

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
"Казанский (Приволжский) федеральный университет"  
Институт вычислительной математики и информационных технологий



подписано электронно-цифровой подписью

### Программа дисциплины

Практикум на электронно-вычислительных машинах Б1.В.ОД.4

Направление подготовки: 02.03.02 - Фундаментальная информатика и информационные технологии

Профиль подготовки: Системный анализ и информационные технологии

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

**Автор(ы):**

Тагиров Р.Р.

**Рецензент(ы):**

Андрианова А.А.

**СОГЛАСОВАНО:**

Заведующий(ая) кафедрой: Латыпов Р. Х.

Протокол заседания кафедры No \_\_\_\_ от " \_\_\_\_ " \_\_\_\_\_ 201\_\_ г

Учебно-методическая комиссия Института вычислительной математики и информационных технологий:

Протокол заседания УМК No \_\_\_\_ от " \_\_\_\_ " \_\_\_\_\_ 201\_\_ г

Регистрационный No 944719

## Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) старший преподаватель, б/с Тагиров Р.Р. кафедра системного анализа и информационных технологий отделение фундаментальной информатики и информационных технологий , Ravil.Tagirov@kpfu.ru

### 1. Цели освоения дисциплины

Дисциплина "Практикум на ЭВМ" в 1 и 2 семестрах предполагает изучение основ разработки алгоритмов и реализации программ с помощью объектно-ориентированных средств. Примеры демонстрируются средствами языка C++.

3 семестр предназначен для освоения студентами инструментальных средств разработки программ. Программа дисциплины содержит список тем заданий для выполнения в дисплейном классе.

### 2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел "Б1.В.ОД.4 Дисциплины (модули)" основной образовательной программы 02.03.02 Фундаментальная информатика и информационные технологии и относится к обязательным дисциплинам. Осваивается на 1, 2 курсах, 1, 2, 3 семестры.

Данная дисциплина читается на 1 и 2 курсе в 1, 2 и 3 семестре для студентов обучающихся по направлению "Фундаментальная информатика и информационные технологии".

### 3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОПК-2 (профессиональные компетенции)	способностью применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий

В результате освоения дисциплины студент:

1. должен знать:

- проблематику создания алгоритмов решения задач и описания их с помощью языков программирования
- состав и назначение основных этапов решения задач на ЭВМ - от разработки алгоритма до отладки и документирования программ

2. должен уметь:

- ориентироваться в различных средах программирования, уметь использовать готовые библиотеки функций
- ориентироваться в составе и назначении средств математического обеспечения ЭВМ

3. должен владеть:

- теоретическими знаниями о принципах построения программ, их отладки, модификации и сопровождения
- навыками использования современных методологий и технологий создания программ и комплексов

- теоретическими знаниями по темам курса "Информатика" и уметь их применять для практического решения задач
- навыками алгоритмизации и применения стандартных математических методов и математического обеспечения ЭВМ для решения различных задач и отладки программ

4. должен демонстрировать способность и готовность:

- применять полученные знания в своей профессиональной деятельности

#### 4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет 6 зачетных(ые) единиц(ы) 216 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 1 семестре; зачет во 2 семестре; зачет в 3 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

#### 4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

##### Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи- ческие занятия	Лабора- торные работы	
1.	Тема 1. Понятие алгоритма и его свойства	1		0	0	4	
2.	Тема 2. Элементарные объекты и их основные типы	1		0	0	4	
3.	Тема 3. Структура простой программы на языке С (C++).	1		0	0	4	
4.	Тема 4. Описание и определение функций	1		0	0	4	Контрольная работа
5.	Тема 5. Массивы	1		0	0	4	Контрольная работа
6.	Тема 6. Указатели и адреса	1		0	0	4	Контрольная работа
7.	Тема 7. Структуры и объединения	1		0	0	4	
8.	Тема 8. Препроцессор	1		0	0	4	
9.	Тема 9. Файловый ввод-вывод	1		0	0	4	

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи- ческие занятия	Лабора- торные работы	
10.	Тема 10. Динамические линейные структуры данных	2		0	0	12	Контрольная работа
11.	Тема 11. Нелинейные структуры данных	2		0	0	12	Контрольная работа
12.	Тема 12. Объектно-ориентированное программирование	2		0	0	12	Контрольная работа
13.	Тема 13. Основы архитектуры MVC	3		0	0	12	Контрольная работа
14.	Тема 14. Основы разработки сетевых приложений	3		0	0	12	Контрольная работа
15.	Тема 15. Основы методологии функционального программирования	3		0	0	12	Контрольная работа
.	Тема . Итоговая форма контроля	1		0	0	0	Зачет
.	Тема . Итоговая форма контроля	2		0	0	0	Зачет
.	Тема . Итоговая форма контроля	3		0	0	0	Зачет
	Итого			0	0	108	

## 4.2 Содержание дисциплины

### Тема 1. Понятие алгоритма и его свойства

#### **лабораторная работа (4 часа(ов)):**

Понятие алгоритма и его свойства Типы алгоритмов - линейные, разветвляющиеся, циклические. Формы записи алгоритмов - словесное описание, блок-схемы, диаграммы, программы на языках программирования. Этапы разработки программ - постановка задачи, разработка алгоритма, реализация программы, запуск на тестовых примерах, поиск и исправление ошибок - отладка.

