

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени В.Ф. Уткина

Кафедра электронных вычислительных машин


«СОГЛАСОВАНО»

Декан факультета ВТ
 Д.А. Перепелкин
«__»__ 2020 г.



«ПРИТВЕРЖДАЮ»

Проректор по РОПиМД
 А.В. Корячко
«__»__ 2020 г.

Заведующий кафедрой ЭВМ
 Б.В. Костров
«__»__ 2020 г.

РАБОЧАЯ ПРОГРАММА

дисциплины

**Б1.В.01.01 «Основы алгоритмизации
и объектно-ориентированное программирование»**

Направления подготовки:

02.03.03 «Математическое обеспечение и администрирование
информационных систем»

ОПОП академического бакалавриата
«Математическое обеспечение
и администрирование информационных систем»

Квалификация выпускника – бакалавр

Форма обучения - очная

Рязань 2020 г.

ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности) 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата), утвержденным приказом Минобрнауки России от 23.08.2017г. № 809.

Программу составил
к.т.н., доц. кафедры
«Электронные вычислительные машины»



С.И. Елесина

Программа рассмотрена и одобрена на заседании кафедры ЭВМ
«14» 06 2020 г., протокол № 10

Заведующий кафедрой
«Электронные вычислительные машины»,
д.т.н., проф. кафедры ЭВМ



Б.В. Костров

1 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Рабочая программа по дисциплине «Основы алгоритмизации и объектно-ориентированное программирование» является составной частью основной профессиональной образовательной программы (ОПОП) по направлению подготовки бакалавров 02.03.03 – «Математическое обеспечение и администрирование информационных систем», разработанной в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 02.03.03 – «Математическое обеспечение и администрирование информационных систем» (уровень бакалавриата), утвержденным приказом Минобрнауки России от 23.08.2017 г. № 809.

Цель дисциплины: формирование базовых профессиональных компетенций по разработке программного обеспечения на языке программирования высокого уровня.

Задачи, решение которых обеспечивает достижение цели:

- знание современных технологий программирования (структурное, модульное, объектно-ориентированного);
- умение ставить задачу и разрабатывать алгоритм ее решения;
- умение составлять и оформлять программы на языках программирования;
- умение использовать принципы объектно-ориентированного программирования;
- знание интегрированных сред разработки программ;
- владение навыками разработки и отладки программ на алгоритмическом языке программирования высокого уровня C/C++.

Перечень планируемых результатов обучения по дисциплине

Выпускник, освоивший программу бакалавриата, должен обладать следующими компетенциями:

Код компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ПК-1	Способен проектировать программное обеспечение с использованием современных инструментальных средств	<p>ПК-1.1. Знает основные принципы и приемы построения архитектуры программного обеспечения; методы и средства проектирования программного обеспечения, в том числе баз данных.</p> <p>ПК-1.2. Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.</p> <p>ПК-1.3. Имеет практический опыт построения</p>

		архитектуры программного обеспечения, проектирования структур данных, баз данных и программных интерфейсов с использованием современных инструментальных средств
--	--	--

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к основной части блока Б1 (Б1.В.01) основной образовательной программы подготовки бакалавров направления 02.03.03 «Математическое обеспечение и администрирование информационных систем».

Дисциплина изучается по очной форме обучения на 1-2 курсах в 1-3 семестрах.

Дисциплина базируется на следующих дисциплинах «Математика» и «Информатика», изучаемых в общеобразовательной школе.

Знания, полученные в результате освоения дисциплины будут полезны обучающемуся при изучении дисциплин: «Параллельное программирование», «Рекурсивно-логическое программирование», «Визуальное программирование», «Структуры и алгоритмы компьютерной обработки данных».

Материал дисциплины «Основы алгоритмизации и объектно-ориентированное программирование» формирует методологические и организационные основы для выполнения обучающимися практик и выпускной квалификационной работы.

3 Объём дисциплины и виды учебной работы (часы)

Объём дисциплины в зачетных единицах (ЗЕ): 11 ЗЕ, 396 часов.

Вид учебной работы	Всего часов	1 семестр	2 семестр	3 семестр
Общая трудоемкость дисциплины	396	180	108	108
1. Контактная работа обучающихся с преподавателем (всего), в том числе:	165,25	66,35	48,55	48,35
лекции	64	32	16	16
практические занятия	48	16	16	16
лабораторные работы	48	16	16	16
консультации	4	2	-	2
иная контактная работа (промежуточная аттестация)	1,25	0,35	0,55	0,35
2. Самостоятельная работа обучающихся (всего), в том числе:	132,7	69	50,7	13
курсовой проект (работа)	11,7	-	11,7	-
иная самостоятельная работа	121	69	39	13
3. Контроль	98,05	44,65	8,75	44,65
Вид промежуточной аттестации обучающегося	Экзамен, зачет	экзамен	зачет	экзамен

4 Содержание дисциплины, структурированное по темам с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Содержание дисциплины, структурированное по темам

Тема 1. Общие принципы разработки программного обеспечения.

Метод проектирования программных средств. Понятия алгоритма, его свойства. Способы представления алгоритмов. Базовые алгоритмические структуры: следование,

разветвление, повторение. Способы их изображения. Методы тестирования и отладки программ. Документирование. ГОСТ ЕСПД.

Средства для разработки программ. Основные этапы решения задач на ЭВМ. Интегрированная среда программирования Visual C++. Основные характеристики среды, интерфейс, основные команды меню. Средства тестирования и отладки, встроенные в системы программирования.

Тема 2. Основные элементы языка программирования C/C++

Краткая история и особенности языка C/C++. Структура программы. Структура языка: алфавит, лексика, синтаксис языка. Константы в языке. Понятие и классификация типов данных. Базовые типы данных в C++. Скалярные (простые) типы данных. Перечислимый тип. Определение переменных. Преобразования типов.

Тема 3. Операции языка. Выражения. Математические функции.

Операции языка C++, приоритеты и их направленность выполнения. Операция присваивания. Арифметические и логические выражения. Классификация операций по приоритетам.

Тема 4. Операторы языка C++.

Операторы языка C++ и их синтаксис. Функции ввода и вывода данных. Блок и составной оператор. Условный оператор и оператор варианта. Разработка разветвляющихся алгоритмов. Операторы циклов. Использование шаблонов при создании циклов. Табулирование функций. Обработка последовательностей. Обработка n-арных операций. Итерационные алгоритмы.

Тема 5. Препроцессор языка C++

Директивы препроцессора. Директива включения файла. Макроопределения. Условная компиляция. Использование команд препроцессора для генерации исходного текста программы.

Тема 6. Указатели, адресная арифметика.

Адрес ячейки памяти. Переменная указатель. Адресная арифметика, типы указателей и операции над ними. Инициализация указателей. Типичные ошибки при работе с указателями. Указатели типа void.

Тема 7. Массивы, строки и указатели.

Одномерные и многомерные массивы, расположение элементов массива в памяти. Первоначальная инициализация одномерных и многомерных массивов. Правила вычисления адреса элементов массива. Указатели на указатели. Указатель на массив и массивы указателей. Алгоритмы сортировки массивов. Основные алгоритмы работы с многомерными массивами. Строки: определение, инициализация, функции для работы со строками.

Тема 8. Функции и классы памяти. Рекурсия

Определение, описание и вызов функций. Тип функции. Понятие фактических и формальных параметров. Прототипы функции. Механизмы передачи параметров по адресу и по значению. Оператор возврата из функции return. Вызов процедур и функций на исполнение. Понятие блока в Паскале. Область действия и время жизни глобальных и локальных переменных. Пространства имен. Классы памяти. Функции, возвращающие указатели. Указатели на функции. Интерпретация сложных описаний. Рекурсивный подход

к программированию. Организация рекурсивного алгоритма. Примеры рекурсивных алгоритмов.

Тема 9. Структуры и объединения

Структурированные типы данных, их классификация. Структуры. Массивы структур, вложенные структуры, указатели на структуры. Битовые поля структур. Объединения.

Тема 10. Функции с переменным числом параметров. Перегрузка функций

Описание и определение функции, количество параметров которой не определено. Механизмы определения числа и типов параметров. Inline-функции. Механизм перегрузки функции. Примеры использования перегруженных функций. Автоматизация создания функций для обработки разнотипных данных.

Тема 11. Поточковый ввод-вывод. Файлы.

