

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В.ЛОМОНОСОВА**

Механико-математический факультет

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

Работа на ЭВМ и программирование

Специальность: математика 010101

Квалификация (степень) выпускника: специалист

Форма обучения: очная, дневная

Автор: доцент кафедры вычислительной математики механико-математического факультета МГУ, к.ф.-м.н. В.Д.Валединский

Москва
2013

I. Название дисциплины: Работа на ЭВМ и программирование

II. Цели и задачи дисциплины:

А. Цели дисциплины:

- изучение основных понятий, фактов и технологий, связанных с применением вычислительной техники в науке, производстве и повседневной практике;
- освоение современных технологий программирования и использования вычислительной техники;
- ознакомление с состоянием дел в области компьютерных технологий, вычислительной техники и их применении при решении актуальных задач.
- получение теоретических знаний в области построения и использования алгоритмов в научной и научно-производственной деятельности.

Б. Задачи дисциплины:

- изучение языков программирования и технологии написания и сопровождения программных продуктов (на примере языков С, С++);
- изучение базовых способов представления данных и алгоритмов их обработки;
- изучение теоретических результатов, связанных с оценками трудоемкости базовых алгоритмов обработки данных, теорией компиляции, теоретическими основами программирования;
- изучение принципов функционирования вычислительной техники, вычислительных комплексов, компьютерных сетей и других сопутствующих устройств.
- формирование устойчивых навыков по использованию вычислительной техники в практической работе, в частности, программировании, сопровождении программных продуктов, настройке компьютерных систем и т.п.

В результате освоения дисциплины обучающийся должен:

1) знать

- базовые классические алгоритмы и методы представления и обработки данных;
- синтаксис и семантику нескольких языков программирования (например, С, С++)
- базовые принципы и процедуры функционирования вычислительных устройств, комплексов, компьютерных сетей
- принципы и технологии разработки программных продуктов и сопровождения программного обеспечения;
- методологию разработки новых алгоритмов для решения практических задач.

2) уметь

- организовать рабочее место и настроить компьютерную технику и программное обеспечение для решения своих рабочих, научных и технических задач;
- составлять, отлаживать, сопровождать программы в рамках определенной системы программирования и операционной системы;
- разрабатывать новые алгоритмы и использовать имеющиеся библиотеки программ для решения прикладных задач;
- проводить оценку и исследование эффективности программного продукта по отношению к конкретным задачам и требованиям;
- организовывать планирование разработки программного продукта в соответствии с поставленными целями и задачами.

III. Место дисциплины / практики в структуре ООП:

Б. Информация о месте дисциплины в учебном плане:

- 1, 2 курсы;
- 1, 2, 3, 4 семестры.

В. Перечень дисциплин, которые должны быть освоены для начала освоения и параллельно данной дисциплине:

- для первого семестра достаточно знаний в рамках школьных курсов математики и информатики;
- для второго и последующих семестров необходимы базовые понятия из математического анализа (производная, интеграл, числовой ряд, ряд Тейлора), алгебры (матрицы), аналитической геометрии (уравнения прямой, плоскости, расстояния, углы).

Г. Общая трудоемкость: 512 академических часов.

Д. Формы промежуточной аттестации: зачет, экзамен.

IV. Формы проведения занятий:

- форма занятий с указанием суммарной трудоемкости по каждой форме:
аудиторная работа, лекции – 86 часов;
аудиторная работа, семинары – 170 часов;
самостоятельная работа – 256 часов; (см. учебный план)
- формы текущего контроля: домашние задания, контрольные работы, прием расчетных работ, автоматизированные тесты.

V. Распределение трудоемкости по разделам и темам, а также формам проведения занятий с указанием форм текущего контроля и промежуточной аттестации
ПО НЕДЕЛЯМ :

Раздел 1. Базовые приемы программирования.

№ п/п	Наименование разделов и тем дисциплины	Трудоемкость (в ак. часах) по формам занятий			Формы контроля
		Аудиторная работа (с разбивкой по формам и видам)		Самостоятельная работа	
		Лекции	Семинары		
1	Тема 1.1. Рабочее окружение программиста		2	2	Домашнее задание
2	Тема 1.2. Базовые конструкции языка С		4	4	Домашнее задание
3	Тема 1.3. Процедуры ввода-вывода		2	2	Домашнее задание
4	Тема 1.4. Работа с файлами		4	4	Домашнее задание
5	Тема 1.5. Индуктивные алгоритмы на последовательностях		2	2	Прием расчетных работ
6	Тема 1.6. Компиляция, сборка, запуск		4	4	Прием расчетных работ
7	Тема 1.7. Массивы и указатели в языке С		2	2	Прием расчетных работ
8	Тема 1.8. Динамическое выделение памяти		4	4	Контрольная-тест
9	Тема 1.9. Алгоритмы обработки массивов		2	2	Домашнее задание
10	Тема 1.10. Методы		4	4	Прием

	сортировки				расчетных работ
11	Тема 1.11. Указатель на функцию		2	2	Прием расчетных работ
12	Тема 1.12. Оценка скорости работы программы		4	4	Прием расчетных работ
13	Тема 1.13. Методы отладки		2	2	Контрольная-тест
14	Тема 1.14. Битовые операции		4	4	Прием расчетных работ
15	Тема 1.15. Решение геометрических задач		2	2	Прием расчетных работ
16	Тема 1.16. Структурные типы в языке C		4	4	Прием расчетных работ
17	Тема 1.17. Дополнительные конструкции языка C		2	2	Прием расчетных работ
18	Зачетная контрольная		4	4	зачёт
	Итого:		54	54	

Раздел 2. Решение прикладных математических задач.

№ п/п	Наименование разделов и тем дисциплины	Трудоемкость (в ак. часах) по формам занятий			Формы контроля
		Аудиторная работа (с разбивкой по формам и видам)		Самостоятельная работа	
		Лекции	Семинары		
1	Тема 2.1. Вычислительная погрешность		2	2	Домашнее задание
2	Тема 2.2. Решение уравнений		4	4	Домашнее задание
3	Тема 2.3. Вычисление интегралов		2	2	Прием расчетных работ
4	Тема 2.4. Минимизация функций		4	4	Прием расчетных работ
5	Тема 2.5. Вычисление элементарных функций		2	2	Прием расчетных работ
6	Тема 2.6. Приближение функций		4	4	Контрольная работа
7	Тема 2.7. Работа с матрицами		2	2	Прием расчетных работ

					работ
8	Тема 2.8. Приведение матриц к треугольному виду		4	4	Прием расчетных работ
9	Тема 2.9. Решение систем линейных уравнений		2	2	Прием расчетных работ
10	Тема 2.10. Вычисление определителя и ранга		4	4	Прием расчетных работ
11	Тема 2.11. Функции ввода-вывода со строками		2	2	Контрольная-тест
12	Тема 2.12. Обработка текстовых файлов		4	4	Прием расчетных работ
13	Тема 2.13. Алгоритмы поиска в строках		2	2	Прием расчетных работ
14	Тема 2.14. Решение уравнений со сложными функциями		4	4	Прием расчетных работ
15	Тема 2.15. Автоматический выбор шага интегрированияР		2	2	Прием расчетных работ
16	Зачетная контрольная		4	4	зачёт
	Итого:		36	64	

Раздел 3. Схемы хранения данных и алгоритмы их обработки.

№ п/п	Наименование разделов и тем дисциплины	Трудоемкость (в ак. часах) по формам занятий			Формы контроля
		Аудиторная работа (с разбивкой по формам и видам)		Самостоятельная работа	
		Лекции	Семинары		
1	Тема 3.1. Объектный подход к программированию	2	2	4	Домашнее задание
2	Тема 3.2. Основные конструкции языка C++	4	2	6	Домашнее задание
3	Тема 3.3. Структуры данных. Стек, Дек, Очередь.	2	2	4	Прием расчетных работ
4	Тема 3.4. Параметризованные классы в C++	4	2	6	Прием расчетных работ
5	Тема 3.5. Списочные структуры данных	2	2	4	Контрольная работа
6	Тема 3.6. Бинарное дерево поиска	4	2	6	Прием расчетных работ

					работ
7	Тема 3.7. Произвольные деревья	2	2	4	Прием расчетных работ
8	Тема 3.8. Деревья поиска с логарифмической глубиной	4	2	6	Прием расчетных работ
9	Тема 3.9. В-деревья	2	2	4	Прием расчетных работ
10	Тема 3.10. Хеш-множества	4	2	6	Контрольная работа
11	Тема 3.11. Файловые системы. FAT.	2	2	4	Прием расчетных работ
12	Тема 3.12. Файловые системы. NTFS, EXT3.	4	2	6	Прием расчетных работ
13	Тема 3.13. Оценки трудоемкости для алгоритмов сортировки	2	2	4	Прием расчетных работ
14	Тема 3.14. Быстрые алгоритмы сортировки	4	2	6	Прием расчетных работ
15	Тема 3.15. Формальные грамматики	2	2	4	Прием расчетных работ
16	Тема 3.16. Синтаксический разбор и компиляция	4	2	6	Прием расчетных работ
17	Тема 3.17. Компиляция в объектный код	2	2	4	Контрольная работа
18	Тема 3.18. Компиляция базовых алгоритмических конструкций	4	2	6	экзамен
	Итого:	54	36	90	

Раздел 4. Современные сетевые технологии.

