. Динамические структуры данных. Очереди.

34. Динамические структуры данных. Линейный двунаправленный список.
35. Динамические структуры данных. Бинарное дерево.
36. Алгоритм быстрой сортировки массива.
37. Объектно-ориентированный подход к программированию. Основные принципы.
38. Классы и их описание. Состав класса.
39. Конструкторы и деструкторы.
40. Перегрузка операций для классов.
41. Наследование классов.
42. Строковый класс.
43. Контейнерные классы.

Примерные практические задания к экзамену

1. Написать функцию, получающую массив целых положительных чисел и возвращающую сумму всех четных элементов массива. Функция должна вызываться из основной программы.

2. Написать функцию, получающую массив дробных чисел и возвращающую количество элементов массива, которые больше его начального элемента. Функция должна вызываться из основной программы.

3. Программа должна корректно работать с русскими символами.

Описать структуру, содержащую следующую информацию о студенте:

– фамилия;

– пол;

– номер группы (формат номера – одна буква и 4 цифры).

Создать массив структур, заполнить его данными. Используя перегруженную операцию вывода в поток `<<`, вывести массив с информацией на экран.

Запросить с клавиатуры курс, найти и вывести на экран информацию обо всех студентах, обучающихся на данном курсе, учитывая, что курс

показывает вторая из цифр в номере группы. Поиск должен быть оформлен в виде отдельной функции.

4. Программа должна корректно работать с русскими символами.

Описать структуру, содержащую следующую информацию о телевизоре:

– страна-производитель;

– модель (12 символов, в номере модели присутствуют и буквы, и цифры);

– цена.

Создать массив структур, заполнить его данными. Используя перегруженную операцию вывода в поток << , вывести массив с информацией на экран.

Запросить с клавиатуры год производства, найти и вывести на экран информацию обо всех телевизорах этого года производства (год показывают последние 4 символа в номере модели). Поиск должен быть оформлен в виде отдельной функции.

5. Определить класс для работы с геометрическими трехмерными векторами. Перегрузить операции ввода и вывода в поток. Перегрузить операции сложения, вычитания, умножения векторов, умножения вектора на число. Определить методы для нахождения векторного произведения и вычисления длины вектора.

Критерии выставления оценки студенту на экзамене по дисциплине

«Информатика и основы программирования»

Баллы (рейтинговой оценки)	Оценка экзамена (стандартная)	Требования к сформированным компетенциям
86-100	<i>«отлично»</i>	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в

		ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
76-85	<i>«хорошо»</i>	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
61-75	<i>«удовлетворительно»</i>	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
0-60	<i>«неудовлетворительно»</i>	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

Шкала соответствия рейтинга по дисциплине и оценок

1 семестр

№	Наименование контрольного мероприятия	Форма контроля	Весовой коэффициент (%)	Максимальный балл	Минимальное требование для допуска к семестровой аттестации
1	Выполнение индивидуальных заданий	Индивидуальное задание	10	5	0
2	Выполнение индивидуальных заданий	Индивидуальное задание	10	5	0
3	Теоретический материал	Тестирование	15	5	0
4	Выполнение индивидуальных заданий	Индивидуальное задание	10	5	0

5	Теоретический материал	Тестирование	15	5	0
6	Экзамен	Экзамен	40	5	0

2 семестр

№	Наименование контрольного мероприятия	Форма контроля	Весовой коэффициент (%)	Максимальный балл	Минимальное требование для допуска к семестровой аттестации
1	Выполнение лабораторных работ	Лабораторная работа	10	5	0
2	Выполнение лабораторных работ	Лабораторная работа	10	5	0
3	Выполнение лабораторных работ	Лабораторная работа	10	5	0
4	Выполнение лабораторных работ	Лабораторная работа	10	5	0
5	Выполнение лабораторных работ	Лабораторная работа	10	5	0
6	Выполнение лабораторных работ	Лабораторная работа	10	5	0
7	Выполнение лабораторных работ	Лабораторная работа	10	5	0
8	Выполнение лабораторных работ	Лабораторная работа	10	5	0
9	Выполнение лабораторных работ	Лабораторная работа	10	5	0
10	Теоретический материал	Тестирование	10	5	0
11	Экзамен	Экзамен	0	–	–

3 семестр

№	Наименование контрольного мероприятия	Форма контроля	Весовой коэффициент (%)	Максимальный балл	Минимальное требование для допуска к семестровой аттестации
1	Выполнение лабораторных работ	Лабораторная работа	10	5	0
2	Выполнение лабораторных работ	Лабораторная работа	20	5	0
3	Выполнение лабораторных работ	Лабораторная работа	20	5	0
4	Выполнение лабораторных работ	Лабораторная работа	20	5	0
5	Выполнение лабораторных работ	Лабораторная работа	15	5	0
6	Выполнение лабораторных работ	Лабораторная работа	15	5	0

7	Экзамен	Экзамен	0	–	–
---	---------	---------	---	---	---

Текущая аттестация студентов. Текущая аттестация студентов по дисциплине «Информатика и основы программирования» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация по дисциплине «Информатика и основы программирования» проводится в форме контрольных мероприятий (выполнение индивидуальных заданий, выполнение лабораторных работ и тестирование) по оцениванию фактических результатов обучения студентов и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине);
- степень усвоения теоретических знаний (тестирование);
- уровень овладения практическими умениями и навыками по всем видам учебной работы (индивидуальные задания, лабораторные работы);
- результаты самостоятельной работы (индивидуальные задания, лабораторные работы).

Оценочные средства для текущей аттестации

№ п/п	Код ОС	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
1	ПР-1	Тестирование	Средство контроля, позволяющее проверить наличие у студентов сформировавшегося понятийного аппарата и навыков решения простейших практических задач. Поскольку при тестировании от студента требуется выбрать правильный ответ из нескольких вариантов, преимуществом этого метода является простота оценки результатов.	Комплект тестовых заданий

2	ПР-6	Лабораторная работа	Средство для закрепления и практического освоения материала по определенному разделу, теме. Закрепление и освоение происходит не только в ходе выполнения работы, но и при составлении отчета по выполненной работе и его защите.	Комплект лабораторных заданий
3	ПР-13	Индивидуальные задания	Средство проверки умений применять полученные знания по заранее определенной методике для решения задач или заданий по модулю или дисциплине в целом. Показывает степень формирования у студентов практических умений и навыков.	Комплект индивидуальных заданий

Индивидуальные задания

Вариант 1

1. Представление чисел в ЭВМ

1. Представить числа, заданные в одной из систем счисления, в других системах счисления (двоичная, десятичная, шестнадцатеричная):

$$157_{10}, 2780_{10}, 11001001_2, 10010110_2, F2_{16}, 135_{16}$$

2. Записать 16-битовое представление целых чисел в памяти ЭВМ:

а) 285; б) –113

3. Записать 32-битовое представление дробных чисел в памяти ЭВМ:

а) 17; б) –3,25; в) 87,137

2. Вычисления в цикле

1. Составить алгоритм расчета по заданному натуральному значению n суммы $\sum_{i=1}^n \frac{x_i - y_i}{x_i + y_i}$, где $x_1 = 1, y_1 = 1, x_i = y_{i-1} + 1, y_i = 3x_{i-1} + 2y_{i-1}$.

2. Составить алгоритм вычисления суммы ряда для заданного значения x с заданной точностью 10^{-4} . Особое внимание обратить на эффективность алгоритма.

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + K, \quad \text{для } x \in R$$

3. Поиск и сортировка в массивах

1. Составить алгоритм для упорядочения заданной последовательности целых положительных чисел следующим образом: сначала должны располагаться все нечётные числа в порядке убывания, а затем все чётные в порядке возрастания. Например: из последовательности {2, 5, 3, 4, 7, 16, 1} должно получиться {7, 5, 3, 1, 2, 4, 16}. Необходимо учесть, что: 1) последовательность может не содержать нечётных чисел; 2) последовательность может не содержать чётных чисел.

