ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет автоматики и вычислительной техники

“УТВЕРЖДАЮ”

Декан АВТФ

профессор, д.т.н. Гужов Владимир Иванович

“___”_____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Программирование на языке высокого уровня

ООП: направление 230100.62 Информатика и вычислительная техника

Шифр по учебному плану: ОПД.Ф.2

Факультет: автоматики и вычислительной техники очная форма обучения

Курс: 2, семестр: 3 4

Лекции: 54

Практические работы: - Лабораторные работы: 36

Курсовой проект: 4 Курсовая работа: 3 РГЗ: -

Самостоятельная работа: 145

Экзамен: 3 4 Зачет: -

Всего: 249

Новосибирск

2011

Рабочая программа составлена на основании _Государственного образовательного стандарта высшего профессионального образования по направлению (специальности): 552800 Информатика и вычислительная техника.(№ 35 тех/бак от 13.03.2000)

ОПД.Ф.2, дисциплины федерального компонента

Рабочая программа обсуждена на заседании кафедры Вычислительной техники протокол № 6 от 31.08.2011

Программу разработал

доцент, к.т.н. Васюткина Ирина Александровна

Заведующий кафедрой

профессор, д.т.н. Губарев Василий Васильевич

Ответственный за основную образовательную программу

профессор, д.т.н. Губарев Василий Васильевич

профессор, д.т.н. Фроловский Владимир Дмитриевич

1. Внешние требования

Таблица 1.1

Шифр дисциплины	Содержание учебной дисциплины	Часы
ОПД.Ф.2	ОПД.Ф.05 Программирование на языке высокого уровня: основные этапы решения задач на ЭВМ; критерии качества программы; жизненный цикл программы; постановка задачи и спецификация программы; способы записи алгоритма; программа на языке высокого уровня; стандартные типы данных; представление основных управляющих структур программирования; теорема структуры и структурное программирование; анализ программ; утверждения о программах; корректность программ; правила вывода для основных структур программирования; инвариантные утверждения; процедуры и функции; массивы; утверждения о массивах; записи; файлы; индуктивные функции на последовательностях (файлах, массивах); динамические структуры данных; линейные списки: основные виды и способы реализации; линейный список как абстрактный тип данных; модульные программы; рекурсивные определения и алгоритмы; программирование рекурсивных алгоритмов; способы конструирования и верификации программ. Квалификационные требования ГОСВПО по направлению 231000 - "Информатика и вычислительная техника" спнциальность 230100.62 бакалавр вычислительной техники и технологий: Для компетентного и ответственного решения профессиональных задач бакалавр: готов участвовать во всех фазах проектирования и разработки объектов профессиональной деятельности; готов участвовать в разработке всех видов документации на программные, аппаратные и программно-аппаратные комплексы; способен использовать современные методы, средства и технологии разра-ботки объектов профессиональной деятельности; готов участвовать в проведении научных исследований и выполнении тех-нических разработок в своей профессиональной области;	**249**

	способен изучать специальную литературу и другую научно-техническую информацию, достижения отечественной и зарубежной науки и техники в области своей профессиональной деятельности; умеет на научной основе организовать свой труд, владеет современными информационными технологиями, применяемыми в сфере его профес-сиональной деятельности; способен в условиях развития науки и изменяющейся социальной прак-тики к переоценке накопленного опыта, анализу своих возможностей, уме-ет приобретать новые знания, используя современные информационные об-разовательные технологии; Бакалавр должен знать: методические и нормативные материалы по проектированию и разработке объектов профессиональной деятельности; технологию проектирования и разработки объектов профессиональной дея-тельности; перспективы и тенденции развития информационных технологий; технические характеристики и экономические показатели лучших отечественных и зарубежных образцов объектов профессиональной деятельно-сти; порядок, методы и средства защиты интеллектуальной собственности; методы анализа качества объектов профессиональной деятельности; современные средства вычислительной техники, коммуникаций и связи; основные требования к организации труда при проектировании объектов профессиональной деятельности; правила, методы и средства подготовки технической документации.	

2. Особенности (принципы) построения дисциплины

Таблица 2.1

Особенности (принципы) построения дисциплины

Особенность (принцип)	Содержание
Основания для введения дисциплины в учебный план по направлению или специальности	Дисциплина "Программирование на языке высокого уровня." включена в учебный план по направлению 230100 (552800) "Информатика и вычислительная техника" (бакалавр) в цикл "Общепрофессиональные дисциплины" ОПД.Ф.2.
Адресат курса	Группы специальности 230100.62 (552800) "Информатика и вычислительная техника" бакалавр техники и технологий
Основная цель (цели) дисциплины	Дисциплина изучается после освоения программирования на языке С. В основу курса положены следующие принципы: - основной целью курса является изучение и освоение принципов и приемов объектно-ориентированного подхода при проектировании программных систем; - изучение этапов жизненного цикла программного обеспечения; - понятия качества и надежности разрабатываемого ПО. - Дисциплина охватывает очень большой объем материала и может читаться с усилением акцентов на разных разделах, в зависимости от уровня подготовки студентов и их пожеланий. - Дисциплина рассчитана на студентов, владеющих приемами программирования и знающих язык программирования высокого уровня С. - Возможно самостоятельное изучение дисциплины студентами по индивидуальному графику в случае достаточной подготовки студента и при согласовании графика с преподавателем. - Курс имеет практическую часть - лабораторные работы (36 часов в 3 и 4 семестре). На лабораторных работах студенты реализуют реальные программы с использованием теоретических положений, изученных на лекциях с использованием реальных сред разработки. - Курсовая работа выполняются по индивидуальному заданию. Примерные варианты заданий приведены в программе.
Ядро дисциплины	Требования по ГОСВПО к обязательному минимуму содержания по направлениюю 654600 - "Информатика и вычислитнльная техника"

	ОПД.Ф.05 Программирование на языке высокого уровня: основные этапы решения задач на ЭВМ; критерии качества программы; жизненный цикл программы; постановка задачи и спецификация программы; способы записи алгоритма; программа на языке высокого уровня; стандартные типы данных; представление основных управляющих структур программирования; теорема структуры и структурное программирование; анализ программ; утверждения о программах; корректность программ; правила вывода для основных структур программирования; инвариантные утверждения; процедуры и функции; массивы; утверждения о массивах; записи; файлы; индуктивные функции на последовательностях (файлах, массивах); динамические структуры данных; линейные списки: основные виды и способы реализации; линейный список как абстрактный тип данных; модульные программы; рекурсивные определения и алгоритмы; программирование рекурсивных алгоритмов; способы конструирования и верификации программ.
Связи с другими учебными дисциплинами основной образовательной программы	Дисциплина изучается после курса "Информатика" . Полученные в дисциплине знания принципов объектно-ориентированного программирования и методов разработки ООП программ являются основой для дисциплины "Технологии программирования"
Требования к первоначальному уровню подготовки обучающихся	Дисциплина изучается после курса "Информатика" и предполагает знание языка программирования С и базовых знаний и навыков по алгоритмизации.
Особенности организации учебного процесса по дисциплине	

3. Цели учебной дисциплины

Таблица 3.1

После изучения дисциплины студент будет

иметь представление	
1	о технологии разработки объектно-ориентированных программ
2	о повторном использовании унаследованного кода
19	об архитектурных особенностях ПК
20	о характере программ, разрабатываемых на языке ассемблере
21	об использовании инструментальных средств разработки и отладки ассемблерных программ
22	о системе команд 32-битных процессоров i80x86 и принципах их двоичного кодиро-вания
знать	
3	особенности языка C++
4	понятия информационных технологий и место объектно-ориентированных технологий в информационных технологиях
5	особенности информационных технологий, основанных на объектно-ориентированных языках
6	какие программные средства используются для построения объектно-ориентированных программ на C++
7	применение объектно-ориентированных библиотек ввода-вывода
8	назначение и особенности библиотеки STL
9	преимущества использования объектно-ориентированного подхода при создании сложных программных продуктов
23	этапы создания программы на языке ассемблера (жизненный цикл программы)
24	программную модель 32- разрядных процессоров i80x86
25	представление числовой и символьной информации в компьютере на разных стадиях выполнения программы
26	способы инициализации (задания) разнообразного типа данных программы
27	виды адресации операндов в памяти, особенности 32-битного режима адресации
28	структуру программ .EXE и .COM и их отображение (образ) в памяти
29	программирование типовых управляющих структур языка C++
30	основные приёмы программирования задач, связанных с вводом/выводом числовой и символьной информации в компьютере
31	концепцию модульного программирования
32	особенности организации и выполнения подпрограмм (процедур) в пользовательских программах
33	структуру и организацию макросов
34	Организацию многомодульных программ
35	типы прерываний реального режима и процедуры их обслуживания
36	структуру прикладных обработчиков программных и аппаратных прерываний
37	организацию интерфейса между ассемблером и языком C++
уметь	