№ п/п	Наименование разделов и тем дисциплины	Трудоемкость (в ак. часах) по формам занятий			Формы контроля
		Аудиторная работа (с разбивкой по формам и видам)		Самостоятельная работа	
		Лекции	Семинары		
1	Тема 4.1. Уровни сетевого взаимодействия	2	2	4	Домашнее задание
2	Тема 4.2. Канальный	2	2	4	Прием

	уровень				расчетных работ
3	Тема 4.3. Сетевой уровень (IPv4)	2	2	4	Прием расчетных работ и ум
4	Тема 4.4. Сетевой уровень (IPv6)	2	2	4	Прием расчетных работ
5	Тема 4.5. Транспортный уровень	2	2	4	Прием расчетных работ
6	Тема 4.6. Протокол TCP	2	2	4	Прием расчетных работ
7	Тема 4.7. Socket интерфейс	2	2	4	Прием расчетных работ
8	Тема 4.8. Технология клиент-сервер	2	2	4	Прием расчетных работ
9	Тема 4.9. Реализация клиента и сервера в одном процессе	2	2	4	Прием расчетных работ
10	Тема 4.10. Реализация клиента и сервера в разных процессах	2	2	4	Прием расчетных работ
11	Тема 4.11. Система доменных имен (DNS)	2	2	4	Прием расчетных работ
12	Тема 4.12. Протоколы высших уровней	2	2	4	Прием расчетных работ
13	Тема 4.13. Проблема и протоколы маршрутизации	2	2	4	Прием расчетных работ
14	Тема 4.14. Алгоритмы маршрутизации	2	2	4	Прием расчетных работ
15	Тема 4.15. Подготовка математических текстов в системе TeX	2	2	4	Прием расчетных работ
16	Тема 4.16. Конструкции и команды TeX	2	2	4	Прием расчетных работ
	Итого:	32	32	64	

VI. Содержание дисциплины - аудиторная и самостоятельная работа:

Раздел 1. Базовые приемы программирования

Тема 1.1. Рабочее окружение программиста.

Семинар (2ч): Многопользовательский интерфейс. Входное имя, пароль. Консоль командной строки. Текстовые редакторы. Основные команды shell. Настройка клавиатуры, языка, рабочего стола. Организация пользовательского каталога. Файловые менеджеры. Другие необходимые приложения.

Задания для самостоятельной работы: изучение материалов темы

Тема 1.2. Базовые конструкции языка C

Семинар (4ч): Базовые типы данных (целый, вещественный), логические значения, кодовые таблицы и символьные значения. Модификаторы signed, unsigned. Объявления переменных и массивов. Области видимости. Классы памяти (локальные, глобальные, static, extern) Управляющие конструкции, разветвления, циклы. Функции, возвращаемые значения. Прототипы, сигнатуры. Структура программы. Использование нескольких файлов. Заголовочные файлы. Примеры простейших программ.

Задания для самостоятельной работы: изучение материалов темы, подготовка и компиляция примеров программ по образцу.

Тема 1.3. Процедуры ввода-вывода.

Семинар (2ч): Библиотеки ввода-вывода в C. Функции чтения и записи с терминалом и клавиатурой. Форматный ввод-вывод. Спецификации преобразования базовых типов при вводе-выводе. Пример кода с вводом-выводом.

Задания для самостоятельной работы: изучение материалов темы, подготовка и компиляция и выполнение примеров программ по образцу.

Тема 1.4. Работа с файлами

Семинар (4ч): Библиотеки ввода-вывода в C. Функции открытия, закрытия файлов. Проверка успешности операции ввода-вывода по отношению к файлу. Понятие позиции ввода-вывода. Последовательный ввод-вывод. Примеры кода с вводом-выводом. Чтение последовательности неизвестной длины из файла.

Задания для самостоятельной работы: изучение материалов темы, подготовка и компиляция и выполнение примеров программ по образцу.

Тема 1.5. Индуктивные алгоритмы на последовательностях

Семинар (2ч): Понятие об индуктивном алгоритме и рекуррентных последовательностях. Построение рекуррентных соотношений для различных задач обработки последовательностей. Пример кода с индуктивной обработкой последовательности. Разбор алгоритмов решения различных задач на последовательности (см. примеры задач). Выдача индивидуальных заданий по теме последовательности.

Задания для самостоятельной работы: набор задач для самостоятельного выполнения из списка 30-40 задач.

Тема 1.6 Компиляция, сборка, запуск

Семинар (4ч): Последовательность прохождения задачи – компиляция, сборка, запуск. Использование нескольких файлов для текста программы. Командная строка для компиляции и сборки. Практическое выполнение и проверка индивидуальных заданий по теме последовательности.

Задания для самостоятельной работы: решение и реализация задач из набора задач по теме последовательности.

Тема 1.7. Массивы и указатели в языке C.

Семинар (2ч): Объявление массивов в C. Инициализация массивов. Статическое выделение памяти. Указатели. Связь массивов и указателей. Операция индексации. Операции с

указателями. Примеры кода с созданием и использованием массивов. Практическое выполнение и проверка индивидуальных заданий по теме последовательности.

Задания для самостоятельной работы: изучение материалов темы, решение и реализация задач из набора задач по теме последовательности.

Тема 1.8 Динамическое выделение памяти.

Семинар (4ч): Понятие о распределении памяти в вычислительной системе (куча, стек).

Статическое и динамическое выделение памяти. Функции захвата и освобождения памяти (malloc, free). Работа с массивами заранее неизвестного размера. Понятие об утечках памяти. Примеры кода с динамическим выделением памяти. Автоматизированный тест по теме последовательности.

Задания для самостоятельной работы: набор задач для самостоятельного выполнения по теме массивы.

Тема 1.9. Алгоритмы обработки массивов

Семинар (2ч): Разбор алгоритмов некоторых типичных задач по теме массивы. Примеры кода и распределения функциональности при решении задач по теме массивы. Практическое выполнение и проверка индивидуальных заданий по теме массивы.

Задания для самостоятельной работы: набор задач для самостоятельного выполнения по теме массивы

Тема 1.10. Методы сортировки

Семинар (4ч): Классические алгоритмы сортировки – выбор максимального, пузырьковая, просеивание, быстрая, пирамидальная, подсчетом, битовые и пр. Обсуждение трудоемкости и затрат дополнительной памяти. Практическое выполнение и проверка индивидуальных заданий по теме массивы.

Задания для самостоятельной работы: набор задач для самостоятельного выполнения по теме сортировки.

Тема 1.11 Указатель на функцию.

Семинар (2ч): Указатель на функцию в языке C. Передача функции в процедуру через параметры. Использование указателя на функцию для процедуры сравнения в реализации алгоритмов сортировки. Библиотечная функция сортировки qsort. Практическое выполнение и проверка индивидуальных заданий по теме массивы и сортировки.

Задания для самостоятельной работы: набор задач для самостоятельного выполнения по теме сортировки.

Тема 1.12 Оценка скорости работы программы

Семинар (4ч): Таймер в вычислительных системах. Библиотечная функция clock и ее использование для замера времени выполнения программы. Тестирование времени работы алгоритмов сортировки на массивах разной длины. Практическое подтверждение теоретических оценок трудоемкости алгоритмов. Практическое выполнение и проверка индивидуальных заданий по теме массивы и сортировки.

Задания для самостоятельной работы: набор задач для самостоятельного выполнения по теме сортировки.

Тема 1.13 Методы отладки

Семинар (2ч): Системная диагностика отказов при выполнении программы. Арифметическое исключение. Отказ по доступу к памяти. Логические и арифметические ошибки в алгоритмах. Использование простейших тестовых выводов для трассировки программы. Замечание о буферизации вывода. Использование отладчиков для поиска и исправления ошибок (на примере gdb). Контрольный тест по теме массивы.

Задания для самостоятельной работы: изучение материалов темы

Тема 1.14 Битовые операции

Семинар (4ч): Битовые операции в языке C, логические операции, сдвиги. Установка и чтение определенного бита к целому числу. Использование битовых операций для индикации состояний объектов, в задачах о делимости и пр. Практическое выполнение и проверка индивидуальных заданий по теме битовые операции и делимость.

Задания для самостоятельной работы: изучение материалов темы, набор задач для самостоятельного выполнения по теме битовые операции и делимость.

Тема 1.15 Решение геометрических задач

Семинар (2ч): Формулировки задач по работе с точками и отрезками на прямой и плоскости. Использование формул аналитической геометрии. Разбор алгоритмов решения некоторых задач на точки и отрезки. Практическое выполнение и проверка индивидуальных заданий по теме битовые операции и делимость.

Задания для самостоятельной работы: изучение материалов темы, набор задач для самостоятельного выполнения по теме геометрические задачи.

Тема 1.16 Структурные типы в языке C

Семинар (4ч): Структурные типы в языке C. Использование структурных типов для представления геометрических объектов (точка, отрезок, и т.п.) и реализации алгоритмов решения геометрических задач. Практическое выполнение и проверка индивидуальных заданий по теме геометрические задачи.

Задания для самостоятельной работы: изучение материалов темы, набор задач для самостоятельного выполнения по теме геометрические задачи.

Тема 1.17 Дополнительные конструкции языка C

Семинар (2ч): директивы препроцессора (include, define, ifdef, и др.) Использование макросов с параметрами и без для параметризации программы. Условная компиляция в зависимости от заданных определений. Практическое выполнение и проверка индивидуальных заданий по теме геометрические задачи.

Задания для самостоятельной работы: изучение материалов темы.

Тема 1.18 Итоговое зачетное занятие.

Семинар (4ч): Практическое выполнение и проверка индивидуальных заданий по всем не сданным ранее темам. Контрольные зачетные задания по «долгам» за семестр.

Раздел 2. Решение прикладных математических задач.

Тема 2.1. Вычислительная погрешность

Семинар (2ч): Формат представления вещественных чисел (IEEE 754). Машинная точность. Нормализованные и ненормализованные числа. Абсолютная и относительная погрешность. Распространение относительно погрешности при арифметических операциях. Примеры неустойчивых вычислений. Проблема сравнения вещественных чисел на равенство в вычислительных алгоритмах.

Задания для самостоятельной работы: изучение материалов темы.

Тема 2.2. Решение уравнений

Семинар (4ч): Алгоритмы решения уравнений – метод биссекции, Ньютона, хорд. Примеры кода для реализации этих методов. Проблема расходимости алгоритмов.

Задания для самостоятельной работы: индивидуальные задания по теме уравнения.

Тема 2.3. Вычисление интегралов

Семинар (2ч): Алгоритмы вычисления определенных интегралов на конечных отрезках. Квадратурные формулы прямоугольников, трапеций, Симпсона, Гаусса. Проблема автоматического достижения заданной точности при равномерном разбиении. Примеры программного кода. . Практическое выполнение и проверка индивидуальных заданий по теме уравнения.