### Тема 2. Элементарные объекты и их основные типы

#### **лабораторная работа (4 часа(ов)):**

Элементарные объекты и их основные типы (целые, вещественные, символьные, логические, длинные и большие целые, вещественные повышенной точности), элементарные операции над объектами определенных типов. Выделение памяти объектам и начальная инициализация. Основы логики высказываний, логические операции и выражения, операции сравнения.

### Тема 3. Структура простой программы на языке C (C++).

#### **лабораторная работа (4 часа(ов)):**

Структура простой программы на языке C (C++). Стандартный заголовок и главная функция. Описание объектов - глобальные и локальные. Обнуление глобальных объектов. Основные выражения и типы операторов. Простые и составные операторы. Простейшие средства ввода данных и вывода результатов через клавиатуру и дисплей.

#### **Тема 4. Описание и определение функций**

##### ***лабораторная работа (4 часа(ов)):***

Описание (объявление прототипа) и определение функций. Варианты передачи параметров. Параметры по умолчанию. Возврат результатов через оператор возврата и через параметры. Изменяемые параметры. Рекурсия (прямая и косвенная) и способы её реализации. Вызовы и передача значений параметров, пропуск последних параметров. Перегрузка имен.

#### **Тема 5. Массивы**

##### ***лабораторная работа (4 часа(ов)):***

Массивы, их описание и размещение в оперативной памяти. Работа с отдельными элементами - операции доступа. Многомерные массивы. Символьные строки и стандартное соглашение о строках. Описание строк, их ввод-вывод и обработка. Функции обработки строк. Стандартные функции обработки строк - длина, копирование, конкатенация и сравнение.

#### **Тема 6. Указатели и адреса**

##### ***лабораторная работа (4 часа(ов)):***

Указатели и адреса. Ссылки. Операции над адресами - получение адреса, доступ к значению по адресу, адресная арифметика. Получение динамической памяти и освобождение. Связь адресов, указателей и массивов. Основные действия над массивами - ввод-вывод, сортировка, линейный и двоичный поиск. Оценка сложности алгоритмов с массивами.

#### **Тема 7. Структуры и объединения**

##### ***лабораторная работа (4 часа(ов)):***

Структуры и объединения. Аналогии и отличия от массивов. Описание и использование полей (элементов). Члены-переменные и члены-функции, начальная инициализация через функцию-конструктор. Операции доступа к элементам через объект и через указатель. Операции над составными объектами. Использование структур внутри структур.

#### **Тема 8. Препроцессор**

##### ***лабораторная работа (4 часа(ов)):***

Препроцессор. Этапы обработки текста программы в среде программирования - препроцессинг, компиляция, редактирование связей, получение выполняемого модуля. Использование объектных и динамически загружаемых библиотек. Описание переменных препроцессора, присваивание им значений и замена в тексте. Макросы - аналоги функций.

#### **Тема 9. Файловый ввод-вывод**

##### ***лабораторная работа (4 часа(ов)):***

Файловый ввод-вывод. Форматный ввод-вывод. Функции ввода-вывода в стиле Си - обработка символов, строк, блоков байтов. Операции и методы в стиле Си++. Описание файлов для ввода, вывода, для обновления. Прямой доступ к памяти. Определение текущей позиции в файле и ее изменения при вводе-выводе или без ввода-вывода.

#### **Тема 10. Динамические линейные структуры данных**

##### ***лабораторная работа (12 часа(ов)):***

Понятие односвязного списка: операции просмотра, вставки, удаления элементов списка. Понятие двусвязного списка: операции просмотра, вставки, удаления элементов списка. Понятие стека и очереди как частных случаев линейных структур данных (операции добавления и удаления элементов). Типовые примеры использования стеков и очередей в алгоритмах. Разбор выражений как пример использования стеков и очередей в алгоритмах.

#### **Тема 11. Нелинейные структуры данных**

##### ***лабораторная работа (12 часа(ов)):***

Понятие дерева сортировки: операции обхода элементов дерева, вставки и удаления элементов из дерева. Использование рекурсивных и нерекурсивных реализаций операций с деревом. Понятия сбалансированного дерева и алгоритм балансировки дерева. Понятие графа. Способы хранения графа в программе. Алгоритмы обхода графов в ширину и в глубину. Алгоритм Дейкстры для поиска кратчайшего пути в графе и интерпретация его результатов. Алгоритм Флойда для поиска всех кратчайших путей в графе и интерпретация его результатов. Алгоритм Крускала поиска каркаса графа минимального веса. Алгоритм Прима поиска каркаса графа минимального веса. Алгоритм проверки наличия цикла в графе.

## **Тема 12. Объектно-ориентированное программирование**

### **лабораторная работа (12 часа(ов)):**

Понятие класса как способ описания модели взаимодействующих в программе объектов. Инкапсуляция как способ описания объекта как "черного ящика" для последующего использования. Механизмы поддержания инкапсуляции (конструкторы, деструкторы, контролируемый доступ к переменным класса, генерация и обработка исключений). Особенности реализации классов с выделением памяти - необходимость определения конструктора копирования и оператора присваивания. Полиморфизм раннего связывания посредством перегрузки функций и операций. Перегрузка конструкторов. Особенности переопределения операций с разным порядком операндов. Особенности переопределения операций ввода и вывода. Шаблоны классов и функций для генерации сходных структур данных и алгоритмов их поддержки. Библиотека стандартных шаблонов STL как набор объектно-ориентированных инструментов стандартных структур данных.

