

Министерство образования и науки Российской Федерации
 Федеральное государственное бюджетное образовательное учреждение
 высшего образования
 «Владимирский государственный университет
 имени Александра Григорьевича и Николая Григорьевича Столетовых»
 (ВлГУ)



Проректор по учебно-методической работе

А.А. Панфилов

« 15 » 02 2016 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«ПРОГРАММИРОВАНИЕ»
 (наименование дисциплины)

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Профиль/программа подготовки _____

Уровень высшего образования _____ бакалавр _____

Форма обучения _____ очная _____

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежу- точного контро- ля (экс./зачет)
1	5/180	36		36	72	Экзамен(36)
2	4/144	36	18	18	72	Зачет, КП
Итого	9/324	72	18	54	144	Экзамен(36), Зачет, КП

Владимир, 2016

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целями освоения дисциплины «Программирование» являются формирование у студентов знаний по основам алгоритмизации вычислительных процессов и общим принципам программирования, представлению основных структур программ и данных; создание фундаментальной основы знаний, необходимой при проектировании программ для вычислительных систем; изучение основных этапов решения задач на ЭВМ; изучение синтаксических и семантических конструкций языков программирования высокого уровня; освоение студентами основных приемов программирования типовых задач, а также изучение архитектуры вычислительной машины, принципов обработки машинных команд, особенностей реализации обработки различных типов данных; освоение студентами соответствия между операторами языков высокого уровня и командами языка ассемблера; получение навыков программирования процессора на языке ассемблер.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Программирование» относится к базовой части ОПОП по направлению 09.03.01 – «Информатика и вычислительная техника». Дисциплина логически, содержательно и методически тесно связана с рядом теоретических дисциплин и практик ОПОП.

Для успешного изучения дисциплины «Программирование» студенты должны быть знакомы с дисциплинами «Математика» и «Введение в специальность», которые формируют необходимые для изучения программирования способности к обобщению и анализу информации, знания математического анализа и алгоритмов, структурных блоков ЭВМ, способов представления данных в ЭВМ, способность использовать персональный компьютер и системы программирования для разработки программного обеспечения, готовность понимать актуальность совершенствования языков программирования, подходов к проектированию программных систем, программного обеспечения в аспектах технического и научного прогресса.

Без освоения дисциплины «Программирование» невозможна дальнейшая успешная подготовка студентов по направлению 09.03.01. Дисциплина «Программирование» определяет саму возможность изучения практически всех последующих дисциплин, поскольку в процессе изучения используются ЭВМ и языки высокого уровня, как средства и инструменты для исследований и получения результатов, для решения специализированных задач.

Дисциплина «Программирование» является основой следующих дисциплин: «Языки программирования», «Операционные системы», «Системное программное обеспечение», «Базы данных», «Графические информационные технологии», «Технология программирования» и ряде других дисциплин, связанных с изучением или использованием программного обеспечения ЭВМ.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

- **способность** к самоорганизации и самообразованию (ОК7);
- **способность** осваивать методики использования программных средств для решения практических задач (ОПК2);
- **способность** разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина" (ПК1).

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

Знать: основные этапы решения задач на ЭВМ, базовые структуры алгоритмов, способы записи алгоритмов; основные синтаксические конструкции языков программирования высокого уровня, структуры данных и принципы обработки данных различного типа; особенности структурного программирования, принципы модульного проектирования программ, понятия подпрограммы и связи по управлению, способы передачи данных в подпрограммы; особенности архитектуры процессора, назначение основных регистров процессора, алгоритм работы процессора и особенности представления и обработки данных целого и вещественного типов; организацию оперативной памяти при различных модулях программ, назначение стека и принципы работы со стекком; основные команды языка ассемблера, стандартные соглашения при проектировании разноразличных программ; состав систем программирования и основы использования дополнительных пакетов и библиотек (ПК1).

Уметь: ставить задачу и разрабатывать алгоритм ее решения на современных ЭВМ, выполнять анализ и декомпозицию сложной задачи в соответствии с принципами структурного программирования, осознанно выбирать адекватные методы решения задач, используя, в том числе, существующие алгоритмы; использовать системы программирования и предоставляемые пакеты библиотек; выполнять компиляцию, отладку и тестирование составленных программ; разрабатывать основные программные документы (ОПК2, ПК1).

Владеть: языками процедурного программирования, навыками разработки и отладки программ на алгоритмическом языке высокого уровня С++, ассемблер; навыками работы с технической и справочной литературой, методикой разработки, компилирования и отладки программ на языках высокого уровня в системах программирования, средствами систем программирования (ОК7, ОПК2).

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 9 зачетных единиц, 324 часов.

№ п/п	Раздел (тема) дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем учебной работы, с применением интерактивных методов (в часах / %)	Формы текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	Контрольные работы	СРС	КП / КР		
1	2	3	4	5	6	7	8	9	10	11	12
1	Алгоритмизация задач										
1.1	Введение. Классификация языков программирования. Этапы создания ПО	1	1	2						1/50	
1.2	Понятие алгоритма. Способы описания алгоритмов. Принцип пошаговой детализации. Структуры алгоритмов.		2	2		4		6		1/17	
1.3	Система программирования.		3	2				2			
2	Синтаксические конструкции										
2.1	Понятие алгоритмического языка. С++ Алфавит языка. Переменные и константы. Типы данных	1	4	2		4		6		1/17	
2.2	Структура программы.		5	2				2		1/50	
2.3	Линейные программы. Операторы организации разветвляющихся структур		6	2		4		10		1/17	Рейтинг-контроль №1
2.4	Операторы организации циклической обработки		7	2				2		1/50	
2.5	Неструктурные алгоритмы. Преобразование неструктурных алгоритмов в структурные.		8	2		4		6		1/17	