Задания для самостоятельной работы: индивидуальные задания по теме интегралы.

Тема 2.4. Минимизация функций

Семинар (4ч): Алгоритмы минимизации функции одной переменной. Методы поиска по последовательности точек, по измельчению сетки, метод золотого сечения. Проблема контроля достигнутой точности. Примеры программного кода. . Практическое выполнение и проверка индивидуальных заданий по теме уравнения и интегралы.

Задания для самостоятельной работы: индивидуальные задания по теме интегралы.

Тема 2.5. Вычисление элементарных функций

Семинар (2ч): представление элементарных функций степенными рядами. Численное суммирование рядов до заданной точности. Проблема области определения (большие аргументы) для элементарных функций. Примеры программного кода. . Практическое выполнение и проверка индивидуальных заданий по теме уравнения, интегралы и минимизация.

Задания для самостоятельной работы: индивидуальные задания по теме элементарные функции.

Тема 2.6. Приближение функций

Семинар (4ч): Кусочная интерполяция (линейная, квадратичная, Эрмитова), Многочлен Лагранжа. Постановка задачи приближения функции одной переменной и примеры программного кода. Практическое выполнение и проверка индивидуальных заданий по ранее пройденным темам.

Задания для самостоятельной работы: индивидуальные задания по теме приближения функций.

Тема 2.7. Работа с матрицами

Семинар (2ч): Представление матриц а языка С – на базе одномерного массива, на базе массива указателей на строки. Передача матриц через параметры функций. Ввод-вывод матриц. Динамическое выделение памяти под матрицы. Примеры программного кода с матричными операциями. Итоговая контрольная работа по теме элементарные вычисления.

Задания для самостоятельной работы: индивидуальные задания по теме матрицы.

Тема 2.8. Приведение матриц к треугольному виду

Семинар (4ч): Алгоритмы приведения матрицы к ступенчатому и диагональному виду (метод Гаусса). Примеры кода. Вычислительная устойчивость алгоритмов. Методы с выбором главного элемента (по строкам, по матрице). Понятие о других методах приведения (отражения, вращения). Практическое выполнение и проверка индивидуальных заданий по теме матрицы.

Задания для самостоятельной работы: индивидуальные задания по теме матрицы..

Тема 2.9. Решение систем линейных уравнений

Семинар (2ч): Алгоритмы решения систем линейных уравнений типа метода Гаусса. Влияние обусловленности системы на точность вычислений. Примеры с матрицей Гильберта. Практическое выполнение и проверка индивидуальных заданий по теме матрицы.

Задания для самостоятельной работы: индивидуальные задания по теме матрицы..

Тема 2.10. Вычисление определителя и ранга, обратной матрицы

Семинар (4ч): Алгоритмы вычисления определителя, ранга и обратной матрицы приведением к ступенчатому или диагональному виду. Практическое выполнение и проверка индивидуальных заданий по теме матрицы.

Задания для самостоятельной работы: индивидуальные задания по теме матрицы..

Тема 2.11. Функции ввода-вывода со строками

Семинар (2ч): дополнительные стандартные функции ввода-вывода, ориентированные на работу со строками. Примеры программного кода. Контрольный тест по теме матрицы.

Задания для самостоятельной работы: изучение материалов темы.

Тема 2.12. Обработка текстовых файлов

Семинар (4ч): Обработка текстовых файлов – посимвольная и построчная. Примеры программного кода. Разбор подходов к решению заданий по теме обработка текстов.

Задания для самостоятельной работы: индивидуальные задания по теме обработка текстов.

Тема 2.13. Алгоритмы поиска в строках

Семинар (2ч): Базовые алгоритмы поиска символов и образцов в строках. Библиотечные функции работы со строками. Примеры программного кода. Практическое выполнение и проверка индивидуальных заданий по теме обработка текстов.

Задания для самостоятельной работы: индивидуальные задания по теме обработка текстов.

Тема 2.14. Решение уравнений со сложными функциями

Семинар (4ч): Формулировка задачи о решении уравнения с интегральной функцией (несобственный интеграл). Разбор подходов к решению. Контроль точности и алгоритмы с автоматическим выбором шага интегрирования.

Задания для самостоятельной работы: индивидуальные задания по теме сложное уравнение.

Тема 2.15. Автоматический выбор шага интегрирования

Семинар (2ч): Обсуждение алгоритмов автоматического выбора шага и отсека области интегрирования при вычислении несобственных интегралов. Практическое выполнение и проверка индивидуальных заданий по теме сложное уравнение.

Задания для самостоятельной работы: индивидуальные задания по теме сложное уравнение.

Тема 2.16 Итоговое зачетное занятие.

Семинар (4ч): Прием заданий, не сданных ранее в семестре. Зачетная контрольная работа.

Раздел 3. Схемы хранения данных и алгоритмы их обработки.

Тема 3.1. Объектный подход к программированию

Лекции (2ч): Понятия объекта, внутреннего представления, интерфейса.

Объектно-ориентированные языки. Доступ к данным и методам. Скрытие данных и методов. Иерархии и наследования.

Семинары (2ч): Добавление методов к структурным типам. Классы в C++. Примеры программного кода.

Задания для самостоятельной работы: изучение материалов темы.

Тема 3.2. Основные конструкции языка C++

Лекции (4ч): Классы, методы, инкапсуляция, полиморфизм. Особенности C++ по сравнению с языком C. Ссылочный тип. Конструкторы, деструкторы. Конструктор копирования. Типы доступа (public, private, friend). Перегрузка операций. Примеры организации классов на примере геометрических объектов. Примеры программного кода.

Семинары (2ч): Разбор материалов лекций. Реализация простейших классов в программе.

Задания для самостоятельной работы: индивидуальные задания по теме реализация простейших классов.

Тема 3.3. Структуры данных. Стек, Дек, Очередь.

Лекции (2ч): Постановка задачи о схемах хранения данных. Непрерывные реализации на базе массива. Стек, дек, очередь. Дисциплина доступа. Интерфейсы. Примеры реализаций. Контроль успешности операций. Исключения в C++, перехват исключений.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме реализация простейших классов.

Задания для самостоятельной работы: индивидуальные задания по теме реализация простейших классов.

Тема 3.4. Параметризованные классы в C++

Лекции (4ч): Параметризация кода под произвольный тип данных. Инstrukция template. Параметризованные реализации стека, дека и очереди.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме реализация простейших классов. Переопределение операций для реализованных классов.

Задания для самостоятельной работы: индивидуальные задания по теме реализация простейших классов.

Тема 3.5. Списочные структуры данных

Лекции (2ч): Ссылочная реализация линейных наборов данных. Однонаправленный и двунаправленный списки. Кольцевые списки. Параметризованные реализации.

Семинары (2ч): Разбор материалов лекций. Контрольная тестовая работа по теме реализация простейших классов.

Задания для самостоятельной работы: индивидуальные задания по теме списочные структуры.

Тема 3.6. Бинарное дерево поиска

Лекции (4ч): Бинарное дерево поиска. Определения. Примеры. Обходы. Процедуры добавления, удаления, поиска элемента по ключу. Сложность поиска. Примеры кода для реализации дерева.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме списочные структуры.

Задания для самостоятельной работы: индивидуальные задания по теме списочные структуры.

Тема 3.7. Произвольные деревья

Лекции (2ч): Произвольные (ветвящиеся) деревья. Представление в виде класса. Обходы. Процедуры добавления, удаления, поиска элемента.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме списочные структуры.

Задания для самостоятельной работы: индивидуальные задания по теме списочные структуры.

Тема 3.8. Деревья поиска с логарифмической глубиной

Лекции (4ч): Сбалансированные AVL деревья. Красно-черные деревья. Определения, свойства. Процедуры добавления, удаления, поиска элемента. Теоремы о глубине AVL дерева и красно-черного дерева. Примеры реализации.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме списочные структуры.

Задания для самостоятельной работы: индивидуальные задания по теме списочные структуры.

Тема 3.9. В-деревья

Лекции (2ч): В и В+ деревья. Определения, свойства. Процедуры добавления, удаления, поиска элемента. Теорема о трудоемкости операций с В-деревом. Реализации над дисковой памятью.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме списочные структуры.

Задания для самостоятельной работы: индивидуальные задания по теме списочные структуры.

Тема 3.10. Хеш-множества

Лекции (4ч): Хеш-множества. Определения. Реализации по методу списков и методу последовательных проб. Примеры хеш-функций. Совершенная хеш-функция. Пример построения минимальной совершенной хеш-функции. Теорема о средней трудоемкости работы с хеш-множеством по методу проб.

Семинары (2ч): Разбор материалов лекций. Контрольная работа по теме списочные структуры.

Задания для самостоятельной работы: индивидуальные задания по теме деревья.

Тема 3.11. Файловые системы. FAT.

Лекции (2ч): Основные концепции и понятия работы файловых систем. Разделы диска, BR, MBR. Система FAT32. Области FAT, DIR, DATA, их назначение и формат. Запись dir entry. Принципы работы с кластерами при выполнении файловых операций.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме деревья.

Задания для самостоятельной работы: индивидуальные задания по теме деревья.

Тема 3.12. Файловые системы. NTFS, EXT3.

Лекции (4ч): Файловая система NTFS, область MFT. Атрибуты файлов. Метафайлы, их роль в системе. Файловые записи, назначение, формат. Основные атрибуты файлов. Управление распределением файловых кластеров. Резидентные и нерезидентные атрибуты.

Файловая система EXT3. Группы блоков. Суперблок, таблица дескрипторов. Запись INODE. Организация доступа и использования файлов в системах UNIX. Управление распределением блоков файла, таблица блоков, косвенные ссылки. Организация каталогов. Жесткие и символические ссылки.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме деревья.

Задания для самостоятельной работы: индивидуальные задания по теме деревья.

Тема 3.13. Оценки трудоемкости для алгоритмов сортировки

Лекции (2ч): Теорема об оценке снизу для сортировки сравнениями. Оценки трудоемкости и памяти для реализации быстрых сортировок.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме деревья.

Задания для самостоятельной работы: индивидуальные задания по теме деревья.