10	создавать приложения на C++ с использованием программной оболочки Visual Studio .Net 2008
11	конструировать классы на C++
12	использовать переопределение операций
13	применять наследование
14	обрабатывать исключительные ситуации в программах
15	разрабатывать и применять шаблоны классов и функций
38	выполнить отладку и тестирование программы в отладчике Turbo Debugger на уровне машинных команд
39	пользоваться сервисными функциями DOS и BIOS в прикладных программах реального режима
40	пользоваться основными конструкциями языка ассемблера при разработке прикладных программ
41	произвести декомпозицию задачи на отдельные модули с последующим формирова-нием их в виде самостоятельных объектных модулей
иметь опыт (владеть)	
16	использования готовых библиотек классов
17	документирования разработанных программных продуктов
18	тестировании и отладки программ
42	машинно-ориентированного способа мышления при разработке ассемблерных программ
43	владения практическими приёмами составления программ на языке C++ с ассемблерными модулями

4. Содержание и структура учебной дисциплины

Лекционные занятия Таблица 4.1

(Модуль), дидактическая единица, тема	Часы	Ссылки на цели
Семестр: 3		
Модуль: C++ как расширение языка C.		
Дидактическая единица: Ссылка, параметры функции по умолчанию, перегрузка функции, константа, операторы распределения памяти, пространство имен		
Основные концепции объектно-ориентированного программирования. Расширения языка C.	2	10, 18, 3, 4, 6, 7, 9
Модуль: Основы объектно-ориентированного программирования на C++		
Дидактическая единица: Класс, конструктор, деструктор, данные- члены, функции-члены класса, статические данные и функции, указатель this, друзья класса, перегрузка операторов.		
Классы. Функции-члены и данные-члены. Интерфейсы и реализация. Конструкторы и деструкторы: конструктор без параметров (по умолчанию); конструктор копирования. Указатель	4	10, 11, 18, 3, 5, 6, 9

this. Статические члены: функции и данные. Указатели на члены. Константные члены-функции и константные объекты. Функции-друзья. Перегрузка бинарных и унарных операций. Примеры перегруженных операций: индексирования, вызова функций, инкремента и декремента префиксных и постфиксных; перегрузка new, delete. Преобразование типов, определяемых пользователем с помощью конструкторов и операций преобразования. Неявное преобразование типов.		
Модуль: Потоки ввода-вывода в C++		
Дидактическая единица: Поток ввода-вывода, форматирование потока, перегрузка потока, предопределенные потоки, файловые потоки.		
Заголовочные файлы. Предопределенные объекты и потоки. Операции помещения и извлечения в поток. Форматирование. Флаги форматирования. Манипуляторы. Ошибки потоков. Файловый ввод-вывод с применением потоковых классов. Режимы работы с файловыми потоками.	2	10, 11, 18, 2, 3, 5, 7
Модуль: Наследование в C++		
Дидактическая единица: Наследование, виртуальные функции, полиморфизм, абстрактные классы.Динамические структуры данных на классах; линейные списки: основные виды и способы реализации.		
Наследование классов и производные классы. Конструкторы, деструкторы и наследование. Иерархии классов. Виртуальные функции. Полиморфизм. Абстрактные классы и чистые виртуальные функции. Множественное наследование. Виртуальные базовые классы. Контроль доступа. Вложенные классы. Включение объектов в классы. Определение типа объектов во время выполнения программы (RTTI).	4	1, 10, 11, 13, 2, 3, 9
Модуль: Обработка исключений в C++		
Дидактическая единица: Исключение, генерация исключения, обработка исключения.		
Обработка ошибок в стандартном С. Распознавание ситуаций. Объектно-ориенти-рованная обработка исключений. Применение try, catch, throw. Раскрутка стека. Стандартные исключения в C++. Работа с конструкторами и исключениями. Функции terminate(), unexpected().	2	1, 10, 11, 13, 14, 16, 18, 3

Модуль: Шаблоны в С++		
Дидактическая единица: Шаблон функции, шаблон класса, итератор		
Шаблоны функций. Шаблоны классов. Параметры шаблонов. Наследование и шаблоны. Примеры построения шаблонов. Итераторы.	2	1, 10, 11, 15, 18, 2, 3, 7
Модуль: Библиотека шаблонов STL.		
Дидактическая единица: Классы контейнеры, последовательные контейнеры, ассоциативные контейнеры.		
Векторы, списки и очереди. Контейнерные классы. Используемые алгоритмы.	2	1, 10, 13, 15, 16, 18, 2, 3, 8, 9
Семестр: 4		
Модуль: АРХИТЕКТУРНЫЕ ОСОБЕННОСТИ ПК		
Дидактическая единица: Ассемблер, архитектура ПК, реальный и защищнные режимы работы, типы данных, программная модель		
Введение язык ассемблера и архитектуру ПК .Характеристика типов данных в ПК IBM PC. Программная модель IBM PC. Режимы адресации и форматы команд.	6	19, 20, 21, 22, 24
Модуль: ФУНДАМЕНТАЛЬНЫЕ ПОНЯТИЯ ЯЗЫКА АССЕМБЛЕРА		
Дидактическая единица: Синтаксис ассемблерной строки, трансляция и компоновка программ, директивы резервирования и инициализации данных, выражения и операторы		
Синтаксис ассемблера. Разработка программы на ассемблере. Структуры данных и их инициализация в программе. Выражения и операторы.	6	21, 23, 26
Модуль: СТРУКТУРЫ АССЕМБЛЕРНЫХ ПРОГРАММ		
Дидактическая единица: Сегментые директивы, шаблон программы, образ программы .EXE и .COM на диске и в памяти, упрощенные сегментные директивы		
Стандартные директивы управления сегментами. Упрощенные директивы описания сегментов. Образы программ на диске и в памяти (формат .EXE и .COM). Использование сегментов данных дальнего типа.	4	28
Модуль: СИТЕМА КОМАНД 32-РАЗРЯДНЫХ ПРОЦЕССОРОВ		
Дидактическая единица: Система команд, режимы адресации операндов в командах		
Команды передачи данных. Способы адресации операндов. Арифметические команды и машинная арифметика. Команды логических операций, сдвига и вращения. Операции над битами и байтами. Команды передачи управления и организации	8	22, 27, 29, 40

циклов. Программирование типовых управляющих структур C/C++ Команды обработки строк.		
Модуль: МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ		
Дидактическая единица: Программный модуль, процедура, макрос, макрокоманда, многомодульная программа, библиотеки макросов и бъектных модулей.		
Процедуры и многомодульные программы. Макросы. Библиотеки макросов. Директива Include	4	31, 32, 33, 34, 41
Модуль: СИСТЕМА ПРЕРЫВАНИЙ РЕАЛЬНОГО РЕЖИМА		
Дидактическая единица: Прерывания и их виды, контроллер прерываний, процедура обработки программных и аппаратных прерываний.		
Система прерываний .Прикладные обработчики прерываний.	4	35, 36
Модуль: ИНТЕРФЕЙС BORLAND C++ И TURBO ASSEMBLER.		
Дидактическая единица: Программная среда, основные принципы взаимодействия между Turbo assembler и C++, соответствие типов данных, модели памяти, процедура вызова из C++ ассемблерных функций и наоборот функций C++ из программы на ассемблере.		
Принципы взаимодействия между Turbo Assembler и Borland C++. Практические приёмы построения двуязычных программ. Демонстрационная программа с использованием взаимных вызовов.	4	37, 43

Лабораторная работа Таблица 4.2

(Модуль), дидактическая единица, тема	**Учебная деятельность**	**Часы**	**Ссылки на цели**
Семестр: 3			
Модуль: Основы объектно-ориентированного программирования на C++			
Дидактическая единица: Класс, конструктор, деструктор, данные-члены, функции-члены класса, статические данные и функции, указатель this, друзья класса, перегрузка операторов.			
Разработка классов, создание конструкторов и деструкторов. Использование статических	Изучает материал темы Выполняет индивидуальное	4	1, 10, 11, 18, 3, 6

