

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет автоматики и вычислительной техники
Заочный факультет

УТВЕРЖДАЮ
Декан АВТФ

профессор, д.т.н. Гужов Владимир Иванович

“ ____ ” _____ г.

УТВЕРЖДАЮ
Декан ЗФ

профессор, д.т.н. Темлякова Зоя Савельевна

“ ____ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Программирование на языке высокого уровня

ООП: специальность 230101.65 Вычислительные машины, комплексы, системы и сети

Шифр по учебному плану: ОПД.Ф.3

Факультет: заочный заочная форма обучения

Курс: 2 3, семестр: 4 5 6

Лекции: 24

Практические работы: - Лабораторные работы: 24

Курсовой проект: - Курсовая работа: - РГЗ: -

Самостоятельная работа: 212

Экзамен: 6 Зачет: 5

Всего: 260

Новосибирск

2011

Рабочая программа составлена на основании Государственного образовательного стандарта высшего профессионального образования по направлению (специальности): 654600 Информатика и вычислительная техника.(№ 224 тех/дс от 27.03.2000)

ОПД.Ф.3, дисциплины федерального компонента

Рабочая программа обсуждена на заседании кафедры Вычислительной техники протокол № 1 от 28.01.2011

Программу разработал

доцент, к.ф.м.н.

Гейнц Оксана Михайловна

Заведующий кафедрой

профессор, д.т.н.

Губарев Василий Васильевич

Ответственный за основную образовательную программу

профессор, д.т.н.

Губарев Василий Васильевич

1. Внешние требования

Таблица 1.1

Шифр дисциплины	Содержание учебной дисциплины	Часы
ОПД.Ф.3	Квалификационные требования ГОСВПО по направлению 230100 - "Информатика и вычислительная техника" специальность 230101.65 специалист по информатике и вычислительной технике: Для компетентного и ответственного решения профессиональных задач специалист: 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 разработки объектов профессиональной деятельности; готов участвовать в проведении научных исследований и выполнении технических разработок в своей профессиональной области; 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 профессиональной деятельности; способен в условиях развития науки и изменяющейся социальной практики к переоценке накопленного опыта, анализу своих возможностей, умеет приобретать новые знания, используя современные информационные образовательные технологии; Специалист должен знать: методические и нормативные материалы по проектированию и разработке объектов профессиональной деятельности;	260

	технологии проектирования и разработки объектов профессиональной деятельности; перспективы и тенденции развития информационных технологий; стандарты и технические условия; технические характеристики и экономические показатели лучших отечественных и зарубежных образцов объектов профессиональной деятельности; 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. Выдержка из ГОС: ОПД.Ф.06 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.	
--	--	--

2. Особенности (принципы) построения дисциплины

Таблица 2.1

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 230101 "Вычислительные машины, комплексы, системы и сети" (квалификация - инженер) в цикл "Общепрофессиональные дисциплины" ОПД.Ф.06.
Адресат курса	Студенты 3-го курса заочного факультета специальности 230101 "Вычислительные машины, комплексы, системы и сети".
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 (24 часа в 3 и 4 семестре). На лабораторных работах студенты реализуют реальные программы с использованием теоретических положений, изученных на лекциях с использованием реальных сред разработки. - Контрольные работы выполняются по индивидуальному заданию. Примерные варианты заданий приведены в программе.
Ядро дисциплины	Требования по ГОСВПО к обязательному минимуму содержания по направлению 654600 - "Информатика и вычислительная техника" ОПД.Ф.06 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	Форма занятий и объём в часах:		
	6 семестр	4 семестр	5 семестр
	Лекции: 8	4	8+ 4
	(обзорные+установочные)	(установочные)	(обзорные)
	Лабораторные работы 12		12
	Выполняемые задания		Контрольная работа
	Контрольная работа		
	Самостоятельная работа в часах 106		110
	Общее кол-во часов на ДО		