1	2	3	4	5	6	7	8	9	10	11	12
3	Структурные типы данных										
3.1	Массивы. Обработка массивов.	1	9	2				2		1/50	
3.2	Символьные массивы. Строки.		10	2		4		4		1/17	
3.3	Структуры.. Перечислимые типы		11-12	2				6		1/50	Рейтинг-контроль №2
4	Модульное программирование										
4.1	Процедуры и функции	1	12	2		4		4		1/17	
4.2	Модули		13	2				2		1/50	
4.3	Файлы. Файловые типы. Обработка компонентов файла		14-15	4		4		4		1/13	
5	Приемы программирования										
5.1	Динамические типы данных. Статическая и динамическая память. Указатели. Программирование с использованием динамической памяти	1	16-17	4		4		6		1/13	
5.2	Задачи сортировки и поиска. Методы сортировки		18	2		4		10		1/17	Рейтинг-контроль №3
Итого за 1 семестр				36		36		72		15/21	Экзамен
6	Архитектура и функционирование вычислительной машины										
6.1	Представление информации. Программное управление.	2	1	2				2		1/50	
6.2	Уровни рассмотрения машины. Структура вычислительной машины.		2	2	2			4		1/25	
6.3	Архитектура учебной ЦВМ. Представление данных. Система команд		3	2		4		2		2/33	
6.4	Программирование для УЦВМ.		4-5	4	2	4		4		2/20	
6.5	Архитектура IBM PC.		6	2	2			6		1/25	Рейтинг-контроль №1

1	2	3	4	5	6	7	8	9	10	11	12
6.6	Регистры процессора. Система прерываний.		7	2				2		1/50	
6.7	Сегменты.		8	2	2			4		1/25	
7	Программирование на языке ассемблера										
7.1	Трансляция и компоновка. Инициализация данных. Режимы адресации данных.	2	9	2				2		1/50	
7.2	Команды языка ассемблера		10-13	8	4	4		20		2/13	Рейтинг-контроль №2
7.3	Подпрограммы.		14-15	4	2	4		10		1/10	
7.4	Приемы программирования.		16	2	2			6		1/50	
7.5	Использование средств языка ассемблера в языках высокого уровня.		17-18	4	2	2		10		1/13	Рейтинг-контроль №3
Итого за 2 семестр				36	18	18		72	КП	15/21	Зачет
Всего				72	18	54		144	КП	30/21	Экзамен, зачет

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При освоении дисциплины используются следующие сочетания видов учебной работы с методами и формами активизации познавательной деятельности бакалавров для достижения запланированных результатов обучения и формирования компетенций:

- изучение теоретического материала дисциплины на лекциях с использованием компьютерных технологий;

- самостоятельное изучение теоретического материала дисциплины с использованием Интернет-ресурсов, информационных баз, методических разработок, специальной учебной и научной литературы;

- закрепление теоретического материала при проведении лабораторных работ и практических занятий с использованием современной вычислительной техники и систем программирования, выполнения проблемно-ориентированных, поисковых, творческих заданий;

- самостоятельная работа студента, направленная на углубление и закрепление знаний, развитие практических умений, комплекса универсальных (общекультурных) и профессиональных компетенций, повышение творческого потенциала бакалавров, заключается в работе бакалавров с лекционным материалом, поиске и анализе литературы и электронных источников информации по заданной проблеме, переводе материалов из тематических информационных ресурсов с иностранных языков, изучении тем, вынесенных на самостоятельную проработку, подготовке докладов и презентаций по результатам выполненной работы, изучении теоретического материала к лабораторным занятиям, подготовке к экзамену.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

1 Семестр.

6.1. Оценочные средства для текущего контроля успеваемости

Список вопросов на рейтинг-контроле:

Вопросы рейтинга-контроля №1

1. В чем состоит назначение компилятора.
2. Приведите классификацию языков программирования по назначению.
3. Опишите отличительные черты структурного и объектно-ориентированного подходов.
4. Перечислите этапы создания программного обеспечения.
5. Дайте понятие алгоритма.
6. Назовите базовые структуры алгоритма.
7. Назовите дополнительные структуры алгоритма.
8. Переменные и константы. Приведите их определения и отличия, назначение.
9. Дайте понятие идентификатора. Для чего он нужен? Как соотносятся переменная и идентификатор?
10. Приведите правила представления (записи) идентификаторов.
11. Что определяет тип переменной?
12. Что такое явное и неявное преобразование типов. Приведите примеры.
13. Выражение. Приведите виды выражений. Опишите приоритеты операций.
14. Как организуются процедуры ввода-вывода? Каковы особенности их использования? Что такое «форматирование вывода на экран»?
15. Опишите оператор присваивания. Что такое «совместимость типов»? Приведите примеры.
16. Опишите структуру программы на языке C++.
17. В каких случаях компилятор выдает сообщение Unknown identifier?
18. В каких случаях компилятор выдает сообщение Type mismatch?
19. Представьте алгоритм решения задачи в виде словесного описания и схемы алгоритма: выведите на экран таблицу умножения, форматированную в виде треугольника.
20. Представьте алгоритм решения задачи в виде словесного описания и схемы алгоритма: выведите на экран первые 20 чисел последовательности Фибоначчи – в 5 рядов друг под другом и по 4 числа в ряду.

Задача:

Составить алгоритм решения задачи и программу на языке C. Требования:

- проверка физического смысла вводимых данных;
- вывод вводимых исходных данных, если они принадлежат области возможных значений, в противном случае вывод сообщения и завершение работы программы.

- вывод результатов работы программы.
- использование стандартных операторов ввода/ вывода.

Вариант 1

Ввести три числа a, b, c , определить могут ли они являться сторонами треугольника. Если треугольник – прямоугольный, то рассчитать площадь, иначе – рассчитать периметр.

Вариант 2

Ввести три числа x, y, R (x, y – координаты точки на плоскости, R – радиус окружности с центром в точке $(0,0)$). Если точка лежит на или внутри окружности, то рассчитать площадь круга, если вне окружности, то длину окружности.