## **Тема 13. Основы архитектуры MVC**

### **лабораторная работа (12 часа(ов)):**

Архитектура "Модель-Представление-Контроллер" как основа построения гибких приложений. Роль каждой из подсистем в работе всей системы. Принципы использования интерфейсов и виртуальных функций для обеспечения гибкого взаимодействия между подсистемами. Демонстрация принципа инкапсуляции на уровне взаимодействия подсистем.

## **Тема 14. Основы разработки сетевых приложений**

### **лабораторная работа (12 часа(ов)):**

Клиент-серверная архитектура приложений. Стек протоколов OSI. Взаимодействие с помощью протоколов TCP и UDP. Алгоритм функционирования серверного приложения: прослушивание порта, формирование сокета, обеспечение параллельной обработки диалогов с несколькими клиентами. Алгоритм функционирования клиентского приложения: подключение к серверу, прием и отсылка сообщений. Сериализация как способ обмена сложной структурированной информацией между клиентом и сервером.

## **Тема 15. Основы методологии функционального программирования**

### **лабораторная работа (12 часа(ов)):**

Основные принципы функционального программирования. Описание функции как переменной. Способ описание функции в лямбда-выражения. Использование смешанного стиля программирования - внедрение лямбда-выражений в нефункциональные языки. Особенности передачи параметров. Особенности написания рекурсивных функций в виде лямбда-выражений.

## **4.3 Структура и содержание самостоятельной работы дисциплины (модуля)**

N	Раздел дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
4.	Тема 4. Описание и определение функций	1		подготовка к контрольной работе	12	Контрольная работа

№	Раздел дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
5.	Тема 5. Массивы	1		подготовка к контрольной работе	12	Контрольная работа
6.	Тема 6. Указатели и адреса	1		подготовка к контрольной работе	12	Контрольная работа
10.	Тема 10. Динамические линейные структуры данных	2		подготовка к контрольной работе	12	Контрольная работа
11.	Тема 11. Нелинейные структуры данных	2		подготовка к контрольной работе	12	Контрольная работа
12.	Тема 12. Объектно-ориентированное программирование			подготовка к контрольной работе	12	Контрольная работа
13.	Тема 13. Основы архитектуры MVC	3		подготовка к контрольной работе	12	Контрольная работа
14.	Тема 14. Основы разработки сетевых приложений	3		подготовка к контрольной работе	12	Контрольная работа
15.	Тема 15. Основы методологии функционального программирования	3		подготовка к контрольной работе	12	Контрольная работа
	Итого				108	

### 5. Образовательные технологии, включая интерактивные формы обучения

Обучение происходит в форме лабораторных занятий, а также самостоятельной работы студентов. Курс является практическим дополнением для дисциплин 'Основы программирования' и 'Языки программирования'. Главная цель курса - получение практических навыков программирования в различных современных стилях и технологиях. Лабораторные занятия проходят в интерактивной форме

обсуждения решения различных задач или в активной форме самостоятельного решения задач студентами. Каждый студент должен решить как можно больше задач, чтобы навыки программирования стали практически интуитивными. На занятиях объясняются основные приемы и синтаксические конструкции языка. Важно сразу задавать вопросы в случаях непонимания, уметь модифицировать

программы, не бояться появления ошибок, так как их поиск и анализ позволяет глубже осознать процессы, которые происходят при выполнении программ.

Главный принцип самостоятельной работы - решить как можно больше задач, так как прикладные навыки программирования являются важной базой для последующего изучения информационных технологий и изъяснение с компьютером на его языке должно стать своеобразным рефлексом ИТ-специалиста. В целях систематизации знаний каждому студенту рекомендуется сделать справочник типовых программных кодов и справочник своих типовых ошибок. Это позволит легче находить приемы для решения задачи и быстрее отлаживать свои программы.

Подготовка к контрольным работам подразумевает решение большого количества задач на темы контрольной работы для формирования 'автоматических' навыков программирования. Также имеет смысл систематизировать и детально рассмотреть те компоненты собственных справочников типовых программных кодов и типовых ошибок, которые студент сделал по теме контрольной работы. Это поможет систематизировать полученные знания.

Зачеты имеют практическую форму, позволяя продемонстрировать за короткий промежуток времени знания и навыки в области алгоритмизации и программирования. Поэтому важно тренироваться решать задачи на время, особое внимание уделять своим типовым ошибкам, внимательности при отладке программ. В целом, по каждой теме рекомендуется при подготовке к зачету решить не менее 3 задач.