--			
Форма отчётности: экзамен (6 семестр)		зачет(5 семестр)	

3. Цели учебной дисциплины

Таблица 3.1

После изучения дисциплины студент будет

иметь представление	
1	о технологии проектирования сложных модульных программ
2	о процессах распределения памяти в программе
3	многообразии структур данных и их использовании
4	об основных компонентах баз данных и их организации
23	об архитектурных особенностях ПК
24	о характере программ, разрабатываемых на языке ассемблера
25	об использовании инструментальных средств разработки и отладки ассемблерных программ
26	о системе команд 32-битных процессоров i80x86 и принципах их двоичного кодирования
знать	
5	операции побитовой обработки машинного слова
6	способы объявления и программирования структурированных переменных и массивов структур
7	указатели и правила адресной арифметики,
8	правила и функции динамического распределения памяти программы
9	программирование обработки динамических переменных и массивов
10	особенности рекурсивных функций и правил их программирования
11	программирование структур данных на базе массива - стека, очереди, списков, списка, дерева
12	программирование динамических структур данных - таблиц массивов, стека, очереди, связанных списков, двоичных деревьев, иерархических структур
17	критерии качества программ, способы верификации программ
27	этапы создания программы на языке ассемблера (жизненный цикл программы)
28	программную модель 32- разрядных процессоров i80x86
29	представление числовой и символьной информации в компьютере на разных стадиях выполнения программы
30	способы инициализации (задания) разнообразного типа данных программы
31	виды адресации операндов в памяти, особенности 32-битного режима адресации
32	структуру программ .EXE и .COM и их отображение (образ) в памяти
33	программирование типовых управляющих структур языка C++
34	основные приёмы программирования задач, связанных с вводом/выводом числовой и символьной информации в компьютере
35	концепцию модульного программирования
36	особенности организации и выполнения подпрограмм (процедур) в пользовательских программах
37	структуру и организацию макросов
38	Организацию многомодульных программ
39	типы прерываний реального режима и процедуры их обслуживания

40	структуру прикладных обработчиков программных и аппаратных прерываний
41	организацию интерфейса между ассемблером и языком C++
уметь	
13	разрабатывать в среде компилятора модульные программы с использованием проекта
14	выбирать различные модели управления памятью программы в среде компилятора
15	управлять выполнением программы с помощью меню программы
16	использовать динамическую память программы для организации динамических массивов и структур,
18	обрабатывать исключительные ситуации в программах
19	разрабатывать и применять шаблоны классов и функций
42	выполнить отладку и тестирование программы в отладчике Turbo Debugger на уровне машинных команд
43	пользоваться сервисными функциями DOS и BIOS в прикладных программах реального режима
44	пользоваться основными конструкциями языка ассемблера при разработке прикладных программ
45	произвести декомпозицию задачи на отдельные модули с последующим формированием их в виде самостоятельных объектных модулей
иметь опыт (владеть)	
20	использования готовых библиотек классов
21	документирования разработанных программных продуктов
22	тестировании и отладки программ
46	машинно-ориентированного способа мышления при разработке ассемблерных программ
47	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: 4		
Модуль: Критерии качества программ. Способы верификации программ		
Дидактическая единица: Процедуры и функции. Программирование рекурсивных алгоритмов; способы конструирования и верификации программ.		
Критерии качества программы. Утверждения о программах, корректность программ, инвариантные утверждения. Способы конструирования и верификации программ.	0,5	17
Семестр: 5		
Модуль: Машинно - ориентированные операции и битовая арифметика		