Вариант 3

Ввести три числа v_0 – скорость в нулевой момент времени, a – ускорение, t – время. Определить скорость тела при равноускоренном движении или перемещение, если движение равнозамедленное.

($\mathbf{v} = \mathbf{v}_0 + \mathbf{a}t$, $\mathbf{S} = \mathbf{v}_0 t + \mathbf{a}t^2/2$, величины \mathbf{v} , \mathbf{v}_0 , \mathbf{a} , \mathbf{S} являются векторными)

Вариант 4

Ввести координаты двух точек x_1, y_1 и x_2, y_2 . Если обе точки принадлежат I или III четверти плоскости, то $z = x_1 + x_2$, иначе $z = (x_1 + x_2)(y_1 + y_2)$.

Вариант 5

Ввести координаты точки x_0, y_0 . Если точка лежит на какой-либо оси, то $z = 0$, иначе $z = x_0^2 y_0^2$.

Вариант 6.

Прямая задана уравнением $y = kx + b$. Ввести коэффициенты прямой k и b , координаты точки x_0, y_0 . Если точка лежит на прямой и функция возрастающая, то $z = x_0 + y_0$, если точка лежит на прямой и функция убывает, то $z = x_0 - y_0$, иначе $z = 0$.

Вопросы рейтинга-контроля №2

1. Какие стандартные управляющие структуры используются в структурном программировании для реализации разветвляющихся алгоритмов?
2. Какие управляющие структуры используются в языке C++ для организации разветвляющихся алгоритмов?
3. Приведите синтаксис оператора условной передачи управления и поясните порядок его выполнения (включая его сокращенную форму).
4. Приведите синтаксис оператора выбора и поясните порядок его выполнения.
5. Дайте определение составного оператора. Каким образом он используется в программе?
6. Как разрешается неоднозначность, которая может возникнуть при использовании вложенных операторов if ?
7. Перечислите виды циклов и приведите их схемы.
8. Приведите функциональную схему счетного цикла и назначение ее отдельных частей.
9. Какие управляющие структуры используются в языке C++ для организации цикла с предусловием?

10. Какие управляющие структуры используются в языке C++ для организации цикла с постусловием?
11. Укажите, сколько операторов можно разместить в теле цикла?
12. Можно ли в теле цикла изменять значение параметра цикла? Ответ пояснить.
13. Можно ли вне тела цикла использовать значение параметра цикла? Ответ пояснить.
14. Дайте понятие массива. Приведите примеры объявления массива в программе всеми известными способами.
15. Опишите принципы работы с одномерными массивами.
16. Опишите принципы работы с многомерными массивами.
17. Дайте определение строкового типа данных. Приведите примеры объявления переменных строкового типа и их инициализацию.
18. Перечислите процедуры и функции для работы со строковым типом данных.
19. Дайте определение символьного массива. В чем состоит отличие символьного массива от переменной строкового типа?
20. Опишите особенности обработки символьного массива.

Задача

Составить алгоритм программы. Написать код на языке C++, реализующие алгоритм с использованием счетного цикла и цикл с предусловием (для нечетных вариантов) и цикл с постусловием (для четных вариантов).

Вариант 1

Сгенерировать матрицу, содержащую целые числа от 0 до 100 размером NхM. Найти сумму и количество чисел, больших 50.

Вариант 2

Сгенерировать матрицу, содержащую целые числа от -100 до 100 размером NхM. Найти сумму и количество чисел отрицательных.

Вариант 3

Сгенерировать матрицу, содержащую целые числа от 0 до 100 размером NхM. Найти произведение и количество чисел, больших 50.

Вариант 4

Сгенерировать матрицу, содержащую целые числа от -10 до 10 размером NхM. Найти произведение и количество чисел отрицательных.

Вариант 5

Сгенерировать матрицу, содержащую целые числа от -100 до 100 размером NхM. Найти номера строк матрицы, в которых встречаются элементы, равные 0 и число нулевых элементов матрицы.

Вариант 6

Сгенерировать матрицу, содержащую целые числа от -100 до 100 размером NхM. Найти номера столбцов матрицы, хотя бы один элемент которых равен 0 и число таких столбцов..

Вопросы рейтинга-контроля №3

1. Каково назначение подпрограмм? Дайте определение подпрограммы.

2. Назовите виды подпрограмм, их сходства и отличия.
3. Каким образом объявляются процедуры и функции. Что такое прототип функции?
4. Как выглядит вызов (обращение к) процедур и функций в программе?
5. Что такое область видимости переменных? Какова область видимости глобальных и локальных переменных? Приведите поясняющие примеры.
6. В чем состоит назначение локальных переменных?
7. В чем заключается суть явной передачи данных в подпрограмму? Какие возможны варианты явной передачи данных?
8. В чем заключается суть неявной передачи данных в подпрограмму? Какие возможны варианты неявной передачи данных?
9. Что такое список параметров процедуры/функции? Дайте определения формальных и фактических параметров подпрограммы. Приведите примеры.
10. Какие существуют способы передачи параметров в подпрограмму?
11. Опишите механизм передачи параметров по значению.
12. Опишите механизм передачи параметров по ссылке.
13. Каковы достоинства и недостатки вложенных подпрограмм? Какой механизм организации позволяет использовать вложенные подпрограммы?
14. Что такое файловая переменная? Какова ее структура и какую информацию она содержит?
15. Опишите принципы программной обработки файла (основные действия)
16. Перечислите процедуры и функции для работы с текстовыми файлами.
17. Перечислите процедуры и функции для работы с типизированными файлами.
18. Опишите принципы работы с файлами записей.
19. В чем заключаются задачи сортировки и поиска? Какие вы знаете алгоритмы сортировки?
20. Дайте характеристики статической и динамической памяти.
21. Что такое указатель? Для чего он нужен? Приведите примеры.
22. Перечислите основные операции при работе с динамической памятью. Приведите примеры.