2. Составить алгоритм, который для двумерного массива целых чисел размера $m \times n$ находит значение величины $z = \min + \max$, где \min – наименьший элемент среди всех столбцов, содержащих больше положительных чисел, чем отрицательных, \max – наибольший элемент среди всех строк, содержащих хотя бы один ноль. Предусмотреть случаи: 1) отсутствия в массиве столбцов, содержащих больше положительных чисел, чем отрицательных; 2) отсутствия в массиве строк, содержащих хотя бы один ноль.

Вариант 2

1. Представление чисел в ЭВМ

1. Представить числа, заданные в одной из систем счисления, в других системах счисления (двоичная, десятичная, шестнадцатеричная):

213_{10} , 3212_{10} , 10011011_2 , 11010010_2 , $C9_{16}$, 162_{16}

2. Записать 16-битовое представление целых чисел в памяти ЭВМ:

а) 299; б) –121

3. Записать 32-битовое представление дробных чисел в памяти ЭВМ:

а) 21; б) 5,75; в) –34,247

2. Вычисления в цикле

1. Составить алгоритм расчета по заданному натуральному значению n суммы $\sum_{i=1}^n \frac{d_i}{c_i^2 - d_i^2}$, где $c_1 = 2$, $d_1 = 1$, $c_i = 3c_{i-1} - d_{i-1}$, $d_i = c_{i-1} + 2$.

2. Составить алгоритм вычисления суммы ряда для заданного значения x с заданной точностью 10^{-4} . Особое внимание обратить на эффективность алгоритма.

$$\ln(1+x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{n+1}}{n+1} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + K, \text{ для } -1 < x \leq 1$$

3. Поиск и сортировка в массивах

1. Составить алгоритм для упорядочения заданной последовательности ненулевых чисел следующим образом: сначала должны располагаться все отрицательные числа в порядке убывания, а затем все положительные в порядке убывания. Например: из последовательности $\{2, 5, -3, 4, 7, -7, -1\}$ должно получиться $\{-1, -3, -7, 7, 5, 4, 2\}$. Необходимо учесть, что: 1) последовательность может не содержать отрицательных чисел; 2) последовательность может не содержать положительных чисел.

2. Составить алгоритм, который для двумерного массива целых положительных чисел размера $m \times n$ находит значение величины $z = \min + \max$, где \min – наименьший элемент среди всех столбцов, содержащих хотя бы одно чётное число, \max – наибольший элемент среди всех строк, содержащих больше нечётных чисел, чем чётных. Предусмотреть случаи: 1) отсутствия в массиве столбцов, содержащих хотя бы одно чётное число; 2) отсутствия в массиве строк, содержащих больше нечётных чисел, чем чётных.

Вариант 3

1. Представление чисел в ЭВМ

1. Представить числа, заданные в одной из систем счисления, в других системах счисления (двоичная, десятичная, шестнадцатеричная):

$$139_{10}, 2534_{10}, 11101100_2, 10110101_2, 5E_{16}, 183_{16}$$

2. Записать 16-битовое представление целых чисел в памяти ЭВМ:

а) 422; б) -270

3. Записать 32-битовое представление дробных чисел в памяти ЭВМ:

а) 31; б) -6,125; в) 54,278

2. Вычисления в цикле

1. Составить алгоритм расчета по заданному натуральному значению n суммы $\sum_{i=1}^n \frac{p_i}{p_i + q_i}$, где $p_1 = 0, q_1 = 1, p_i = p_{i-1} + 2q_{i-1}, q_i = q_{i-1} + 2p_{i-1}$

2. Составить алгоритм вычисления суммы ряда для заданного значения x с заданной точностью 10^{-4} . Особое внимание обратить на эффективность алгоритма.

$$\operatorname{arcth} x = \sum_{n=0}^{\infty} \frac{1}{(2n+1)x^{2n+1}} = \frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + K, \text{ для } |x| > 1$$

3. Поиск и сортировка в массивах

1. Составить алгоритм для упорядочения заданной последовательности целых положительных чисел следующим образом: сначала должны располагаться все чётные числа в порядке убывания, а затем все нечётные в порядке убывания. Например: из последовательности $\{2, 5, 3, 4, 7, 16, 1\}$ должно получиться $\{16, 4, 2, 7, 5, 3, 1\}$. Необходимо учесть, что: 1) последовательность может не содержать нечётных чисел; 2) последовательность может не содержать чётных чисел.

2. Составить алгоритм, который для двумерного массива целых чисел размера $m \times n$ находит значение величины $z = \min + \max$, где \min – наименьший элемент среди всех строк, не содержащих отрицательных чисел, \max – наибольший элемент среди всех столбцов, содержащих больше положительных чисел, чем отрицательных. Предусмотреть случаи: 1) отсутствия в массиве строк, не содержащих отрицательных чисел; 2) отсутствия в массиве столбцов, содержащих больше положительных чисел, чем отрицательных.

Вариант 4

1. Представление чисел в ЭВМ

1. Представить числа, заданные в одной из систем счисления, в других системах счисления (двоичная, десятичная, шестнадцатеричная):

$$193_{10}, 3120_{10}, 11010011_2, 10100110_2, 5B_{16}, 172_{16}$$

2. Записать 16-битовое представление целых чисел в памяти ЭВМ:

а) 312; б) –196

3. Записать 32-битовое представление дробных чисел в памяти ЭВМ:

а) 21; б) 4,375; в) –71,923

2. Вычисления в цикле

1. Составить алгоритм расчета по заданному натуральному значению n суммы $\sum_{i=1}^n \frac{1}{a_i - b_i}$, где $a_1 = 1$, $b_1 = 3$, $a_i = a_{i-1} + b_{i-1}$, $b_i = a_{i-1} - 2b_{i-1}$.

2. Составить алгоритм вычисления суммы ряда для заданного значения x с заданной точностью 10^{-4} . Особое внимание обратить на эффективность алгоритма.

$$\operatorname{sh} x = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + K, \quad \text{для } x \in R$$

3. Поиск и сортировка в массивах

1. Составить алгоритм для упорядочения заданной последовательности ненулевых чисел следующим образом: сначала должны располагаться все положительные числа в порядке убывания, а затем все отрицательные в порядке возрастания. Например: из последовательности $\{-1, 5, -3, 4, 7, -7, 2\}$ должно получиться $\{7, 5, 4, 2, -7, -3, -1\}$. Необходимо учесть, что: 1) последовательность может не содержать отрицательных чисел; 2) последовательность может не содержать положительных чисел.

2. Составить алгоритм, который для двумерного массива целых положительных чисел размера $m \times n$ находит значение величины $z = \min + \max$, где \min – наименьший элемент среди всех столбцов, содержащих только нечётные числа, \max – наибольший элемент среди всех строк, содержащих хотя бы одно чётное число. Предусмотреть случаи: 1) отсутствия в массиве столбцов, содержащих только нечётные числа; 2) отсутствия в массиве строк, содержащих хотя бы одно чётное число.

Критерии оценки индивидуальных заданий

5 баллов – расчетное задание выполнено полностью верно или составленный алгоритм является эффективным и дает верный результат при любых исходных данных, студент способен ответить на все вопросы по заданию;

4 балла – расчетное задание выполнено с 1-2 ошибками или составленный алгоритм является достаточно эффективным и дает верный результат при любых исходных данных, студент не допускает существенных неточностей при ответе на вопросы по заданию;

3 балла – расчетное задание выполнено с 3-4 ошибками или составленный алгоритм недостаточно эффективный, но дает верный результат за исключением особых случаев исходных данных, студент способен в целом пояснить выполненное задание с отдельными неточностями в деталях;

0 баллов (задание не зачтено) – расчетное задание содержит большое количество ошибок или составленный алгоритм не дает верного результата или крайне неэффективный, студент не способен дать пояснения по заданию.

Задания для лабораторных работ

Лабораторная работа № 1. Ввод/вывод информации

Обе программы должны быть реализованы двумя способами:
а) используя функции ввода/вывода в стиле Си; б) потоки в стиле С++.

1. Написать программу, запрашивающую дробное значение x с клавиатуры, рассчитывающую значение выражения $y = 3x^2 - 5x + 2$ и выводящую на экран значения x и y .