членов.	задание на лаб. работу		
Модуль: Перегрузка операций в С++			
Дидактическая единица: Поток ввода-вывода, форматирование потока, перегрузка потока, предопределенные потоки, файловые потоки.			
Переопределение операций. Организация и переопределение операций ввода-вывода.	Изучает материал темы Выполняет индивидуальное задание на лаб. работу	4	10, 11, 12, 16, 18, 3, 7
Модуль: Наследование в С++			
Дидактическая единица: Наследование, виртуальные функции, полиморфизм, абстрактные классы.Динамические структуры данных на классах; линейные списки: основные виды и способы реализации.			
Наследование. Обработка исключительных ситуаций. Создание динамического списка объектов, связанных наследованием.	Изучает материал темы Выполняет индивидуальное задание на лаб. работу	6	1, 10, 11, 13, 14, 16, 18, 3, 9
Модуль: Шаблоны в С++			
Дидактическая единица: Шаблон функции, шаблон класса, итератор			
Разработка шаблонных классов. Использование их для хранения данных.	Изучает материал темы Выполняет индивидуальное задание на лаб. работу	4	10, 11, 12, 15, 16, 18, 3, 8, 9
Семестр: 4			
Модуль: Лабораторный практикум			
Дидактическая единица: Система команд 32-разрядных процессоров и принципы их кодирования			
Ассемблирование и отладка готовых программ. Изучение системы команд, их кодирование, способы адресации операндов.	- изучает материал по трансляции программ и производит исполнение программы Hello.asm в интерактивном режиме; - следуя инструкциям, выполняет пошаговое исполнение одной из демонстрационных программ в отладчике TD; -в заданную для исследования программу вводит синтаксические	4	21, 22, 24, 26, 27, 28, 38

	ошибки, производит трансляцию программы с последующим анализом ошибок, указанных в тексте листинга; - изучает (с помощью преподавателя) принципы двоичного кодирования команд и применяет их на примере заданных для исследования команд; - сравнивает полученные результаты кодирования с машинными кодами, представленными в окне CPU отладчика TD, и устраняет выявленные ошибки		
Дидактическая единица: Сервисные функции ввода-вывода информации, обаботка строковых переменных.			
Функции ввода-вывода символьной информации. Обработка строковых переменных.	- изучает методический материал по использованию функций DOS для ввода-вывода символьной информации (справоч-ные данные, спец. демонстрационный файл DosInt21); - знакомится с понятием расширенного ASCII-кода клавиш и его использованием для управлением хода программы (справочные данные, спец. демонстрационный файл ScanAscii); - изучает методический материал по при-менению строковых команд для обработки строк символов	4	21, 32, 38, 39, 40, 42

	(справочные данные, демонстрационные файлы); - в соответствии с заданием, применяя базовые алгоритмы поиска и сортировки дан-ных, разрабатывает собственную программу; - производит отладку программы с помо-щью отладчика TD		
Дидактическая единица: Сервисные функции BIOS			
Функции BIOS для работы с экраном и клавиатурой	- знакомится со справочными данными по функциям BIOS; - запускает на исполнение демонстрацион-ные программы, анализируя особенности использования функций BIOS для выполнения необходимых действий; - разрабатывает (в соответствии с заданием) собственную программу и производит её отладку	4	33, 40, 42
Дидактическая единица: Форматы числовых данных в компьютере			
Преобразование форматов числовых данных в операциях ввода-вывода	-знакомится с методическим материалом по преобразованию форматов числовой информации в операциях ввода-вывода; -знакомится с принципами построения многомодульных программ; - анализирует поставленную задачу и производит её	6	25, 30, 31, 34, 41, 42

	декомпозицию; -создаёт список переменных задачи, с последующим закреплением каждой из них как за отдельными модулями (локальные пере-менные), так и за несколькими (общие); -создаёт программные модули задачи; -создаёт исполняемый файл многомодульной программы и производит её отладку; -формирует из вспомогательных объект-ных модулей свою собственную библиотеку и заново производит трансляцию		

5. Самостоятельная работа студентов

Семестр- 3, Курсовая работа

Курсовые работы выполняются студентами по индивидуальным заданиям.

1. Простые структуры данных в памяти (начальный балл 3,0)

Задан объект со списком полей - запись хранимой таблицы. Структура данных таблица должна быть полностью динамической и также реализована в виде класса. Обязательные операции - добавление, удаление, включение и извлечение по логическому номеру, сортировка, включение с сохранением порядка, загрузка и сохранение объектов в бинарном файле, поиск по различным критериям. Предполагается, что операции сравнения хранимых объектов переопределены стандартным образом (в виде операций <,> и т.д.).

Программа должна реализовывать указанные выше действия. Протестировать структуру данных на простом типе данных (например, int, double) и сложном, из выбранного задания. Программа тестирования должна содержать меню, обеспечивающее выбор операций.

Содержание записи (строки) таблицы.

1. Список студентов факультета. Основная таблица: Фамилия, дата рождения, дата поступления, дата отчисления, адрес, группа. Редактирование с выбором по группе и шаблону фамилии.
2. Доска объявлений. Категория объявления: куплю-продам, рубрика: транспорт, недвижимость, материалы и т.д. Текст объявления - строка (абзац) переменной длины, дата подачи, контактный телефон. Поиск по заданной рубрике и по шаблону искомого слова в тексте объявления. Сортировка по дате подачи.
3. Движение поездов. Номер поезда, дни недели отправления, время отправления, время в пути (часов, минут). Станция отравления, назначения, промежуточные станции. Промежуточные станции запоминаются в динамическом массиве номеров станций. Поиск всех поездов, следующих до заданной станции.

4. Учет посещаемости индивидуальных и коллективных занятий. Фамилия, группа, динамический массив дат посещения занятий. Операции добавления даты посещения для выбранного слушателя и для выбранной группы слушателей. Вывод списка студентов, посещавших занятия по заданной дате. Добавление и удаление групп. Сортировка по количеству посещений и по фамилии.
5. Справочник склада. Наименование товара, категория товара, количество, дата поступления, цена и процент торговой надбавки. Сортировка по всем параметрам. Составление фактуры: выбор нескольких товаров, количества (с уменьшением его на складе), подсчет общей суммы и торговой надбавки.
6. База данных междугородных переговоров. Город, код города, тариф, дата, время переговоров, продолжительность в минутах, телефон абонента, дата оплаты. Сортировка по дате переговоров. Вывод списка неоплаченных переговоров и суммы оплаты (дата оплаты пуста).
7. База данных пункта проката. Предмет проката, стоимость проката, дата, время получения и возврата предмета, фамилия. Несданные предметы - пустая дата возврата. Подсчет суммы оплаты за сданный предмет, подсчет дохода пункта за заданный год/месяц.

Вид структуры данных

1. Динамический массив указателей на структуры.
2. Односвязный список структур.
3. Двусвязный циклический список структур.
4. Динамический массив структур с уплотнением при удалении и с расширением при переполнении.
5. Дерево, вершина которого содержит два указателя на поддеревья, счетчик количества вершин и структурированную переменную (начальный балл 4).

Дополнение: добавить в структуру данных вложенный класс, реализующий функции итератора (дополнительный балл 0,5).

2. Последовательный двоичный файл (начальный балл 3.5)

Двоичный файл содержит записи переменной длины - строки таблицы с заданной структурой столбцов (см. простые структуры данных). Формат записи предполагает ее переменную размерность. Реализовать набор операций над записями без загрузки одновременно всей таблицы в память (поэлементная загрузка СД): добавление строки таблицы, извлечение, удаление, вставка по логическому номеру и редактирование (обновление) строки, вставка с сохранением порядка, сортировка, постраничный просмотр, сжатие файла. При изменении размерности записи она переписывается в конец файла.

Программа должна реализовывать указанные выше действия. Протестировать структуру данных на простом типе данных (например, int, double) и сложном, из выбранного по заданию. Программа тестирования должна содержать меню, обеспечивающее выбор операций.

Вид структуры данных

1. Динамический массив указателей на структуры.
2. Односвязный список структур.
3. Двусвязный циклический список структур.
4. Динамический массив структур с уплотнением при удалении и с расширением при переполнении.
5. Дерево, вершина которого содержит два указателя на поддеревья, счетчик количества вершин и структурированную переменную (начальный балл 4).