Типы потоков, основные функции работы с потоками. Организация работы с файлами. Файлы, способы организации и методы доступа. Работа с текстовыми и бинарными файлами. Форматированный ввод-вывод. Символьный ввод-вывод. Строчный ввод-вывод. Блочный ввод-вывод. Прямой доступ.

Тема 12. Указатели и динамические структуры данных

Динамическое распределение памяти в Си. Выделение и освобождение памяти. Создание одномерных и многомерных динамических массивов. Манипулирование блоков памяти. Динамические структуры данных: стеки, списки. Работа с динамическими структурами данных.

Тема 13. Простейший графический интерфейс в Visual C++.

Основные компоненты библиотеки VCL. Разработка графического интерфейса. Формы. Меню. Модальные окна. Диалоги. Сообщения. Компоненты ввода и отображения информации. Кнопки, индикаторы и управляющие элементы. Отображение графической и мультимедиа информации.

Тема 14. Базовые принципы ООП.

Объекты и классы. Свойства и функции класса. Базовые принципы объектно-ориентированного программирования. Базовые конструкции объектно-ориентированных программ. Поддержка объектно-ориентированного программирования в языке C++.

Тема 15. Объекты и классы в языке C++.

Понятие класса и объекта. Указатель на текущий объект. Ограничение доступа к компонентам объекта. Конструкторы и деструкторы. Встроенные функции. Организация внешнего доступа к компонентам класса. Дружественные классы и функции. Вложенные классы. Статические компоненты класса. Создание и уничтожение объектов. Диаграммы классов.

Тема 16. Наследование. Производные классы.

Наследование классов и функций. Основные правила построения производных классов и работа с ними. Конструкторы и деструкторы. Виртуальные функции. Абстрактные классы. Множественное наследование. Управление доступом к компонентам класса. Виртуальные деструкторы.

Тема 17. Перегрузка операторов.

Основные принципы перегрузки операторов. Функция operator. Переопределение бинарных и унарных операторов. Преобразование типа. Присваивание и инициализация. Функция operator[]. Функция operator() .Функция operator->. Переопределение операторов new и delete.

Тема 18. Параметризованные классы.

Параметризованные классы (шаблоны). Шаблоны функций (передача типа в функцию в виде параметра). Методы использования шаблонов. Переопределение шаблонов функций. Передача дополнительных аргументов в шаблоны. Совместное использование шаблонов и принципов наследования.

Тема 19. Обработка исключений.

Понятие исключения. Определение типа исключений. Получение дополнительной информации об исключениях. Иерархия исключений. Исключения, которые не являются ошибками. Спецификация функций, обрабатывающих исключения. Взаимодействие программ на языке C++.

Тема 20. Стандартная библиотека (STL).

Контейнеры, Алгоритмы. Итераторы. Функциональные объекты.

Тема 21. Разработка объектно-ориентированного ПО.

Основные концепции программирования. Основные этапы разработки ООП (начало, развитие, построение и передача). Принципы разработки ООП по этапам средствами UML.

4.2. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах).

Название раздела	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем						Самостоятельная работа обучающихся	Контроль
		Всего	Лекции	Практические занятия	Лабораторные работы	Консультации	Иные виды контактной работы		
Общие принципы разработки программного обеспечения	14,5	4,5	2	2	-	0,5	-	6	4
Основные элементы языка программирования C/C++	14,5	4,5	2	-	2	0,5	-	6	4
Операции языка. Выражения. Математические функции	22	8	4	2	2	-	-	8	6
Операторы языка C++	32	14	6	4	4	-	-	10	8
Препроцессор языка C++	9,5	2,5	2	-	-	0,5	-	4	3
Указатели, адресная арифметика	13,5	2,5	2	-	-	0,5	-	8	3
Массивы, строки и указатели	26	10	6	2	2	-	-	10	6
Функции и классы памяти. Рекурсия	31	14	6	6	2	-	-	10	7

Название раздела	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем						Самостоятельная работа обучающихся	Контроль
		Всего	Лекции	Практические занятия	Лабораторные работы	Консультации	Иные виды контактной работы		
Структуры и объединения	19	6	2	-	4	-	-	9	4
Функции с переменным числом параметров. Перегрузка функций.	18	8	4	2	2	-	-	8	2
Потоковый ввод-вывод. Файлы	21	10	4	2	4	-	-	9	2
Указатели и динамические структуры данных	27	16	4	4	8	-	-	9	2
Простейший графический интерфейс в Visual C++.	23	12	2	8	2	-	-	9	2
Базовые принципы ООП	7	2	2	-	-	-	-	4	1
Объекты и классы в языке C++	14	6	2	-	4	-	-	2	6
Наследование. Производные классы.	22	10	4	2	4	-	-	3	9
Перегрузка операторов	16,5	6,5	2	4	-	0,5	-	4	6
Параметризированные классы	16,5	6,5	2	4	-	0,5	-	4	6
Обработка исключений.	16,5	6,5	2	-	4	0,5	-	4	6
Стандартная библиотека (STL).	18	8	2	2	4	-	-	4	6
Разработка объектно-ориентированного ПО	15,25	6,5	2	4	-	0,5	-	3,7	5,05
Промежуточная аттестация	1,25	1,25	-	-	-	-	1,25	-	-
Итого	396	165,25	64	48	48	4	1,25	132,7	98,05

Виды лабораторных, практических и самостоятельных работ

Тема	Вид работы	Наименование и содержание работы	Трудоемкость, часов
Тема 1. Общие принципы разработки программного обеспечения.	Практическое занятие	Создание консольного приложения в интегрированной среде Visual C++.	2
		Ввод и редактирование программ в ИС Visual C++.	
	Самостоятельная работа	Изучение конспекта лекций	3
		Изучение методических указаний, подготовка к практическим занятиям	3
		Подготовка к экзамену	4

Тема 2. Основные элементы языка программирования C/C++	Лабораторная работа	Подготовка программ к выполнению и выполнение программ в ИС Visual C++. Программирование линейных программ	2
	Самостоятельная работа	Изучение конспекта лекций	3
		Изучение методических указаний, подготовка к лабораторной работе Подготовка к экзамену	3 4
Тема 3. Операции языка. Выражения. Математические функции	Практическое занятие	Операции в языке C++	2
	Лабораторная работа	Работа с двоичными масками	2
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям и лабораторной работе Подготовка к экзамену	3 5 6
Тема 4. Операторы языка C++.	Практическое занятие	Разработка разветвляющихся алгоритмов	2
		Обработка последовательностей. Обработка n-арных операций и итерационные циклы	2
	Лабораторная работа	Условный оператор и оператор варианта	2
		Табулирование функций	2
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям и лабораторным работам Подготовка к экзамену	2 8 8
Тема 5. Препроцессор языка C++	Самостоятельная работа	Изучение конспекта лекций и разбор примеров	4
		Подготовка к экзамену	3
Тема 6. Указатели, адресная арифметика.	Самостоятельная работа	Изучение конспекта лекций и самостоятельное решение примеров	8
		Подготовка к экзамену	3
Тема 7. Массивы, строки и указатели	Практическое занятие	Разработка алгоритмов для обработки массивов	2
	Лабораторная работа	Строки	2
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям и лабораторным работам Подготовка к экзамену	5 5 6
Тема 8. Функции и классы памяти.	Практическое занятие	Механизм передачи параметров по адресу и по значению	2

Рекурсия		Интерпретация сложных описаний	2
		Рекурсия	2
	Лабораторная работа	Применение функций при обработке двумерных массивов	2
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям и лабораторным работам Подготовка к экзамену	5 5 7
Тема 9. Структуры и объединения	Лабораторная работа	Объединения	2
		Структуры	2
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к лабораторным работам Подготовка к экзамену	3 6 4
Тема 10. Функции с переменным числом параметров. Перегрузка функций	Практическое занятие	Функции с переменным числом параметров.	2
	Лабораторная работа	Перегрузка функций	2
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям и лабораторным работам Подготовка к зачету	2 6 2
Тема 11. Поточный ввод-вывод. Файлы.	Лабораторная работа	Работа с текстовыми файлами	4
	Практическое занятие	Работа с двоичными файлами. Блочный ввод вывод. Прямой доступ	2
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям и лабораторным работам Подготовка к зачету	3 6 2
Тема 12. Указатели и динамические структуры данных	Лабораторная работа	Однонаправленный список	4
		Стек. Обратная польская запись	4
	Практическое занятие	Динамические массивы	4
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям и лабораторным работам Подготовка к зачету	3 6 2
Тема 13. Простейший	Практическое занятие	Разработка графического интерфейса в Visual C++	8