2. Написать программу, которая запрашивает два целых числа a и b с клавиатуры, рассчитывает значения выражений $c = -2ab + 5$ и $d = (a - 1)(3b + 1)c^2$ и выводит на экран числа, как введенные с клавиатуры, так и рассчитанные.

Лабораторная работа № 2. Условный оператор

1. Написать программу, которая приглашает ввести цифру 2 и выдает на экран надпись «правильно», если введено число 2, или «неправильно» в противном случае.

2. Написать программу, которая приглашает ввести два числа, сравнивает их и выдает сообщение: «Первое число больше», «Первое число

меньше» или «Числа равны». Программа должна использовать вложенный условный оператор.

3. Написать программу, запрашивающую число x и рассчитывающую значение следующего выражения: $y = \frac{3x^2 - 5x + 1}{(2x + 7)(x - 2)}$.

4. Написать программу, которая запрашивает значение x и проверяет его принадлежность области $x \in [-5; 2] \cup [3; 4] \cup (7; +\infty)$.

Лабораторная работа № 3. Циклы

1. Написать программу, рассчитывающую и выводящую на экран значения функции $y = \frac{(5x - 3)(x + 5)}{2x^2 - x - 1}$ на отрезке $x \in [a; b]$ с шагом h . Границы отрезка и шаг должны запрашиваться с клавиатуры.

2. Написать программу, находящую сумму всех целых нечетных чисел от 5 до 25.

3. Написать программу, находящую значение произведения $\frac{1}{2} \cdot \frac{4}{5} \cdot \frac{7}{8} \cdot \frac{16}{17}$.

4. Написать программу, которая в цикле запрашивает с клавиатуры значение x в градусах, рассчитывает функцию $y = \sin x$ и выдает ответ на экран. При вводе с клавиатуры значения 45° программа должна сперва выдать ответ, а затем завершить работу.

Лабораторная работа № 4. Массивы

1. Написать программу, находящую сумму элементов массива, которые больше 5. В качестве примера взять данные: 3, 2.4, -1.5, 5, 3.1, 7.2, -6.1, 10.2

2. Написать программу, находящую количество элементов массива, которые меньше среднего значения его элементов. В качестве примера взять данные из задания 1.

3. Написать программу, которая для массива целочисленных значений находит **номер** его первого положительного элемента.

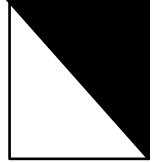
Примечание. Обратить внимание на случай отсутствия положительных чисел в массиве!

4. Написать программу, находящую два наименьших элемента массива.

Примечание. Обратит внимание на случай, когда эти элементы имеют одинаковые значения!

Лабораторная работа № 5. Двумерные массивы

1. Написать программу, находящую в квадратной матрице порядка n наибольший из элементов, расположенных в закрашенной части, включая диагональ (см. рисунок).



2. Написать программу, заполняющую квадратную матрицу порядка n числами 0 и 1 по образцу.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

3. Написать программу, которая определяет, сколько строк двумерного массива размера $m \times n$ содержат хотя бы одно положительное число.

Лабораторная работа № 6. Функции

1. Написать программу, которая запрашивает катеты прямоугольного треугольника и находит его гипотенузу. Расчет гипотенузы должен быть реализован в программе в виде отдельной функции.

2. Написать функцию, рассчитывающую объем и площадь поверхности шара по заданному радиусу. Расчет обеих величин должен производиться в **одной** функции.

3. Написать функцию, получающую массив целых чисел и возвращающую максимальный элемент массива.

4. Написать функцию, получающую двумерный массив чисел и возвращающую сумму положительных элементов в четных столбцах.

Лабораторная работа № 7. Обработка символьной и строковой информации

1. Написать программу, запрашивающую ввод одного символа с клавиатуры и определяющую, является ли введенный символ цифрой.

2. С клавиатуры вводится русский текст длиной не более 80 символов. Написать программу, подсчитывающую количество букв 'ц' в тексте.

3. Написать программу, запрашивающую с клавиатуры английское слово и преобразующую его так, чтобы заглавные и строчные буквы в слове чередовались, причем первой должна быть заглавная.

Примечание. Программа должна работать правильно независимо от того, в каком регистре вводились буквы.

4. Библиотечная функция `strcat` имеет прототип (объявление):
`char* strcat(char *a, char *b);`

Функция добавляет строку `b` в конец строки `a`. Нуль-символ в строке `a` заменяется первым символом строки `b`, и новый нуль-символ добавляется уже в конец объединенной строки. Функция возвращает указатель на получившуюся строку, т.е. на строку `a`.

Написать свою функцию, работающую аналогично библиотечной. Программа должна демонстрировать работоспособность написанной функции.

Лабораторная работа № 8. Структуры. Перегрузка операций

Для работы с комплексными числами определить:

- 1) структуру, полями которой являются действительная и мнимая части числа;
- 2) функцию ввода комплексного числа (в любом виде);
- 3) функцию вывода комплексного числа в общепринятой записи с использованием мнимой единицы;
- 4) функцию для нахождения модуля комплексного числа;
- 5) перегруженную операцию сложения, включая:
 - сложение двух комплексных чисел;
 - сложение комплексного числа с действительным и наоборот, учитывая, что любое действительное число можно представить как комплексное с нулевой мнимой частью;
 - операцию `+=` для комплексного числа с комплексным и комплексного с действительным;

б) перегруженную операцию сравнения двух комплексных чисел на равенство $==$.

Продемонстрировать в программе использование вышеуказанных функций и операций.

Лабораторная работа № 9. Массивы структур

Описать структуру, содержащую следующую информацию о студенте: фамилия, имя, отчество; номер группы; год рождения.

Программа должна включать следующие функции:

- вывод структуры с информацией о студенте на экран (в любой удобной форме);

- сортировка массива структур в алфавитном порядке по фамилиям студентов (подсказка: для сравнения строк при сортировке удобно использовать функцию `strcmp`);

- поиск в массиве структур по заданному номеру группы; функция должна возвращать индекс найденного элемента массива либо указатель на него; должно быть предусмотрено особое возвращаемое значение, если искомый элемент не найден; **функция не должна выводить сообщений на экран.**

Функция `main` должна:

- считать из файла данные о студентах в массив структур (тестовый файл должен содержать информацию не менее чем о 10 студентах), контроль ошибок при открытии файла и чтении данных обязателен;

- вывести их на экран, используя написанную функцию;

- отсортировать массив, используя написанную функцию, и вывести его на экран;

- запросить с клавиатуры номер группы для поиска; используя написанную функцию поиска, найти и вывести на экран информацию о ВСЕХ студентах данной группы (подсказка: так как функция поиска за один вызов находит только одного студента, её вызов должен быть в цикле,

причем при каждом новом вызове поиск должен начинаться с того места, где закончился предыдущий).

Лабораторная работа № 10. Реализация стека

Написать программу, реализующую стек целых чисел. Необходимо определить следующие функции для работы со стеком:

- создание стека;
- добавление элемента в стек;
- удаление элемента из стека;
- вывод стека на экран;
- очистка стека.

Используя написанные функции, сформировать стек, а затем удалить из него минимальный элемент.

Лабораторная работа № 11. Реализация очереди

Описать структуру, содержащую следующую информацию о пациенте: фамилия, имя, отчество; возраст; диагноз.

Написать программу, которая организует очередь пациентов. Диалог осуществляется с помощью меню, позволяющего выполнить следующие действия:

- чтение данных из файла (запрашивается имя файла, содержащего информацию о пациентах; данные считываются из файла и добавляются в очередь);

- вывод данных в файл (запрашивается имя файла, в который выводится информация обо всех пациентах в порядке очереди, сами пациенты при этом из очереди не извлекаются);

- добавление в очередь (запрашивается ввод данных о пациенте с добавлением его в конец очереди);

- извлечение из очереди (пациент извлекается из начала очереди, на экран выводится информация о нем);

- просмотр очереди (на экран выводится информация обо всех пациентах в порядке очереди, сами пациенты при этом из очереди не извлекаются);

- поиск по месту в очереди (запрашивается номер в очереди, выводится информация о пациенте на данном месте);
- поиск по диагнозу (запрашивается диагноз, выводится информация обо всех пациентах с этим диагнозом с указанием места в очереди);
- очистка очереди (здесь главное освободить память, выделенную для элементов очереди; выводить информацию о пациентах не нужно);
- завершение работы (не забудьте при этом выполнить очистку очереди, если нужно).