Дополнение: добавить в структуру данных вложенный класс, реализующий функции итератора (дополнительный балл 0,5).

3. Шаблон иерархической структуры данных в памяти (начальный балл 4,0)

Для заданной двухуровневой структуры данных, содержащей указатели на объекты (или сами объекты) - параметры шаблона, разработать полный набор операций (добавление, включение и извлечение по логическому номеру, сортировка, включение с сохранением порядка, загрузка и сохранение строк в бинарном файле, балансировка - выравнивание размерностей структур данных нижнего уровня). Предполагается, что операции сравнения хранимых объектов переопределены стандартным образом (в виде операций <,> и т.д.). Программа должна использовать шаблонный класс с объектами-строками и реализовывать указанные выше действия над текстом любого объема, загружаемого из файла.

Программа должна реализовывать указанные выше действия. Протестировать структуру данных. Программа тестирования должна содержать меню, обеспечивающее выбор операций.

1. Шаблон структуры данных - двухуровневый массив указателей на объекты. Массив верхнего уровня - статический, массивы нижнего уровня - динамические, размерность - параметр конструктора, последовательность указателей в каждом массиве ограничена NULL. Если после включения указателя массив заполняется полностью, то создается еще один массив указателей, в который переписывается половина указателей из старого.
2. Шаблон структуры данных - двухуровневый массив указателей на объекты с типом - параметром шаблона. Массив верхнего уровня - статический. Массивы нижнего уровня - динамические, размерность каждого следующего в 2 раза больше размерности предыдущего, последовательность указателей в каждом массиве ограничена NULL.
3. Шаблон структуры данных - односвязный список, каждый элемент которого содержит статический массив указателей на объекты. Последовательность указателей в каждом массиве ограничена NULL. При переполнении текущего массива указателей создается новый элемент списка, в который переписывается половина указателей из текущего.
4. Шаблон структуры данных - односвязный список, каждый элемент которого содержит динамический массив указателей на объекты. Размерность массива указателей в каждом последующем элементе списка в 2 раза больше, чем в предыдущем. Последовательность указателей в каждом массиве ограничена NULL. При переполнении текущего массива указателей последний указатель переносится в следующий элемент списка.
5. Шаблон структуры данных - двусвязный циклический список, каждый элемент которого содержит статический массив указателей на объекты. Последовательность указателей в каждом массиве ограничена NULL. При переполнении текущего массива указателей создается новый элемент списка, в который переписывается половина указателей из текущего.
6. Шаблон структуры данных - односвязный список, каждый элемент является заголовком односвязного списка. Элемент списка второго уровня содержит указатель на объект. (При включении элемента последним в список предусмотреть ограничение длины текущего списка и переход к следующему).
7. Шаблон структуры данных - массив указателей на заголовки списков. Элемент списка содержит указатель на объект. (При включении элемента последним в список предусмотреть ограничение длины текущего списка и переход к следующему).
8. Шаблон структуры данных - дерево. Конечная вершина дерева содержит статический массив указателей на объекты (ограниченный NULL-указателем), промежуточная вершина содержит счетчик вершин дерева и два указателя на правое и левое поддерево. Значения в дереве не упорядочены. Естественная нумерация значений производится путем обхода дерева по принципу "левое поддерево - вершина - правое поддерево". Если при включении указателя в найденный массив последний переполняется, он становится промежуточной вершиной. (начальный балл 4,5).

4. Класс - структура данных в двоичном файле (начальный балл - 4,5)
Класс двоичного файла, производный от fstream. Двоичный файл содержит заданную структуру данных. Программа должна представлять собой простой текстовый редактор, использующий структуру данных для промежуточного хранения редактируемого файла. Должны быть реализованы операции создания и заполнения двоичного файла из заданного текстового и сохранения содержимого двоичного файла в текстовом.
Программа должна реализовывать указанные выше действия. Протестировать структуру данных. Программа тестирования должна содержать меню, обеспечивающее выбор операций.
1. Класс - двоичный файл, производный от fstream. Файл содержит массив указателей на строки, представленные записями переменной длины: целый счетчик и последовательность символов строки (без 0-ограничителя). Формат файла: в начале - файла, размерность массива указателей, текущее количество указателей, адрес (смещение) массива указателей в файле.
2. Класс - двоичный файл, производный от fstream. Файл содержит односвязный список строк в формате записей переменной длины со счетчиком. Формат файла: в начале файла - заголовок списка. Элемент списка содержит файловый указатель на следующий и саму строку в виде записи переменной длины.
3. Класс - двоичный файл, производный от fstream. Файл содержит двусвязный циклический список строк в формате записей переменной длины со счетчиком. Формат файла: в начале файла - заголовок списка. Элемент списка содержит файловые указатели на следующий и предыдущий и саму строку в виде записи переменной длины.
4. Класс - двоичный файл, производный от fstream. Файл содержит дерево, конечная вершина которого содержит строку, представленную записью переменной длины: целый счетчик и последовательность символов строки (без 0-ограничителя). Промежуточная вершина содержит указатели на правое и левое поддерево и число вершин в поддереве. Формат файла: в начале - файла, указатель на корневую вершину.

5. Шаблон - структура данных в двоичном файле (начальный балл - 5)
Класс двоичного файла, производный от fstream. Двоичный файл содержит заданную структуру данных с типом хранимых объектов - параметром шаблона. Программа должна выполнять операции создания файла, просмотра, добавления, удаления, обновления и сортировки объектов заданного при генерации типа. Реализовать две версии программы для различных хранимых объектов (например, текстовых строк и объектов класса из лабораторной работы). Предполагается, что операции сравнения хранимых объектов переопределены стандартным образом (в виде операций <,> и т.д.). Для позиционирования в потоке можно использовать методы seekg(long,mode) и long tellg(), для хранимого объекта переопределен метод fstream << object который записывает объект с текущей позиции в открытый двоичный поток и метод fstream >> object, который читает содержимое объекта с текущей позиции открытого потока.
Программа должна реализовывать указанные выше действия. Протестировать структуру данных на простом типе данных (например, int, double и т.д.) и сложном. Программа тестирования должна содержать меню, обеспечивающее выбор операций.
1. Шаблон - двоичный файл, содержащий двусвязный циклический список объектов. Тип хранимого в файле объекта - параметр шаблона. В начале файла расположен файловый указатель на первый элемент списка. Элемент списка содержит 2 файловых указателя на следующий и предыдущий элементы, за которыми следует объект.
2. Шаблон - двоичный файл, содержащий односвязный список объектов. Тип хранимого в файле объекта - параметр шаблона. В начале файла расположен файловый указатель на первый элемент списка. Элемент списка содержит файловый указатель на следующий элемент, за которым следует объект.

3. Шаблон - двоичный файл, содержащий массив указателей на объекты. Тип хранимого в файле объекта - параметр шаблона. В начале файла расположены: размерность массива указателей (int), текущее количество указателей (int) и смещение (адрес) массива указателей. Если происходит заполнение массива указателей, то его размерность увеличивается в 2 раза и он переписывается в конец файла.
4. Шаблон - двоичный файл, содержащий двоичное дерево объектов. Тип хранимого в файле объекта - параметр шаблона. В начале файла расположен указатель на корневую вершину. Вершина содержит два файловых указателя на правое и левое поддерево и сам объект, который непосредственно следует за указателями.

6. Коллекция объектов произвольных типов в памяти (начальный балл - 4)
Необходимо разработать интерфейс для объединения в структуру данных множества объектов различных классов - абстрактный базовый класс объектов object, для которого предусмотреть виртуальные методы: загрузки объекта из текстовой строки, выгрузки объекта в текстовую строку в динамической памяти, добавления объекта в последовательный двоичный файл, чтения объекта из последовательного двоичного файла, возврата уникального идентификатора класса, возврата указателя на строку с именем класса, сравнения двух объектов, "сложения "(объединения) двух объектов, создание динамической копии объекта. Сделать классы хранимых объектов производными от абстрактного базового класса object (например, класс Float, String и класс из лабораторной работы).
Коллекция представляет собой структуру данных, хранящую указатели на объекты произвольного типа - объекты базового класса. Программа должна обеспечивать просмотр коллекции, добавление объекта выбранного типа, редактирование и удаление объекта, сложение и сравнение объектов при условии совпадения типов с помещением результата в ту же коллекцию, сохранение и загрузку объектов из текстового (или двоичного) файла.
Программа должна реализовывать указанные выше действия. Протестировать структуру данных на простом типе (например, Int или String). Программа тестирования должна содержать меню, обеспечивающее выбор операций.
Варианты структуры данных для организации коллекции:
1. Динамический массив указателей на записи.
2. Двусвязный список.
3. Двусвязный циклический список.
4. Список с организацией стека.
5. Список с организацией очереди.
6. Дерево, вершина которого содержит два указателя на поддеревья, счетчик количества вершин и указатель на объект (начальный балл 4,5).