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

### **Тема 1. Понятие алгоритма и его свойства**

### **Тема 2. Элементарные объекты и их основные типы**

### **Тема 3. Структура простой программы на языке C (C++).**

### **Тема 4. Описание и определение функций**

Контрольная работа , примерные вопросы:

1. Написать и протестировать функцию проверки того, что прямоугольная матрица имеет строку, состоящую только из нулевых элементов.
2. Написать и протестировать функцию получения количества столбцов прямоугольной матрицы, имеющих сумму элементов, равную 0.
3. Написать и протестировать функцию получения номера строки прямоугольной матрицы с максимальным количеством нулевых элементов.
4. Написать и протестировать функцию получения количества строк, в которых есть нулевые элементы.
5. Написать и протестировать функцию получения номера столбца прямоугольной матрицы, в котором нет нулевых элементов.
6. Написать и протестировать функцию получения номера столбца прямоугольной матрицы, в котором элементы образуют возрастающую последовательность.
7. Написать и протестировать функцию получения номера строки прямоугольной матрицы, в которой все элементы одинаковы.
8. Написать и протестировать функцию получения номера столбца прямоугольной матрицы с максимальным количеством ненулевых элементов.
9. Написать и протестировать функцию получения номера последней по порядку строки прямоугольной матрицы, в которой нет нулевых элементов.
10. Написать и протестировать функцию получения номера последнего по порядку столбца прямоугольной матрицы, в котором сумма элементов отрицательна.

### **Тема 5. Массивы**

Контрольная работа , примерные вопросы:

1. Написать и протестировать рекурсивную функцию, которая проверяет отсутствие четных чисел в массиве целых чисел. 2. Написать и протестировать рекурсивную функцию, которая находит количество элементов массива целых чисел, в записи которых есть цифра 8. 3. Написать и протестировать рекурсивную функцию, которая определяет номер последнего вхождения в целочисленный массив заданного элемента. Если такого элемента нет, должна быть возвращена -1. 4. Написать и протестировать рекурсивную функцию, которая определяет номер первого вхождения заданного элемента в целочисленный массив. Если такого элемента нет, должна быть возвращена -1. 5. Написать и протестировать рекурсивную функцию, которая определяет номер последнего отрицательного элемента в целочисленном массиве. Если такого элемента нет, должна быть возвращена -1. 6. Написать и протестировать рекурсивную функцию, которая определяет количество локальных максимумов целочисленного массива (локальный максимум больше всех своих соседей). 7. Написать и протестировать рекурсивную функцию, которая определяет значение первого по порядку локального минимума целочисленного массива (локальный минимум меньше всех своих соседей). 8. Написать и протестировать рекурсивную функцию, которая определяет порядковый номер последнего по порядку локального максимума целочисленного массива (локальный максимум больше всех своих соседей). 9. Написать и протестировать рекурсивную функцию, которая определяет самый большой по значению локальный минимум в целочисленном массиве (локальный минимум меньше всех своих соседей). 10. Написать и протестировать рекурсивную функцию, которая определяет является ли целочисленный массив "пилообразным" (массив образует "пилу", если каждый его элемент является либо локальным минимумом, либо локальным максимумом (локальный минимум меньше всех своих соседей, локальный максимум 0 больше всех своих соседей)).

#### **Тема 6. Указатели и адреса**

Контрольная работа , примерные вопросы:

1. Дана символьная строка. Написать и протестировать функцию получения слова максимальной длины, начинающегося и заканчивающегося одной и той же буквой. 2. Дана символьная строка. Написать и протестировать функцию получения количества слов, состоящих более чем из 5 букв. 3. Дана символьная строка. Написать и протестировать функцию, которая распечатывает все слова, в которых есть буквы 'a' или 'b'. 4. Дана символьная строка. Написать и протестировать функцию получения слова максимальной длины из тех, в записи которых используется только одна буква (любая). 5. Дана символьная строка. Написать и протестировать функцию получения количества слов, длина которых не превышает 5 букв и которые начинаются с буквы 'a'. 6. Дана символьная строка. Написать и протестировать функцию получения порядкового номера слова максимальной длины. 7. Дана символьная строка. Написать и протестировать функцию получения количества слов, в записи которых нет буквы 'a'. 8. Дана символьная строка. Написать и протестировать функцию получения порядкового номера первого по порядку слова, которое является симметричным. 9. Дана символьная строка. Написать и протестировать функцию получения номера первого по порядку слова, в котором есть буква 'a'. 10. Дана символьная строка. Написать и протестировать функцию получения количества слов минимальной длины.

#### **Тема 7. Структуры и объединения**

#### **Тема 8. Препроцессор**

#### **Тема 9. Файловый ввод-вывод**

#### **Тема 10. Динамические линейные структуры данных**

Контрольная работа , примерные вопросы:

1. Функция подсчета количества четных элементов в односвязном списке. 2. Функция дублирования максимального элемента односвязного списка. 3. Функция реверса односвязного списка. 4. Функция сортировки односвязного списка. 5. Функция удаления максимального элемента из односвязного списка. 6. Функция подсчета количества четных элементов в двусвязном списке. 7. Функция дублирования максимального элемента двусвязного списка. 8. Функция реверса двусвязного списка. 9. Функция сортировки двусвязного списка. 10. Функция удаления максимального элемента из двусвязного списка.