Задачи

Составьте алгоритм решения и код на языке C++ (обязательно использование подпрограмм).

Вариант 1

Дана матрица размера $N \times N$. Найти сумму элементов строк матрицы и записать их в отдельном массиве. Подпрограммой оформить суммирование элементов отдельной строки: <подпрограмма> SumStr (<номер строки матрицы>, <размерность>, ...).

Вариант 2

Дана матрица размера $N \times N$, число M . Определить количество элементов со значением M в каждой строке матрицы, результат – в отдельном массиве. Подпрограммой оформить подсчет элементов со значением M в отдельной строке:

<подпрограмма> NumStr (<строка матрицы>, <размерность>, <значение M >, ...).

Вариант 3

Дана матрица размера $N \times N$. Найти число положительных элементов строк матрицы и записать их в отдельном массиве. Подпрограммой оформить подсчет положительных элементов в отдельной строке: <подпрограмма> NPositiveStr (<номер строки матрицы>, <размерность>, ...).

Вариант 4

Дана матрица размера $N \times N$. Заменить все отрицательные элементы матрицы 0 и найти число замен в каждой строке, записать их в отдельном массиве. Подпрограммой оформить подсчет 0-х элементов отдельной строки: <подпрограмма> NNulStr (<номер строки матрицы>, <размерность>, ...).

Вариант 5

Дана матрица размера $N \times N$. Найти произведение четных элементов каждой строки, записать их в отдельном массиве. Подпрограммой оформить подсчет произведения четных элементов отдельной строки: <подпрограмма> MulEvenStr (<номер строки матрицы>, <размерность>, ...).

Вариант 6

Дана матрица размера $N \times N$. Найти сумму нечетных элементов каждой строки, записать их в отдельном массиве. Подпрограммой оформить подсчет суммы нечетных элементов отдельной строки: <подпрограмма> AddOddStr (<номер строки матрицы>, <размерность>, ...).

6.2. Оценочные средства для промежуточной аттестации студентов по итогам освоения дисциплины

Вопросы к экзамену

1. Понятие алгоритма. Требования к алгоритму. Свойства алгоритма.
2. Структуры алгоритмов. Способы записи алгоритмов.
3. Этапы решения задач на ЭВМ.
4. Принципы структурного программирования.
5. Типы данных и константы языка C++.
6. Типизированные указатели.
7. Ссылочный тип.
8. Перечислимый тип.
9. Структура программы на языке C++.
10. Процедуры ввода-вывода. Поточный ввод-вывод.
11. Оператор присваивания. Основные арифметические операции C++.
12. Выражения. Основные операции языка C++ .
13. Условный оператор. Оператор выбора.
14. Операторы безусловного перехода.
15. Циклические структуры языка C++. Виды циклов.
16. Циклические программы. Вложенные циклы.
17. Глобальные и локальные переменные.
18. Функции. Механизм передачи параметров.
19. Вложенные функции. Рекурсия.
20. Область видимости и время жизни переменной.
21. Фактические и формальные параметры.
22. Перегрузка функций.
23. Шаблоны функций.
24. Понятие модуля. Преимущества модульного программирования. Структура модуля.
25. Пример модуля. Способ использования модуля.
26. Массивы в языке C++. Связь указателей и массивов в C++.

27. Символьный тип данных. Строковые массивы. Способы обработки.
28. Основные операции над строками. Функции для работы со строками.
29. Структуры. Работа со структурами. Примеры.
30. Массивы структур. Особенности обработки. Примеры.
31. Файлы. Виды файлов. Файловая переменная. Общая схема работы с файлами.
32. Файлы. Виды файлов. Общие для всех файлов функции обработки.
33. Текстовые файлы. Функции обработки для текстовых файлов.
34. Бинарные файлы. Функции для работы с бинарными файлами.
35. Статическая и динамическая память.
36. Работа с динамической памятью. Примеры.
37. Алгоритмы и методы сортировки: оценка эффективности алгоритма.
38. Сортировка выбором.
39. Сортировка включением.
40. Сортировка обменом.
41. Шейкер-сортировка.
42. Быстрая сортировка Хоара.
43. Алгоритмы и методы поиска в отсортированном массиве данных.

6.3 Задания для самостоятельной работы студентов:

Раздел 1.

Составить ответы на следующие вопросы

1. Какие существуют способы записи алгоритма?
2. Метод нисходящего проектирования, в чем его особенности?

Задача 1

Составить словесный алгоритм

Вариант 1 Нахождение корней линейного уравнения $kx+b=0$.

Вариант 2 Нахождение корней квадратного уравнения $ax^2 + c=0$.

Вариант 3 Нахождение корней кубического уравнения $ax^3 + c=0$.

Вариант 4 Определить, являются ли числа a , b , c сторонами остроугольного треугольника.

Вариант 5 Определить, являются ли числа a , b , c сторонами прямоугольного треугольника.

Вариант 6 Определить, являются ли числа a , b , c сторонами тупоугольного треугольника.

Раздел 2.

Составить ответы на следующие вопросы

- 1) Какие существуют способы записи алгоритма?
- 2) Метод нисходящего проектирования, в чем его особенности?
- 3) Линейные операторы языка C++.
- 4) Простые типы данных языка C++.
- 5) Структура программы на языке C++.
- 6) Стандартные библиотеки и их подключение.
- 7) Что такое идентификатор, переменная, константа?