Дополнение: добавить в коллекцию вложенный класс, реализующий функции итератора (дополнительный балл 0,5).
Семестр- 3, Индив. работа
Выполнение индивидуального задания на курсовую работу
Семестр- 3, Подготовка к занятиям
Модуль 1. - 10ч.
C++ как расширение языка C.
Модуль 2. - 10 ч.
Основы объектно-ориентированного программирования на C++
Модуль 3. - 10 ч.
Потоки ввода-вывода в C++.
Модуль 4. - 15 ч.
Наследование в C++

Модуль 5. 10 ч.
Обработка исключений в C++
Модуль 6. 10 ч.
Шаблоны в C++

Семестр- 4, Курсовой проект

Курсовой проект "Разработка прикладных обработчиков прерываний и резидентных программ".
Объём часов самостоятельной для выполнения КП - 20.

Цели:
1. Изучение вопросов, связанных с
- процедурными особенностями программных и аппаратных прерываний;
- организацией резидентных программ.
2. Разработка собственных прикладных обработчиков прерываний и резидентных программ в соответствии с индивидуальным заданием.
Методические материалы и индивидуальные задания на проектирование изложены в руководстве к выполнению курсовой работы, расположенном на сайте кафедры ВТ НГТУ: ftp:/ermak.cs.nstu.ru/afaval/assembler

Требования к оформлению и содержанию курсовой работы.
1. Курсовая работа должна иметь титульный лист и состоять из 12 - 15 листов печатного текста шрифтом, равный 12 пунктам.
2. Содержание КР включает следующие обязательные параграфы (разделы):
" Задание.
" Структуру программы и обоснование выбора её отдельных модулей (составных обработчиков прерываний) должна учитывать следующие факторы:
- вычислительные процедуры, используемые в программе, должны соответствовать заданному алгоритму реализации;
- резидентная программы должна обязательно включать прикладной обработчик клавиатурного прерывания Int new_09h, быть защищенной от повторной установки, обладать способностью самовыгрузки из памяти;
- прикладной аппаратный обработчик прерываний должны строиться с учётом свойства нереентерабельности DOS;
- программа должна иметь хорошо продуманным экранным интерфейсом.
" Справочное описание используемых в программе системных обработчиков, сервисных функций DOS и BIOS.
" Функциональное описание программы и результаты её демонстрации.
" Листинги программ.
3. Необходимые компьютерные файлы на дискете (флэш-памяти) для демонстрации программ.

Семестр- 4, Индив. работа

Консультации по темам курса и вопросам выполнения курсового проекта - 8 часов.

Семестр- 4, Подготовка к занятиям

Подготовка к лекционным занятиям (18 лекций) -36 часов.
Подготовка к лабораторным работам (4 работы) - 24 часа.

6. Правила аттестации студентов по учебной дисциплине

3-ий семестр.

Для аттестации студентов по дисциплине используется балльно-рейтинговая система. Рейтинг студента по дисциплине определяется как сумма баллов за работу в семестре (текущий рейтинг) и баллов, полученных в результате итоговой аттестации (экзамен).

В таблице приведено максимальное количество баллов, которое может набрать студент по видам учебной деятельности в течение семестра и диапазоны баллов, соответствующие минимальному и максимальному количествам баллов. Максимальная сумма баллов за семестр составляет 100 баллов (текущий рейтинг - 60 баллов, итоговая аттестация - 40 баллов).

Правила текущей аттестации:

1. В течение третьего семестра необходимо представить и защитить 4 лабораторных работы, курсовую работу (курсовой проект) в сроки, установленные учебным графиком (см. таблицу).
2. К защите лабораторной работы допускаются студенты, выполнившие работу в полном объеме (все задания согласно варианту) и оформившие отчет по работе в соответствии с требованиями.
3. На защите предлагается два теоретических вопроса и один практический вопрос (по теме работы).
4. Максимальное количество баллов 10-12 выставляется, если студент полностью ответил на все вопросы, без серьезных замечаний и недочетов.
5. Количество баллов 7-9 выставляется, если студент ответил на два вопроса из трех, причем один из вопросов - практический.
6. Минимальное количество баллов 5-6 выставляется, если студент ответил на два вопроса из трех частично, с серьезными замечаниями, недочетами.
7. При несвоевременной защите лабораторных работ оценка студента снижается на 2 балла, за каждый пропущенный срок защиты, согласно таблицы.
8. К защите курсовых работ допускаются студенты, выполнившие курсовую работу в полном объеме (все задания согласно варианту) и оформившие отчет по работе в соответствии с требованиями.
9. На защите предлагается два теоретических вопроса и один практический вопрос (по ходу выполнения работы).
10. Максимальное количество баллов 88-100 выставляется, если студент полностью выполнил задание на курсовую работу, отлично оформил пояснительную записку и ответил на все вопросы, без серьезных замечаний и недочетов.
11. Количество баллов 71-87 выставляется, если студент выполнил задание на курсовую работу, оформил пояснительную записку, но были допущены небольшие недочеты в работе или ее оформлении, ответил на два вопроса из трех, причем один из вопросов - практический.
12. Минимальное количество баллов 50-70 выставляется, если студент выполнил задание на курсовую работу, оформил пояснительную записку, но имеются существенные замечания по работе и ее оформлению, ответил на два вопроса из трех частично, с серьезными замечаниями, недочетами.
13. Пересдача лабораторной работы, курсовой работы назначается, если студент не ориентируется в учебном материале, не может объяснить ход и результаты выполнения работы. В случае пересдачи работы происходит потеря баллов (максимальное количество баллов составляет 8-9 или 71 в зависимости от вида работы).
14. В случае представления и защиты курсовой работы с опозданием от учебного графика происходит потеря баллов (опоздание на 1 неделю - потеря 5 баллов, опоздание на 2 недели - 10 баллов, 3 недели - потеря 15 баллов и более - потеря до 50% баллов от максимально возможного).

15. В течение семестра со студентами проводится 3 контрольных среза знаний в тестовой форме по темам теоретического курса. 1-й - модули 1-2, 2-й - модули 3-4, 3-й - модули 5-7.

16. Максимальный балл 6 выставляется, если студент ответил правильно на 88% и более вопросов теста; 5 балла, если 71% - 87% верных ответов; 4 балла, если 60% - 70% верных ответов, 3 балла - 50% - 59%.

Правила итоговой аттестации:

1. К экзамену допускаются студенты, защитившие все лабораторные работы, курсовую работу, сдавшие срезы знаний в тестовой форме, и набравшие в сумме не менее 50% (30 баллов) по результатам текущего рейтинга.

2. Экзамен проводится в письменном виде, предлагается одна задача и один теоретический вопрос (темы и образцы экзаменационных задач и теоретические вопросы приведены в п.9).

3. Максимальное количество 35-40 баллов выставляется, если все задания выполнены полностью, без серьезных замечаний.

4. Количество баллов 28-34 выставляется, если дан полный ответ на теоретический вопрос, а в решении задачи допущены незначительные недочеты.

5. Минимальное количество баллов 20-27 выставляется, если дан неполный ответ на теоретический вопрос и в решении задачи имеются серьезные недочеты.

6. Возможно получить "автомат" (отлично) по дисциплине без сдачи экзамена, если студент в течение семестра выполняет дополнительные задания повышенной сложности и набирает свыше 90 баллов по текущему рейтингу.