Нужно обращать внимание, что при любом выполняемом действии очередь может как уже существовать, так и еще нет!

Лабораторная работа № 12. Реализация линейного списка

Картотека в бюро обмена квартир организована как линейный список. Сведения о каждой квартире содержат:

- количество комнат;
- этаж;
- площадь;
- адрес.

Составить программу, которая обеспечивает:

- начальное формирование картотеки;
- ввод заявки на обмен;
- поиск в картотеке подходящего варианта: при равенстве количества комнат и этажа и различии площадей в пределах 10% выводится соответствующая карточка и удаляется из списка, в противном случае поступившая заявка включается в список;
- вывод всего списка.

Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

Лабораторная работа № 13. Описание класса

1. Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

2. Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

3. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность отдельного изменения составных частей адреса, создания и уничтожения объектов этого класса.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

Лабораторная работа № 14. Использование строкового класса

1. Имеется текстовая строка. Необходимо удалить из строки начальные и завершающие пробелы, а также заменить все последовательности из нескольких пробелов одним пробелом.

2. Имеется текстовая строка. Необходимо найти самое длинное слово в строке. Слова в строке отделяются друг от друга пробелами (возможно несколькими подряд) или знаками препинания (возможно с предшествующими или последующими пробелами).

3. Имеется текстовая строка. Необходимо преобразовать её так, чтобы слова в строке располагались в обратном порядке; порядок букв в словах должен остаться неизменным; знаки препинания и пробелы должны находиться между теми же словами, что и в исходной строке.

4. Имеется текстовая строка, состоящая из нескольких слов, и две заданные подстроки из одного слова каждая. Необходимо заменить в тексте

все слова, совпадающие с первой подстрокой, на вторую подстроку. Замена должна производиться только при совпадении целого слова, а не его части. Обратите внимание, что длины подстрок могут не совпадать.

Лабораторная работа № 15. Использование контейнерных классов

1. Выполнить задание лабораторной работы № 9, используя контейнерный класс.

2. Выполнить задание лабораторной работы № 10, используя контейнерный класс.

3. Выполнить задание лабораторной работы № 11, используя контейнерный класс.

4. Выполнить задание лабораторной работы № 12, используя контейнерный класс.

5. Создать картотеку книг. Предусмотреть возможность работы с произвольным числом книг, поиска книги по различным признакам (например, по автору или по году издания), добавления книг в картотеку, удаления книг из нее, сортировки книг по разным полям. Написать программу, демонстрирующую работу с картотекой. Программа должна содержать меню, позволяющее осуществить проверку всех методов.

Критерии оценки лабораторных работ

5 баллов – написанная программа дает верный результат при любых исходных данных, алгоритм составлен и реализован эффективно, без замечаний, отчет по работе содержит полную информацию, студент способен ответить на все вопросы по составленному алгоритму и его реализации;

4 балла – написанная программа дает верный результат при любых исходных данных, алгоритм составлен и реализован достаточно эффективно, с незначительными замечаниями, к отчету по работе нет принципиальных замечаний, студент не допускает существенных неточностей при ответе на вопросы по алгоритму и его реализации;

3 балла – написанная программа дает верный результат за исключением особых случаев исходных данных, составленный алгоритм и/или его реализация недостаточно эффективны, отчет содержит минимальную информацию, студент способен в целом пояснить алгоритм и его реализацию с отдельными неточностями в деталях;

0 баллов (задание не зачтено) – программа не дает верного результата, алгоритм (реализация алгоритма) составлен неверно или крайне неэффективно, отчет по работе отсутствует или не содержит необходимой информации, студент не способен дать пояснения по алгоритму и его реализации.

Тестовые задания

1. Какая из следующих записей - правильный комментарий в C++?

- а) /*комментарий */
- б) {комментарий}
- в) /* Комментарий */
- г) ** Комментарий **

2. Укажите правильную форму записи цикла do while

- а)

```
do {
    /*блок операторов*/;
} while (/*условие выполнения цикла*/);
```
- б)

```
do {
    /*блок операторов*/;
} while (/*условие выполнения цикла*/)
```
- в)

```
do {
    /*блок операторов*/;
} while {/*условие выполнения цикла*/}
```

3. Какими знаками заканчивается большинство строк кода в Си++?

- а) : (двоеточие)
- б) ; (точка с запятой)
- в) , (запятая)
- г) . (точка)

4. Программа, переводящая входную программу на исходном языке в исполняемую программу, называется:

- а) компилятор
- б) интерпретатор
- в) редактор
- г) сканер

**5. Чему будет равна переменная a, после выполнения этого кода:
int a; for(a = 0; a < 10; a++) { } ?**

- а) 9
- б) 10
- в) 1
- г) 0

6. Какие служебные символы используются для обозначения начала и конца блока кода?

- а) { }
- б) < >
- в) begin end
- г) ()

7. Какой служебный знак ставится после оператора case ?

- а) ;
- б) .
- в) :
- г) -

8. Чтобы подключить заголовочный файл в программу на C++, например iostream необходимо написать:

- а) #include (iostream)
- б) #include <iostream.h >
- в) include #iostream.h;
- г) #include <iostream>

9. До каких пор будут выполняться операторы в теле цикла while (x < 100)?

- а) пока x равен 100
- б) пока x меньше или равен 100
- в) пока x больше 100
- г) пока x строго меньше 100

10. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?

- а) contingency
- б) default

- в) all
- г) other

11. Простые типы данных в C++.

- а) целые – int, вещественные – float или double, символьные – char
- б) целые – int, вещественные – float или double, символьные – string
- в) целые – bool, вещественные – float или double, символьные – string
- г) целые – int, вещественные – float или real, символьные – char

12. Каков результат работы следующего фрагмента кода?

```
int x = 0;
switch(x) {
    case 1: cout << "Один";
    case 0: cout << "Ноль";
    case 2: cout << "Привет мир";
}
```

- а) Один
- б) НольПривет мир
- в) Привет мир
- г) Ноль

13. Укажите правильное определение функции main в соответствии со спецификацией стандарта ANSI

- а) int main(void)
- б) void main(void)
- в) int main()
- г) void main()

14. Какой из следующих операторов - оператор сравнения двух переменных?

- а) =
- б) equal
- в) :=
- г) ==

15. Какой из перечисленных типов данных не является типом данных в C++?

- а) real
- б) float
- в) double
- г) int

16. Какой из циклов является циклом с постусловием?

- а) while
- б) for
- в) do while

17. Какое значение, по умолчанию, возвращает программа операционной системе в случае успешного завершения?

- а) 0
- б) -1
- в) 1
- г) программа не возвращает значение

18. Язык программирования C++ разработал

- а) Никлаус Вирт
- б) Бьерн Страуструп
- в) Кен Томпсон
- г) Дональд Кнут

19. Какую функцию должны содержать все программы на C++?

- а) program()
- б) system()
- в) start()
- г) main()

20. Какой цикл называется циклом с предусловием?

- а) while
- б) do while
- в) for

21. Какой из ниже перечисленных операторов, не является циклом в C++?

- а) while
- б) for
- в) repeat until
- г) do while

22. Что будет напечатано в результате выполнения программы?

```
#include<iostream>
int main() {
    for (int i = 0; i < 4; ++i) {
        switch (i) {
            case 0 : std::cout << "0";
            case 1 : std::cout << "1"; continue;
            case 2 : std::cout << "2"; break;
            default : std::cout << "D"; break;
        }
        std::cout << ".";
    }
    return 0;
}
```

- a) 0112.D.
- б) 01.2.D.
- в) 011.2.D
- г) 0.1.2.