графический интерфейс в Visual C++.	Лабораторная работа	Работа с графикой в Visual C++	2
	Самостоятельная работа	Изучение конспекта лекций Изучение методических указаний, подготовка к практическим занятиям и лабораторным работам Подготовка к зачету	2 7 2
Тема 14. Базовые принципы ООП	Самостоятельная работа	Изучение конспекта лекций. Подготовка к зачету	4 1
Тема 15. Объекты и классы в языке C++	Лабораторная работа	Введение в объектно-ориентированное программирование. Понятие объекта и класса	4
	Самостоятельная работа	Изучение конспекта лекций и подготовка к лабораторной работе. Подготовка к экзамену	2 6
Тема 16. Наследование. Производные классы.	Лабораторная работа	Инкапсуляция данных и наследование Виртуальные функции	2 2
	Практическое занятие	Абстрактные классы	2
	Самостоятельная работа	Изучение конспекта лекций и подготовка к ЛР и ПЗ Подготовка к экзамену	3 9
Тема 17. Перегрузка операторов	Практическое занятие	Разработка программы «Перегрузка операторов»	4
	Самостоятельная работа	Изучение конспекта лекций и подготовка к ПЗ Подготовка к экзамену	4 6
Тема 18. Параметризованные классы	Практическое занятие	Разработка программы «Шаблоны»	4
	Самостоятельная работа	Изучение конспекта лекций и подготовка к ПЗ Подготовка к экзамену	4 6
Тема 19. Обработка исключений	Лабораторная работа	Разработка программы «Обработка исключений»	4
	Самостоятельная работа	Изучение конспекта лекций и подготовка к ЛР Подготовка к экзамену	4 6
Тема 20. Стандартная библиотека (STL)	Лабораторная работа	Контейнер для динамического массива	2
		Контейнерный класс для реализации множества	2
	Практическое занятие	Алгоритмы в STL	2
	Самостоятельная работа	Изучение конспекта лекций и подготовка к ЛР и ПЗ Подготовка к экзамену	4 6
Тема 21. Разработка объектно-ориентированного ПО	Практическое занятие	Применение UML при разработке объектно-ориентированного ПО	4
	Самостоятельная работа	Изучение конспекта лекций и подготовка к ПЗ Подготовка к экзамену	3,7 5,05

Курсовое проектирование

В курсовом проекте рассматриваются основные разделы дисциплины, которые были рассмотрены на момент выполнения курсовой работы: динамические структуры, структуры, массивы, ввод-вывод, создание графического пользовательского интерфейса.

Тема курсового проекта: «Разработка приложений с использованием динамических структур данных». Все варианты заданий связаны с разработкой таблицы данных с использованием линейных однонаправленных списков.

Разрабатываемая программа должна обязательно выполнять следующие запросы:

- заполнение пустой таблицы (создание списка);
- сохранение таблицы (списка) в файле;
- чтение таблицы (списка) из файла;
- вывод таблицы на экран;
- добавление элементов в таблицу (в список);
- удаление элементов из таблицы (из списка);
- а также все запросы, которые указаны в индивидуальном задании.

Вызовы запросов должны осуществляться через систему меню с использованием средств графического интерфейса Visual C++. Необходимо предусмотреть контроль ошибок пользователя при вводе данных. Результаты некоторых запросов (по согласованию с преподавателем на этапе уточнения технического задания) должны выводиться в виде графиков или диаграмм.

Все элементарные действия (заполнение списка, запись элемента в список и т.д.) должны быть оформлены в виде подпрограмм, а все (или некоторые) объявления и подпрограммы должны быть оформлены в виде модуля (модулей).

Весь перечень заданий приведен в методических указаниях к курсовому проекту № 4692.

Пример варианта индивидуального задания «Абоненты библиотеки»

Информация об абонентах библиотеки следующая:

- номер читательского билета;
- ФИО;
- год рождения;
- пол;
- подразделение (кафедра, номер группы);
- должность;
- отметка о перерегистрации;
- имеются книги на срок;
- дата возврата книг.

Написать программу, которая выполняет следующие запросы:

- вывод информации об абоненте по номеру читательского билета;
- упорядочение таблицы по ФИО;
- вывод списка абонентов-должников определенного факультета;
- вывод списка абонентов, которые не прошли перерегистрацию;
- вывод процентного соотношения сотрудников и студентов среди абонентов.

5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

- 1) Программирование: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост. С.И. Елесина. Рязань, 2011. 40 с. № 4493. (24 экз.)
- 2) Программирование: методические указания к курсовому проекту / Рязан. гос. радиотехн. ун-т; сост. С.И. Елесина. Рязань, 2013. 24 с. № 4692 (16 экз.)
- 3) Основы разработки объектно-ориентированного программного обеспечения. Часть 1 [Электронный ресурс]: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост.: А.М. Гостин, А.Н. Сапрыкин. Рязань, 2014. 16 с. — Режим доступа: <http://elib.rsreu.ru/ebs/download/554>
- 4) Основы разработки объектно-ориентированного программного обеспечения. Часть 2 [Электронный ресурс]: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост.: А.М. Гостин, А.Н. Сапрыкин. Рязань, 2016. 16 с. — Режим доступа: <http://elib.rsreu.ru/ebs/download/558>
- 5) Основы разработки объектно-ориентированного программного обеспечения. Часть 3 [Электронный ресурс]: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост.: А.М. Гостин, А.Н. Сапрыкин. Рязань, 2017. 16 с. — Режим доступа: <http://elib.rsreu.ru/ebs/download/567>

6 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Фонд оценочных средств приведен в Приложении к рабочей программе дисциплины (см. документ «Оценочные материалы по дисциплине «Основы алгоритмизации и объектно-ориентированное программирование»).

7 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная учебная литература:

- 1) Каширин И.Ю. От С к С++ : Учеб.пособие / И. Ю. Каширин, В. С. Новичков. - М.:Горячая линия-Телеком, 2005. - 324с. (78 экз.)
- 2) Керниган, Б. В. Язык программирования С [Электронный ресурс] / Б. В. Керниган, Д. М. Ричи. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 313 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73736.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)
- 3) Стенли, Липпман Язык программирования С++ [Электронный ресурс] : полное руководство / Липпман Стенли, Лажойе Жози ; пер. А. Слинкин. — Электрон. текстовые данные. — Саратов : Профобразование, 2017. — 1104 с. — 978-5-4488-0136-5. — Режим доступа: <http://www.iprbookshop.ru/63964.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)
- 4) Александров, Э. Э. Программирование на языке С в Microsoft Visual Studio 2010 [Электронный ресурс] / Э. Э. Александров, В. В. Афонин. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 570 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73712.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)

5) Страуструп Б. Язык программирования C++ для профессионалов [Электронный ресурс]/ Страуструп Б.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 670 с.— Режим доступа: <http://www.iprbookshop.ru/73737.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)

Дополнительная учебная литература:

1) Котлинская, Г.П. Программирование на языке СИ : Справ.пособие / Г. П. Котлинская, О. И. Галиновский. - Минск: Вышэйшая школа, 1991. - 156с. (9 экз.)

2) Лисицин, Д. В. Объектно-ориентированное программирование [Электронный ресурс] : конспект лекций / Д. В. Лисицин. — Электрон. текстовые данные. — Новосибирск : Новосибирский государственный технический университет, 2010. — 88 с. — 978-5-7782-1454-5. — Режим доступа: <http://www.iprbookshop.ru/44970.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)

3) Зырянов, К. И. Программирование на C++ [Электронный ресурс] : учебное пособие / К. И. Зырянов, Н. П. Кисленко. — Электрон. текстовые данные. — Новосибирск : Новосибирский государственный архитектурно-строительный университет (Сибстрин), ЭБС АСВ, 2017. — 129 с. — 978-5-7795-0817-9. — Режим доступа: <http://www.iprbookshop.ru/85873.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)

4) Букунов, С. В. Основы объектно-ориентированного программирования [Электронный ресурс] : учебное пособие / С. В. Букунов, О. В. Букунова. — Электрон. текстовые данные. — СПб. : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017. — 196 с. — 978-5-9227-0713-8. — Режим доступа: <http://www.iprbookshop.ru/74339.html> - ЭБС "IPRbooks", по паролю (дата обращения: 24.06.2019)

5) Приемы объектно-ориентированного проектирования. Паттерны проектирования. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес – С-Пб.: Питер, 2010. - 366 с.