Дидактическая единица: Стандартные типы данных		
Битовая структура переменной. Понятие битового поля и маски. Ис-пользование битовой структуры переменной для фиксации элемен-тов множеств, событий и состояний, управления внешними устрой-ствами. Использование восьмеричной и шестнадцатеричной систем счисления для установки битовых полей и масок. Машинно - ориен-тированные операции и битовая арифметика: логические операции И, ИЛИ, НЕ, исключающее ИЛИ. Программирование машинной об-работки битовых полей, маскирования битов, упаковки данных.	1	5
Модуль: Указатели и управление памятью		
Дидактическая единица: представление основных управляющих структур программирования.		
Адресный указатель и его смысл. Объявление указателя в программе. Тип указателя. Жизненный цикл указателя: объявление, связывание, косвенная выборка значения. Указатель, как формальный параметр и результат функции. Явное и неявное преобразование типа значения через указатель. Адресная арифметика. Указатели и массивы. Про-граммирование типовых приемов обработки одномерного массива через указатель. Программирование типовых приемов обработки строки текста через указатель. Массив указателей. Представление двумерного массива с помощью массива указателей. Программиро-вание типовых приемов обработки двумерного массива, представ-ленного в виде массива указателей.	1,5	16, 2, 7, 8
Модуль: Структурированные переменные и массивы структур. Организация памяти программы и динамическая память		
Дидактическая единица: Теорема структуры и структурное программирование; анализ программ; утверждения о программах; корректность программ;; правила вывода для основных структур программирования; инвариантные утверждения.		
Структура и объединение, как пользовательский тип данных. Сход-ство и различие структуры и объединения. Поле структуры, объеди-нения. Программирование работы с отдельным полем структуры, объединения. Одномерный массив структур. Правила доступа к по-лям структур в массиве. Программирование типовых операций обра-ботки одномерного массива структур. Указатель на структуру. Программирование доступа к отдельным полям структуры через указатель. Программирование типовых приемов обработки одномерного массива структур через указатель на массив.	0,5	12, 16, 6

Модуль: Динамические переменные и массивы.		
Дидактическая единица: Массивы; утверждения о массивах; записи; файлы.		
Структура памяти программы. Модели организации памяти про-граммы Tyni, Small, Medium, Large, Huge. Близкий, дальний и нор-мализованный указатели и их связь с моделями памяти. Динамическая память программы. Функции управления динамиче-ской памятью программы. Динамический одномерный массив. Строка - как динамический одномерный массив. Динамический двумерный массив и программирование его создания. Уничтожение динамических переменных и массивов. Надежные массивы и их программная реализация.	1	16, 9
Модуль: Программирование рекурсивных алгоритмов		
Дидактическая единица: Процедуры и функции. Программирование рекурсивных алгоритмов; способы конструирования и верификации программ.		
Рекурсия. Определение рекурсии. Рекурсивные структуры данных, функции. Особенности проектирования рекурсии. Динамика рекур-сии, рекурсивный спуск и возврат. Дно рекурсии. Передача парамет-ров и результата между рекурсивными вызовами. Программирование рекурсивных алгоритмов задач поиска, движения по лабиринту, расстановки фигур.	1	10, 12
Модуль: Введение в структуры данных		
Дидактическая единица: Линейные списки: основные виды и способы реализации; линейный список как абстрактный тип данных.		
Понятие структуры данных. Элемент, как алгоритмическая компо-нента структуры данных. Статические и динамические структуры данных Неупорядоченные структуры данных, доступ к элементам по логическому номеру. Упорядоченные структуры данных, доступ к элементу по ключевому значению. Классификация структур данных: стек, очередь, списки, деревья, ие-рархические таблицы, иерархические структуры. Основные операции структур данных: поиск, вставка нового элемен-та, удаление элемента.	0,5	16, 3, 4
Модуль: Статические структуры данных		
Дидактическая единица: Линейные списки: основные виды и способы реализации; линейный		