№п/п	Вид учебной работы	Макс. кол-во баллов	Диапазоны баллов	Срок защиты (неделя семестра)
3 семестр:				
1.	Лабораторная работа №1	10	5-10	неделя проведения 2-й лаб. работы по учебному расписанию
2.	Лабораторная работа №2	10	5-10	неделя проведения 3-й лаб. работы по учебному расписанию
3.	Лабораторная работа №3	10	5-10	неделя проведения 4-й лаб. работы по учебному расписанию
4.	Лабораторная работа №4	12	6-12	18
5.	Тестовый опрос 1	6	3- 6	6
6.	Тестовый опрос 2	6	3-6	12
7.	Тестовый опрос 3	6	3-6	18
Итого по текущему рейтингу:		60	30-60	
8.	Экзамен	40	20-40	
Итого за семестр:		100	50-72 (удовл.)	

4-ый семестр

Процесс формирования учебного рейтинга отражён в таблице. Определяющий фактор - максимальная сумма баллов за курс составляет 100 баллов, а именно, доэкзаменационный или текущий рейтинг - 60 баллов, итоговый - 40 баллов.

Правила текущей аттестации по лабораторному практикуму.

1. В течение 5-го семестра необходимо выполнить и защитить 4 лабораторных работы и курсовой проект в сроки, установленные графиком (см. таблицу 1).

2. К защите лаб_работы необходимо представить машинописный экземпляр отчёта - один на учебную бригаду.

3. Абсолютное значение проставляемых балов зависит от характера выполнения работы (один или несколько заходов), содержания и качества представленных отчётов, своевременности их представления, а также содержательности ответов на вопросы, заданные при защите.

4. Минимальный зачётный балл (из указанного в таблице для данной работы) выставляется, если отчёт по работе не содержит краткого описания используемого алгоритма, листинг программы плохо откомментирован, а студент испытывает значительные за-труднения при объяснении содержательной части программы.

5. Студент направляется на дополнительную проработку материала, если при защите работы он получает баллы меньше нижнего предела диапазона, установленного для данной работы.

Правила аттестации по курсовому проекту.

1 Курсовой проект должен быть выполнен и защищён не позднее 17 недели текущего семестра.

2. Требования к оформлению пояснительной записки изложены в § 14 методического пособия по КП.

3. Минимальный зачётный балл - 11 - выставляется при существенных отступлениях от предъявленных требований к оформлению пояснительной записки, плохой экранный интерфейс демонстрируемых программ, а также при отсутствии объяснений (или невразумительных ответов) на некоторые вопросы преподавателя в процессе защиты проекта.

Таблица 1.

№	Вид учебной деятельности	Максимальное кол-во баллов	Диапазоны баллов	Срок сдачи
1	Работа №5	6	3 - 6	Во время текущей лаб. работы или на ближайшем практическом занятии.
2	Работа №6	10	5 - 10	-- // -- // -- // --
3	Работа №7	10	5 - 10	-- // -- // -- // --
4	Работа №8	12	6 - 12	-- // -- // -- // --. Дополнительно в часы, выделенные преподавателем на зачётной неделе.
5	Курсовой проект (КП)	22	11 - 22	14 - 17 недели.
	Итого: Текущий рейтинг	60	30 - 60	

Дополнительные возможности для повышения текущего рейтинга.

1.- Выполнение работы сверх предусмотренной основной программой курса.

Разработка программы в формате "C++ ассемблер"

2.- Учитывая объективную сложность выполнения курсового проекта одновременно с выполнением текущих заданий по изучению и освоению программирования на языке ассемблера, студенты премируются дополнительными баллами за досрочное завершение работы (см. таблицу 2)

Таблица 2

Дополнительные возможности по-вышения текущего рейтинга

Работа, не предусмотренная програм-мой курса. "Разработка программы в формате "C/C++ - ассемблер"		
Максимальное кол-во баллов	Диапазоны баллов	Срок сдачи
14	8 - 114	14 - 17 недели

Досрочное завершение:

- заданий лабораторного практику-ма	5	5
- курсового проекта	5	5
Итого, дополнительно к текущему рей-тингу, определённому выше	24	18 - 24
Суммарный текущий рейтинг сту-дентов, наиболее успешно занимаю-щихся по данной дисциплине	84	50 - 84
Диапазон рейтинга студентов, претендующих на получение оценки "автоматом"		
ОТЛИЧНО		80 - 84
ХОРОШО		70 - 79

Экзамен и итоговая аттестация.

1. К экзамену допускаются студенты, сдавшие лабораторные работы, курсовой проект и, набравшие не менее 50% от максимального значения текущего рей-тинга (т.е. 30 баллов).

2. Экзаменационный билет содержит два вопроса: теоретический вопрос по курсу и задачу на составление программы или подпрограммы.

3. Максимальное количество баллов за сдачу экзамена - 40 (15 - за ответ на теоретический вопрос и 25 - за выполнение задачи).

4. Студент, пропустивший 50% лекционных часов, не может получить за теоретическую часть билета более 10 баллов.

5. Экзамен считается сданным, если студент набрал 20 баллов.

6. Студент не может получить минимально необходимое количество баллов (20 баллов) за экзамен, если он не справился с разработкой программы.

7. Студенты, имеющие текущий (доэкзаменационный) рейтинг в диапазоне 70 - 84 балла и не получившие (по каким-либо причинам) разрешение от деканата на выставление оценки "хорошо или отлично" без проведения экзаменационных испытаний, освобождаются на экзамене от выполнения 2-го вопроса.

Итоговая оценка в этом случае проставляется на основе суммарного рейтинга:
"(70 - 84 балла) + экз_балл за теоретический вопрос".

Вид учебной деятельности	Максимальное кол-во баллов	Диапазоны баллов
Срок сдачи		
Экзамен	40	20 - 40

Итого: Рейтинг по дисциплине (оценка за курс).

100

50-72 (удовл)

73-87 (хор)

88-100 (отл)

7. Список литературы

7.1 Основная литература

В печатном виде

1. Подбельский В. В. Язык Си++ : [учебное пособие для вузов по направлениям "Прикладная математика" и "Вычислительные машины, комплексы, системы и сети"] / В. В. Подбельский. - М., 2007. - 559 с. : ил., табл. - Рекомендовано МО.
2. Подбельский В. В. Язык Си++ : учебное пособие для вузов по направлениям "Прикладная математика" и "Вычислительные машины, комплексы, системы и сети" / В. В. Подбельский. - М., 2006. - 559 с. : ил., табл. - Рекомендовано МО.
3. Шеферд Д. Программирование на Microsoft Visual C++ .NET : мастер-класс [пер. с англ.] / Джордж Шеферд по материалам Дэвида Круглински. - М., 2007. - 892 с. : ил. + 1 CD-ROM.
4. Хабибуллин И. . Программирование на языке высокого уровня C/C++ : учебное пособие для вузов по направлению 654600 "Информатика и вычислительная техника" / И. Ш. Хабибуллин. - СПб, 2006. - 485 с. : ил. - Рекомендовано УМО.
5. Дейтел Х. М. Как программировать на C++ / Х. М. Дейтел, П. Дж. Дейтел ; пер. с англ. под ред. В. В. Тимофеева. - М., 2007. - 799 с. : ил.
6. Павловская Т. А. C/C++. Структурное программирование : практикум / Т. А. Павловская, Ю. А. Щупак. - СПб. [и др.], 2007. - 238 с. : ил.. - На тит. л.: Издательская программа 300 лучших учебников для высшей школы.
7. Синицын С. В. Программирование на языке высокого уровня : учебник / С. В. Синицын, А. С. Михайлов, О. И. Хлытчиев. - М., 2010. - 392, [1] с. : ил., табл. - Рекомендовано УМО.
8. Павловская Т. А. C/C++. Программирование на языке высокого уровня : [учебник по направлению "Информатика и вычислительная техника"] / Т. А. Павловская. - СПб. [и др.], 2010. - 460 с. : табл., ил. - Рекомендовано МО.
9. Юров В. И. Assembler : учебное пособие для вузов по направлению подготовки специалистов " Информатика и вычислительная техника" / В. И. Юров. - СПб. [и др.], 2006. - 636 с. : ил., табл. - Рекомендовано МО.
10. Финогенов К. Г. Использование языка Ассемблера : учебное пособие для вузов по специальности 351400 "Прикладная информатика (по областям)" / К. Г. Финогенов. - М., 2004. - 438 с. : ил. - Рекомендовано УМО.
11. Афанасьев В. А. Assembler IBM PC. Ч. 1. лабораторный практикум : учебное пособие [для 2 курса АВТФ по специальности 2201 "Вычислительные машины, комплексы, системы и сети"] / В. А. Афанасьев ; Новосиб. гос. техн. ун-т. - Новосибирск, 2003. - 115 с. : табл.
12. Афанасьев В. А. Assembler IBM PC : конспект лекций / В. А. Афанасьев ; Новосиб. гос. техн. ун-т. - Новосибирск, 2008. - 257, [1] с. : ил.