Тема 3.14. Быстрые алгоритмы сортировки

Лекции (4ч): Быстрые сортировки – слияние, быстрая сортировка, пирамидальная сортировка, сортировка подсчетом. Теорема о средней трудоемкости быстрой сортировки. Сортировка слиянием без использования дополнительной памяти. Модификации сортировок подсчетом и радикальных сортировок. Многопоточковые сортировки.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме дерева.
Задания для самостоятельной работы: индивидуальные задания по теме дерева.

Тема 3.15. Формальные грамматики

Лекции (2ч): Постановка задачи компиляции алгоритмических языков. Формальные грамматики. Определения. Способы описания. Примеры формальных грамматик для языков арифметических выражений, описаний переменных, прототипов функций и т.п..

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме дерева.
Задания для самостоятельной работы: индивидуальные задания по теме дерева.

Тема 3.16. Синтаксический разбор и компиляция

Лекции (4ч): Синтаксический разбор. Понятие о LR разборах. LR(k) грамматика. Стек разбора, сдвиг, свертка. Формальное построение алгоритма LR(k) разбора заданного выражения. Критерий возможности свертки. Критерий успешности разбора. Компилятор модельного языка.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме дерева.
Задания для самостоятельной работы: индивидуальные задания по теме дерева.

Тема 3.17. Компиляция в объектный код

Лекции (2ч): проблема компиляции алгоритмического языка в код конкретной вычислительной системы. Дерево разбора. Построение кода для линейных участков программы для системы команд стекового процессора.

Семинары (2ч): Разбор материалов лекций. Выполнение и проверка заданий по теме дерева.
Задания для самостоятельной работы: индивидуальные задания по теме дерева.

Тема 3.18. Компиляция базовых алгоритмических конструкций

Лекции (4ч): компиляция управляющих конструкций (условных и безусловных переходов), компиляция вызовов функций, распределение памяти на стеке. Компиляция выражений с преобразованием типов.

Семинары (2ч): Разбор материалов лекций. Итоговая проверка заданий по теме дерева.

Раздел 4. Современные сетевые технологии.

Тема 4.1. Уровни сетевого взаимодействия

Лекции (2ч): Организация компьютерных сетей, LAN, WAN. Аппаратные средства, станция, адаптер, среда связи, мост, шлюз, маршрутизатор. Пакеты, протоколы. Модель взаимодействия ISO OSI, ее соотношение с реальной практикой. Горизонтальные и вертикальные взаимодействия. Стек протоколов.

Семинары (2ч): Разбор материалов лекций. Обсуждение практических заданий по сетевым взаимодействиям.

Задания для самостоятельной работы: индивидуальные задания по разработке простейшей базы данных.

Тема 4.2. Канальный уровень

Лекции (2ч): Задачи канального уровня. Доступ к среде передачи. MAC, Mac-address. Ethernet. Формат пакета, дисциплина CSMA/CD, ограничение снизу на длину пакета. Сети WiFi, сервисы, зоны доступа, взаимодействие станций, дисциплина CSMA/CA.

Семинары (2ч): Разбор материалов лекций. Проверка заданий по подготовке наборов записей для базы данных.

Задания для самостоятельной работы: индивидуальные задания по разработке простейшей базы данных.

Тема 4.3. Сетевой уровень (IPv4)

Лекции (2ч): IP адресация v4. Классы адресов, широковещательные адреса, маска подсети, Задачи протокола IP формат пакета. Взаимодействие с канальным уровнем, протоколы ARP и RARP. Протокол ICMP v.4. Команды и сообщения. Запрос PING.

Семинары (2ч): Разбор материалов лекций. Реализация хранения и загрузки базы данных.

Задания для самостоятельной работы: индивидуальные задания по разработке простейшей базы данных.

Тема 4.4. Сетевой уровень (IPv6)

Лекции (2ч): Версия IP v.6. Причины появления и отличия от IP v.4. Изменение адресации, формата пакета и идеологии обработки пакетов. Версия ICMP. Проблема взаимодействия разных версий IP в единой глобальной сети.

Семинары (2ч):): Разбор материалов лекций. Реализация поиска записей в базе данных.

Задания для самостоятельной работы: индивидуальные задания по разработке простейшей базы данных.

Тема 4.5. Транспортный уровень.

Лекции (2ч): Задачи транспортного уровня. Идентификация абонента по номеру порта.

Порты стандартных сетевых служб и сервисов. Протокол UDP, назначение, особенности, формат пакета.

Семинары (2ч): Разбор материалов лекций. Реализация языка запросов к базе данных.

Задания для самостоятельной работы: индивидуальные задания по разработке простейшей базы данных.

Тема 4.6. Протокол TCP

Лекции (2ч): Гарантированная доставка данных. Протокол TCP. Алгоритм обмена данными, буферы передачи и приема, подтверждения, таймера, процедуры установления и разрыва соединения. Окно приема и передачи. Формат пакета.

Семинары (2ч): Разбор материалов лекций. Тестирование реализация базы данных с языком запросов.

Задания для самостоятельной работы: индивидуальные задания по разработке простейшей базы данных.

Тема 4.7. Socket интерфейс

Лекции (2ч): Библиотечные функции поддержки TCP/IP. Создание сокета, задание адресов, типов протокола. Использование DNS службы (gethostbyname). Проблема big-little endian при передаче данных по сети (htonl, ntohl и т.д.). Настройка использования сетевых интерфейсов. Базовые функции передачи и приема данных (read, write, sendto, recvfrom). Контроль успешности вызовов.

Семинары (2ч): Разбор материалов лекций. Формулировка требований к сетевой базе данных и способов ее реализации.

Задания для самостоятельной работы: индивидуальные задания по разработке сетевой базы данных.

Тема 4.8. Технология клиент-сервер

Лекции (2ч): Взаимодействие клиент – сервер, Запросы и отклики. Методика реализации с использованием протоколов UDP и TCP. Разработка форматов обмена для данных переменной длины.

Семинары (2ч): Разбор материалов лекций. Реализация простейших сетевых взаимодействий и передачи сообщений по выбранным протоколам.

Задания для самостоятельной работы: индивидуальные задания по разработке сетевой базы данных.

Тема 4.9. Реализация клиента и сервера в одном процессе

Лекции (2ч): Реализация клиента и сервера по протоколам UDP и TCP. Реализация таймеров. Системные вызовы connect, listen, accept. Проверка готовности каналов ввода вывода с помощью функции select. Стратегия обмена с ожиданием и без ожидания.

Семинары (2ч): Разбор материалов лекций. Реализация простейших сетевых взаимодействий с взаимным обменом соотношений между клиентом и сервером..

Задания для самостоятельной работы: индивидуальные задания по разработке сетевой базы данных.

Тема 4.10. Реализация клиента и сервера в разных процессах

Лекции (2ч): Порождение нового процесса системным вызовом fork. Идентификаторы процессов PID и PPID. Реализация сервера с выделением отдельного процесса каждому новому клиенту. Проблема взаимного исключения при параллельном доступе к ресурсам и данным, алгоритм читателей и писателей. Примеры программного кода.

Семинары (2ч): Разбор материалов лекций. Реализация простейших сетевых взаимодействий с взаимным обменом соотношений между клиентом и сервером..

Задания для самостоятельной работы: индивидуальные задания по разработке сетевой базы данных.

Тема 4.11. Система доменных имен (DNS)

Лекции (2ч): Доменная система имен. Распределенная база DNS серверов. Формат DNS запросов и ответов. Итеративный и рекурсивный варианты обслуживания запросов.

Семинары (2ч): Разбор материалов лекций. Реализация сетевых взаимодействий с взаимным обменом соотношений между клиентом и сервером для работы с базой данных..

Задания для самостоятельной работы: индивидуальные задания по разработке сетевой базы данных.

Тема 4.12. Протоколы высших уровней

Лекции (2ч): протоколы FTP, HTTP, SMTP. Форматы сообщений. Обработка отказов. Стандартные порты и типичные реализации.

Семинары (2ч): Разбор материалов лекций. Реализация сетевых взаимодействий с взаимным обменом соотношений между клиентом и сервером для работы с базой данных..

Задания для самостоятельной работы: индивидуальные задания по разработке сетевой базы данных.

Тема 4.13. Проблема и протоколы маршрутизации

Лекции (2ч): Маршрутизация, основные принципы и понятия. Зоны маршрутизации. Друхуровневая модель маршрутизации. Обзор некоторых алгоритмов и подходов к маршрутизации.

Семинары (2ч): Разбор материалов лекций. Итоговое тестирование сетевой базы данных с одиночным сервером и одиночным клиентом.

Задания для самостоятельной работы: индивидуальные задания по разработке сетевой базы данных.

Тема 4.14. Алгоритмы маршрутизации

Лекции (2ч): Алгоритм Дейкстры определения кратчайшего пути в графе. Алгоритм вектора расстояний. Свойства, достоинства и недостатки этих алгоритмов. Проблема осцилляций И «счета до бесконечности» при изменении загрузки, решение этих проблем.

Семинары (2ч): Разбор материалов лекций. Формулировка задания по переносу реализации на случай многих клиентов и серверов.

Задания для самостоятельной работы: индивидуальные задания по разработке сетевой базы данных.

Тема 4.15. Подготовка математических текстов в системе TeX

Лекции (2ч): Основные концепции системы TeX для подготовки математических текстов.

Структура документа, основные команды, запись формул, управление шрифтами.

Семинары (2ч): Разбор материалов лекций. Реализация задания по переносу реализации на случай многих клиентов и серверов.

Задания для самостоятельной работы: индивидуальные задания по разработке сетевой базы данных.

Тема 4.16. Конструкции и команды TeX

Лекции (2ч): Разделы документа, автоматические нумерации, таблицы, макрокоманды.

Семинары (2ч): Разбор материалов лекций. Тестирование задания по переносу реализации на случай многих клиентов и серверов. Зачет.

VII. Используемые образовательные, научно-исследовательские и научно-производственные технологии:

А. Образовательные технологии: интерактивные лекции и семинары; решение типовых задач; выполнение творческого задания; дискуссии по теме занятий; активное обсуждение и оценка работы студентов в группе; самостоятельная работа, .