список как абстрактный тип данных.		
Статические структуры для хранения и поиска данных. Представление стека, очереди, списка, двоичного дерева поиска на базе массива. Программирование основных операций для стека на базе массива. Программирование основных операций для циклической очереди на базе массива. Программирование основных операций для неупорядоченного списка на базе массива. Программирование основных операций для упорядоченного списка на базе массива. Программирование основных операций для дерева поиска на базе массива.	1	11, 16
Модуль: Динамические структуры данных		
Дидактическая единица: Динамические структуры данных.		
Динамические связные списки. Элемент списка и его рекурсивная структура. Роль адресного указателя в организации динамического связного списка. Заголовок, текущий элемент списка. Структура динамического односвязного, двухсвязного списка. Программирование итерационного и рекурсивного алгоритмов поиска элемента с заданным значением в связном списке. Односвязный упорядоченный список. Программирование итерационного и рекурсивного алгоритмов вставки и удаления элемента по значению из односвязного упорядоченного списка. Проблема концов списка при программировании операций. Программирование итерационного и рекурсивного алгоритмов вставки и удаления элемента с заданным номером из двухсвязного списка. Проблема концов списка при программировании операций. Организация стека и очереди на базе односвязного списка, программирование основных операций. Динамические связные деревья, как рекурсивные структуры данных. Элемент дерева (узел) и его рекурсивная структура. Двоичное (бинарное) дерево поиска, как упорядоченная с	5	12, 16, 7
Семестр: 6		
Модуль: АРХИТЕКТУРНЫЕ ОСОБЕННОСТИ ПК		
Дидактическая единица: Ассемблер, архитектура ПК, типы данных, программная модель		
Архитектура вычислительных машин. Основные понятия Основные архитектурные отличия процессоров класса Pentium. Реальный и защищённый режимы работы процессоров i80x86. Сегментированная модель	2	23, 24, 25, 26, 27, 28

памяти реального режима. Типы данных в компьютере. Целочисленные формы данных: биты, бай-ты, слова. Десятичные числа. Символьные данные. Указатели адреса. Представление чисел с плавающей точкой (одинарный базовый и двойной форматы). Форматы, используемые в процессоре с плавающей точкой. Программная модель 32-битного процессора, назначение регистров и организация стека. Регистр флагов. Режимы адресации операндов и форматы машинных команд 32-битных процессоров. Механизм 16-битной адресации. Примеры деассемблирования и ассемблирования отдельных команд. Механизм 32-битной адресации		
Модуль: ФУНДАМЕНТАЛЬНЫЕ ПОНЯТИЯ ЯЗЫКА АССЕМБЛЕРА.		
Дидактическая единица: Синтаксис ассемблерной строки, этапы разработки программы, директивы инициализации данных, выражения и операторы		
Программное обеспечение (ассемблеры) для процессоров фирмы Intel. Основные этапы создания программы: текст, ассемблирование, компоновка и отладка. Типы исполняемых файлов. Синтаксис ассемблера. Ассемблерные строки и идентификаторы. Этапы разработки ассемблерной программы Структура данных. Переменные программы (инициализированные и не-инициализированные) и директивы их определения для различных типов переменных. Макроопределения: директивы EQU и =. Выражения (адресные и константные) и операторы. Назначения операторов и синтаксис их использования. Правила выполнения арифметических операций с адресными выражениями. Использование счётчика текущего адреса \$ в ассемблерных программах	2	25, 27, 30
Модуль: СТРУКТУРЫ АССЕМБЛЕРНЫХ ПРОГРАММ		
Дидактическая единица: Стандартные и упрощенные сегментные директивы, шаблоны программ .EXE и .COM и их образ в памяти и на диске		
Структуры ассемблерных программ. Два набора директив управления сегментами и их функциональное назначение. Модели памяти и упрощенные директивы определения сегментов в режимах MASM и IDEAL. Сегментные атрибуты и	1	32