#### **Тема 11. Нелинейные структуры данных**

Контрольная работа , примерные вопросы:

1. Функция сравнения структур двух двоичных деревьев. 2. Функция получения глубины двоичного дерева. 3. Функция подсчета количества узлов двоичного дерева, расположенных на заданном уровне. 4. Функция получения общего количества узлов двоичного дерева. 5. Функция получения максимального четного элемента двоичного дерева. 6. Функция проверки связности графа. 7. Функция проверки того, что заданная вершина является истоком графа. 8. Функция проверки того, что заданная вершина является стоком графа. 9. Функция проверки того, что граф является деревом. 10. Функция обхода графа в ширину. 11. Функция обхода графа в глубину.

## **Тема 12. Объектно-ориентированное программирование**

Контрольная работа , примерные вопросы:

1. Описать класс "предметный указатель". Каждый компонент указателя содержит слово и номера страниц, на которых это слово встречается. Количество номеров страниц, относящихся к одному слову, от 1 до 10. Предусмотреть возможность формирования указателя с клавиатуры и из файла, печати предметного указателя, сохранения в файл, вывода номеров страниц для заданного слова, добавления и удаления элемента из указателя. 2. Описать класс "адресная книжка". Каждая запись в книжке содержит имя адресата, дату рождения и список номеров телефона (домашнего, мобильного и пр. - всего не более 5 номеров). Предусмотреть возможность формирования адресной книжки с клавиатуры и из файла, печати адресной книжки, поиска записи по какому-либо признаку (фамилии, дате рождения или номеру телефона), добавления и удаления записей, сохранения в файл. 3. Описать класс "каталог библиотеки". Каждая запись каталога содержит информацию о книге - название, автор, количество экземпляров, количество экземпляров "на руках". Предусмотреть возможность формирования каталога с клавиатуры и из файла, печати каталога, сохранения в файл, поиска книги по какому-либо признаку (например, автору или названию), добавления книг в библиотеку, удаления книг из нее, фиксации получения или возврата книги читателем. 4. Описать класс "записная книжка". Каждая запись в книжке содержит дату, время и описание (информацию о назначенной встрече, каком-то деле и пр.). Предусмотреть возможность формирования записной книжки с клавиатуры и из файла, печати записной книжки, поиска записи по какому-либо признаку (по описанию, по дате и времени), добавления и удаления записей, сохранения в файл. 5. Описать класс "расписание занятий". Каждая запись содержит день недели, время, название учебной дисциплины, аудиторию. Предусмотреть возможность формирования расписания с клавиатуры и из файла, печати всего расписания и расписания на конкретный день (печать должна быть осуществлена в хронологическом порядке), добавления и удаления записей, сохранения в файл. 6. Описать класс "школьный дневник". Каждая запись дневника содержит дату, название предмета, домашнее задание. Предусмотреть возможность формирования дневника с клавиатуры и из файла, печати всего содержимого дневника, домашних заданий, полученных в конкретный день, или в конкретный день по конкретному предмету, добавления и удаления записей, сохранения в файл. 7. Описать класс "оценки в школьном дневнике". Каждая запись дневника содержит дату, название предмета, полученную оценку. Предусмотреть возможность формирования дневника с клавиатуры и из файла, печати всего содержимого дневника, оценок, полученных в конкретный день, или в конкретный день по конкретному предмету, добавления и удаления записей, сохранения в файл. 8. Описать класс "расписание приема клиентов". Каждая запись содержит дату, время, фамилию клиента. Время приема одного клиента должно быть равно одному часу. Предусмотреть возможность формирования расписания с клавиатуры и из файла, печати всего расписания, или расписания в конкретный день, добавления и удаления записей, сохранения в файл. При добавлении записи следует учитывать, что время записи должно быть свободно (не существуют уже созданной записи с этим же временем). 9. Описать класс "библиографический список для реферата". Каждая запись списка содержит автора(ов), название библиографического источника, его вид (учебник, статья, монография), год издания. Предусмотреть возможность формирования списка с клавиатуры и из файла, печати всего списка, добавления и удаления записей, сохранения в файл, поиска источников по автору, виду источников, году издания. 10. Описать класс "программа телепередач". Каждая запись программы содержит день недели, время, название канала, название телевизионной программы.

## **Тема 13. Основы архитектуры MVC**

Контрольная работа , примерные вопросы:

Контрольная работа заключается в создании игрового приложения на базе архитектуры "модель-представление-контроллер". Подсистема модели должна обеспечивать хранение и реализацию основного алгоритма игры. Подсистема представления может быть реализована в виде консольного или оконного приложения. Подсистема контроллера является необязательной для реализации. Тематика игровых приложений может быть взята из следующего списка: 1. Игра "Крестики-нолики" 2. Игра "Быки-коровы" 3. Игра "Кошки-мышки" 4. Игра "Морской бой" 5. Игра "Змейка" 6. Игра "Лабиринт" 7. Игра "Тетрис" 8. Игра "Преодоление препятствий" 9. Игра "Сапер" 10. Игра "Шашки"

#### **Тема 14. Основы разработки сетевых приложений**

Контрольная работа , примерные вопросы:

Контрольная работа посвящена разработке комплекса сетевых приложений. Серверная часть представляет хранение данных в виде файла (текстового формата или формата xml) и функций доступа к нему. Клиентская часть осуществляет функции взаимодействия с пользователем. Список предметных областей для разработки: 1. Реализация функциональной опции проверки баланса платежной банковской карты в банкомате. 2. Реализация функциональной опции оплаты счета банковской картой через банкомат. 3. Реализация функциональной опции заказа товаров в онлайн-магазине. 4. Реализация функциональной опции поиска магазинов, где есть в наличии заданный товар. 5. Реализация функциональной опции поиска книги в библиотеке. 6. Реализация функциональной опции онлайн-заказа туристической путевки. 7. Реализация функциональной опции подбора лекарства по заданным симптомам. 8. Реализация функциональной опции поиска статей на заданную тему. 9. Реализация функциональной опции поиска расписания врача. 10. Реализация функциональной опции поиска маршрута общественного транспорта.