- 8) Что такое совместимость типов?
- 9) Явное и неявное преобразование типов.
- 10) Какие управляющие структуры используются в языке C++ для организации разветвляющихся алгоритмов?
- 11) Поясните порядок выполнения оператора *if...else* и его сокращенной формы?
- 12) Поясните порядок выполнения оператора *switch*. Каково здесь назначение оператора *break*?
- 13) Сколько операторов можно написать после ключевых слов *if* и *else*?
- 14) Что такое составной оператор?
- 15) Как разрешается неоднозначность, которая может возникнуть при использовании вложенных операторов *if*?
- 16) Функциональная схема цикла и назначение ее отдельных частей.
- 17) Какие виды циклов существуют, в чем их принципиальные отличия?
- 18) Назначение цикла *for* и его отдельных компонентов.
- 19) Укажите, сколько операторов можно разместить в теле цикла?
- 20) Можно ли в теле цикла изменять значение параметра цикла?
- 21) Можно ли вне тела цикла использовать значение параметра цикла?
- 22) Как работают операторы *while* и *do-while*?
- 23) Какими способами можно инициализировать массив?
- 24) Какими способами можно обращаться к элементам массива?
- 25) Какова общая схема работы с массивом?

Задача 1.

Составить программу, формирующую таблицу умножения, и выводящую результат в виде верхнетреугольной матрицы. Использовать вложенные счетные циклы.

Задача 2.

Дан массив из N действительных чисел. Составить структурный алгоритм для нахождения :

- 1) Первого отрицательного элемента массива. Вывести элемент и его номер в массиве.
- 2) Последнего отрицательного элемента массива. Вывести элемент и его номер в массиве.
- 3) Элемента массива, который является квадратом целого числа. Вывести элемент и его номер в массиве.
- 4) Элемента массива, который является четным целым числом. Вывести элемент и его номер в массиве.
- 5) Сумму элементов с четными номерами. Вывести сумму.
- 6) Числа положительных, отрицательных и нулевых членов массива.

Раздел 3.

Составить ответы на следующие вопросы

- 1) Какова общая схема работы с многомерным массивом?
- 2) Каковы правила организации вложенных циклов?
- 3) Какие существуют способы объявления массива?
- 4) Назовите алгоритм работы и правила использования неструктурного оператора *break*.
- 5) Какие существуют способы организации символьных данных?
- 6) Объявление и инициализация символьных массивов.
- 7) В чем отличие символьной и строковой константы?
- 8) Как хранится строковое значение в памяти ЭВМ?
- 9) Общие принципы работы с символьными данными.

- 10) Библиотеки и функции для работы со строками.
- 11) Какие возможны операции над строками?

Задача 1.

Разработать алгоритм для кодирования двоичной последовательности методом Лемпела-Зива. Исходную последовательность обрабатывать в виде строки. При реализации программы использовать библиотечные функции работы с символами и строками. Результатом работы должно быть: 1) длина кодового слова в словаре фраз, 2) кодовые слова, представленные массивом строк, каждая из которых является кодом фразы словаря.

Задача 2.

(Обратная задаче 1). Разработать алгоритм для декодирования данных, закодированных по методу Лемпела-Зива. Исходные данные 1) длина кодового слова в словаре фраз, 2) количество кодовых слов, 3) последовательность кодовых слов, вводимых с клавиатуры (при тестировании использовать результаты работы задачи 1). В программе предусмотреть вариант некорректного ввода кодового слова с выдачей информационного сообщения. Результат представить в виде одной строки.

Раздел 4.

Составить ответы на следующие вопросы

- 1) Что такое подпрограмма? Каково ее назначение?
- 2) Назовите основные принципы модульности и как они реализованы в C++.
- 3) Что такое область видимости и время жизни объекта программы?
- 4) Глобальные и локальные переменные.
- 5) Формальные и фактические параметры.
- 6) Назовите основные принципы структурного программирования.
- 7) Как происходит передача параметров в подпрограмму, какие аппаратные и программные средства для этого используются?
- 8) В чем принципиальное отличие передачи параметра по значению и по ссылке, в каких случаях используется одно и другое?
- 9) Что такое синоним переменной, чем он отличается от указателя?
- 10) Для чего нужно предварительное объявление функции?
- 11) Что такое файл? Общая схема работы с файлом.
- 12) Что содержит файловая переменная?
- 13) Как работают функции *fopen* и *fclose*?
- 14) Что такое структура? Как обращаться к полям структуры?
- 15) Как выглядит объявление структуры? Как в памяти хранится переменная структурного типа?
- 16) В чем отличие обработки текстового и бинарного файлов?

Задача 1.

Разработать программу кодирования и декодирования данных по методу Лемпела-Зива. Кодлируемая/декодированная последовательность и кодовые слова содержатся в :

- 1) текстовых файлах;
- 2) двоичных файлах.

Задача 2.

Реализовать алгоритм кодирования/ декодирования по методу Лемпела-Зива с использованием функций. В виде функций реализовать следующие алгоритмы: 1) поиск фразы-прототипа для новой фразы, результат – новая фраза, код фразы, 2) формирование фразы при декодировании, результат – новая фраза.

Раздел 5.

Составить ответы на следующие вопросы

- 1) Какие виды памяти Вы знаете? Каково их назначение?
- 2) Что такое динамическая память? В чем заключается ее особенность? Правила работы с динамической памятью.
- 3) Как работают операторы *new* и *delete*?
- 4) Что такое шаблон функции? Для чего он используется?
- 5) На каком механизме базируется реализация шаблонов функций?
- 6) Что такое параметризация шаблона? Чем он параметризируется?
- 7) В чем заключается параметризация шаблона?
- 8) Что такое «метод дихотомии»? Для чего он используется?
- 9) Как оценить эффективность того или иного метода сортировки?
- 10) Для чего используется сортировка? В каких случаях затраты на сортировку оправданы?
- 11) Как осуществляется добавление новых данных в отсортированный массив?

Задача 1. Модифицировать программы кодирования/ декодирования данных по методу Лемпела-Зива с использованием динамической памяти для словаря фраз.

2 Семестр.