8 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Обучающимся предоставлена возможность индивидуального доступа к следующим электронно-библиотечным системам:

Электронно-библиотечная система «IPRbooks», режим доступа – с любого компьютера РГПТУ без пароля, из сети интернет по паролю. – URL: <https://iprbookshop.ru/>.

Электронные образовательные ресурсы:

Дмитрий Полевой. Лекции — C++ и основы ООП (видеоуроки) [Электронный ресурс]. – URL: <http://www.youtube.com/playlist?list=PLE9F6A65165CBC023>

Профессиональные базы данных, информационно-справочные системы:

Программирование на C и C++ (онлайн справочник) [Электронный ресурс]. – URL: <http://www.c-cpp.ru/books/obektno-orientirovannoe-programmirovanie>

9. Методические указания для обучающихся по освоению дисциплины

Указания в рамках лекций

Во время лекции студент должен вести краткий конспект.

Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания. При этом обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции.

Обучающимся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Указания в рамках лабораторных работ

Лабораторные работы составляют важную часть профессиональной подготовки студентов. Основная цель проведения лабораторных работ – формирование у студентов аналитического и творческого мышления путем приобретения практических навыков.

Методические указания к лабораторным работам по дисциплине наряду с рабочей программой и графиком учебного процесса относятся к методическим документам, определяющим уровень организации и качества образовательного процесса. Содержание лабораторных работ фиксируется в рабочей программе дисциплины.

При подготовке к лабораторным работам необходимо просмотреть конспекты лекций и методические указания, рекомендованную литературу по данной теме, а так же подготовиться к ответу на контрольные вопросы.

В ходе выполнения индивидуального задания лабораторной работы студент готовит отчет о работе (с помощью офисного пакета LibreOffice или в другом редакторе доступном студенту, схемы алгоритмов выполняются с помощью векторного графического редактора Microsoft Office Visio). В отчете отражаются цель работы, индивидуальное задание, постановка задачи, разработка алгоритма решения поставленной задачи и программа. Должны быть подготовлены тестовые наборы данных и отражены результаты работы программы.

За 10 минут до окончания занятия преподаватель проверяет объем выполненной за занятие работы и отмечает результат в рабочем журнале. Оставшиеся невыполненными пункты задания студент обязан доделать самостоятельно.

После проверки отчета преподаватель может проводить устный или письменный опрос студентов для контроля усвоения ими основных теоретических и практических знаний по теме занятия (студенты должны знать смысл полученных ими результатов и ответы на контрольные вопросы). По результатам проверки отчета и опроса выставляется оценка за лабораторную работу.

Указания в рамках практических занятий

Перед выполнением практического занятия необходимо внимательно ознакомиться с заданием. Желательно заранее выполнить подготовку проекта в инструментальной среде, чтобы на практическом занятии осталось время для сдачи работы.

Перед сдачей работы рекомендуется ознакомиться со списком вопросов изучаемой темы и попытаться самостоятельно на них ответить, используя конспект лекций и рекомендуемую литературу. Таким образом вы сможете сэкономить свое время и время преподавателя.

Кроме чтения учебной литературы из обязательного списка рекомендуется активно использовать информационные ресурсы сети Интернет по изучаемой теме. Ответы на многие вопросы, связанные с разработкой программ на объектно-ориентированном языке, использованием языковых конструкций, принципов ООП, освоением инструментальной среды, вы можете получить в сети Интернет, посещая соответствующие информационные ресурсы.

Указания в рамках подготовки к промежуточной аттестации

При подготовке к экзамену и зачетам в дополнение к изучению конспектов лекций, учебных пособий, слайдов и другого раздаточного материала предусмотренного рабочей программой дисциплины, необходимо пользоваться учебной литературой, рекомендованной к настоящей рабочей программе. При подготовке к экзамену и зачету нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить по нескольким типовым задач из каждой темы (в том случае если тема предусматривает решение задач). При решении задач всегда необходимо уметь качественно интерпретировать итог решения.

Указания в рамках выполнения курсовой работы

Целью курсовой работы являются разработка и отладка приложения с использованием динамических структур данных, написанного на алгоритмическом языке программирования C++.

Курсовая работа (КР) представляет собой самостоятельную работу по заданной теме. Работа предполагает:

- домашнюю внеаудиторную подготовку;
- выполнение практической части дома или в классе персональных компьютеров на кафедре;
- консультации по КР;
- предъявление промежуточных результатов для проверки и контроля хода выполнения КР;
- написание и оформление пояснительной записки;
- сдачу и защиту КР в сроки согласно учебному графику.

Основные этапы работы над курсовой работой:

1) Постановка задачи: выбираются исходные данные, их типы и структуры, указываются ограничения на исходные данные; выполняется детализация функций программы; определяются формы представления результатов работы программы. Разработанную постановку задачи необходимо согласовать с преподавателем.

2) Разработка алгоритмов: выделение модулей программы; разработка алгоритмов каждого модуля с использованием нисходящего проектирования; представление результатов в виде схем алгоритмов в соответствии с ГОСТ 19.701.

3) Разработка программы: выбор средств языка высокого уровня C++ для написания текста программы по разработанным на предыдущем этапе схемам алгоритмов; описания исходных текстов программ и структур входных, выходных и промежуточных данных.

4) Экспериментальная проверка программы. Результаты экспериментальной проверки разработанной программы при различных исходных данных с указанием: всех использованных вариантов контрольных наборов и полученных результатов для каждого варианта; вывода о степени правильности функционирования программы.

5) Оформление пояснительной записки к КР выполняется с помощью офисного пакета LibreOffice или в другом редакторе доступном студенту, схемы алгоритмов выполняются с помощью векторного графического редактора Microsoft Office Visio.

Указания в рамках самостоятельной работы студентов

Самостоятельная работа студентов по дисциплине играет важную роль в ходе всего учебного процесса. Методические материалы и рекомендации для обеспечения самостоятельной работы студентов готовятся преподавателем и выдаются студентам в виде раздаточных материалов или оформляются в виде электронного ресурса используемого в рамках системы дистанционного обучения ФГБОУ ВО «РГРТУ».

Самостоятельное изучение тем учебной дисциплины способствует:

- закреплению знаний, умений и навыков, полученных в ходе аудиторных занятий;
- углублению и расширению знаний по отдельным вопросам и темам дисциплины;
- освоению умений прикладного и практического использования полученных знаний.

Самостоятельная работа как вид учебной работы может использоваться на лекциях и практических и лабораторных работах, а также иметь самостоятельное значение – внеаудиторная самостоятельная работа обучающихся – при подготовке к лекциям, практическим и лабораторным занятиям, а также к экзамену.

Основными видами самостоятельной работы по дисциплине являются:

- самостоятельное изучение отдельных вопросов и тем дисциплины;
- подготовка к выполнению лабораторных работ и практическим занятиям;
- подготовка к защите лабораторных работ и практических занятий, оформление отчета.

Рекомендации по работе с литературой

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта, изучается дополнительная рекомендованная литература. Литературу по курсу рекомендуется изучать в библиотеке, с использованием доступной электронной библиотечной системы или с помощью сети Интернет (источники, которые могут быть использованы без нарушения авторских прав).

10 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

При проведении занятий по дисциплине используются следующие информационные технологии:

- удаленные информационные коммуникации между студентами и преподавателем, ведущим лекционные и практические занятия, посредством информационной образовательной среды ФГБОУ ВО «РГРТУ», позволяющие осуществлять оперативный контроль графика выполнения и содержания образовательного процесса, решение организационных вопросов, консультирование;
- доступ к современным профессиональным базам данных (в том числе международным реферативным базам данных научных изданий) и информационным справочным системам;
- проведение аудиторных занятий с использованием презентаций и раздаточных материалов в электронном виде;
- выполнение студентами различных видов учебных работ с использованием лицензионного программного обеспечения, установленного на рабочих местах студента в

компьютерных классах и в помещениях для самостоятельной работы, а также для выполнения самостоятельной работы в домашних условиях.