23. Тело любого цикла выполняется до тех пор, пока его условие ...

- a) истинно
- б) у цикла нет условия
- в) ложно
- г) зависит от вида цикла

24. Выберите правильный вариант объявления константной переменной в C++, где type - тип данных в C++ variable - имя переменной value - константное значение

- a) `const type variable = value;`
- б) `const type variable := value;`
- в) `const variable = value;`

25. Каков будет результат выражения `!(1 && !(0 || 1))`?

- a) неоднозначность
- б) true
- в) false

**26. Результат выполнения следующего фрагмента кода:
`!((1 || 0) && 0)`**

- a) результат не может быть заранее определен
- б) 1
- в) 0

**27. Что будет напечатано, после выполнения этого кода:
`cout << (5 << 3); ?`**

- a) 53
- б) 35
- в) 40
- г) 0

28. Укажите неправильно записанную операцию отношения

- a) `>=`
- б) `=!`
- в) `==`
- г) `<=`

29. Какое из следующих значений эквивалентно зарезервированному слову true?

- a) -1
- б) 2

- в) 1
- г) 0

30. При выводе в поток cout можно печатать несколько значений или переменных в одной команде, используя следующий синтаксис:

- а) `cout << "Привет", name, "n";`
- б) `cout << "Привет" + name + "n";`
- в) `cout << "Привет" << name << "n";`
- г) `cout << ("Привет" & name & "n");`

31. В каком случае нужно обязательно использовать фигурные скобки в условном операторе if?

- а) если в теле оператора if нет ни одного оператора
- б) если в теле оператора if всего один оператор
- в) если в теле оператора if два и более операторов
- г) нет правильного ответа

32. Какой из ниже перечисленных вариантов ответа, показывает правильно записанный условный оператор if ?

- а) `if` условное выражение оператор;
- б) `if` условное выражение { оператор; }
- в) `if` { условное выражение } оператор;
- г) `if` (условное выражение) оператор;

**33. Чему равен результат выполнения следующего выражения:
1000 / 100 % 7 * 2 ?**

- а) 1000
- б) 250
- в) 6
- г) 10

**34. Результат выполнения следующего фрагмента кода:
cout << 22 / 5 * 3;**

- а) 1.47
- б) 1
- в) 13.2
- г) 12

35. Логическая операция с более высоким приоритетом выполнения

- а) !
- б) ||
- в) &&

36. Какое значение будет напечатано в результате выполнения программы?


```
#include<iostream>
int main() {
    int x = 0;
    int y = 0;
    if (x++ && y++) y += 2;
    std::cout << x + y << std::endl;
    return 0;
}
```

- a) 0
- б) 1
- в) 2
- г) 4

37. Тело условного оператора if, будет выполняться, если его условие:

- a) ложно (false)
- б) истинно (true)
- в) зависит от ситуации

38. Укажите операцию, приоритет выполнения которой больше остальных

- a) ()
- б) >
- в) +
- г) /

39. Укажите операцию, приоритет выполнения которой ниже остальных

- a) >>
- б) &&
- в) &
- г) ||

40. Какое ключевое слово указывает, что целая переменная не может принимать отрицательные значения?

- a) long
- б) нет такого зарезервированного слова
- в) unsigned
- г) positive

41. Оператор if else позволяет определить действие ...

- a) только для случая ложного условия
- б) для случаев истинного и ложного условий
- в) только для случая истинного условия

42. Какая из следующих логических операций – логическое И?

- а) ||
- б) |
- в) &
- г) &&

43. Что появится на экране, после выполнения этого фрагмента кода?

```
int a = 1, b = 2;  
if (a == b);  
cout << a << " = " << b << endl;
```

- а) a = b
- б) 1 = 2
- в) вывод на экран не выполнится
- г) синтаксическая ошибка

44. Значение 5.9875e17 может быть сохранено в переменной, типа:

- а) bool
- б) long
- в) int
- г) float

45. Какие преобразования типов данных невозможны без потери данных?

- а) float в int
- б) char в float
- в) int в float
- г) float в double

46. Какой заголовочный файл необходимо подключить, чтобы вызвать функцию isalpha()?

- а) conio.h
- б) ctype
- в) cstring
- г) ifstream.h

47. В какой из следующих записей используется операция разыменования?

- а) address(a);
- б) *a;
- в) a;
- г) &a;

48. Какая из следующих функций добавляет одну строку в конец другой?

- а) stradd();
- б) stringadd();

- в) append ();
- г) strcat ();

49. В каком из вариантов ответов объявлен двумерный массив?

- а) array anarray[20][20];
- б) char array[20];
- в) int anarray[20][20];
- г) int array[20, 20];

50. В какой из следующих записей используется операция взятия адреса?

- а) address(a);
- б) *a;
- в) a;
- г) &a;

51. Укажите правильное объявление массива:

- а) int anarray[10];
- б) int anarray{10};
- в) int anarray;
- г) array anarray[10];

52. Укажите правильное объявление указателя в C++

- а) int &x;
- б) int *x;
- в) ptr x;
- г) int x;

53. Какая из следующих записей возвращает значение переменной a, хранящееся в памяти по адресу, на который указывает указатель?

- а) val(a);
- б) *a;
- в) a ;
- г) &a;

54. Какая из следующих функций сравнивает две строки?

- а) compare();
- б) stringcompare();
- в) cmp();
- г) strcmp();

55. Укажите зарезервированное ключевое слово для динамического выделения памяти:

- а) create
- б) new

- в) malloc
- г) value

56. Какое значение будет напечатано, в результате выполнения следующего кода?

```
int sum = 0;
int array[ ][3] = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8}};
for (int i = 0; i < 3 ; ++i)
    for (int j = 2; j < 3 ; j++) sum += array[i][j];
cout << sum;
```

- а) 9
- б) синтаксическая ошибка
- в) 36
- г) 15

57. Укажите статическую строку:

- а) char string[100];
- б) 'Статическая строка'
- в) "Статическая строка"

58. Что такое ссылка?

- а) нет правильного ответа
- б) ссылка является псевдонимом для объекта
- в) оператор
- г) используется для переименования объектов

59. Доступ к отдельным элементам массива осуществляется по:

- а) имени массива и адресу
- б) имени массива и индексу
- в) по адресу и индексу

60. Укажите строку, которая возвращает адрес первого элемента в массиве arr?

- а) &arr;
- б) arr[1];
- в) arr[0];
- г) arr;

61. Объявлена переменная char a; Какое из следующих выражений не верно?

- а) a = "3";
- б) a = '3';
- в) a = 3;

62. Словосочетание "Hello world!" может быть сохранено в символьном массиве размером n элементов. Укажите чему равно n?

- а) 13
- б) 11
- в) 14
- г) 12

63. Память выделяется корректно в строке:

- а) `int a = new int(20);`
- б) `int *a = new int[20];`
- в) `int *a = new sizeof(int * 20);`
- г) `int a = new int[20];`

64. Как правильно высвободить память, выделенную как `char *a = new char[20];`

- а) `delete [] a;`
- б) `delete a;`
- в) `delete a[];`

65. Каков порядковый номер последнего элемента массива, если размер массива 19?

- а) 19
- б) порядковый номер определяется программистом
- в) 18
- г) 20

66. Укажите зарезервированное ключевое слово для высвобождения выделенной памяти!

- а) `clear`
- б) `free`
- в) `delete`
- г) `remove`

67. В какой из следующих строк выполняется обращение к седьмому элементу массива?

- а) `mas(7)`
- б) `mas[8]`
- в) `mas[7]`
- г) `mas[6]`

68. В каком из следующих вариантов ответов выполнен корректный доступ к полю структуры через указатель?

- а) `b*var;`
- б) `b.var;`
- в) `b>var;`
- г) `b->var;`

69. Правильное объявление переменной типа структуры foo:

- a) int foo;
- б) foo;
- в) struct foo;
- г) foo var;

70. При определении структуры необходимо использовать следующее ключевое слово

- a) record
- б) object
- в) struct
- г) structure

71. Какой из следующих классов обрабатывает процесс записи в файл?