6.4. Оценочные средства для текущего контроля успеваемости

Список вопросов на рейтинг-контроле:

Вопросы рейтинга-контроля №1

1. Опишите алгоритм работы УЦВМ.
2. Перечислите характерные черты языка Ассемблер.
3. Как будут размещены элементы 5-мерного массива в памяти УЦВМ?
4. Что такое структура?
5. Перечислите структурные элементы учебной ЦВМ.
6. Назовите режимы адресации в УЦВМ.
7. В каком коде представлены числа в УЦВМ?
8. Значение какого регистра используется при индексной адресации?
9. Перечислите команды, изменяющие естественный порядок выполнения программы.
10. Сколько принципов программного управления Вы знаете?
11. Назовите минимальную единицу информации.
12. Назовите минимальную адресуемую единицу информации.
13. Сколько форматов команд существует в УЦВМ?
14. Сколько программно доступных регистров существует в УЦВМ?

Вопросы рейтинга-контроля №2

1. Перечислите сходные черты учебной ЦВМ и реального процессора.
2. Перечислите различия учебной ЦВМ и реального процессора.
3. Опишите принципы организации связи по управлению в УЦВМ.
4. Опишите принципы организации связи по данным в УЦВМ.

5. Какая модель памяти должна использоваться для программ формата COM?
6. Перечислите сегментные регистры.
7. Назовите режимы адресации реального процессора.
8. Как формируется 20-разрядный адрес в памяти при использовании базовой индексной адресации?
9. Перечислите директивы для объявления типов данных.
10. Напишите объявление 2-мерного массива двойных слов. Размер массива: 7 строк, 4 столбца.

Вопросы рейтинга-контроля №3

1. Напишите фрагмент кода, с помощью которого регистры AX и DX обменяются значениями без привлечения дополнительных регистров и ячеек памяти.
2. Можно ли вызвать подпрограмму без использования команды CALL? Ответ пояснить.
3. Какие процессы происходят при вызове подпрограммы командой CALL?
4. Какие регистры общего назначения предназначены для работы со стекком?
5. Чем определяется формат операндов при использовании команды деления?
6. Напишите команду условного перехода для условия
 - 1) если выше или равно
 - 2) если больше или равно
 - 3) если ниже или равно
 - 4) если меньше или равно
 - 5) если было переполнение
 - 6) если не меньше
 - 7) если был перенос
7. Какой регистр используется при организации циклических вычислений с помощью команд LOOP*?
8. Какие действия выполняет процедура пролога?
9. Перечислите основные соглашения при написании раноязыковых программ.
10. Напишите программу на языке ассемблера, которая определяет сумму элементов с нечетными индексами массива из 10 слов, а результат выводит на экран.

6.5. Оценочные средства для промежуточной аттестации студентов по итогам освоения дисциплины

Вопросы к зачету

1. Алгоритм работы УЦВМ.
2. Организация массива в памяти УЦВМ?
3. Структурные элементы учебной ЦВМ.
4. Режимы адресации в УЦВМ.
5. Способы представления информации в УЦВМ?
6. Принципы индексной адресации.
7. Естественный порядок следования команд и передача управления.
8. Форматы команд УЦВМ.
9. Архитектура УЦВМ.
10. Структура и алгоритм работы реального процессора.
11. Организация подпрограмм в УЦВМ: связь по данным, связь по управлению.
12. Модели памяти.
13. Сегментные регистры.

14. Регистры общего назначения.
15. Режимы адресации реального процессора.
16. Директивы для объявления типов данных.
17. Организация массива.
18. Система команд процессора x86.
19. Способы передачи управления подпрограмме.
20. Работа со стеком.
21. Команды условного перехода.
22. Организация циклических вычислений.
23. Процедуры пролога и эпилога.
24. Основные соглашения при написании разноязыковых программ.

6.6. Задания для курсового проекта:

- 1) получить индивидуальное задание у преподавателя (вычисляемая функция);
- 2) разработать программу на одном из языков высокого уровня (С, С++, Паскаль), реализующую вычисление заданной функции, оформив код в виде **процедуры**;
- 3) разработать процедуру вычисления функции на языке Ассемблер, используя встроенные средства выбранного языка высокого уровня;
- 4) разработать отдельную процедуру вычисления функции на языке Ассемблер и подключить её к проекту на языке высокого уровня;
- 5) реализовать подпрограммы (процедуры или функции) ввода и вывода целых чисел на языке Ассемблер и подключить их к проекту на языке высокого уровня;
- 6) выполнить анализ реализованных процедур вычисления заданной функции по быстрдействию.

Варианты индивидуальных заданий

1. $(2*c - d + 23) / (a/4 - 1)$
2. $(c + 4*d - 123) / (1 - a/2)$
3. $(-2*c + d*82) / (a/4 - 1)$
4. $(2*c + d - 52) / (a/4 + 1)$
5. $(c/4 - d*62) / (a*a + 1)$
6. $(-2*c - d + 53) / (a/4 - 1)$
7. $(2*c - d/4) / (a*a + 1)$
8. $(2 + c - d*23) / (2*a*a - 1)$
9. $(2*c - d/3) / (b - a/4)$
10. $(4*c + d - 1) / (c - a/2)$
11. $(2*c - d*42) / (c + a - 1)$
12. $(25/c - d + 2) / (b + a*a - 1)$
13. $(c - d/2 + 33) / (2*a*a - 1)$
14. $(4*c - d/2 + 23) / (a*a - 1)$
15. $(c*d + 23) / (a/2 - 4*d - 1)$
16. $(c/d + 3*a/2) / (c - a + 1)$
17. $(2*c + d*51) / (d - a - 1)$
18. $(2*c + d/4 + 23) / (a*a - 1)$
19. $(2*c - d/2 + 1) / (a*a + 7)$
20. $(2*c/d + 2) / (d - a*a - 1)$
21. $(12/c - d*4 + 73) / (a*a + 1)$
22. $(2*c/a - d*d) / (d + a - 1)$
23. $(-53/a + d - 4*a) / (1 + a*b)$
24. $(-15*a + b - a/4) / (b*a - 1)$
25. $(-25/a + c - b*a) / (1 + c*b/2)$