их значения для упрощенных директив. Шаблоны (оболочки) программы для .EXE-и .COM- файлов. Образ про-граммы в памяти компьютера. Использование сегментов данных дальнего типа		
Модуль: СИСТЕМА КОМАНД 32-БИТНЫХ ПРОЦЕССОРОВ.		
Дидактическая единица: Система команд, режимы адресации операндов		
Команды передачи данных. Команды общего назначения MOV и их ас-семблерная запись для различных способов адресации операндов. Виды косвенной адресации в режиме формирования 16- битного адреса. Изме-нение типа данных или замена типа переменной (директива Label). Команды загрузки исполнительного и полного адреса. Стековые коман-ды. Команды ввода/вывода. Дополнительные виды адресации в 32- разряд-ных процессорах. Арифметические операции сложения в обратном и дополнительном ко-дах. Перенос и переполнение. Изменение знака числа и операция вычита-ния. Команды сложения и вычитания. Команды преобразования форматов. Программная реализация арифметики двойной (многократной) точности для операций сложения и вычитания. Анализ возможных результатов (зна-чения флагов) исполнения команды сравнения над операндами без знака и со знаком. Команды умножения и деления беззнаковых и знаковых целых чисел. Команды логических операций, сдвига и вращения. Команды проверки и модификации бит и байтов. Команды передачи управления. Безу	2,5	26, 31, 33, 44
Модуль: МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ		
Дидактическая единица: Программный модуль, процедура , макрос, макрокоманда, Многомодульная программа, библиотека объектных модулей.		
Концепция модульного программирования. Возможные подходы в реали-зации многомодульных программ. Процедуры. Вызов, возврат и их типы - внутри- и межсегментные. Орга-низация интерфейса с процедурой (способы передачи параметров между основной программой и процедурой по ссылке и значению). Передача па-раметров процедуре через стек. Многомодульные программы на основе подпрограмм, оформленных в виде модулей, допускающих независимое ассемблирование. Требования к структурной организации программных модулей, вытекающие из принятой модели памяти и соответствующих ей директив	1,5	35, 36, 37, 38, 45

описания сегментов. Внешние и внутренние имена. Директивы EXTRN и PUBLIC. Библиотека объектных модулей и её использование при создании многомодульных программ Макросы (макроопределение, макрокоманда и макрорасширение). Особенности подстановки в макросы фактических параметров различных типов (символьные, числовые и строковые). Микрооператоры подстановки. Локальные метки макроса. Блоки повторения: REPT, IRP и IRPC. Макросы и блоки повторения.		
Модуль: СИСТЕМА ПЕРЕРЫВАНИЙ. РЕАЛЬНОГО РЕЖИМА		
Дидактическая единица: Прерывания, контроллер прерываний, процедуры обработки прерываний		
Типы прерываний Программные прерывания и процедура их обработки. Аппаратные прерывания. Контроллер прерываний и процедура обработки аппаратных прерываний. Прикладные обработчики программных прерываний: перехват прерываний, взаимосвязь прикладных и системных обработчиков. Особенности обработки аппаратных прерываний	1,5	39, 40
Модуль: ИНТЕРФЕЙС МЕЖДУ АССЕМБЛЕРОМ И ЯЗЫКОМ C++.		
Дидактическая единица: Основные принципы взаимодействия между Turbo assembler и C++, процедуры вызова из C++ ассемблерных функций и наоборот.		
Внешние программы (модули) на ассемблере. Программная среда. Принципы взаимодействия между Turbo Assembler и Borland C++. Вызов ассемблерных функций из C++. Передача аргументов из C++ в язык ассемблер. Вызов функций C++ из ассемблера и передача аргументов	1,5	41, 47

Лабораторная работа

Таблица 4.2

(Модуль), дидактическая единица, тема	Учебная деятельность	Часы	Ссылки на цели
Семестр: 5			
Модуль: Лабораторный практикум			
Дидактическая единица: Ввод-вывод информации, обработка строковых переменных			
Программирование главного меню текстового редактора	Выполнение лаб.раб. в соответствии с метод.	4	1, 13, 15