- a) ifstream
- б) ofstream
- в) input_file
- г) cout

72. Укажите правильный доступ к полю структуры:

- a) b>var;
- б) b.var;
- в) b*var;
- г) b-var;

73. Правильное определение структуры в C++:

- a) struct a_struct int a;
- б) struct {int a;}
- в) struct a_struct {int a;}
- г) struct a_struct {int a;};

74. Что значит ключевое слово inline?

- a) все вызовы функции заменяются кодом этой функции
- б) сообщает компилятору использовать функцию только в пределах одного файла с исходным кодом
- в) все вызовы функций заменяются библиотечными функциями
- г) нет правильного ответа

75. Что будет напечатано на экране, после выполнения этого кода?

```
#include <iostream>
int foo(int x);
int foo(int y) { return y+1; }
int main( ) {
    int x = 3;
    int y = 6;
    cout << foo(x);
```

- ```
 return 0;
 }
```
- а) 4
  - б) 9
  - в) 7
  - г) ошибка компиляции

**76. Укажите правильный вызов функции (предполагается, что функция была объявлена ранее):**

- а) `funct;`
- б) `int funct();`
- в) `funct();`
- г) `funct x, y;`

**77. Какие из следующих утверждений верны?**

- а) встроенные функции не должны содержать более 10 строк кода.
- б) встроенные функции должны возвращать значение.
- в) встроенные функции не могут возвращать значения.
- г) компилятор может проигнорировать объявление встроенной функции.

**78. Какая строка содержит зарезервированные слова языка программирования C++?**

- а) `defaulted, goto, return, extern, private, public, protected`
- б) `if, else, for, while do, switch, continue, break`
- в) `char, int, float, doubled, short, long, unsigned, signed`
- г) `sizeof, const, typedef, static, voided, enum, struct, union`

**79. Для чего используются встроенные функции?**

- а) для увеличения скорости работы программы
- б) чтобы уменьшить размер программы
- в) для упрощения файла с исходным кодом
- г) для удаления ненужных функций

**80. Какая из следующих функций встроенная?**

- а) `void foo() inline {}`
- б) `inline void foo() {}`
- в) `inline:void foo() {}`
- г) нет правильного ответа

**81. В каком порядке параметры `argc` и `argv` объявлены в функции `main`?**

- а) массив аргументов, затем количество элементов
- б) количество аргументов, затем массив аргументов
- в) они не объявлены в `main`
- г) существует только один параметр

**82. Какая из переменных хранит количество аргументов, передаваемых в программу?**

- а) argc
- б) count
- в) arglen
- г) argv

**83. Укажите тип возвращаемого значения следующей функции**  
`int func(char x, float v, double t);`

- а) float
- б) double
- в) char
- г) int

**84. Что из нижеперечисленного не является прототипом функции?**

- а) `char x();`
- б) `double func(char x)`
- в) `void func();`
- г) `int func(char x, char y);`

**85. Выберите правильный вариант выделения динамической памяти под переменную X типа float:**

- а) `float *ptr = new float; X = *ptr;`
- б) `float & ptr = new float; X = & ptr;`
- в) `float * ptr = &X; X = new float;`

**86. Что такое массив?**

а) именованный набор переменных, имеющих различные типы данных и располагающихся в памяти последовательно

б) именованный набор переменных и функций, которые располагаются в одной области памяти

в) именованный набор переменных, имеющих один тип данных и располагающихся в памяти последовательно

г) именованный набор переменных имеющих символьный тип данных, и располагающихся в одной области памяти

**87. Как написать следующее выражение на языке С «Переменной a присвоено значение b»?**

- а) `a==b`
- б) `a=b`
- в) `b=a`
- г) `a:=b`

**88. Как написать следующее выражение «Второму элементу массива Myarray присвоено значение пять»?**



- а) `int [1] Myarray=«пять»`
- б) `int Myarray [1] = 5`
- в) `int Myarray [2] = «пять»`
- г) `int Myarray [2] = 5`

**89. Какой размер в байтах имеет переменная вещественного типа `float`**

- а) 2
- б) 4
- в) 8
- г) 10

**90. Дан массив `int L[3][3] = { { 2, 3, 4 }, { 3, 4, 8 }, { 1, 0, 9 } }`; Чему будет равно значение элемента этого массива `L[1][2]`**

- а) 2
- б) 3
- в) 4
- г) 8

**91. Объявление `char *buf`; соответствует**

- а) созданию символьной переменной `buf`
- б) созданию строковой переменной `buf`
- в) созданию указателя `buf` на символьное значение
- г) созданию указателя `buf` на строку

**92. Что называется прототипом функции?**

- а) описание функции, включая ее имя, имена и типы параметров
- б) описание функции, включая ее имя, тип возвращаемого значения, типы параметров
- в) имя функции и тип возвращаемого значения
- г) описание функции, включая ее имя, тип возвращаемого значения, имена и типы параметров, тело функции

**93. Какой размер массива `M` будет после выполнения кода: `char M[] ="Goodlive" ?`**

- а) 10
- б) 9
- в) 8
- г) Не определен

**94. В каком случае необходимо использовать оператор `return` в теле функции?**

- а) всегда
- б) если необходимо, чтобы функция вернула значение
- в) если необходимо обеспечить выход из функции в произвольном месте

г) если указан тип возвращаемого значения, в том числе и void

**95. Каким способом можно задать многострочный комментарий в языке C++**

- а) /\*комментарии к программе\*/
- б) //комментарии к программе//
- в) //комментарии к программе
- г) {комментарии к программе}

**86. Логическое выражение возвращает результат типа**

- а) int
- б) bool
- в) char
- г) logical

**97. Выберите правильный вариант записи на языке Си двойного неравенства  $0 < x < 10$**

- а)  $0 < x < 10$
- б)  $x > 0, x < 10$
- в)  $x > 0 \text{ AND } x < 10$
- г)  $x > 0 \ \&\& \ x < 10$

**98. Тело какого цикла всегда будет выполнено хотя бы один раз, независимо от истинности условия:**

- а) while
- б) do while
- в) for
- г) нет такого цикла в языке Си

**99. В результате выполнения кода:**

```
int i=2;
switch (i) {
 case 1: i += 2;
 case 2: i *= 3;
 case 6: i /= 2;
 default: ;
}
```

- а) переменная i примет значение 6
- б) переменная i примет значение 3
- в) переменная i примет значение 2
- г) тело оператора switch не поменяет значение переменной i

**100. В языке C++ константа 3e02 задаёт число**

- а) 0.03
- б) 30
- в) 300

г) 3

**101. В языке C++ встроенный тип данных char предназначен для хранения**

- а) целых чисел или символов
- б) вещественных чисел
- в) символов
- г) строк

**102. В языке C++ встроенный тип данных int предназначен для хранения**

- а) символов
- б) положительных и отрицательных целых чисел
- в) только положительных целых чисел
- г) вещественных чисел

**103. В языке C++ встроенный тип данных double предназначен для хранения**

- а) символов
- б) вещественных чисел
- в) целых чисел
- г) матриц

**104. В языке C++ идентификаторы Index, INDEX и index обозначают**

- а) одну и ту же переменную
- б) две различных переменных
- в) три различных переменных

**105. Какое объявление переменной в языке C++ является НЕ правильным:**

- а) `int x1;`
- б) `int x1 = 0;`
- в) `int 1x;`

**106. В языке C++ результатом выполнения операции  $5 / 2$  будет число:**

- а) 3
- б) 2
- в) 2.5
- г) 1

**107. В языке C++ результатом выполнения операции  $5 \% 2$  будет число**

- а) 1
- б) 2

- в) 3
- г) 2.5

**108. Какое значение получит переменная x в результате выполнения программы:**

```
int x, y;
x = 0;
y = 0;
x = ++y;
```

- а) 0
- б) 1
- в) 2
- г) результат не определен

**109. Массивы в языке C++ могут быть**

- а) только одномерными
- б) одномерными или двумерными
- в) с любым количеством измерений