В электронном виде

1. Афанасьев В. А. Assembler IBM PC. Ч. 1. лабораторный практикум : учебное пособие [для 2 курса АВТФ по специальности 2201 "Вычислительные машины, комплексы, системы и сети"] / В. А. Афанасьев ; Новосиб. гос. техн. ун-т. - Новосибирск, 2003. - 115 с. : табл.. - Режим доступа: http://www.ciu.nstu.ru/fulltext/textbooks/2003/2003_afanasjev.zip. - На обл. загл.: Assembler IMB PC.

7.2 Дополнительная литература

В печатном виде

1. Романов Е. Л. Язык Си++ в задачах, вопросах и ответах : [учебное пособие] / Е. Л. Романов. - Новосибирск, 2003. - 426, [1] с. : ил.

2. Дьюхарст С. Программирование на C++ : Пер. с англ. / С. Дьюхарст. - Киев, 1993. - 272 с. : ил.
3. Павловская Т. А. C/C++ Программирование на языке высокого уровня : издательская программа 300 лучших учебников для высшей школы в честь СПб. / Т. А. Павловская. - М. [и др.], 2005. - 460 с. : ил. - Рекомендовано МО.
4. Павловская Т. А. C/C++. Программирование на языке высокого уровня : Учебник для вузов / Т. А. Павловская. - СПб., 2002. - 460 с. : ил.. - Алф. указ.: с. 450-460.
5. Фленов М. Е. Программирование на C++ глазами хакера / Михаил Фленов. - СПб., 2009. - 350 с. : ил. + 1 CD-ROM.
6. Шилдт Г. Искусство программирования на C++ : [для программистов] / Герберт Шилдт ; [пер. с англ. Т. Коротяевой]. - СПб., 2005. - 474 с. : ил.
7. Шилдт Г. Программирование на C и C ++ для Windows 95 / Шилдт, Г. - Киев, 1996. - 400 с. : ил.
8. Джамса К. 1001 совет по C/C++ : настольная книга программиста : пер. с англ. / К. Джамса. - М., 1997. - 783 с.
9. Разработка прикладных обработчиков прерываний и резидентных программ в MS DOS : учебно-методическое пособие / сост. В. А. Афанасьев. – Новосиб. гос. техн. ун-т ; кафедра ВТ НГТУ, 2003. – 53 с.
10. Афанасьев В. А. Интерфейс между Turbo Assembler и Borland C++ : учебно-методическое пособие / В. А. Афанасьев. – Новосиб. гос. техн. ун-т ; кафедра ВТ, 2005. – 45 с.
11. Кулаков В. Программирование на аппаратном уровне / В. Кулаков. – [2-е изд.]. – СПб. и др. : Питер : Питер принт, 2003. – 847 с.; 21 см. + 1 компакт-диск. – (Специальный справочник).
12. Голубь Н. Г. Искусство программирования на Ассемблере : лекции и упражнения / Голубь Н. Г. - СПб., 2002. - 644 с. : ил.
13. Абель П. Ассемблер. Язык и программирование для IBM PC / пер. с англ. под ред. С.М. Молявко. - Киев [и др.], 2003. - 734 с. : табл.
14. Секаев В. Г. Основы программирования на Ассемблере : учебное пособие / В. Г. Секаев ; Новосиб. гос. техн. ун-т, Фак. автоматики и вычисл. техники. - Новосибирск, 2010. - 98, [1] с. : табл.

В электронном виде

1. Секаев В. Г. Основы программирования на Ассемблере : учебное пособие / В. Г. Секаев ; Новосиб. гос. техн. ун-т, Фак. автоматики и вычисл. техники. - Новосибирск, 2010. - 98, [1] с. : табл.. - Режим доступа: http://www.ciu.nstu.ru/fulltext/textbooks/2010/10_sekaev.pdf

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

В печатном виде

1. Романов Е. Л. Практикум по программированию на C++ : [учебное пособие] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - СПб., 2004. - 426, [1] с. : ил.
2. Программирование на C++ : учебное пособие / [В. П. Аверкин и др.] ; под ред. А. Д. Хомоненко. - СПб., 2003. - 508 с. : ил. - Рекомендовано УМО.
3. Секаев В. Г. Основы программирования на Ассемблере : учебное пособие / В. Г. Секаев ; Новосиб. гос. техн. ун-т, Фак. автоматики и вычисл. техники. - Новосибирск, 2010. - 98, [1] с. : табл.

В электронном виде

1. Романов Е. Л. Практикум по программированию на C++ : [учебное пособие] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - СПб., 2004. - 426, [1] с. : ил.
2. Секаев В. Г. Основы программирования на Ассемблере : учебное пособие / В. Г. Секаев ; Новосиб. гос. техн. ун-т, Фак. автоматики и вычисл. техники. - Новосибирск, 2010. - 98, [1] с. : табл.. - Режим доступа: http://www.ciu.nstu.ru/fulltext/textbooks/2010/10_sekaev.pdf

8.2 Программное обеспечение

1. Microsoft, eMbedded Visual C++, Интегрированная среда разработки

9. Контролирующие материалы для аттестации студентов по дисциплине

3-ий семестр

В билеты экзамена включается одна задача и один теоретический вопрос. Задача должна быть выполнена с использованием технологии ООП на Си++.
Программа должна быть оформлена в виде класса или системы классов. Если программа работает с данными произвольных типов, то класс должен быть оформлен в виде шаблона. В классах необходимо описание только тех методов, которые используются в поставленной задаче. Если в варианте задания четко определен вид структуры данных, формы представления данных, то отклонения от условий считается грубым нарушением (например, статический или динамический массив, фиксированная или переменная размерность, внешняя или внутренняя форма представления числа и т.п.). При отсутствии такой информации необходимо явно оговорить выбранный Вами способ представления данных.
К тексту программы должно быть приложено описание структур данных, принципов построения алгоритма и перечня особых ситуаций, с которыми может сталкиваться программа (как реализованных в программе, так и не реализованных - "пустая" структура данных, поведение программы при несоответствии формата файла и т.п..). Объем готовой программы - в пределах 1-2 стр.. Стилистика описания, логичность, связность также учитываются при выставлении оценки.

Тематика задач, выносимых на экзамен

Классы типов данных, представленные динамическими структурами данных: целые произвольной размерности, представленные строкой символов-цифр (внешняя форма представления), или последовательностью цифр-тетрад (двоично-десятичное представление), степенные полиномы, представленные динамическим массивом коэффициентов, матрицы произвольной размерности, представленные динамическим (линейным) массивом коэффициентов или массивом указателей на строки матрицы динамические массивы. Разреженная матрица, представленная списком, динамическим массивом или массивом указателей на описатели ненулевых коэффициентов. Переопределение операций сложения, вычитания или умножения с результатом новым объектом. Конструктор копирования.
Шаблоны иерархических структур данных в памяти. Тип хранимого элемента параметр шаблона. Структуры данных: двухуровневые массивы указателей (массивы нижнего уровня - динамические), односвязные и двусвязные циклические списки, содержащие статический или динамический массив указателей на объекты, двоичные деревья, содержащие массивы указателей на объекты. Операции включения по заданному логическому номеру, с сохранением порядка, сортировки выбором и вставками, загрузки хранимых элементов в структуру данных из последовательного потока.
Структуры данных в двоичных файлах (хранимые объекты строки и динамические массивы). Класс, производный от fstream, содержащий массив указателей, односвязный или двусвязный список с элементами динамическими массивами или строками в формате записей переменной длины. Операции включение объекта с сохранением порядка и по логическому номеру, двоичный поиск. Использовать загрузку управляющих структур и поэлементную загрузку хранимых объектов (строк и массивов).
Структуры данных в двоичных файлах (тип хранимых объектов - параметр шаблона). Шаблон класса двоичного файла, содержащего массив указателей, односвязный или двусвязный список или двоичное дерево с элементами, тип которых является параметром шаблона. Операции включение объекта с сохранением порядка и по логическому номеру, двоичный поиск. Использовать загрузку управляющих структур и поэлементную загрузку хранимых объектов.