Б. Научно-исследовательские технологии: изучение литературы, а также научных и научно-популярных статей, блогов и лекций ведущих отечественных и зарубежных специалистов, представленным в Интернете,

VIII. Учебно-методическое обеспечение самостоятельной работы студентов, оценочные средства контроля успеваемости и промежуточной аттестации:

А. Учебно-методические рекомендации для обеспечения самостоятельной работы студентов, в том числе ссылки на методические материалы, размещенные на сайте кафедры:

Основное внимание должно уделяться выработке навыков самостоятельного исследования проблемы, выбору путей ее решения и надежной реализации этих путей в программном продукте, в частности, подбору тестовых примеров для отладки алгоритмов и проверке правильности работы всех составных компонент программы. Так, в 1 и 2 семестрах задания для самостоятельной работы имеют учебный характер, четко и однозначно сформулированы и имеют целью наработку технических навыков программирования с относительно небольшим вкладом в разработку алгоритма решения.

В 3 семестре задания уже имеют комплексный характер и «нечеткую» формулировку в том смысле, что студент должен сам составить строгие спецификации к своей

программе и реализовать их. Кроме этого обращается внимание на вопросы тестирования и верификации полученного программного кода.

Задачи 4 семестра приближаются к производственным. При общей формулировке и требованиях, они могут быть развиты в различных направлениях, например, в сторону повышения эффективности алгоритмов поиска в базе данных, либо в сторону повышения надежности и эффективности сетевых обменов, либо к внедрению разнообразных стратегий во взаимодействие клиента и сервера. При этом роль преподавателя состоит в моделировании поведения «заказчика», а студент выступает в роли «исполнителя», реализующего проект и разрабатывающего для него все конкретные решения и спецификации.

Б. Примерный список заданий для проведения текущей и промежуточной аттестации

Примерные задания для раздела 1.

Тема последовательности.

В следующих задачах предполагается, что в файле записана последовательность чисел неизвестной длины (возможно, пустая). Требуется за один просмотр файла и без запоминания последовательности в массиве определить требуемую характеристику последовательности.

Программа должна содержать функцию, которая получает в качестве параметра имя файла (или указатель на файл) и возвращает требуемое значение.

Функция `main` открывает необходимые файлы, проверяет успешность открытия, обращается к функции для вычисления результата и выводит результат в соответствующий файл.

1. Подсчитать среднее арифметическое чисел из последовательности.
2. Подсчитать среднее геометрическое чисел из последовательности.
3. Подсчитать среднее гармоническое чисел из последовательности.
4. Подсчитать количество чисел, больших предыдущего.
5. Определить есть ли в последовательности число X (для вещественных чисел --- с точностью E).
6. Определить номер последнего числа, равного X (для вещественных чисел --- с точностью E).
7. Определить все ли элементы последовательности равны между собой (для вещественных чисел --- с точностью E).
8. Определить является ли последовательность возрастающей, убывающей?
9. Определить удовлетворяют ли элементы последовательности данному трехчленному рекуррентному соотношению
10. Определить количество различных элементов целой неубывающей последовательности.
11. Определить общее количество элементов в постоянных участках целой последовательности.
12. Определить порядковый номер первого числа, равного максимуму по всей целой последовательности.
13. Определить номер последнего числа, равного минимуму по всей целой последовательности.
14. Определить количество чисел, равных минимуму из всей целой последовательности.
15. Найти величину максимального отклонения элементов последовательности от их среднего арифметического.
16. Найти количество возрастающих участков последовательности.
17. Найти сумму четных элементов во всех возрастающих участках целой последовательности.

18. Определить каких участков в последовательности больше --- возрастающих или невозрастающих.
19. Найти количество элементов в наибольшем постоянном участке целой последовательности.
20. Найти количество элементов в постоянном участке целой последовательности с наибольшей суммой элементов этого участка.
21. Найти длину возрастающего участка последовательности с наибольшим количеством элементов.
22. Найти наибольшую сумму возрастающего участка последовательности (т.е. максимум из сумм элементов по каждому возрастающему участку).
22. Найти среднее арифметическое локальных экстремумов последовательности.
23. Найти максимальное расстояние (количество элементов) между двумя соседними локальными минимумами последовательности.
24. Найти среднее арифметическое значений элементов целой последовательности, учитывая значения в постоянных участках только один раз.
25. Найти среднее арифметическое, взяв по одному элементу из каждого постоянного участка целой последовательности.
26. Найти максимальную сумму подряд идущих элементов последовательности.
27. Последовательность чисел представляет собой коэффициенты многочлена по возрастанию степеней. Вычислить многочлен и его производную в заданной точке x .
28. Последовательность чисел представляет собой коэффициенты многочлена по убыванию степеней. Вычислить многочлен и его производную в заданной точке x .

Тема массивы.

Решения следующих задач должны содержать функцию, которая получает в качестве параметров имя массива и его длину (или нескольких массивов, если этого требуют условия задачи) и выполняет необходимые действия.

При решении не разрешается создавать или резервировать в программе дополнительную память, соизмеримую по размерам с объемом исходных данных. То есть, нельзя создавать дополнительные массивы, если это явно не оговорено в задаче.

0. Все задачи из предыдущего списка (для последовательностей) могут быть переформулированы для массивов.
1. Симметричны ли значения элементов массива целых чисел?
2. Переставить элементы массива в обратном порядке.
3. Циклически сдвинуть элементы массива на одну позицию вправо.
4. Сравнить два неупорядоченных целочисленных массива A и B как числовые множества без повторения элементов: $A=B$ и $A \subset B$.
5. Для двух целочисленных массивов построить третий массив, являющийся их объединением как числовых множеств без повторения элементов. Указать длину получившегося массива.
6. Для двух целочисленных массивов построить третий массив, являющийся их пересечением как числовых множеств без повторения элементов. Указать длину получившегося массива.
7. Определить какое число встречается в массиве целых чисел наибольшее количество раз.
8. Удалить из целочисленного массива одинаковые значения, т.е. если какое-то значение встречается несколько раз (в разных местах массива), то оставить только первый такой элемент, а остальные удалить из массива. Оставшиеся элементы сдвинуть к началу массива, и указать их количество.
9. Сократить подряд идущие одинаковые элементы целочисленного массива до одного элемента. То есть, если в массиве встречается несколько одинаковых

- элементов, стоящих рядом, то оставить только один из них, а остальные удалить из массива. Оставшиеся элементы сдвинуть к началу массива, и указать их количество.
10. Удалить из массива все отрицательные значения, а оставшиеся уплотнить (сдвинуть) с сохранением исходного порядка к началу массива. Указать количество оставшихся значений.
 11. Циклически сдвинуть элементы массива на K позиций вправо с затратой $O(N)$ действий (N -длина массива)
 12. Каждый элемент $a[i]$ массива заменить на сумму элементов исходного массива вплоть до него самого включительно, т.е. от 0 до i -го.
 13. Каждый элемент массива заменить на полусумму его соседних элементов (кроме первого и последнего)
 14. Назовем x -отрезком группу подряд идущих элементов массива, каждый из которых равен x . Для заданного числа x заменить элементы каждого x -отрезка на полусумму элементов, прилегающих к этому отрезку справа и слева. Если x -отрезок расположен в начале или конце массива, считать второй крайний элемент равным нулю.
 15. Сгруппировать положительные элементы массива в его начале, а отрицательные -- в конце с сохранением их порядка.
 16. Назовем массив из N целых чисел счастливым, если существует такое $0 < k < N$, что сумма элементов с индексами от 0 до $k-1$ совпадает с суммой элементов с индексами от k до $N-1$. Определить является ли данный массив счастливым.
 17. Назовем массив из целых чисел плотным, если множество значений элементов массива полностью заполняет некоторый отрезок $[a, b]$ (рассматриваются целые значения). Определить является ли данный массив плотным.
 18. Получить массив биномиальных коэффициентов для степени N , последовательно вычисляя строки треугольника Паскаля (можно использовать только один массив).
 19. Элементы массива не убывают. Двоичным поиском определить позицию, где в этот массив можно вставить данное число x .
 20. Даны два неубывающих массива. Построить третий неубывающий массив, который является объединением первых двух (элементы могут повторяться).
 21. Выполнить следующее преобразование. Элементы с четными индексами сгруппировать в начале массива с сохранением их исходного порядка относительно друг друга, элементы с нечетными индексами сгруппировать в конце массива также с сохранением их исходного порядка.
 22. Выполнить следующее преобразование массива длины N . Элементы с индексами $i \leq [(N+1)/2]$ переместить на позиции с четными индексами с сохранением их исходного порядка относительно друг друга, оставшиеся элементы ($i > [(N+1)/2]$) разместить на позициях с нечетными индексами также с сохранением их исходного порядка. Т.е. начальная и конечная половины массива "перемешиваются" чередованием элементов.

Тема сортировки.

При решении следующих задач нужно реализовать отдельную функцию для сортировки массива чисел, функцию для проверки массива на упорядоченность, функцию для чтения массива из файла и функцию для генерирования случайного массива указанной длины.

Нужно составить тест для проверки трудоемкости алгоритмов сортировки. Для этого реализуется один из методов "медленной" сортировки, один из методов "быстрой" сортировки и сортировка с использованием библиотечной функции `qsort`. Для заданного N генерируется неупорядоченный массив из N чисел, который сортируется всеми тремя методами. Замеряется время работы каждой сортировки (например, с помощью функции `clock()` или любых других подходящих). Тест повторяется несколько раз с удвоением длины массива N .

1. Простая сортировка обмeнами.
2. Пузырьковая сортировка.
3. Сортировка просеиванием.
4. Вставка с последовательным поиском.
5. Сортировка слиянием..
6. Быстрая сортировка (quicksort).
7. Пирамидальная сортировка (heapsort)
8. Линейная сортировка (подсчетом) целого массива.
9. Сортировка целого массива группировкой с последовательным упорядочиванием битов.

Тема битовые операции и делимость.