**110. Индексация массивов в языке C++ начинается**

- а) с единицы
- б) с нуля
- в) с любого индекса, определяемого программистом

**111. Если одномерный массив в языке C++ состоит из N элементов, то его индекс может принимать значения**

- а) от 0 до N-1
- б) от 1 до N
- в) от 0 до N
- г) от 1 до N-1

**112. Для вывода информации на экран в языке Си используется функция**

- а) output( )
- б) write( )
- в) printf( )
- г) print( )

**113. В языке Си для хранения текстовых строк используется**

- а) массив переменных типа char;
- б) специальный строковый тип данных string;
- в) массив переменных типа string;

**114. Что появится на экране в результате выполнения программы:**

```
int x, y;
x=10;
```

```
y=20;
printf("x = %i", y);
```

а) x = 10  
б) x = 20  
в) x = 0  
г) x = %i

**115. С точки зрения языка C++ выражение является истинным, если**

- а) это выражение равно 0  
б) это выражение не равно 0  
в) это выражение равно 1  
г) это выражение не равно 1

**116. С точки зрения языка C++ выражение является ложным, если**

- а) это выражение равно 0  
б) это выражение не равно 0  
в) это выражение равно 1  
г) это выражение не равно 1

**117. Какое значение получит переменная y в результате выполнения программы:**

```
int x, y;
x = 0;
y = 0;
if (!x) y = 1;
```

- а) 0  
б) 1  
в) -1  
г) значение не определено

**118. Какое значение получит переменная y в результате выполнения программы:**

```
int x=1, y=1;
while (x < 1) {
 x++;
 y++;
}
```

- а) 1  
б) 2  
в) 0  
г) значение не определено

**119. Какое значение получит переменная y в результате выполнения программы:**

```
int x=1, y=1;
```

```
while (x < 2) {
 x++;
 y++;
}
```

- a) 1
- б) 2
- в) 0
- г) значение не определено

**120. Какое значение получит переменная y в результате выполнения программы:**

```
int x=1, y=1;
do {
 x++;
 y++;
} while (x < 1);
```

- a) 1
- б) 2
- в) 3
- г) значение не определено

**121. Какое значение получит переменная y в результате выполнения программы:**

```
int x, y;
y = 1;
for(x=0; x<3; x++) y*= 2;
```

- a) 1
- б) 2
- в) 4
- г) 8

**122. Какое значение получит переменная x в результате выполнения программы:**

```
int x, y;
y = 1;
for(x=0; x<3; x++) y*= 2;
```

- a) 0
- б) 1
- в) 2
- г) 3

**123. В языке C++ существует оператор break. Он служит для того, чтобы**

a) досрочно прекратить выполнение содержащего его ближайшего цикла while, do ... while или for или оператора switch

- б) досрочно прекратить выполнение текущей итерации содержащего его ближайшего цикла while, do ... while или for
- в) досрочно завершить программу

**124. В языке C++ существует оператор continue. Он служит для того, чтобы**

- а) досрочно прекратить выполнение содержащего его ближайшего цикла while, do ... while или for или оператора switch
- б) досрочно прекратить выполнение текущей итерации содержащего его ближайшего цикла while, do ... while или for
- в) досрочно завершить программу

**125. Оператор return в языке C++ служит для**

- а) возвращения функцией значения и прекращения её работы
- б) прекращения функцией работы без возвращения ею значения
- в) возвращения функцией значения без прекращения её работы

**126. Если функции имеют одинаковое имя, но разное количество или тип параметров, то такие функции называются**

- а) вложенными
- б) глобальными
- в) перегруженными
- г) недопустимыми

**127. В языке C++ областью действия локальной переменной по умолчанию является**

- а) вся программа
- б) тот модуль программы, в котором она объявлена
- в) та функция, в которой она объявлена
- г) тот блок, в котором она объявлена

**128. Основное отличие динамического размещения данных от статического состоит в том, что**

- а) статические данные размещаются на диске, динамические – в оперативной памяти
- б) статические данные размещаются в момент старта программы, динамические – в ходе её выполнения
- в) динамические данные размещаются в момент старта программы, статические – в ходе её выполнения

**129. Указателем называется**

- а) переменная, хранящая в качестве значения какой-либо адрес в памяти
- б) любая локальная переменная
- в) любая глобальная переменная
- г) переменная, объявленная со спецификатором pointer

**130. В языке C++ имя одномерного массива эквивалентно**

- а) начальному элементу массива
- б) указателю на начальный элемент массива
- в) последнему элементу массива
- г) указателю на последний элемент массива

**131. Если в функцию в качестве параметра передана переменная по значению, то изменение этой переменной внутри тела функции**

- а) приведёт к её изменению и вне тела функции;
- б) не приведёт к её изменению вне тела функции;
- в) может привести к её изменению вне тела функции при соответствующих настройках компилятора
- г) может привести к её изменению вне тела функции при желании программиста

**132. Если в функцию в качестве параметра передан указатель на переменную, то изменение этой переменной внутри тела функции**

- а) приведёт к её изменению и вне тела функции;
- б) не приведёт к её изменению вне тела функции;
- в) может привести к её изменению вне тела функции при соответствующих настройках компилятора
- г) может привести к её изменению вне тела функции при желании программиста

**133. Если в функцию в качестве параметра передана переменная по ссылке, то изменение этой переменной внутри тела функции**

- а) приведёт к её изменению и вне тела функции;
- б) не приведёт к её изменению вне тела функции;
- в) может привести к её изменению вне тела функции при соответствующих настройках компилятора
- г) может привести к её изменению вне тела функции при желании программиста

**134. В языке C++ возвращаемое значение функции**

- а) может быть указателем
- б) не может быть указателем
- в) может быть указателем, только если это указатель на массив
- г) может быть указателем, только если это указатель на другую функцию

**135. Укажите оператор выбора в языке C ++**

- а) case
- б) choice
- в) switch
- г) default



**136. В сложных выражениях последовательность выполнения операций определяется**

- а) только приоритетом операций
- б) только скобками
- в) скобками, приоритетом операций, а при одинаковом приоритете ассоциативностью операций
- г) только ассоциативностью операций

**137. В чем разница между фактическими и формальными параметрами функций?**

- а) формальные параметры могут использоваться только вне тела функции, а фактические используются как вне функции, так и внутри нее
- б) нет разницы, это одни и те же параметры
- в) и формальные, и фактические параметры используются вне тела функции
- г) формальные параметры определены в теле функции, а фактические – значение, с которыми функция вызывается

**138. Для чего предназначен оператор continue в языке C++?**

- а) пропускает остаток тела цикла и переходит к следующей итерации
- б) пропускает цикл и переходит к следующему оператору в теле программы
- в) определяет условие продолжения цикла
- г) продолжает выполнение текущей итерации цикла

**139. Укажите верное утверждение:**

- а) строки в Си представляются как массивы элементов типа char, заканчивающиеся символом '0'
- б) строки в Си представляются как массивы элементов типа char, заканчивающиеся символом '\0'
- в) строки в Си представляются как массивы элементов типа char, заканчивающиеся символом »
- г) строки в Си представляются как массивы элементов типа char, заканчивающиеся символом '.'

**140. Что выполняет операция ++ в языке C++?**

- а) уменьшает значение операнда на единицу
- б) уменьшает значение операнда на два
- в) увеличивает значение операнда на два
- г) увеличивает значение операнда на единицу

**141. Выберите правильный вариант объявления константной переменной в C++, где type – тип данных; variable – имя переменной; value – константные значения.**

- а) type variable = value;

- б) `const type variable = value; +`
- в) `const variable = value;`
- г) `type const variable = value;`

**142. В каком случае выражение будет вычисляться быстрее?**

- а) `X = X + 1;`
- б) `X += 1;`
- в) `X++;`
- г) Все три выражения будут выполнены с одинаковой скоростью.