	требованиями		
Модуль: Лабораторная работа №2			
Дидактическая единица: Массивы; утверждения о массивах; записи; файлы.			
Разработка основных функций текстового редактора	Выполнение лаб.раб. в соответствии с метод. требованиями	4	1, 12, 3, 4
Модуль: Лабораторная работа №3			
Дидактическая единица: Теорема структуры и структурное программирование; анализ программ; утверждения о программах; корректность программ;; правила вывода для основных структур программирования; инвариантные утверждения.			
Создание и комплексная отладка многомодульной программы текстового редактора.		4	1, 13, 17
Семестр: 6			
Модуль: Лабораторный практикум			
Дидактическая единица: Ввод-вывод информации, обработка строковых переменных			
Функции DOS ввода-вывода символьной информации. Обработка строковых переменных .	- изучает методический материал по использованию функций DOS для ввода-вывода символьной информации ; - знакомится с понятием расширенного ASCII-кода клавиш и его использованием для управлением хода программы ; - изучает методический материал по применению строковых команд для обработки строк символов ; - в соответствии с заданием, применяя базовые алгоритмы поиска и сортировки данных, разрабатывает собственную программу; - производит отладку	4	25, 36, 40, 43, 44, 46

	программы с помощью отладчика TD		
Дидактическая единица: Сервисные функции BIOS.			
Применение сервисных функций драйвера BIOS для работы с экраном и клавиатурой.	- знакомится со справочными данными по функциям BIOS; - запускает на исполнение демонстрационные программы, анализируя особенности использования функций BIOS для выполнения необходимых действий; - разрабатывает (в соответствии с заданием) собственную программу и производит её отладку	4	37, 44, 46
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 декомпозицию; -создаёт список переменных задачи, являющихся общими для нескольких модулей; -создаёт программные модули задачи; -создаёт исполняемый файл многомодульной программы и	4	29, 34, 35, 38, 45, 46

	производит её отладку; -формирует из вспомогательных объект-ных модулей свою собственную библиотеку и заново производит трансляцию		
--	--	--	--

5. Самостоятельная работа студентов

Семестр- 5, Подготовка к зачету

На подготовку к зачету выделяется 18 часов.

Семестр- 5, Контрольные работы

Цель контрольной работы: приобретение и закрепление теоретических знаний и практических навыков в разработке алгоритмов и программ на уровне прикладного и системного программирования.

В течение семестра студент выполняет контрольную работу в соответствии с индивидуальным заданием. Контрольная работа включает разработку алгоритмов и программ по пяти темам:

- машинно-ориентированное программирование,
- рекурсивные алгоритмы,
- массивы указателей,
- динамический связный список,

Отчет по контрольной работе для задач 1-4 должен содержать описание алгоритма решения задачи, текст программы с комментариями, результаты тестирования программы на ЭВМ. Работа программ должна быть продемонстрирована преподавателю.

Объем часов самостоятельной работы - 20 часов

Семестр- 5, Подготовка к занятиям

Объем часов самостоятельной работы:

самостоятельная работа с литературой для усвоения материала	18 часов	
при подготовк к 3-м лабораторным работам		36
часов		

Семестр- 6, Контрольные работы

Контрольная работа на тему "Жизненный цикл программы. Система команд процессора и принципы их кодирования"

Объем часов самостоятельной работы - 12.

Семестр- 6, Подготовка к экзамену

Примерный перечень вопросов и задач приведен в разделе контролирующих иатериалов.

1. Объем часов самостоятельной работы для подготовки к экзамену составляет - 66 часов (фактически

это часы самостоятельной работы с литературой в течение семестра)

Семестр- 6, Подготовка к занятиям

Объем часов самостоятельной работы:

при	подготовк	к	3-м	лабораторным	работам	
12						
при	подготовке	к	обзорным	лекциям	(8	часов)
16						

6. Правила аттестации студентов по учебной дисциплине

5-ый семестр.