#### **Тема 15. Основы методологии функционального программирования**

Контрольная работа , примерные вопросы:

Контрольная работа проверяет навыки записи классических алгоритмов обработки данных в функциональном стиле. Требуется записать предложенный в задании алгоритм в виде набора лямбда-выражений. Список задач. 1. Проверка целого числа на простоту. 2. Проверка, входит ли заданная цифра в запись целого числа. 3. Проверка того, что массив образует симметричную последовательность. 4. Проверка того, что массив образует возрастающую последовательность. 5. Сортировка массива методом выбора минимального элемента. 6. Алгоритм бинарного поиска для нахождения местонахождения заданного элемента в массиве. 7. Слияние двух отсортированных массивов в один. 8. Получение элементов, присутствующих в двух массивах одновременно. 9. Получение элементов, присутствующих в одном массиве и не присутствующих в другом. 10. Проведение транспонирования квадратной матрицы. 11. Получить произведение двух прямоугольных матриц. 12. Получить количество слов в символьной строке. 13. Получить количество слов символьной строки с заданной длиной. 14. Проверить, есть ли в символьной строке слова-палиндромы.

#### **Итоговая форма контроля**

зачет (в 1 семестре)

#### **Итоговая форма контроля**

зачет (в 2 семестре)

#### **Итоговая форма контроля**

зачет (в 3 семестре)

Примерные вопросы к зачету:

1 семестр

1. Выделение памяти, ввод и освобождение памяти для одномерного массива.
2. Выделение памяти, ввод и освобождение памяти для матрицы.
3. Проверка целого числа на простоту.
4. Проверка, входит ли заданная цифра в запись целого числа.

5. Алгоритм Евклида поиска наибольшего общего делителя двух целых чисел.
6. Поиск k-ого числа Фибоначчи.
7. Поиск суммы элементов одномерного массива, удовлетворяющих условию.
8. Поиск произведения элементов одномерного массива, удовлетворяющих условию.
9. Поиск количества элементов одномерного массива, удовлетворяющих условию.
10. Проверка того, что массив образует симметричную последовательность.
11. Проверка того, что массив образует возрастающую последовательность.
12. Поиск максимального (минимального) элемента массива.
13. Поиск максимального (минимального) элемента массива из удовлетворяющих условию.
14. Поиск количества максимальных (минимальных) элементов в массиве.
15. Получение двух максимальных (минимальных) элементов массива.
16. Проверка того, что все элементы в массиве присутствуют в нем один раз.
17. Сортировка массива методом выбора минимального элемента.
18. Сортировка массива методом пузырька.
19. Сортировка массива методом вставок.
20. Алгоритм бинарного поиска для нахождения местонахождения заданного элемента в массиве.
21. Слияние двух отсортированных массивов в один.
22. Получение элементов, присутствующих в двух массивах одновременно.
23. Получение элементов, присутствующих в одном массиве и не присутствующих в другом.
24. Проверка симметричности квадратной матрицы.
25. Проведение транспонирования квадратной матрицы.
26. Поворот квадратной матрицы на 90°.
27. Поменять местами в квадратной матрице элементы главной и побочной диагоналей в каждой строке (в каждом столбце).
28. Получить сумму элементов квадратной матрицы, находящихся выше (ниже) главной (побочной) диагоналей.
29. Проверка того, что квадратная матрица является верхней треугольной (нижней треугольной).
30. Получить сумму двух прямоугольных матриц.
31. Получить произведение двух прямоугольных матриц.
32. Получить строку (столбец) прямоугольной матрицы, удовлетворяющую условию.
33. Поменять местами две строки (два столбца) прямоугольной матрицы, удовлетворяющие условию.
34. Получить количество слов в символьной строке.
35. Получить слово максимальной (минимальной) длины в символьной строке.
36. Получить количество слов символьной строки, начинающихся и заканчивающихся одной и той же буквой.
37. Получить количество слов символьной строки с заданной длиной.
38. Получить количество слов символьной строки, в которых есть не менее 3 букв 'a'.
39. Проверить, есть ли в символьной строке слова-палиндромы.
40. Удалить из символьной строки все лишние пробелы.

## 2 семестр

1. Опишите алгоритм построения всех кратчайших путей в графе с помощью алгоритма Флойда. Пусть существует функция, реализующая этот алгоритм, например, имеющая следующий прототип:  
`void Floyd (int n, int** a, int**& T, int**&H),`

где  $n$  - количество вершин графа,  $a$  - матрица смежности графа,  $T$  - матрица кратчайших расстояний в графе,  $H$  -

матрица кратчайших путей в графе.