26. $(4*a - 1 + b/2) / (b*c - 5)$
27. $(8*b + 1 - c) / (a/2 + b*c)$
28. $(4*a - b - 1) / (c/b + a)$
29. $(4*b/c - 1) / (12*c + a - b)$
30. $(b + c*b - a/4) / (a*b - 1)$
31. $(a + c/b - 28) / (4*b*a + 1)$
32. $(c/b - 24 + a) / (2*a*c - 1)$
33. $(2*b - a + b*c) / (c/4 - 1)$
34. $(41 - d/4 - 1) / (c/b + a*d)$
35. $(a - b*4 - 1) / (c/31 + a*b)$
36. $(b/a + 4*c) / (c - b + 1)$

6.7 Задания для самостоятельной работы студентов:

Раздел 6

1. Для каждой из предложенных задач записать алгоритм решения задачи на C++; схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; распределение памяти и текст программы на машинном языке УЦВМ (Запись алгоритма на C++ и блок-схема должны быть согласованы между собой и составлены таким образом, чтобы машинная программа УЦВМ была их реализацией на низком уровне)
 - а) Даны три переменные. Определить наибольшее значение.
 - б) Даны три переменные. Результату присвоить значение 1, если переменные упорядочены по возрастанию; иначе - 0.
 - в) Определить n-й член ряда Фибоначчи: $F[1]=1, F[2]=1, F_n=F[n-1]+F[n-2]$.
 - г) Вычислить $y = 1 + x + 2*x^2 + 3*x^3 + 4*x^4 + 5*x^5$.
 - д) Известно, что две переменные имеют одинаковые значения, а третья - отличное от них. Найти ее значение.
 - е) Составить программу перемножения двух чисел без применения команды умножения.
 - ж) Составить программу целочисленного деления двух чисел без применения команды деления. Если деление невозможно, то переменной, называемой индикатором ошибки, присвоить значение 1, иначе - 0.
 - з) Составить программу для нахождения наибольшего общего делителя двух чисел.
 - и) Вычислить факториал числа n, где $n \geq 1$.
 - к) Вычислить n-ю степень числа, где $n \geq 1$.
 - л) Определить максимальное число, факториал которого может быть представлен в машине.
 - м) Вычислить $y = \min(x^2, \max(y, 10))$.
 - н) Вычислить число размещений из m по n по формуле $A(n, m) = m(m-1)(m-2), \dots, (m-(n-1))$, где $m \geq n$.
 - о) Вычислить $y = \max(x-1, y^2, z \text{ div } 3)$.
 - п) Даны три переменные с неравными значениями, рассматриваемые как координаты точек на числовой оси. Определить длину минимального отрезка, включающего все три точки.
2. Составить на языке Ассемблер учебной ЦВМ программу решения каждой задачи из задания 1, провести ее ассемблирование, изучить полученную при этом программу на машинном языке.
3. Для каждой из предложенных задач записать алгоритм решения задачи на C++ (с использованием массивов); схему алгоритма (блок-схему) решения задачи, согласованную с

записью на C++; текст программы на языке Ассемблер УЦВМ (Запись алгоритма на C++ и блок-схема должны быть согласованы между собой и составлены таким образом, чтобы программа для УЦВМ была их реализацией на низком уровне)

- а) Найти значение минимального элемента массива.
- б) Найти индекс максимального элемента массива.
- в) Найти скалярное произведение двух векторов.
- г) Найти сумму элементов главной диагонали квадратной матрицы.
- д) Найти сумму положительных элементов массива.
- е) Найти индекс элемента с заданным значением. Предполагается, что в массиве есть такой элемент и он единственный.
- ж) Определить количество положительных и отрицательных элементов массива.
- з) Определить количество элементов массива, которым предшествуют элементы с меньшими значениями.
- и) Каждому элементу массива, начиная со второго, присвоить значение максимального элемента из числа ему предшествующих и его самого.
- к) Дана прямоугольная матрица. Найти сумму элементов строки с заданным номером.
- л) Определить, какие два последовательных элемента массива наименее отличаются друг от друга. Найти индекс первого элемента пары.
- м) Построить массив, элементы которого суть суммы последовательных пар элементов исходного массива.
- н) Определить количество элементов массива, значения которых превышают заданное.
- о) Массив, элементы которого принадлежат множеству $\{0,1\}$, рассматривается как представление целого числа. Определить значение числа, заданного таким способом.
- п) Дан массив, элементы которого принадлежат множеству $\{0,1\}$. Определить длину первой последовательности рядом стоящих единиц.

4. Для каждой из предложенных в задании 1 задач записать алгоритм решения задачи на C++ с использованием подпрограмм; схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; текст программы на языке Ассемблер УЦВМ (Запись алгоритма на C++ и блок-схема должны быть согласованы между собой и составлены таким образом, чтобы программа для УЦВМ была их реализацией на низком уровне)

Раздел 7.

1. Для каждой из предложенных задач записать алгоритм решения задачи на C++; схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; текст программы на языке ассемблер (реальной ЭВМ) (Запись алгоритма на C++ и блок-схема должны быть согласованы между собой и составлены таким образом, чтобы программа на языке ассемблер)

- а) Даны три переменные. Определить наибольшее значение.
- б) Даны три переменные. Результату присвоить значение 1, если переменные упорядочены по возрастанию; иначе - 0.
- в) Определить n-й член ряда Фибоначчи: $F[1]=1, F[2]=1, F_n=F[n-1]+F[n-2]$.
- г) Вычислить $y = 1 + x + 2*x^2 + 3*x^3 + 4*x^4 + 5*x^5$.
- д) Известно, что две переменные имеют одинаковые значения, а третья - отличное от них. Найти ее значение.
- е) Составить программу перемножения двух чисел без применения команды умножения.
- ж) Составить программу целочисленного деления двух чисел без применения команды деления. Если деление невозможно, то переменной, называемой индикатором ошибки, присвоить значение 1, иначе - 0.
- з) Составить программу для нахождения наибольшего общего делителя двух чисел.