Правила текущей аттестации:

1. В течение пятого семестра необходимо представить и защитить 3 лабораторные работы, контрольную работу в сроки, установленные учебным графиком (см. таблицу).
2. К защите допускаются студенты, выполнившие лабораторные работы, контрольную 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 ходу выполнения работы).
4. Максимальное количество баллов (12 или 22-24 в зависимости от вида работы) выставляется, если студент полностью ответил на все вопросы, без серьезных замечаний и недочетов.
5. Количество баллов 7-11 или 16-21 (в зависимости от вида работы) выставляется, если студент полностью ответил на два вопроса из трех, причем один из вопросов - практический.
6. Минимальное количество баллов 6 или 12-15 (в зависимости от вида работы) выставляется, если студент ответил на два вопроса из трех частично, с серьезными замечаниями, недочетами.
7. Передача лабораторной работы, контрольной работы назначается, если студент не ориентируется в учебном материале, не может объяснить ход и результаты выполнения работы. В случае передачи работы происходит потеря баллов (максимальное количество баллов составляет 10 или 20 в зависимости от вида работы).
8. В случае представления и защиты работ с опозданием от учебного графика происходит потеря баллов (опоздание на 1 месяц - потеря 2 или 4 баллов в зависимости от вида работы, опоздание на 2 месяца - потеря 4 или 8 баллов, 3 месяца и более - потеря 50% баллов от максимально возможного).

Правила итоговой аттестации:

1. К зачету допускаются студенты, сдавшие лабораторные работы, контрольную работу, и набравшие не менее 50% (30 баллов) по результатам текущего рейтинга.
2. Зачет проводится в письменном виде, предлагается две задачи и один теоретический вопрос.
3. Максимальное количество 36-40 баллов выставляется, если все задания выполнены полностью, без серьезных замечаний.
4. Количество баллов 30-35 выставляется, если успешно выполнены два задания из трех, причем один из отвеченных вопросов - теоретический.
5. Минимальное количество баллов 20-29 выставляется, если выполнены два задания из трех, но с серьезными ошибками, замечаниями, недочетами.
6. Возможно получить зачет по дисциплине без сдачи зачета, если студент в течение семестра выполняет дополнительные задания повышенной сложности и набирает свыше 90 баллов по текущему рейтингу.

Таблица

№

п/п	Вид учебной работы	Максимальное кол-во баллов	Диапазоны баллов
-----	--------------------	----------------------------	------------------

Срок представления и защиты (неделя семестра)

Пятый семестр:

1. Лабораторная работа №1	12	6-12
январь (февраль)		
2. Лабораторная работа №2	12	6-12
январь (февраль)		
3. Лабораторная работа №3	12	6-12
январь (февраль)		
4. Контрольная работа	24	12-24
5 семестр		
Итого по текущему рейтингу:	60	30-60
5. Зачет	40	20-40
Итого за пятый семестр:	100	50-100 (зачет)

6- 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.

1. В течение 6-го семестра необходимо выполнить контрольную работу и 4 лабораторные работы (см. таблицу 1).

2. К защите контрольной и лабораторных работ необходимо представить машинописный экземпляр отчёта.

3. Абсолютное значение проставляемых баллов 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.

Таблица 1.

№	Вид учебной деятельности	Максимальное кол-во баллов	
Диапазоны баллов	Срок сдачи		
1	Контрольная работа	15	6 - 15
до сессии			
2 Работа №1	15	8 - 15	Во
время сессии			
3 Работа №2	15	8 - 15	-
- // -- // -- // --			
4 Работа №3	15	8 - 15	-
- // -- // -- // --			
Итого: Текущий рейтинг		60	30 - 60

Экзамен и итоговая аттестация.

1. К экзамену допускаются студенты, сдавшие лабораторные работы, курсовой проект и, набравшие не менее 50% от максимального значения текущего рейтинга (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).