Обучающимся по данной дисциплине предоставляется доступ к дистанционным курсам, расположенным в системе дистанционного обучения ФГБОУ ВО «РГРТУ»: «Основы алгоритмизации и объектно-ориентированное программирование» [Электронный ресурс]. URL: <http://cdo.rsreu.ru/course/view.php?id=2266>.

Система дистанционного обучения ФГБОУ ВО «РГРТУ» доступна как из внутренней информационной системы организации, так и из глобальной сети Интернет.

Перечень лицензионного программного обеспечения:

- 1) Операционная система Windows XP Professional (лицензия Microsoft DreamSpark Membership ID 700102019);
- 2) Операционная система Windows 7 (лицензия DreamSpark Membership ID 700102019);
- 3) LibreOffice 4.4 – лицензия LGPLv3;
- 4) Microsoft Office Visio (лицензия Microsoft DreamSpark Membership ID 700102019).
- 5) Интегрированная среда разработки Visual Studio C++ (лицензия DreamSpark Membership ID 700102019);
- 6) Свободная интегрированная среда разработки приложений Dev-C++. Лицензия: GNU GPL. Сайт: <http://orwelldevcpp.blogspot.ru/>

Перечень профессиональных баз данных (в том числе международным реферативным базам данных научных изданий) и информационных справочных систем:

- 1) Информационно-правовой портал ГАРАНТ.РУ [Электронный ресурс]. – URL: <http://www.garant.ru>. – Режим доступа: свободный доступ (дата обращения 02.02.2017).
- 2) Справочная правовая система КонсультантПлюс [Электронный ресурс]. – URL: <http://www.consultant.ru/online/>. – Режим доступа: свободный доступ (будние дни – 20.00 - 24.00, выходные и праздничные дни – круглосуточно) (дата обращения 02.02.2017).

11 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Для освоения дисциплины необходимы:

- 1) для проведения лекционных занятий необходима аудитория с достаточным количеством посадочных мест, соответствующая необходимым противопожарным нормам и санитарно-гигиеническим требованиям;
- 2) для проведения практических занятий и лабораторных работ необходим класс персональных компьютеров с установленными операционными системами Microsoft Windows XP (или выше) и лицензионным программным обеспечением LibreOffice, Microsoft Office Visio, Dev-C++, Visual Studio C++.
- 3) для проведения лекций аудитория должна быть оснащена проекционным оборудованием.

ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Рязанский государственный радиотехнический университет
имени В.Ф. Уткина»

КАФЕДРА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

**Б1.В.01 «Основы алгоритмизации
и объектно-ориентированное программирование»**

Направления подготовки:

02.03.01 «Математика и компьютерные науки»

ОПОП академического бакалавриата
«Математика и компьютерные науки»

Квалификация (степень) выпускника – бакалавр

Форма обучения – очная

Рязань, 2020 г.

1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимися в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации.

Промежуточная аттестация проводится в форме зачета, экзамена и защиты курсового проекта. Форма проведения зачета и экзамена - тестирование, письменный опрос по теоретическим вопросам и выполнение практического задания.

2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной:

Описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
<i>3 балла</i> (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
<i>2 балла</i> (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
<i>1 балл</i> (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
<i>0 баллов</i>	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

Описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	Критерий
<i>3 балла</i> (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
<i>2 балла</i> (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов

<i>1 балл</i> (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
<i>0 баллов</i>	выставляется студенту, который не смог ответить на вопрос

Описание критериев и шкалы оценивания практического задания:

Шкала оценивания	Критерий
<i>3 балла</i> (эталонный уровень)	Задача решена верно
<i>2 балла</i> (продвинутый уровень)	Задача решена верно, но имеются неточности в логике решения
<i>1 балл</i> (пороговый уровень)	Задача решена верно, с дополнительными наводящими вопросами преподавателя
<i>0 баллов</i>	Задача не решена

Описание критериев и шкалы оценивания курсовой работы

Шкала оценивания	Критерий
<i>Оценка «отлично»</i> (эталонный уровень)	Курсовая работа (КР) выполнена в полном объеме, нет замечаний по разработке алгоритмов и программ, работа выполнена самостоятельно, пояснительная записка к КР оформлена аккуратно, соблюдались сроки сдачи и защиты КР, при защите КР студент ответил на все предложенные вопросы
<i>Оценка «хорошо»</i> (продвинутый уровень)	Курсовая работа выполнена в полном объеме, присутствуют незначительные замечания по разработке алгоритмов и программ, проект выполнен самостоятельно, пояснительная записка к КР оформлена аккуратно, соблюдались сроки сдачи и защиты КР, при защите КР студент ответил не на все предложенные вопросы (правильных ответов не менее 75%)
<i>Оценка «удовлетворительно»</i> (пороговый уровень)	Курсовая работа выполнена в полном объеме, присутствуют ошибки при разработке алгоритмов и программ, КР выполнена самостоятельно, по оформлению пояснительной записки к КР имеются замечания, частично соблюдались сроки сдачи и защиты КР, при защите КР студент ответил не на все предложенные вопросы (правильных ответов не менее 50%)
<i>Оценка «неудовлетворительно»</i>	Курсовая работа выполнена не в полном объеме, присутствуют грубые ошибки при разработке алгоритмов и программ, КР выполнена не самостоятельно, по оформлению пояснительной записки к КР имеются замечания, не соблюдались сроки сдачи и защиты КР, при защите КР студент ответил не на все предложенные вопросы (правильных ответов менее 50%)

На промежуточную аттестацию выносятся тест, два теоретических вопроса и 1 задача. Максимально студент может набрать 12 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в

традиционную форму по системе «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно», «зачтено», «не зачтено».

Оценка «отлично» выставляется студенту, который набрал в сумме 12 баллов (выполнил все задания на эталонном уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «хорошо» выставляется студенту, который набрал в сумме от 8 до 11 баллов при условии выполнения всех заданий на уровне не ниже продвинутого. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «удовлетворительно» выставляется студенту, который набрал в сумме от 4 до 7 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «неудовлетворительно» выставляется студенту, который набрал в сумме менее 4 баллов или не выполнил все предусмотренные в течение семестра практические задания.

Оценка «зачтено» выставляется студенту, который набрал в сумме не менее 4 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «не зачтено» выставляется студенту, который набрал в сумме менее 4 баллов или не выполнил все предусмотренные в течение семестра практические задания.

3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

	<i>Контролируемые разделы (темы) дисциплины</i>	<i>Код контролируемой компетенции (или её части)</i>	Вид, метод, форма оценочного мероприятия
1	Общие принципы разработки программного обеспечения	ПК-1	Экзамен, курсовая работа
2	Основные элементы языка программирования C/C++	ПК-1	Экзамен, курсовая работа
3	Операции языка. Выражения. Математические функции	ПК-1	Экзамен, курсовая работа
4	Операторы языка C++	ПК-1	Экзамен, курсовая работа
5	Препроцессор языка C++	ПК-1	Экзамен, курсовая работа
6	Указатели, адресная арифметика	ПК-1	Экзамен, курсовая работа
7	Массивы, строки и указатели	ПК-1	Экзамен, курсовая работа
8	Функции и классы памяти. Рекурсия	ПК-1	Экзамен, курсовая работа

9	Структуры и объединения	ПК-1	Экзамен, курсовая работа
10	Функции с переменным числом параметров. Перегрузка функций.	ПК-1	Зачет
11	Потоковый ввод-вывод. Файлы	ПК-1	Зачет, курсовая работа
12	Указатели и динамические структуры данных	ПК-1	Зачет, курсовая работа
13	Простейший графический интерфейс в Visual C++.	ПК-1	Зачет, курсовая работа
14	Базовые принципы ООП	ПК-1	Зачет
15	Объекты и классы в языке C++	ПК-1	Зачет
16	Наследование. Производные классы.	ПК-1	Экзамен
17	Перегрузка операторов	ПК-1	Экзамен
18	Параметризованные классы	ПК-1	Экзамен
19	Обработка исключений.	ПК-1	Экзамен
20	Стандартная библиотека (STL).	ПК-1	Экзамен
21	Разработка объектно-ориентированного ПО	ПК-1	Экзамен

4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация в форме экзамена и зачета

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ПК-1	Способен проектировать программное обеспечение с использованием современных инструментальных средств

Типовые тестовые вопросы:

Вопрос 1

Выберите верное определение алгоритма:

- 1) Алгоритм – набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата.
- 2) Алгоритм – набор упорядоченных действий исполнителя для достижения некоторого результата.
- +3) Алгоритм – набор инструкций, описывающих порядок действий компьютера для достижения некоторого результата.
- 4) Алгоритм – набор инструкций для написания программного кода и составления схемы алгоритма при разработке программы.