Напишите функцию (на языке программирования, псевдокоде или на естественном языке), которая на основании

матрицы кратчайших расстояний распечатывает все истоки графа (те вершины, из которых можно построить путь

во все другие вершины графа).

2. Опишите алгоритм построения всех кратчайших путей в графе с помощью алгоритма Флойда. Пусть существует

функция, реализующая этот алгоритм, например, имеющая следующий прототип:

```
void Floyd (int n, int** a, int**& T, int**&H),
```

где  $n$  - количество вершин графа,  $a$  - матрица смежности графа,  $T$  - матрица кратчайших расстояний в графе,  $H$  -

матрица кратчайших путей в графе.

Напишите функцию (на языке программирования, псевдокоде или на естественном языке), которая на основании

матрицы кратчайших расстояний определяет, является ли граф связным (связным называется граф, из каждой

вершины которого можно построить путь в любую другую вершину графа).

3. Опишите алгоритм построения каркаса минимального веса для графа алгоритмом Краскала. Пусть существует

функция, реализующая этот алгоритм, например, имеющая следующий прототип:

```
void Kraskal (int n, int** a, int**& K),
```

где  $n$  - количество вершин графа,  $a$  - матрица смежности графа,  $K$  - матрица смежности для построенного

каркаса.

Напишите функцию (на языке программирования, псевдокоде или на естественном языке), которая на основании

каркаса минимального веса исходного графа проверяет, является ли граф деревом (деревом является граф,

который является связным и не имеет циклов).

4. Опишите алгоритм построения всех кратчайших путей в графе с помощью алгоритма Флойда. Пусть существует

функция, реализующая этот алгоритм, например, имеющая следующий прототип:

```
void Floyd (int n, int** a, int**& T, int**&H),
```

где  $n$  - количество вершин графа,  $a$  - матрица смежности графа,  $T$  - матрица кратчайших расстояний в графе,  $H$  -

матрица кратчайших путей в графе.

Напишите функцию (на языке программирования, псевдокоде или на естественном языке), которая на основании

матрицы кратчайших расстояний в графе проверяет, является ли граф деревом (деревом является граф, который

является связным и не имеет циклов, или связный граф с количеством ребер на единицу меньше количества

вершин).

5. Дано дерево сортировки. Напишите рекурсивную функцию, которая находит количество листовых узлов дерева

(узлы, у которых нет ни левого, ни правого потомков).

6. Дано дерево сортировки. Напишите рекурсивную функцию, которая находит количество узлов дерева, которые имеют левых потомков.
7. Дано дерево сортировки. Напишите рекурсивную функцию, которая находит количество узлов дерева, которые имеют правых потомков.
8. Дано дерево сортировки. Напишите функцию получения разницы между максимальным и минимальным элементами дерева.
9. Дано дерево сортировки. Напишите рекурсивную функцию получения четных элементов дерева.
10. Дано дерево сортировки. Напишите рекурсивную функцию получения максимального четного элемента дерева.
11. Дано дерево сортировки. Напишите рекурсивную функцию получения количества узлов на заданном уровне.
12. Дан непустой односвязный список. Написать функцию, которая добавляет в начало списка новый элемент, равный количеству максимальных элементов в списке.
13. Дан непустой односвязный список. Написать функцию, которая удаляет из списка все вхождения его минимального элемента.
14. Дан непустой односвязный список. Написать функцию, которая удаляет из списка первый и последний элементы списка.
15. Дан непустой односвязный список. Написать функцию, которая добавляет новый последний элемент в список, равный среднему арифметическому элементов списка.
16. Дан непустой двусвязный список. Написать рекурсивную функцию проверки списка на симметричность.
17. Дан непустой двусвязный список. Написать функцию реверса списка.
18. Дан непустой двусвязный список. Написать функцию разбиения списка на два - четных и нечетных элементов.
19. Дан непустой двусвязный список. Написать функцию удаления из списка всех нечетных элементов.
20. Дан непустой двусвязный список. Написать функцию сортировки списка методом вставок.

### 3 семестр

Зачет проводится в виде реализации индивидуального проекта, в котором игровое приложение реализовано со

следующими обязательными компонентами:

1. Архитектура MVC.
2. Реализация компоненты модели в виде dll-библиотеки.
3. Реализация контроллера в виде tcp-сервера.
4. Реализация представления в виде клиентского приложения с оконным интерфейсом.
5. Реализация алгоритмов функционирования модели в функциональном стиле программирования.
6. Создание комплекта тестов для алгоритмов компоненты модели.
7. Создание комплекта тестов для алгоритмов компоненты представления.
8. Использование файлов различных форматов для хранения информации.