1. Написать функции, записывающие 0 или 1 в указанный бит данного целого числа и оставляющие остальные биты без изменения.
2. Проверить четность количества единиц в двоичном представлении данного целого числа.
3. Найти первые N целых чисел, у которых младший байт является зеркальным отражением следующего байта.
4. Определить позицию самой старшей единицы в битовом представлении данного целого числа.
5. Написать функции, позволяющие работать с целым числом (long) как с ``массивом" четырех однобайтовых чисел (char).
6. Написать функцию, которая зеркально переворачивает битовое представление целого числа (аналогично перестановке массива в обратном порядке).
7. Написать функцию, которая циклически сдвигает битовое представление целого числа на указанное количество позиций вправо или влево (аналогично циклическому сдвигу массива).
8. Написать функцию, которая переставляет байты целого числа (int) в обратном порядке.
9. Возвести число в степень N за не более чем $2 \log_2 N$ умножений.
10. Вывести в файл все подмножества множества $\{1, \dots, N\}$.
11. Вывести в файл все k-элементные подмножества мн-ва $\{1, \dots, N\}$.
12. Найти наибольший общий делитель двух целых чисел (алгоритм Евклида).
13. Определить четность произвольной перестановки N чисел.
14. Вычислить первые N простых чисел.
15. Разложить натуральное число на простые множители
16. Вывести значение целого числа N в ``словесной форме".
17. Вычислить представление числа $1/N$ в виде десятичной дроби (начало и период) (то же для числа M/N).

Тема геометрические задачи

В следующих задачах предполагается, что в файле записано несколько пар чисел, которые можно рассматривать как координаты множества точек на плоскости или как координаты множества концов отрезков на прямой.

1. Множество точек определяет ломаную. Имеет ли она самопересечения?
2. Множество точек определяет многоугольник. Является ли он выпуклым?
3. Множество точек определяет многоугольник. Определить угол, под которым данный многоугольник виден из заданной точки.
4. Множество точек определяет многоугольник. Для данной точки определить где она расположена относительно этого многоугольника: внутри, снаружи, на границе.
5. Множество точек определяет выпуклый многоугольник.

Найти его минимальный и максимальный диаметры (вписанная и описанная окружности).

6. Дано множество отрезков на прямой. Покрывает ли объединение этих отрезков заданный отрезок $[a, b]$?
7. Два множества точек задают два многоугольника (без самопересечений, но не обязательно выпуклые). Определить расстояние между этими многоугольниками
8. Два множества точек задают два выпуклых многоугольника, не лежащих один внутри другого. Определить расстояние между этими многоугольниками с линейной оценкой трудоемкости по суммарному количеству вершин.
9. Дано множество точек. Найти центр и радиус минимального круга, который содержит все эти точки.
10. Дано множество отрезков на прямой. Выбрать из него и вывести те отрезки, объединение которых дает отрезок наибольшей длины.
11. Даны центры равномерно растущих кругов на плоскости. При столкновении друг с другом столкнувшиеся круги прекращают свой рост. Найти радиусы кругов, когда процесс роста остановится полностью.
12. Дано множество точек на плоскости. Построить выпуклую оболочку этого множества.
13. Множество точек определяет выпуклый многоугольник. Построить многоугольник, который получится, если линию, задающую каждую сторону, отодвинуть в перпендикулярном ей направлении на величину h .

Примерные задания для раздела 2.

Тема простейшие вычислительные задачи

В следующих задачах требуется оформить решение в виде функции, получающей в качестве параметра указатель на функцию `double (*f)(double x)`

1. Вычислить корень уравнения $f(x)=0$ на данном отрезке с заданной точностью ϵ
 - А. методом деления пополам;
 - Б. методом Ньютона (касательных);
 - В. методом хорд (секущих).

Написать тест, в котором вывести количество итераций и сравнить результат с точным ответом.

2. Вычислить интеграл от данной функции $f(x)$ на данном отрезке $[a, b]$ по составной формуле
 - А. прямоугольников с N отрезками;
 - Б. трапеций с N отрезками;
 - В. Симпсона с N отрезками;
 - Г. Гаусса (двухточечной) с N отрезками.

Сравнить с точным ответом.

3. Вычислить минимальное значение данной функции $f(x)$ с заданной точностью ϵ методом построения параболы с поиском трех точек с помощью
 - А. измельчения геометрической прогрессии;
 - Б. метода золотого сечения;
 - В. измельчением равномерной сетки.

4. Написать программу вычисления значения одной из элементарных функций (\sin , \cos , \exp , \log) в заданной точке x и с заданной точностью ϵ суммированием ряда Тейлора. Сравнить с точным ответом

5. Написать программу, которая по значениям $x_1 \dots x_n$ и $y_1 \dots y_n$ вычисляет в данной точке x приближенное значение функции с помощью
 - А. интерполяционного многочлена Лагранжа;
 - Б. кусочно-линейной интерполяции;

- В. кусочно-квадратичной интерполяции;
- Г. Эрмитовой интерполяции (по $f(x_i)$ и $f'(x_i)$);
- Д. наилучшего среднеквадратического приближения линейной функцией.

Тема матрицы

1. Реализовать функции умножения матрицы на вектор, умножения двух прямоугольных матриц, функции для создания и вывода матриц.
2. Определить ранг вещественной $N \times M$ матрицы.
3. Найти определитель вещественной $N \times N$ матрицы.
4. Для квадратной матрицы вычислить обратную.
5. Решить систему линейных уравнений.

Тема обработка текстового файла

1. Реализовать стандартные функции работы со строками `strcmp`, `strcpy`, `strstr`, `strcat`, `strset` и др.
2. Заменить в файле каждую последовательность заданных одинаковых символов на один такой символ.
3. Заменить всюду в файле один заданный набор символов на другой (с учетом разницы в их длине).
4. Вывести все слова из данного файла в другой файл в порядке их появления по одному слову на строке.
5. Вывести номера строк исходного файла и номер позиции в строке, где встречается заданное слово.
6. Определить максимальную, минимальную и среднюю длину слов из данного файла а также частоту (процент) появления каждого символа.
7. Вывести все слова из данного файла в алфавитном порядке.
8. Разрезать "длинные" строки в файле по пробелам на более короткие (не более заданной длины).
9. Удалить из файла часть текста между двумя ``скобками'', где скобка --- это заданный набор символов. Например, убрать из файла комментарии в стиле C.
10. Реализовать инструкцию типа '#include', т.е. вставить содержимое файла 'filename' в то место файла, где встречается строка '#include filename'.
11. Реализовать инструкции типа '#define' и '#undef', т.е. выполнить указанные подстановки в области их задания.
12. Реализовать инструкции типа '#ifdef - #else - #endif', т.е. оставить в файле требуемый текст в зависимости от условия.
13. Отформатировать абзацы текста в заданных границах и с красной строкой (без переноса слов). Абзац --- фрагмент текста между пустыми строками.

Тема сложная задача

Для заданной точности ϵ и различных значений параметра A найти все решения нелинейного уравнения, содержащего интегралы с особенностями. Например,

$$\int_0^{\sqrt{2\pi}} e^{-t^2/2} dt = \alpha.$$

Задания различаются формой функции с интегралом.

Примеры контрольных работ для 1 курса

Контрольная по теме последовательности.

Задание представляет собой строгую формулировку задач на тему последовательности, приведенные выше, например

Найти номер последнего числа, равного X . Первое число во входном файле - X , далее исходная последовательность целых чисел.

Корректный результат --- целый номер (отсчет от 1), при отсутствии X в последовательности или для пустой последовательности ответ 0.

Исходные данные (последовательность чисел) задаются в файле input.txt, при этом количество чисел в последовательности заранее неизвестно. Ответ надо вывести в указанной форме в файл "output.txt".

Программа должна решать указанную задачу последовательно читая числа последовательности и вычисляя требуемый результат за один просмотр данной последовательности.

При реализации алгоритма нельзя создавать ни в функции "main", ни в других функциях рабочие массивы, длина которых явно или неявно зависела бы от длины исходных данных.

Программа должна содержать отдельную функцию для обработки последовательности, получающую в качестве параметров указатель на предварительно открытый файл с данными, а также, возможно, другие параметры, необходимые для решения конкретной задачи; при этом открытие и закрытие файла данных, а также вывод результата должен происходить вне этой функции, например, это можно делать в функции "main".

Если программа не может корректно решить задачу (файлы не открылись, при чтении возникли ошибки, количество чисел в последовательности не позволяет корректно найти результат и т.п.) то результат не вычисляется, и функция "main" должна вернуть значение -1. При успешном решении задачи функция "main" должна вернуть 0.

Контрольная по теме массивы.

Задание представляет строгую формулировку задач на тему массивы, приведенные выше, например

Переставить элементы массива вещественных чисел в обратном порядке. Результат является корректным при наличии в массиве хотя бы одного числа.

Ответ: последовательно записанные элементы преобразованного массива.

Исходные данные (набор чисел) задаются в файле "input.txt", при этом первое число в файле задает длину массива чисел, а последующие числа есть элементы массива, который требуется обработать. Результат решения задачи надо вывести в указанной форме в файл "output.txt".

Программа должна решать указанную задачу, динамически создавая в функции \verb"main" единственный массив для хранения данных на требуемое количество элементов и заполняя его числами, записанными в файле.

При реализации алгоритма нельзя создавать ни в функции "main", ни в других функциях дополнительные рабочие массивы, длина которых явно или неявно зависела бы от длины исходных данных.

Программа должна содержать отдельную функцию для обработки массива получающую в качестве параметров указатель на начальный элемент массива и длину массива, а также, возможно, другие параметры, необходимые для решения конкретной задачи; при этом функция, осуществляющая обработку массива, не должна заниматься его вводом/выводом.

Если программа не может корректно решить задачу (файлы не открылись, при чтении возникли ошибки, количество чисел в файлах не соответствует условию задачи, отказы в выделении памяти и т.п.) то результат не вычисляется, и функция "main" должна вернуть значение -1. При успешном решении задачи функция "main" должна вернуть 0.

Контрольная по теме матрицы.