Вопрос 2

Выберите все свойства алгоритма:

- +1) Массовость
- 2) Идеальность
- +3) Понятность

- +4) Дискретность
- 5) Лаконичность
- +6) Конечность
- +7) Определенность
- 8) Уникальность
- 9) Работоспособность
- +10) Эффективность.

Вопрос 3

Что такое и зачем нужна отладка?

- 1) Отладка – этап разработки компьютерной программы, на котором обнаруживают, локализуют и устраняют ошибки. Она необходима для того, чтобы пользователь мог увидеть ошибки в программе и сообщить о них разработчику.
- +2) Отладка – этап разработки компьютерной программы, на котором обнаруживают, локализуют и устраняют ошибки. Она необходима для того, чтобы разработчик мог найти ошибки в программе и устранить их.
- 3) Отладка – этап разработки компьютерной программы, на котором отлаживают совместную работу программного обеспечения.

Вопрос 4

ЕСПД – что это такое?

- +1) Единая система программной документации – комплекс государственных стандартов Российской Федерации, устанавливающих взаимосвязанные правила разработки, оформления и обращения программ и программной документации.
- 2) Единая система программной документации – комплекс сопроводительной программной документации для разрабатываемых программ.
- 3) Единая система программной документации – комплекс документации для разработчиков, в которой представлены вспомогательные материалы для программиста

Вопрос 5

Что такое тип данных?

- +1) Определяет возможный диапазон значения переменных, а также операции и функции с ними
- 2) Определяет внешний вид значений при записи в коде программы
- 3) Определяет внешний вид значений, возможный диапазон и операции с ними
- 4) Определяет внутреннее представление данных

Вопрос 6

Переменные – это:

- +1) величины, которые могут менять свое значение в процессе выполнения программы
- 2) величины, которые не могут менять своего значения в процессе выполнения программы
- 3) обозначают строки программы, на которые передается управление вовремя выполнения программы

Вопрос 7

Указатель – это переменная, которая содержит в качестве своего значения _____ другой переменной.

- 1) Индекс
- +2) Адрес
- 3) Код
- 4) Двоичное представление

Вопрос 8

Основным блоком в программе консольного приложения на языке Си является:

- 1) Самая большая функция в программе
- +2) Функция `main()`
- 3) Блок переменных и констант

Вопрос 9

В языке Си оператор присваивания имеет полную и краткую формы записи. Выберите из списка ниже верное использование данного оператора.

- +1) `a = 10;`
- +2) `b += 12.5;`
- 3) `c == 'c';`
- 4) `d =+ 7.123;`

Вопрос 10

Необходимо найти наибольшее из двух чисел. Выберите верный способ.

- +1) `max = a > b ? a : b;`
- 2) `max = a > b : a ? b;`
- 3) `max = a ? b : a : b;`

Вопрос 11

Какой из следующих операторов - оператор сравнения двух переменных?

- 1) `:=`
- 2) `equal`
- + 3) `==`
- 1) `=`

Вопрос 12

Укажите операцию, приоритет выполнения которой больше остальных

- 1) `/`
- +2) `++`
- 3) `()`
- 4) `*`
- 5) `+`

Вопрос 13

Выберите правильный синтаксис функций:

- 1) Любой_тип имя_функции (список параметров)
- 2) Возвр._тип имя_функции();

- 3) Возвр._тип имя функции() {тело функции}
- +4) Возвр._тип имя_функции (список параметров) { тело функции}

Вопрос 14

Выберите правильное утверждение для классов памяти:

- 1) Область действия регистровой переменной – вся программа
- 2) Статическая глобальная переменная ничем не отличается от статической локальной переменной
- +3) Область действия внешней переменной – вся программа
- 4) Область действия автоматической переменной – файл

Вопрос 15

В каком заголовочном файле хранятся средства для описания и использования функций с переменным числом аргументов?

- 1) stdlib.h
- 2) ctype.h
- 3) stdio.h
- +4) stdarg.h
- 5) string.h

Вопрос 16

В каких из следующих фрагментов кода верно приведена инициализация двумерного массива?

- +1) float Example1[2][2] = { {3.2, 5.6} , {1.8, 9.3} };
- 2) int Example2[][] = { {1, 2, 3, 4} , {5, 6, 7, 8} };
- +3) int Example3[][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
- 4) float Example4[3][] = { {1.4, 5.8, 6.9} ; {7.4, 5.5, 2.4} }.

Вопрос 17

Какие из следующих утверждений про объединения и структуры верны?

- 1) При объявлении объединения его нельзя инициализировать;
- 2) Для переменных типа объединение места в памяти выделяется столько, сколько нужно элементу объединения, имеющему наименьшую длину в байтах;
- +3) При объявлении можно смешивать объединения и структуры;
- +4) Каждый раз только один элемент объединения доступен для обращения.

Вопрос 18

Укажите правильное объявление указателя.

- 1) int x;
- 2) ptr x;
- +3) int *x;
- 4) int &x;

Вопрос 19

Какой из следующих записей используется операция разыменования?

- 1) a;
- +2) *a;

- 3) address(a);
- 4) &a;

Вопрос 20

Как правильно освободить память, после выполнения данного кода?

- ```
char *a; a = new char [20];
```
- 1) delete a;
  - 2) delete a[];
  - +3) delete [] a;

### Вопрос 21

*Какие функции используются для форматированного ввода-вывода?*

- +1) fscanf()
- 2) fopen()
- 3) printf()
- +4) fprintf()

### Вопрос 22

*Выберете два прототипа для блочного ввода и вывода данных.*

- 1) int fputs(char \*s, FILE \*f);
- +2) int fread(void \*ptr, int size, int n, FILE \*f);
- 3) int fscanf (FILE \*fp, const char \*fmt, ...);
- +4) int fwrite(void \*ptr, int size, int n, FILE \*f);

### Вопрос 23

*Выберете правильно записанный прототип открытия файла.*

- +1) FILE \*fopen(const char \*filename, const char \*mode);
- 2) int fprintf (FILE \*fp, char \*fmt, ...);
- 3) int \*fopen(const char \*filename, const char \*mode);
- 4) FILE \*fopen(const char \*filename);

### Вопрос 24

*Что относится к символьному и строчному вводу- выводу из нижеперечисленных прототипов?*

- 1) int fwrite(void \*ptr, int size, int n, FILE \*f);
- 2) int fread(void \*ptr, int size, int n, FILE \*f);
- +3) int fputs(char \*s, FILE \*f);
- +4) char \*fgets(char \*s, int n, FILE \*f);
- 5) int feof(FILE \* filename);
- +6) int fputc(int c, FILE\*fp);
- 7) int fscanf (FILE \*fp, const char \*fmt, ...);

### Вопрос 25

*Расположите процесс создания приложения в верном порядке:*

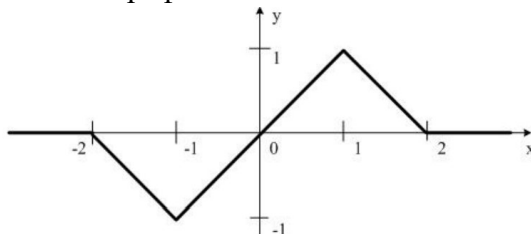
- 1) Написание программного кода, описание свойств элементов, доступных только во время работы приложения, описание реакций на событие появления окна, нажатие на кнопку и других;

- 2) Формирование окна программы - расположение необходимых элементов, задание размеров, изменение свойств;
- 3) Отладка программы.

Ответ: 2, 1, 3

### Типовые практические задания:

1. Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика.



2. **Оператор for.** Написать программу, которая выводит таблицу степеней двойки от нулевой до десятой. Ниже приведен рекомендуемый вид экрана во время работы программы.