В качестве области разработки выступают игровые приложения (могут совпадать с темами игр контрольной

работы 1):

1. Игра "Крестики-нолики"
2. Игра "Быки-коровы"
3. Игра "Кошки-мышки"
4. Игра "Морской бой"
5. Игра "Змейка"
6. Игра "Лабиринт"
7. Игра "Тетрис"
8. Игра "Преодоление препятствий"
9. Игра "Сапер"
10. Игра "Шашки"
11. Игра "Реверси"
12. Игра "Парные картинки".
13. Игра "Пазл"
14. Игра "Го"
15. Игра "Линии"
16. Игра "Судоку"
17. Игра "2048"
18. Игра "Маджонг"
19. Игра "Нарды"
20. Игра "Пятнашки"

### 7.1. Основная литература:

1. Гагарина Л.Г. Программирование на языке высокого уровня. Программирование на языке C++: учеб. Пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. - М. : ИД 'ФОРУМ' : ИНФРА-М, 2018. - 512 с. - 480 с. Режим доступа: <http://znanium.com/bookread2.php?book=918098>
2. Кучунова Е. В. Программирование. Процедурное программирование: Учебное пособие / Кучунова Е.В., Олейников Б.В., Чередниченко О.М. - Красноярск.:СФУ, 2016. - 92 с. - Режим доступа: <http://znanium.com/bookread2.php?book=978627>
3. Корнеев В.И. Программирование графики на C++. Теория и примеры : учеб. Пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. - М. : ИД 'ФОРУМ' : ИНФРА-М, 2017. - 517 с. - Режим доступа: <http://znanium.com/bookread2.php?book=562914>
4. Воронцова Е.А. Программирование на C++ с погружением: практические задания и примеры кода - М.:НИЦ ИНФРА-М, 2016. - 80 с. - Режим доступа: <http://znanium.com/bookread2.php?book=563294>
5. Колдаев В. Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2015. - 416 с. - Режим доступа: <http://znanium.com/bookread2.php?book=484837>
6. Программирование на СИ#: Учебное пособие / Медведев М.А., Медведев А.Н. - 2-е изд., стер. - М.:Флинта, Изд-во Урал. Ун-та, 2017. - 64 с. - Режим доступа: <http://znanium.com/bookread2.php?book=948428>
7. Гуриков С.Р. .Введение в программирование на языке Visual C# : учеб. Пособие / С.Р. Гуриков. - М. : ФОРУМ : ИНФРА-М, 2018. - 447 с. - Режим доступа: <http://znanium.com/bookread2.php?book=967691>

## 7.2. Дополнительная литература:

1. Чистякова В. И. Алгоритмы и структуры данных : учебник / В.В. Белов, В.И. Чистякова. - М.: КУРС : НИЦ ИНФРА-М, 2017. - 240 с. - Режим доступа: <http://znanium.com/bookread2.php?book=766771>
2. Канцедал С.А. Алгоритмизация и программирование : Учебное пособие / С.А. Канцедал. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. - 352 с. - Режим доступа: <http://znanium.com/bookread2.php?book=429576>
3. Абрамян М. Э. Практикум по программированию на языке Паскаль: Массивы, строки, файлы, рекурсия, линейные динамические структуры, бинарные деревья: учебное пособие / М.Э. Абрамян. - Ростов н/Д: Издательство ЮФУ, 2010. - 276 с. - Режим доступа: <http://znanium.com/bookread2.php?book=549917>
4. Объектно-ориентированное программирование с примерами на С#: Учебное пособие / Хорев П.Б. - М.: Форум, НИЦ ИНФРА-М, 2016. - 200 с. - Режим доступа: <http://znanium.com/bookread2.php?book=529350>

## 7.3. Интернет-ресурсы:

Википедия - <http://ru.wikipedia.org>

Интернет-журнал по ИТ - <http://www.rsdn.ru>

Интернет-портал образовательных ресурсов по ИТ - <http://www.intuit.ru>

Интернет-портал ресурсов по ИТ (Microsoft) - <http://msdn.microsoft.com/ru-ru/ms348103.aspx>

Портал ресурсов по математике, алгоритмике и ИТ - <http://algotlist.manual.ru/>

## 8. Материально-техническое обеспечение дисциплины(модуля)

Освоение дисциплины "Практикум на электронно-вычислительных машинах" предполагает использование следующего материально-технического обеспечения:

Учебно-методическая литература для данной дисциплины имеется в наличии в электронно-библиотечной системе "ZNANIUM.COM", доступ к которой предоставлен студентам. ЭБС "ZNANIUM.COM" содержит произведения крупнейших российских учёных, руководителей государственных органов, преподавателей ведущих вузов страны, высококвалифицированных специалистов в различных сферах бизнеса. Фонд библиотеки сформирован с учетом всех изменений образовательных стандартов и включает учебники, учебные пособия, УМК, монографии, авторефераты, диссертации, энциклопедии, словари и справочники, законодательно-нормативные документы, специальные периодические издания и издания, выпускаемые издательствами вузов. В настоящее время ЭБС ZNANIUM.COM соответствует всем требованиям федеральных государственных образовательных стандартов высшего профессионального образования (ФГОС ВПО) нового поколения.

Лабораторные занятия по дисциплине проводятся в компьютерном классе.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 02.03.02 "Фундаментальная информатика и информационные технологии" и профилю подготовки Системный анализ и информационные технологии .

Автор(ы):

Тагиров Р.Р. \_\_\_\_\_

"\_\_" \_\_\_\_\_ 201\_\_ г.

Рецензент(ы):

Андрианова А.А. \_\_\_\_\_

"\_\_" \_\_\_\_\_ 201\_\_ г.