Задание состоит в реализации простейшего преобразования элементов матрицы, например

В файле input.txt записаны два целых числа $n > 1$ и $m > 1$ - число строк и столбцов матрицы, а далее по строкам записана сама матрица вещественных чисел. Требуется прочитать эту матрицу из файла, создав массив строго на указанное количество чисел. Далее требуется написать функцию, которая получает эту матрицу в качестве параметра и выполняет с ней следующее преобразование: находит строки, имеющие максимальную и минимальную суммы элементов, и меняет эти строки местами.

При написании алгоритма можно считать, что такие строки определяются однозначно. Вызовом этой функции нужно преобразовать исходную матрицу и затем вывести ее в файл `output.txt` по строкам (выводить только элементы матрицы, без указания размерности).

Контрольная по теме обработка текстов

Задание состоит в простейшем преобразовании текстовых файлов, например,

В файле `input.txt` записан некоторый текст, общая длина которого заранее не известна (можно считать, что длина каждой строки не превосходит 4000 символов).

Требуется поставить в начале каждой строки (в том числе и пустой) ее порядковый номер, отделяя его одним дополнительным пробелом от содержимого исходной строки (нумерация строк начинается с единицы). Преобразованный текст нужно вывести в файл `output.txt`.

Примерные задания для раздела 3.

Тема реализация простейших классов.

Требуется разработать и реализовать один или несколько простейших классов на C++ и написать тесты для проверки работы этих классов.

Требования к реализации

0. Если кто еще не умеет, то должен научиться писать реализацию в нескольких файлах, использовать собственные заголовочные файлы, уметь составлять простейшие make-файлы и использовать их для сборки, а также уметь собирать и отлаживать программу из командной строки (в том числе с использованием `gdb`).

1. Все реализации должны удовлетворять согласованным неформальным интерфейсам и иметь заголовочный `.h` файл с прокомментированным описанием методов класса. Разработка формального описания классов предоставляется студенту, при этом должны быть предусмотрены способы реакции и оповещения о некорректных или ошибочных ситуациях.

2. К заданию должны быть написаны тесты, т.е. отдельные программы которые вызывают требуемые функции с разнообразными наборами данных и (желательно автоматически) проверяют правильность результата.

Возможные варианты задания

Данные формулировки являются только примерами, вполне возможна их модификация, расширение, уточнение и пр.

A. Класс вектор в R^3 с основными операциями аналитической геометрии.

Конструктор по умолчанию,
конструктор по трем координатам,
конструктор копирования,
= присваивание,
- противоположный вектор,
+- сумма, разность,
* умножение на число,
* скалярное произведение,
% векторное произведение,
длина вектора,
нормировка, и т.д.

B. Класс динамический массив вещественных чисел.

Конструктор по умолчанию,
конструктор копирования,
конструктор по заданному набору (обычному массиву и длине),
изменение размера,

= присваивание,
+ добавление другого массива,
добавление в конец,
вставка по индексу,
удаление по индексу,
[] доступ по индексу,
получить размер,
получить указатель на набор данных.

C. Классы точка (вектор) и отрезок в R^2 .

Конструктор по умолчанию,
конструктор копирования,
конструктор по координатам,
конструктор отрезка по двум точкам.
Пересечение отрезков,
сдвиг точки, проекции точек и отрезков на оси или на отрезки,
симметрии, повороты и т.п.

D. Классы точка и треугольник на плоскости.

Конструктор по умолчанию,
конструктор копирования,
конструктор по координатам,
конструктор треугольника по трем точкам.
Периметр, площадь, основания высот, медиан, биссектрис, центры
вписанной и описанной окружностей, углы при вершинах и т.д.

E. Классы точка и треугольник на плоскости.

Конструктор по умолчанию,
конструктор копирования,
конструктор по координатам,
конструктор треугольника по трем точкам.
Разнообразные преобразования на плоскости
(сдвиг, поворот, гомотетия, аффинное, и т.п.). Сравнение треугольников на
равенство, подобие и т.д.

F. Класс рациональное число a/b с рациональной арифметикой.

Конструктор по умолчанию,
конструктор копирования,
конструктор по вещественному числу и по числителю и знаменателю.
Арифметические операции, сокращение дроби, преобразование к `double` и т.п.

G. Класс целое число с "длинной" арифметикой

Конструктор по умолчанию,
конструктор копирования,
конструктор по целому числу.
Арифметические операции (деление --- целая часть),
преобразование к `double` и т.п.

H. Класс "небольшое" множество целых чисел (динамический массив
с возможностью линейного поиска, добавления, удаления конкретного значения).

Конструктор по умолчанию,
конструктор копирования,
конструктор по массиву целых чисел.
Добавление и удаление значения, поиск, объединение, пересечение, разность множеств.

I. Класс ASCII строка.

Конструктор по умолчанию,
конструктор копирования,
конструктор по массиву символов.
Конкатенация строк, добавление в конец, поиск подстроки и символа,
длина строки, вырезание подстроки и т.п.

График выполнения задания

неделя 1: разработка и согласование описания (интерфейса) класса.

неделя 2: реализация и первичный тест работоспособности.

неделя 3: доработка более сложных методов и полный тест.

Тема списочные структуры

Требуется разработать и реализовать параметризованный класс списочного типа. Класс должен содержать конструктор без параметров, конструктор копирования. Набор методов класса должен соответствовать идеологии работы со списками, т.е. должно быть введено понятие текущей позиции и реализованы функции перемещения этой позиции и доступа и изменения списка в окрестности текущей позиции.

В классе должны быть реализованы оператор присваивания и предусмотрено переопределение одной-двух операций по смыслу задачи (например, сложение, как объединение множеств данных). В классе должна быть реализована операция сортировки хранящегося набора данных, а также распечатки всего набора данных в файл или на экран в наглядной форме.

Контроль за возможностью доступа к хранимым значениям и за другими возможными отказами выполняется через механизм исключений (throw --- try-catch)

Релизация тестируется для двух-трех различных типов хранимых объектов - простого и "сложного".

Варианты структур данных

- A. Однонаправленный список
- B. Двухнаправленный список
- C. Кольцевой однонаправленный список
- D. Кольцевой двухнаправленный список
- E. Динамический массив как список блоков
- F. Стек на базе списка
- G. Очередь на базе списка
- H. Дек на базе кольцевого списка

Типы объектов

Работа параметризованной реализации проверяется и тестируется на различных типах данных:

1. базовые типы (числа int, double)
2. простейшие структурные типы (точка, вектор R^3)
3. сложные классы --- классы из своего первого задания.

Для тестирования в списочном классе нужно предусмотреть параметризованную функцию распечатки содержимого списка, и соответственно реализовать функции распечатки для каждого конкретного используемого типа.

Работа над заданием предполагает еженедельный контроль по следующему графику (естественно, его можно выполнять с опережением)

График выполнения задания

неделя 1: разработка и согласование описания (интерфейса) класса.

неделя 2: реализация и первичный тест, проверка работы механизма исключений

неделя 3: тест на сложных типах данных

Тема деревья и хеш-множества

Задания этого раздела предполагают реализацию множеств с быстрым доступом (быстрые деревья или хеширование). Соответственно тесты должны показывать, что использование таких реализаций действительно эффективно, т.е. должны предусматривать работу с большими наборами данных. Кроме этого, для деревьев нужно проверить, что они действительно имеют логарифмическую глубину. То есть, написать отдельную функцию, которая вычисляет глубину дерева, и протестировать как меняется глубина при добавлении большого количества элементов.

Для хеш-множеств нужно будет подобрать хеш-функцию, наилучшим образом рассеивающую элементы множества. Таким образом, надо предусмотреть вывод соответствующей информации (длины хеш-списков, количество коллизий и т.п.) при работе с вашей реализацией.

Отображения

Требуется построить параметризованный класс, который реализует отображение где ключом является строка, а значением --- некоторый другой класс. Интерфейс отображения должен поддерживать следующие операции:

- добавить пару (ключ, значение);
- искать значение по указанному ключу;
- удалить ключ и соответствующее значение;
- получить количество хранящихся ключей;
- итератор по множеству ключей и значений.

В качестве примеров и тестов можно рассмотреть отображение строк на целые числа и отображения строк на строки (как в словаре). Тесты должны предусматривать возможность загрузки множества из файла или автоматической генерации.

Варианты заданий различаются способом реализации отображения.

Сбалансированное AVL дерево поиска.

Красно-черное дерево поиска.

B-дерево.

Хеш-множество по методу списков.

Хеш-множество по методу линейных проб.

Упорядоченное множество строк

Такое множество позволяет выбирать упорядоченные подмножества и определять следующий или предыдущий элементы для какого-либо элемента множества (строки). Реализация должна поддерживать следующий интерфейс

- добавить строку в множество;
- удалить строку;
- искать данную строку;
- получить количество элементов в множестве;
- итератор по части множества.

Это означает, что после указания некоторой строки, мы можем перебирать последовательно упорядоченные строки множества от указанной в одну или другую сторону.

Для тестов следует предусмотреть загрузку из файла и автоматическую генерацию множества. Варианты задания различаются основой реализации множества.

Сбалансированное дерево для хранения указателей на строки.

Красно-черное дерево поиска для хранения указателей на строки.

B-дерево для хранения указателей на строки.

Вопросы и задачи к экзамену

На экзамен выносятся следующие темы лекционного курса.