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

Итого: Рейтинг по дисциплине (оценка за курс).

50 - 100

Оценка за курс в баллах, зачётных единицах ECTS

и в традиционной форме

..., C₋; удовл)

50-72 (E,

..., B; хор)

73-87 (C,

..., A⁺; отл)

88-100 (B⁺,

7. Список литературы

7.1 Основная литература

В печатном виде

1. 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.
2. 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.
3. 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.
4. Афанасьев В. А. Assembler IBM PC : конспект лекций / В. А. Афанасьев ; Новосиб. гос. техн. ун-т. - Новосибирск, 2008. - 257, [1] с. : ил.
5. Подбельский В. В. Программирование на языке Си : учебное пособие для вузов по направлениям: "Прикладная математика и информатика", "Информатика и вычислительная техника", специальностям "Прикладная математика", "Вычислительные машины, комплексы, системы и сети управления" / В. В. Подбельский, С. С. Фомин. - М., 2007. - 600 с. : ил., табл. - Рекомендовано МО.
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. 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.
8. 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.
9. 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.
10. Сеницын С. В. Программирование на языке высокого уровня : учебник / С. В. Сеницын, А. С. Михайлов, О. И. Хлытчиев. - М., 2010. - 392, [1] с. : ил., табл. - Рекомендовано УМО.
11. 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.
12. Подбельский В. В. Программирование на языке Си : учебное пособие для вузов / В. В. Подбельский, С. С. Фомин. - М., 2005. - 600 с. : ил., табл. - Рекомендовано МО.

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 IBM PC.

7.2 Дополнительная литература

В печатном виде

1. Абель П. Ассемблер. Язык и программирование для IBM PC / пер. с англ. под ред. С.М. Молявко. - Киев [и др.], 2003. - 734 с. : табл.
2. Романов Е. Л. Практикум по программированию на C++ : [учебное пособие] / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - СПб., 2004. - 426, [1] с. : ил.
3. Романов Е. Л. Язык Си++ в задачах, вопросах и ответах : [учебное пособие] / Е. Л. Романов. - Новосибирск, 2003. - 426, [1] с. : ил.
4. Дьюхарст С. Программирование на C++ : Пер. с англ. / С. Дьюхарст. - Киев, 1993. - 272 с. : ил.

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.

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

В электронном виде

1. Афанасьев В. А. Учебно-методический комплекс для практикума по дисциплине "Программирование" (раздел "ASSEMBLER IBM PC") [Электронный ресурс] : [электронный учебно-методический комплекс] / В. А. Афанасьев ; Новосиб. гос. техн. ун-т. - Новосибирск, [2004]. - Режим доступа: <ftp://tkvt.cs.nstu.ru/teacher/afaval/assembler>. - Загл. с экрана.

8.2 Программное обеспечение

1. НГТУ, Романов Е. Л. Си/Си ++. От дилетанта до профессионала [Электронный ресурс] : электронное учебное пособие ;, Образцы программ, тестовые задания на Си/Си++
2. Microsoft, Visual Studio, Интегрированная среда разработки

9. Контролирующие материалы для аттестации студентов по дисциплине

5-ий семестр

В билеты зачета 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 значение 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-поток в библиотеке. Виды потоков.
34. Библиотека классов-поток. Форматирование в потоках.
35. Библиотека классов-поток. Состояние потока. Обработка ошибок в потоках.
36. Библиотека классов-поток. Работа с текстовыми файлами.
37. Библиотека классов-поток. 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.

6-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. Использование 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 баллов", 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 вернуть в регистре DL (CF=1). Если число не найде-но, установить CF=0.
5. Напишите программу, которая подсчитывает число положительных, нулевых и отрицательных чисел в буфере Array, включающем 128 слов. Соответствующие значения поместить в переменные: Num_positive, Num_zerro, Num_negative.
6. Выполните реверсирование ASCII-строки длиной 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 точку "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.