*Таблица степеней двойки*

|       |     |
|-------|-----|
| 0     | 1   |
| 1     | 2   |
| 2     | 4   |
| 3     | 8   |
| 4     | 16  |
| 5     | 32  |
| ..... | ... |

3. **Оператор for.** Написать программу, которая выводит таблицу значений функции  $y=|x|$  Диапазон изменения аргумента от -4 до 4, шаг приращения аргумента 0,5.

4. **Оператор цикла с постусловием.** Написать программу, вычисляющую сумму и среднее арифметическое последовательности положительных чисел, которые вводятся с клавиатуры. Ниже приведен рекомендуемый вид экрана во время выполнения программы (данные, введенные пользователем, выделены полужирным шрифтом).

*Вычисление среднего арифметического последовательности положительных чисел.*

*Для завершения ввода введите ноль.*

**45**

**23**

**15**

**0**

*Введено чисел: 3*

*Сумма чисел: 83*

*Среднее арифметическое: 27.67*

5. **Оператор цикла с постусловием.** Написать программу, которая определяет минимальное число во введенной с клавиатуры последовательности положительных чисел (длина последовательности неограниченна). Ниже приведен рекомендуемый вид экрана во

время выполнения программы (данные, введенные пользователем, выделены полужирным шрифтом).

*Определение минимального числа в последовательности положительных чисел.*

*Для завершения ввода введите ноль.*

**12**

**75**

**23**

**0**

*Минимальное число: 9*

**6. Оператор цикла с предусловием.** Напишите программу, которая выводит на экран таблицу значений функции  $y = 2x^2 - 5x - 8$  в диапазоне от -4 до 4. Шаг изменения аргумента 0,5.

**7. Оператор цикла с предусловием.** Напишите программу, которая проверяет, является ли введенное пользователем целое число простым. Рекомендуемый вид экрана во время выполнения программы приведен ниже. Данные, введенные пользователем, выделены полужирным шрифтом.

*Введите целое число и нажмите <Enter>*

**45**

*45 — не простое число.*

**8.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить произведение элементов массива, расположенных между максимальным и минимальным элементами.

**9.** В одномерном массиве, состоящем из  $n$  целых элементов, вычислить сумму элементов массива, расположенных между первым и последним нулевыми элементами.

**10.** Дана целочисленная квадратная матрица. Определить произведение элементов в тех строках, которые не содержат отрицательных элементов (оформить в виде функции);

**11.** Дана целочисленная квадратная матрица. Определить сумму элементов в тех строках, которые содержат нули (оформить в виде функции);

**12.** Написать программу, сортирующую массив (порядок и метод сортировки выбрать самостоятельно).

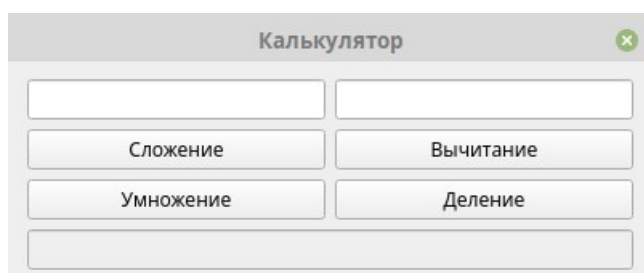
**13.** Сведения о деталях, хранящихся на складе, содержат следующие атрибуты: название, количество, стоимость одной детали. Вывести информацию о детали, суммарная стоимость для которой максимальна.

**14.** Написать программу, которая считывает текст из файла и выводит на экран только предложения, содержащие введенное с клавиатуры слово.

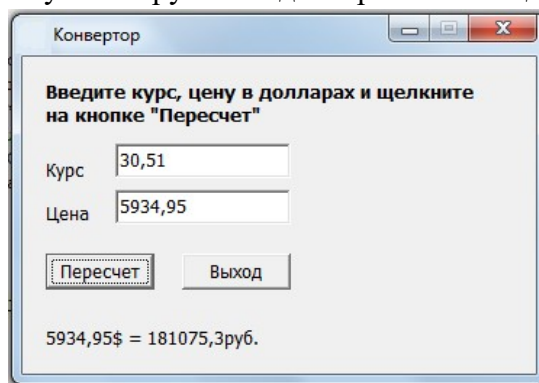
**15.** Написать программу, которая считывает текст из файла и выводит на экран только цитаты, то есть предложения, заключенные в кавычки.

**16.** Написать программу, которая считывает текст из файла и выводит на экран только предложения, состоящие из заданного количества слов. Контрольные вопросы

**17.** Необходимо разработать программу – калькулятор, осуществляющую умножение двух введенных чисел. Вид формы:



18. Необходимо разработать программу – конвертор, осуществляющую конвертирование введенной суммы в рублях в доллары. Рекомендуемый вид формы:



19. Написать программу вычисления величины дохода по вкладу. Процентная ставка (% годовых) и время хранения (дней) задаются во время работы программы. Рекомендуемый вид формы как задании 2.

20. Написать программу вычисления пересчета величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах. Рекомендуемый вид формы как задании 2.

Задание 21. Постройте иерархию классов по заданному примеру. Что будет выведено на экран в результате выполнения программы?

```
#include <iostream.h>
#include <string.h>
class book {
 /* описание классов */
};
int main(void) {
 /* описание программы */
}
```

Задание 22. Постройте иерархию классов по заданному примеру. Что будет выведено на экран в результате выполнения программы?

```
#include <iostream.h>
#include <string.h>
class A {
 /* описание классов */
};
int main(void) {
 /* описание программы */
}
```

Задание 23. Какие функции классов А и В могут быть виртуальными? Опишите их.

```
#include <iostream.h>
#include <string.h>
class A {
 /* описание класса A */
};
class B: public A {
 /* описание класса B */
};
```

Задание 24. Опишите класс triangle как абстрактный, используя виртуальную функцию show.

```
#include <iostream.h>
#include <graphics.h>
class triangle {
 /* описание класса triangle */
};
```

Задание 25. Создайте класс C, который наследовал бы свойства классов A и B и измените описания классов A и B так, чтобы класс W стал бы одним общим экземпляром базового класса.

```
#include <iostream.h>
#include <string.h>
class W {
 /* описание классов */
};
```

Задание 26. Найдите ошибки, связанные с наследованием атрибутов классов.

```
#include <iostream.h>
class BASE {
 /* описание классов */
};
int main(void) {
 /* описание программы */
}
```

Задание 27. Исправьте ошибку путем определения виртуального деструктора.

```
class book {
 /* описание классов */
};
int main(void) {
 /* описание программы */
}
```

Задание 28. Определите класс STACK как шаблон. Как будут использованы объекты класса STACK в программе?

```
#include <iostream.h>
#include <string.h>
class STACK {
 /* описание класса */
};
```

```
int main(void) {
 /* описание программы */
}
```

Задание 29. Определите функцию min как шаблон. Что будет выведено на экран в результате выполнения программы?

```
#include <iostream.h>
#include <string.h>
int min(int x, int y) {
 /* описание функции */
}
int main(void) {
 /* описание программы */
}
```

Задание 30. Переопределите параметризованный шаблон функции min так, чтобы программа выполнялась без ошибки.

```
#include <iostream.h>
#include <string.h>
template <class Time> Time min(Time x, Time y) {
 /* описание функции */
};
int main(void) {
 /* описание программы */
}
```

Задание 31. Добавьте в шаблон класса дополнительный аргумент, определяющий ограничение на максимальный размер массива, и модифицируйте программу.

```
#include <iostream.h>
#include <string.h>
template <class Time> class array {
 /* описание класса */
};
```

Задание 32. Модифицируйте программу так, чтобы она работала с массивами действительного типа.

```
#include <iostream.h>
#include <classlib\arrays.h>
int main(void) {
 /* описание программы */
}
```

Задание 33. Добавьте в программу пропущенные команды для обработчика исключений на неверный индекс массива.

```
#include <iostream.h>
#include <string.h>
class string {
 /* описание класса */
};
int main(void) {
```



```

..... /* описание программы */
}

```

Задание 34. Добавьте в программу обработчики исключений на неверный размер и неверное значение индекса массива.

```

#include <iostream.h>
#include <string.h>
class sz { };
class string {
..... /* описание класса */
};

int main(void) {
..... /* описание программы */
}