1. Стек, дек, очередь. Непрерывные реализации на базе массива.
2. Ссылочные реализации списков (одно- и двунаправленного).
3. Бинарное дерево поиска. Реализация; процедуры добавления, удаления, обхода.
4. Произвольное дерево. Реализация; процедуры добавления, удаления, обхода.
5. Сбалансированное (AVL) бинарное дерево; процедуры добавления, удаления.
6. Теорема о глубине сбалансированного дерева.
7. Красно-черное дерево; процедуры добавления, удаления.
8. Теорема о глубине красно-черного дерева.
9. B-дерево; процедуры добавления, удаления. Трудоемкость операций с B-деревом.
10. Битовая реализация множества.
11. Хеширование. Примеры хеш-функций.
12. Хеш-множество на базе массива списков.
13. Хеш-множество по методу последовательных проб.
14. Оценка среднего количества проб при поиске и добавлении.
15. Совершенная (perfect) хеш-функция; определения, пример построения.
16. Файловая система типа FAT (DOS, Windows). Атрибуты файлов. Области fat, dir, data.
17. Файловая система типа EXT (Unix). Области block group, superblock, descriptor table, inode
18. Файловая система типа NTFS. Файл как набор атрибутов. Файл MFT, метафайлы.
19. Организация ссылок на кластеры в нерезидентном атрибуте Data.
20. Теорема об оценке снизу для трудоемкости сортировки сравнениями.
21. Быстрая сортировка (quicksort); оценка средней трудоемкости.
22. Пирамидальная сортировка (сортировка кучей, heapsort); оценка трудоемкости.
23. Сортировка подсчетом с "линейной" оценкой трудоемкости.
24. Формальные грамматики. Нормальная форма Бэкуса--Наура. Примеры.
25. Понятие о LR разборах. Процедура LR(1) разбора, операции сдвига и свертки.
26. Модельный компилятор модельного алгоритмического языка.
27. Разработка входного языка и архитектуры модельного стекового процессора.
28. Реализация модельного компилятора для случая линейных программ.
29. Компиляция условных и безусловных переходов.
30. Реализация вызовов функций.

Форма проведения экзамена:

1. Письменный ответ на билет с вопросами, относящимися к темам программы.
2. Практическая задача на реализацию алгоритмов из лекционного курса.

Структура письменного билета

1. Определение понятия, формулировка утверждения.
2. Описание интерфейса представления данных на языке C++.
3. Вопросы о сути алгоритмов, доказательств, схем представлений данных.
4. Задачи по выполнению или оценке свойств алгоритмов на модельном наборе данных.

Содержание практической задачи:

Даны некоторые заготовки, создающие некоторые структуры данных с заданными свойствами, например, деревья или списки. Требуется выполнить обход или поиск в этих структурах с целью определения или вычисления некоторых характеристик записанных

там наборов данных. Например, найти в бинарном дереве все вершины, нарушающие условие AVL сбалансированности.

На выполнение письменной части отводится 1 академический час, на реализацию программ 2 академических часа.

Пример письменного билета:

1. Определение B-дерева.
2. Запишите описание класса, реализующего однонаправленный список строк (char*).
3. Для чего предназначена область FAT в соответствующей файловой системе?
4. В чем суть операций сдвига и свертки в процессе LR разбора?
5. Изобразите состояние массива 7,2,3,6,1,4,5 после выстраивания пирамиды при сортировке по возрастанию методом пирамидальной сортировки.
6. Опишите формальной грамматикой язык (множество) десятичных числовых констант в языке C.

Пример практической задачи

Файл "tree.o" содержит откомпилированные заготовки для работы бинарным неупорядоченным деревом, в каждой вершине которого хранятся два целых значения int value, balance. Заготовки содержат функции для создания дерева и распечатки поддеревьев. Файл "tmain.cpp" иллюстрирует работу этих функций. Прочитайте комментарии в файлах "tmain.cpp" и "tree.h". Используя данные заготовки напишите программу решения следующей задачи:

Назовем балансом поддерева число, равное разности между максимальным и минимальным значениями в вершинах этого поддерева (включая корень). Требуется проставить во всех вершинах балансы соответствующих поддеревьев и распечатать полученное дерево. После этого требуется определить какие значения балансов встречаются в дереве и распечатать только различные эти значения в порядке возрастания (т.е., например, если балансы в дереве есть 1,5,1,3,2,2, то печатается 1,2,3,5).

Критерии оценивания ответа:

Теоретический билет считается зачтенным, если в нем дано не менее 4 правильных ответов на заданные вопросы. Верные факты и рассуждения, не имеющие прямого отношения к заданным вопросам, правильным ответом не считаются. Практическая задача должна выдавать правильный ответ и нормально завершаться (без ``падения"). Итоговая оценка формируется на основе зачетных оценок по заданиям семестра, количества правильных ответов на теоретический билет и результатов решения практической задачи.

Примерные задания для раздела 4.

Требуется реализовать сетевое взаимодействие между сервером, моделирующем базу данных, и несколькими клиентами, выполняющими запросы к серверу.

Задание 1 предполагает простое взаимодействие клиента с сервером в ``режиме чтения". Это означает, что сервер при запуске загружает свою базу данных из заданного файла, а клиенты обращаются к серверу только для получения информации по своим запросам. При этом сама логика взаимодействия предполагает ``однократный" принцип работы клиента. То есть соединение создается отдельно под каждый конкретный запрос и разрывается после его обслуживания.

Обычно результатом запроса является несколько записей, либо таблица, построенная на основе нескольких записей. Эти данные должны быть отсортированы клиентом в зависимости от сути конкретной задачи, например, по алфавиту, по возрастанию, по датам и т.п. (обсуждается в каждом конкретном случае).

Сроки и этапы

- генерация случайного массива исходных данных для тестирования --- 1 неделя
- реализация базы данных без сетевой части (загрузка, функции поиска, функции добавления, удаления, изменения) --- 2 недели
- реализация языка запросов к базе --- 2 недели
- реализация сетевого взаимодействия --- 2 недели

Задание 2 предполагает развитие задания 1 в направлении пакетного режима работы и возможности редактирования базы данных со стороны клиента. Редактирование базы предполагает изменение существующей записи, добавление и удаление записей. При этом следует реализовать систему взаимоисключения, так чтобы запись, редактируемая одним клиентом, не могла быть изменена другим клиентом, а также чтобы редактирование базы данных не препятствовало обработке запросов от других клиентов.

Конкретные операции редактирования записей обсуждаются в рабочем порядке для каждой конкретной задачи. Пакетный режим означает, что клиент может автономно выполнить серию запросов, записанных, например, в текстовом файле, и обработать полученные ответы, сохранив их также в каких-то файлах.

Сроки и этапы

- реализация механизма блокировки записей базы при выполнении запросов --- 1 неделя
- реализация пакетного режима работы клиента по сценарию из файла --- 2 недели

Контрольная работа --- простейшее взаимодействие клиента и сервера путем многократных запросов и ответов.

Как пример задания --- сервер ``загадывает" целое число, а клиент пытается его ``отгадать" методом деления пополам (сервер отвечает больше или меньше кго число, чем то, которое предложил клиент).

Задание 3 предполагает активное взаимодействие нескольких серверов и клиентов. Имеется в виду перенаправление запросов, регистрация клиентов на сервере, рассылка сообщений, моделирование почты, формула, чата и т.п.

Сроки и этапы

- реализация регистрации и авторизации клиентов на сервере --- 2 недели
- реализация распределенной базы данных --- 2 недели

Контрольная работа --- простейшее взаимодействие нескольких клиентов и серверов по типу почты, формула, чата, DNS и т.п.

База данных на стороне сервера представляет собой простой массив записей, соответствующих тематике задачи. При запуске сервер считывает записи базы данных из текстового файла, где также для простоты может быть заранее указано общее количество записей. Все операции поиска выполняются для простоты последовательным просмотром данного массива. Использование более сложных структур для хранения данных, как правило, не предполагается, но может быть добавлено по желанию в отдельных случаях и в отдельных задачах.

На сервере должна быть предусмотрена операция сохранения его базы данных в файл в аналогичном текстовом виде для последующего использования.

В качестве примера базы данных здесь приводится только один возможный вариант:

База Студент.

Запись о студенте содержит 4 поля:

```
struct { char name[64]; int group; double rating; char phone[16];};
```

Запрос:

```
select name=[min,max] group=[min,max] rating=[min,max]
```

должен возвращать все записи, удовлетворяющие указанным критериям.

Отдельные поля могут отсутствовать в запросе. Тогда они не проверяются при выборке.

Другие базы описываются аналогично на основе других предметных областей – библиотека, расписание, ежедневник, банк, магазин и т.д.

IX. Учебно-методическое и информационное обеспечение дисциплины:

А. основная литература:

Валединский В.Д., Пронкин Ю.Н.

Вычислительные системы и программирование. Схемы хранения данных.

М.: Изд-во ЦПИ при механико-математическом ф-те МГУ им. М.В.Ломоносова, 2006.

Валединский В.Д., Пронкин Ю.Н.

Вычислительные системы и программирование. Организация вычислительных систем.

М.: Изд-во ЦПИ при механико-математическом ф-те МГУ им. М.В.Ломоносова, 2006.

Кнут Д. Искусство программирования. Т. 1 - 3. М., СПб., Киев: ИД "Вильямс", 2000.

Кормен Т., Лейзерсон Ч., Ривест Р.

Алгоритмы, построение и анализ. М.: МЦНМО, 2000.

Абрамов С.А. и др. Задачи по программированию. — М.: Наука, 1988.

Вирт Н. Алгоритмы и структуры данных. — М.: Мир, 1989.

Вирт Н. Алгоритмы + структура данных = программы. — М.: Мир, 1985.

Семенов Ю. А.

Алгоритмы телекоммуникационных сетей. ч. 1. Алгоритмы и протоколы каналов и сетей передачи данных. М.: Изд-во Бином, 2007, 637 с.

Семенов Ю. А.

Алгоритмы телекоммуникационных сетей. ч. 2. Протоколы и алгоритмы маршрутизации в internet. М.: Изд-во Бином, 200, 829 с.

Б. Дополнительная литература:

Ахо, Сети Р., Ульман Дж.

Компиляторы: принципы, техника реализации и инструменты. М., 2001.

Дейтел Г.

Введение в операционные системы. М.: Мир, 1987.

Синицын С. В., Налютин Н. Ю. Верификация программного обеспечения. М.: Изд-во Бином, 2008, 368 с.

Х. Материально-техническое обеспечение дисциплины:

А. Помещения:

Дисплейные классы 445, 446 во 2 учебном корпусе

Б. Оборудование:

Персональные компьютеры (52 станции), сервер, сетевая аппаратура и программное обеспечение.

В. Иные материалы:

Программное обеспечение: OS LINUX, компилятор GNU GCC,

Автор _____ Валединский В.Д.

Программа утверждена на заседании кафедры,
протокол № __16__ от __15 мая_2013г.

Заведующий кафедрой
Проф. Г.М.Кобельков