Примеры задач

Задача 1.

Класс разреженных матриц. Матрица представлена динамическим массивом ненулевых коэффициентов (строка, столбец - int, значение - double). Переопределить операцию сложения матриц, результат - объект-значение, операнды не меняются.

Задача 2.
Шаблон класса - двусвязный список, хранит элементы типа T. Переопределить операции вставки элемента по логическому номеру, сортировки в списке.

Задача 3.
Шаблон класса - двухсвязный список. Каждый элемент списка содержит статический массив указателей на элементы типа T. Первый пустой указатель в массиве имеет значении NULL. Реализовать операцию удаления элемента по логическому номеру. Освободившийся массив указателей удаляется из списка.

Задача 4.
Шаблон - двоичный файл, содержащий массив файловых указателей на объекты типа T, а также сами объекты. В начале файла лежат размер массива указателей, текущее количество указателей и смещение массива указателей. Данные упорядочены. Реализовать операцию двоичного поиска объекта. Использовать поэлементную загрузку данных из файла.

Экзаменационные вопросы

1. Основные свойства объектно-ориентированного программирования.
2. Новые средства языка C++.
3. Понятие класса в C++ и его роль в ООП. Описание класса в C++. Члены класса.
4. Описание и использование объектов в C++. Указатели на объекты класса.
5. Вызов методов класса. Указатель this.
6. Конструкторы. Конструктор копирования и конструктор без параметров. Деструкторы.
7. Статические элементы класса.
8. Модификаторы доступа к членам класса.
9. Дружественные функции и классы.
10. Указатели на элементы класса.
11. Перегрузка унарных операций. Способы перегрузки.
12. Перегрузка бинарных операций. Способы перегрузки.
13. Перегрузка операции присваивания.
14. Перегрузка операций new и delete
15. Перегрузка вызова функции
16. Перегрузка операции индексирования
17. Перегрузка арифметических операций и операции постфиксного и префиксного типа.
18. Перегрузка операций ввода/вывода в стандартный и файловый поток.
19. Основные понятия наследования. Модификаторы доступа при наследовании и их влияние на члены родительского класса при наследовании
20. Простое наследование. Конструкторы и деструкторы. Порядок создания и удаления объекта.
21. Множественное наследование. Конструкторы и деструкторы. Порядок создания и удаления объекта.
22. Проблемы множественного наследования. Виртуальные классы при множественном наследовании.
23. Виртуальные методы. Виртуальные деструкторы. Понятие полиморфизма. Реализация полиморфизма.
24. Отношение включения и отношение наследования. Закрытое наследование.

25. Локальные и вложенные классы.
26. Исключения в C++. Блоки try, catch, оператор throw. Обработка неловленных исключений.
27. Обработка исключительных ситуаций в классах. Создание пользовательского обработчика.
28. Шаблоны функций в C++. Специализированные шаблоны функций.
29. Шаблоны классов в C++. Определение функций шаблонных классов. Специализированные шаблоны классов.
30. Контейнерные классы в шаблонах. Итераторы.
31. Последовательные контейнеры. Организация хранения данных и методы их обработки.
32. Ассоциативные контейнеры. Организация хранения данных и методы их обработки.
33. Понятие потока ввода/вывода. Иерархия классов-потоков в библиотеке. Виды потоков.
34. Библиотека классов-потоков. Форматирование в потоках.
35. Библиотека классов-потоков. Состояние потока. Обработка ошибок в потоках.
36. Библиотека классов-потоков. Работа с текстовыми файлами.
37. Библиотека классов-потоков. Работа с двоичными файлами.
38. RTTI в C++. Операторы приведения типов.
39. Жизненный цикл ПО
40. Отладка и тестирование ПО.
41. Критерии качества ПО.
42. Модели разработки ПО.
43. Проектирование ПО.

4-ый семестр
Экзаменационные билеты включают 2 вопроса: теоретический и прикладной (разработка программы)
Список теоретических вопросов
1. Представление числовых и символьных данных в ПК. Базовый одинарный формат для вещественных чисел.
2. Сегментированная модель памяти реального режима. Формирование исполнительного (физического) адреса.
3. Программная модель 32-разрядных процессоров. Общие и специальные функции регистров.
4. Функциональное назначение сегментных регистров в ассемблерных программах. Возможности переназначения. Организация стека.
5. Флаги условий и управления процессором. Характеристика команд с точки зрения их влияния на флаги.
6. Форматы команд базового процессора I8086 и основные принципы их двоичного кодирования. Виды адресации.
7. Структуры данных программы и их инициализация (размещение) в памяти.
8. Использования директив макроопределений EQU и =.
9. Операторы языка ассемблера и их использование в адресных и константных выражениях.
10. Стандартные директивы управления сегментами. Шаблоны программ .EXE и .COM.
11. Набор упрощенных директив управления сегментами. Шаблоны программ .EXE и .COM.
12. Организация программы с использованием сегментов дальнего типа.
13. Основные типы команд передачи данных. Примеры записи команд общего назначения для различных способов адресации операндов.

14. Команды загрузки эффективного адреса, табличной трансляции и преобразования форматов (CBW, CWD и CWDE).
15. Арифметические команды. Рассмотреть примеры выполнения команды сложения (вычитания) для конкретных операндов с целью определения состояния флагов результата. Сформулируйте понимание и реализацию принципа арифметики многократной точности.
16. Характеристика команд для выполнения логических операций и команд сдвига. Примеры типового использования команд.
17. Команды безусловной и условной передачи управления. Примеры ассемблерной записи команды JMP для различных типов переходов.
18. Перечень команд условных переходов с указанием условий перехода (состояний флагов).
19. Программирование на ассемблере управляющих структур языка C++:
- условный оператор if_else и его разновидности;
- логический переключатель switch-case.
20. Ассемблерная команда Loop организации цикла и цикл типа For в C++. Дополнительные команды организации циклов в ассемблере. Пример применения.
21. Организация процедуры и её интерфейс с основной программой. Способы передачи параметров между основной программой и процедурой.
22. Макросы. Определения. Примеры. Локальные метки в макросе.
23. Концепция модульного программирования. Организация программных модулей. Директивы PUBLIC и EXTRN.
24. Система прерываний в компьютере. Аппаратные и программные средства, реализующие механизм прерываний. Процедура обработки прерываний в реальном режиме.
25. Резидентные программы и их организация.
26. Нереентерабельность MS DOS и пути её преодоления в обработчиках аппаратных прерываний.

Несколько примеров экзаменационных заданий на разработку программ.

1. Составьте программу извлечения квадратного корня положительного числа (? 65535) руководствуясь алгоритмом: квадратный корень из целого числа равен количеству последовательных нечётных чисел, которое можно из него вычесть.
2. Напишите программу для выдачи на экран сообщения вида "Вы получили за экзаменационную работу N балов", где N - число от 1 до 127, двоичный эквивалент которого определён макроопределением (например, N = 99)
3. Массив знаковых чисел с именем Data_list включает N слов. Напишите программу, которая помещает наибольшее число в регистр AX.
4 Определить номер m (индекс) числа A в массиве из N слов. Значение числа A определить директивой равенства '='. Номер числа возвратить в регистре DL (CF=1). Если число не найде-но, установить CF=0.
5. Напишите программу, которая подсчитывает число положительных, нулевых и отрицатель-ных чисел в буфере Array, включающем 128 слов. Соответствующие значения поместить в пе-ременные: Num_positive, Num_zerro, Num_negative.
6. Выполните реверсирование ASCII-строки длиною N байт. Для реверсирования используйте временный буфер в сегменте кода.
7. Выполнить преобразование ASCIIZ-строки (признаком конца строки является нулевой ASCII-символ), исключив из неё русские прописные буквы от А до Я (80h…9Fh).
8. Составьте программу вывода текста некоторой ASCII строки с некоторым атрибутом в точ-ку "строка-столбец", используя непосредственное программирование видеобуфера. Длина строки не превышает 40 символов. Выход из программы - по нажатию клавиши.

9. Перевести положительное 6 разрядное восьмеричное ASCII-число из буфера Octo_buf (6 байт) в 16-разрядный двоичный код в AX (младший разряд числа располагается на месте старшего байта буфера). Установить флаг CF=1, если произошло переполнение.