**143. Укажите правильный вариант объявления указателя в C++**

- а) `int x;`
- б) `int & x;`
- в) `int * x;`
- г) `ptr x;`

**144. Зачем в C++ используют оператор return?**

- а) чтобы задержать работу программы
- б) функция, в которой он содержится, завершает свое выполнение и управление возвращается в то место программы, из которого вызывалась данная функция
- в) чтобы организовать цикл
- г) чтобы ввести в программу новые значения

**145. В программе на языке C++ объявлены переменные int x, y; Какое выражение позволяет вычислить остаток от деления этих переменных?**

- а) `x % y`
- б) `x div y`
- в) `x mod y`
- г) `x / y`

**146. Определите значение переменной b после выполнения следующих команд:**

- ```
int a, b;  
float c;  
a = 3;  
c = 2.6;  
b = a * c;
```
- а) 6
 - б) 7
 - в) 7.8
 - г) 9

147. Определите значение переменной b после выполнения следующих команд:

```
int a, b;  
a = 5;  
b = 3 * ++a;
```

- а) 15
- б) 17
- в) 18
- г) 21

148. Выберите верные утверждения, относящиеся к массиву:

- а) это совокупность переменных одного типа
- б) переменные хранятся в последовательно расположенных ячейках оперативной памяти
- в) последовательность переменных, которые имеют одинаковое имя
- г) благодаря нумерации переменные располагаются только в строку

149. Что может быть индексом элемента массива

- а) символ
- б) дробное число
- в) только целое положительное число
- г) целое число, большее или равное нулю

150. Пусть `int a [5] = {10, 11, 12, 13, 14}`. Чему равно значение элемента `a [3]`?

- а) 3
- б) 12
- в) 13
- г) 14

151. Укажите правильно организованный цикл для обработки всех элементов массива, размерность которого `n`

- а) `for (int i = 1; i <= n; i++)`
- б) `for (int i = 0; i <= n; i++)`
- в) `for (int i = 1; i <n; i++)`
- г) `for (int i = 0; i <n; i++)`

152. Укажите тип данных, определяет служебное слово `struct`:

- а) массив данных с различной структурой
- б) тип функций, которые могут иметь различную структуру параметров
- в) тип данных, которые могут менять свою структуру
- г) составной объект, к которому могут входить элементы различных типов.

153. Понятие `this` в классе:

- а) указатель `this` является скрытым аргументом метода, превращает функцию в область памяти только для чтения.

б) объект `this` является аргументом метода другого класса, существует во всех методах и указывает на адрес: `this` -> объект

в) указатель `this` является скрытым аргументом метода, существует во всех методах объекта и указывает на его (объект) адрес: `this` -> объект

154. Какие функции есть у любого класса?

- а) конструктор
- б) деструктор
- в) конструктор и деструктор
- г) нет таких

155. Имеет ли деструктор аргументы?

- а) имеет всегда
- б) не имеет никогда
- в) наличие аргументов зависит от свойств класса
- г) наличие аргументов зависит от программиста

156. Что такое деструктор?

а) деструктор – это функция, которая должна открывать динамическую область для экземпляра класса

б) деструктор – это специальная функция-элемент, которая должна отслеживать данные в экземпляре класса в процессе работы

в) деструктор – это специальная функция-элемент, которая должна уничтожать экземпляр класса после завершения его работы

157. Какое значение должен возвращать деструктор?

- а) объект класса
- б) код состояния о правильном удалении класса
- в) указатель на класс
- г) деструкторы не возвращают значение

158. Какая функция выполняет начальную инициализацию данных в классе?

- а) конструктор
- б) деструктор
- в) инициализатор
- г) нет правильного ответа

159. Укажите корректное объявление класса:

- а) `class A { int x; };`
- б) `class B { }`
- в) `public class A { }`
- г) `object A { int x; };`

160. Какого спецификатора доступа в классах нет?

- а) `public`

- б) hidden
- в) private
- г) protected

161. Класс - это:

- а) любой тип данных, определяемый пользователем
- б) тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки
- в) структура, для которой в программе имеются функции работы с нею

162. Членами класса могут быть

- а) как переменные, так и функции, могут быть объявлены как private и как public
- б) только переменные и функции, объявленные как private
- в) только переменные и функции, объявленные как public
- г) переменные, объявленные только как private, и функции, объявленные только как public

163. Что называется конструктором?

- а) метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса
- б) метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)
- в) метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса
- г) метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса

164. Выберите неверное утверждение

- а) конструкторы класса не наследуются
- б) конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций
- в) точка вызова конструктора определяется программистом
- г) конструктор не возвращает значение

165. Выберите правильное утверждение

- а) деструктор – это метод класса, без которого невозможно удаление объекта
- б) деструктор – это отдельная функция головной программы, применяемая для освобождения памяти, занимаемой объектом
- в) деструктор не наследуется
- г) деструктор наследуется, но должен быть перегружен

166. Что называется наследованием?

а) это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы

б) это механизм переопределения методов базового класса

в) это механизм, посредством которого производный класс получает все поля базового класса

г) это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить

167. Выберите правильное объявление производного класса

а) `class MoreDetails:: Details;`

б) `class MoreDetails: public class Details;`

в) `class MoreDetails: public Details;`

г) `class MoreDetails: class(Details);`

168. Дружественная функция – это

а) функция другого класса, среди аргументов которой есть элементы данного класса

б) функция, объявленная в классе с атрибутом `friend`, но не являющаяся членом класса;

в) функция, являющаяся членом класса и объявленная с атрибутом `friend`;

г) функция, которая в другом классе объявлена как дружественная данному

169. Переопределение операций имеет вид:

а) имя_класса, ключевое слово `operation`, символ операции

б) имя_класса, ключевое слово `operator`, символ операции, в круглых скобках могут быть указаны аргументы

в) имя_класса, ключевое слово `operator`, список аргументов

г) имя_класса, два двоеточия, ключевое слово `operator`, символ операции

170. Для доступа к элементам объекта используются:

а) при обращении через имя объекта – точка, при обращении через указатель – операция «->»

б) при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «точка»

в) при обращении через имя объекта – точка, при обращении через указатель – два двоеточия

г) при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «->»

171. Определите, что такое производный класс:

а) класс, построенный на основе другого класса

б) класс, на основе которого построен другой класс

- в) класс, на основе которого построено ни одного класса
- г) класс, у которого кроме элементов являются методы

172. Имеет ли доступ производный класс к закрытым элементам своего базового класса:

- а) всегда имеет
- б) никогда не имеет
- в) имеет при открытом наследовании
- г) имеет при отсутствии спецификатора наследования

173. Определите, что такое множественное наследование:

- а) наследование нескольких производных классов от одного базового
- б) наследования производного класса от нескольких базовых
- в) наследование производного класса с спецификатором `final`
- г) наследование производного класса от класса, который является производным от другого класса

Примеры вариантов тестовых заданий с ответами

1 вариант

№	Вопрос	Ответ
1	<p>Какой из перечисленных типов данных не является типом данных в C++?</p> <p>а) float б) double в) real г) int</p>	в
2	<p>Укажите правильное объявление массива в C++:</p> <p>а) <code>int a[10];</code> б) <code>int a{10};</code> в) <code>int array;</code> г) <code>array a[10];</code></p>	а
3	<p>Чему будет равна переменная a, после выполнения следующего кода на C++: <code>int a; for(a = 0; a < 10; a++) { }</code> ?</p> <p>а) 9</p>	б

	б) 10 в) 1 г) 0	
--	-----------------------	--

2 вариант

№	Вопрос	Ответ
1	Какой из ниже перечисленных операторов, не является оператором цикла в C++? а) while б) for в) repeat until г) do while	в
2	Укажите правильное объявление указателя в C++: а) int &x; б) int *x; в) ptr x; г) int x;	б
3	Чему будет равна переменная b, после выполнения следующего кода на C++: int b; for(b = 10; b>=0; b--) { } ? а) 10 б) 9 в) 0 г) -1	г

Критерии оценки теста

5 баллов – правильно выполнено 86-100% заданий;

4 балла – правильно выполнено 71-85% заданий;

3 балла – правильно выполнено 51-70% заданий;

0 баллов (не зачтено) – правильно выполнено менее 51% заданий.