```

### Типовые теоретические вопросы:

- 1) Метод проектирования программных средств. Основные этапы.
- 2) Тип данных. Классификация типов данных.
- 3) Циклы с известным и неизвестным числом повторений.
- 4) Команда ветвления if. Оператор swith.
- 5) Массивы.
- 6) Понятие блока. Область действия и время жизни переменных. Особенности локальных переменных. Локальные статические переменные. Особенности глобальных переменных.
- 7) Операции в Си.
- 8) Указатели в Си. Инициализация указателей. Операции с указателями.
- 9) Понятие потока. Организация ввода-вывода в стиле Си.
- 10) Массивы и строки в Си. Инициализация.
- 11) Функции в языке Си. Методы передачи аргументов в функцию.
- 12) Прототипы функции. Функция main().
- 13) Директивы препроцессора include и define.
- 14) Классы памяти.
- 15) Динамическое распределение памяти в стиле С. Функция malloc().
- 16) Динамическое распределение памяти в стиле C++. Операторы new и delete.
- 17) Указатели и массивы в Си. Одномерные динамические массивы.
- 18) Структуры и объединения.
- 19) Работа с файлами в языке Си. Символьный ввод-вывод. Строчный ввод-вывод.
- 20) Динамические линейные однонаправленные списки. Стек. Отличия организации стека и списка.
- 21) Что такое метод? Каковы особенности описания методов?
- 22) Что такое свойство объекта, каким образом можно его изменять?
- 23) Что такое событие? Каково назначение обработчика событий?
- 24) Каково назначение сообщений?
- 25) Перечислите основные компоненты окна среды Visual C++ и укажите их назначение.
- 26) Как разместить компонент на форме?

- 27) Какими способами можно изменять свойства компонента?
- 28) Каково назначение обработчиков событий? Каким образом можно инициировать создание процедуры – обработчика события?
- 29) Опишите назначение компонента MainMenu и порядок использования конструктора меню.
- 30) Базовые принципы ООП.
- 31) Правила использования базовых и производных классов.
- 32) Конструкторы и деструкторы. Правила конструирования базовых и производных классов.
- 33) Виртуальные функции. Правила использования виртуальных функций.
- 34) Абстрактные классы. Использование абстрактных классов в программах.
- 35) Множественное наследование.
- 36) Управление доступом к компонентам класса. Правила использования атрибутов.
- 37) Виртуальные деструкторы.
- 38) Параметризованные классы (шаблоны).
- 39) Передача типа в функцию в качестве аргумента. Шаблоны функций.
- 40) Переопределение шаблонов функций. Правила выбора функций.
- 41) Передача дополнительных аргументов в шаблоны.
- 42) Обработка исключений.
- 43) Определение типа исключений.

#### 4.2. Промежуточная аттестация в форме курсовой работы

| Код компетенции | Результаты освоения ОПОП<br>Содержание компетенций                                                   |
|-----------------|------------------------------------------------------------------------------------------------------|
| ПК-1            | Способен проектировать программное обеспечение с использованием современных инструментальных средств |

#### Типовое задание для курсовой работы по дисциплине:

Целью курсового проектирования являются разработка и отладка приложения с использованием динамических структур данных, написанного на алгоритмическом языке программирования C++ в среде визуального программирования Visual C++.

Тема курсовой работы: «Разработка приложения с использованием динамических структур данных».

Курсовая работа (КР) представляет собой самостоятельную работу по заданной теме. Работа предполагает:

- домашнюю внеаудиторную подготовку;
- выполнение практической части дома или в классе персональных компьютеров на кафедре;
- консультации по КР;
- предъявление промежуточных результатов для проверки и контроля хода курсового проектирования;
- написание и оформление пояснительной записки;
- сдачу и защиту КР в сроки согласно учебному графику.

### **Исходные данные**

- Тип компьютера IBM PC совместимый.
- Операционная система Windows XP (не ниже).
- Язык программирования C/C++.
- Среда программирования: Microsoft Visual Studio.
- Текст задания согласно варианту.

### **Общие требования**

Все варианты заданий связаны с разработкой таблицы данных с использованием линейных однонаправленных списков.

Разрабатываемая программа должна обязательно выполнять следующие запросы:

- заполнение пустой таблицы (создание списка);
- сохранение таблицы (списка) в файле;
- чтение таблицы (списка) из файла;
- вывод таблицы на экран;
- добавление элементов в таблицу (в список);
- удаление элементов из таблицы (из списка);
- а также все запросы, которые указаны в индивидуальном задании.

Вызовы запросов должны осуществляться через систему меню с использованием средств визуального программирования. Необходимо предусмотреть контроль ошибок пользователя при вводе данных. Результаты некоторых запросов (по согласованию с преподавателем на этапе уточнения технического задания) должны выводиться в виде графиков или диаграмм. Приложение обязательно должно иметь заставку.

Все элементарные действия (заполнение списка, запись элемента в список и т.д.) должны быть оформлены в виде подпрограмм, а все (или некоторые) объявления и подпрограммы должны быть оформлены в виде модуля (модулей).

### ***Перечень индивидуальных заданий:***

#### **Вариант № 1. Абоненты библиотеки**

Информация об абонентах библиотеки следующая:

- номер читательского билета;
- ФИО;
- год рождения;
- пол;
- подразделение (кафедра, номер группы);
- должность;
- отметка о перерегистрации;
- имеются книги на срок;
- дата возврата книг.

Написать программу, которая выполняет следующие запросы:

- вывод информации об абоненте по номеру читательского билета;
- упорядочение таблицы по ФИО;
- вывод списка абонентов-должников определенного факультета;
- вывод списка абонентов, которые не прошли перерегистрацию;
- вывод процентного соотношения сотрудников и студентов среди абонентов.

## **Вариант № 2. Анкета студента**

Анкета студента имеет следующие пункты:

- ФИО;
- пол;
- факультет;
- номер группы;
- адрес постоянного проживания;
- сведения о получении стипендии;
- вид спорта.

Написать программу, которая выполняет следующие запросы:

- вывод списка студентов определенной группы по алфавиту;
- вывод списка студентов на факультете, которые занимаются спортом;
- вывод списка иногородних студентов на факультете;
- вывод списка студентов на факультете, которые не получают стипендию;
- вывод процентного соотношения мужчин и женщин на факультете.

## **Вариант № 3. Анкета абитуриента**

Анкета абитуриента имеет следующие пункты:

- ФИО;
- адрес постоянного проживания;
- льгота (инвалидность, сирота, целевой набор);
- баллы по ЕГ (математика, физика, русский язык);
- номер направления. По каждой специальности определен проходной балл.

Написать программу, которая выполняет следующие запросы:

- упорядочение таблицы по ФИО абитуриента;
- определение процента иногородних абитуриентов;
- вывод списка абитуриентов, которые поступили на определенное направление;
- вывод номеров направлений в порядке убывания «популярности»;
- вывод процентного соотношения льготников и абитуриентов, которые поступают в вуз на общих основаниях.

Полный список индивидуальных заданий находится в методических указаниях к КР. Дублирование тем для индивидуального исследования в пределах одной учебной группы не допускается.

Защита курсовой работы назначается по итогам проверки предоставленной пояснительной записки, оформленной в соответствии с требованиями и разработанного приложения. Защита осуществляется в форме ответов на вопросы преподавателя.

## **Требования к содержанию пояснительной записки:**

Титульный лист

Задание

Содержание

Введение

1. Постановка задачи
2. Разработка графического интерфейса.
3. Разработка алгоритмов
4. Разработка программы

5. Экспериментальная проверка программы

6. Руководство оператора

Заключение

Список использованных источников

Приложение

**Типовые вопросы на защите курсового проекта:**

- 1) *Как разрабатывался графический интерфейс для вашего приложения?*
- 2) *Какие визуальные и не визуальные компоненты вы использовали?*
- 3) *Есть ли в вашем приложении модальные окна? Для каких целей вы их используете?*
- 4) *Как создавалась заставка для вашего приложения?*
- 5) Как создается однонаправленный динамический список?
- 6) Как в списке происходит поиск данных?
- 7) Как отображаются в вашем приложении табличные данные?
- 8) Как осуществляется чтение и запись данных
- 9) Опишите назначение компонента MainMenu. Как вы используете эту компоненту в проекте?
- 10) Объясните алгоритм(ы) различных функций.
- 11) Как эти алгоритмы реализуются на языке C++.

Составил

к.т.н., доц. кафедры ЭВМ

Елесина С.И.

Зав. кафедрой ЭВМ

д.т.н., проф.

Костров Б.В.