- и) Вычислить факториал числа n , где $n \geq 1$.
- к) Вычислить n -ю степень числа, где $n \geq 1$.
- л) Определить максимальное число, факториал которого может быть представлен в машине.
- м) Вычислить $y = \min(x^2, \max(y, 10))$.
- н) Вычислить число размещений из m по n по формуле $A(n, m) = m(m-1)(m-2)\dots(m-(n-1))$, где $m \geq n$.
- о) Вычислить $y = \max(x-1, y^2, z \text{ div } 3)$.
- п) Даны три переменные с неравными значениями, рассматриваемые как координаты точек на числовой оси. Определить длину минимального отрезка, включающего все три точки.

2. Для каждой из предложенных задач записать алгоритм решения задачи на C++ (с использованием массивов); схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; текст программы на языке ассемблер (реальной ЭВМ) (Запись алгоритма на C++ и блок-схема должна быть согласованы между собой и составлены таким образом, чтобы программа для УЦВМ была их реализацией на низком уровне)

- а) Найти значение минимального элемента массива.
- б) Найти индекс максимального элемента массива.
- в) Найти скалярное произведение двух векторов.
- г) Найти сумму элементов главной диагонали квадратной матрицы.
- д) Найти сумму положительных элементов массива.
- е) Найти индекс элемента с заданным значением. Предполагается, что в массиве есть такой элемент и он единственный.
- ж) Определить количество положительных и отрицательных элементов массива.
- з) Определить количество элементов массива, которым предшествуют элементы с меньшими значениями.
- и) Каждому элементу массива, начиная со второго, присвоить значение максимального элемента из числа ему предшествующих и его самого.
- к) Дана прямоугольная матрица. Найти сумму элементов строки с заданным номером.
- л) Определить, какие два последовательных элемента массива наименее отличаются друг от друга. Найти индекс первого элемента пары.
- м) Построить массив, элементы которого суть суммы последовательных пар элементов исходного массива.
- н) Определить количество элементов массива, значения которых превышают заданное.
- о) Массив, элементы которого принадлежат множеству $\{0, 1\}$, рассматривается как представление целого числа. Определить значение числа, заданного таким способом.
- п) Дан массив, элементы которого принадлежат множеству $\{0, 1\}$. Определить длину первой последовательности рядом стоящих единиц.

3. Для каждой из предложенных в задании 1 задач записать алгоритм решения задачи на C++ (с использованием подпрограмм); схему алгоритма (блок-схему) решения задачи, согласованную с записью на C++; текст программы на языке ассемблер (реальной ЭВМ) (Запись алгоритма на C++ и блок-схема должна быть согласованы между собой и составлены таким образом, чтобы программа для УЦВМ была их реализацией на низком уровне)

4. Составить текст программы на языке ассемблер (реальной ЭВМ) для следующих задач:
- а) Объединить два файла с записью их содержимого в третий (новый) файл.
 - б) Подсчитать побайтовую контрольную сумму файла (4 байта, переполнение результата игнорируется). Записать полученное значение в новый файл.

- в) Сгенерировать бинарный файл, компонентами которого являются значения функции $n!$; n меняется от 0 до 10 с шагом 1. Для представления числа использовать 2 байта (слово).
- г) Даны два файла одинакового размера. Получить третий файл, содержащий разности между компонентами первого и второго файлов. Размер компонентов - 1 байт. Переполнение результата игнорируется.
- д) Дан текстовый файл. Получить новый файл, удалив все символы с кодом 10 из исходного файла.
- е) Скопировать файл, используя буфер размером 1 килобайт. На размер файла ограничений не накладывается.
- ж) Дан текстовый файл. Подсчитать количество вхождений подстроки «abc» с записью результата (2 байта) в новый файл.
- з) Дан файл, размер которого не превышает одного килобайта. Разработать программу, производящую инвертирование компонент файла. Размер компонент - 1 байт. Результат записывается поверх существующей информации в том же файле.
- и) Разбить файл на две равные по размеру части, создав два новых файла. Для получения размера файла использовать подпрограмму `file_seek`.
- к) Разбить файл на две части, записав байты с нечетными номерами в один выходной файл, а с четными - в другой.
- л) Дан текстовый файл. Получить новый файл, заменив символы «1» в исходном файле на «2» и наоборот.
- м) Получить файл, отличающийся от исходного обратным порядком следования компонент. Размер компонент - 1 байт.
- н) Дан текстовый файл. Получить файл на основе исходного, приведя все символы, соответствующие латинским буквам, к верхнему регистру.
- о) Дан текстовый файл, представляющий собой программу на языке ассемблер. Получить файл на основе исходного, удалив все комментарии.
- п) Создать в каталоге 50 пустых файлов с именами «file_??.txt», где ?? - порядковый номер файла. Учесть, что по умолчанию приложения MS-DOS не могут одновременно работать более чем с 16 файлами.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1 Основная литература (Имеется в библиотеке ВлГУ)

1. Златопольский Д. М. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс] / Златопольский Д. М. - 3-е изд. (эл.). - М. : БИНОМ, 2015. - <http://www.studentlibrary.ru/book/ISBN9785996329328.html>
С.М. Окулов
2. Программирование в алгоритмах [Электронный ресурс] / С.М. Окулов. - 5-е изд. (эл.). - М. : БИНОМ, 2014. - (Развитие интеллекта школьников). - <http://www.studentlibrary.ru/book/ISBN9785996323111.html>
Н.А. Давыдова, Е.В. Боровская
3. Программирование [Электронный ресурс] / Н.А. Давыдова, Е.В. Боровская. - М. : БИНОМ, 2015. - <http://www.studentlibrary.ru/book/ISBN9785996326471.html>

7.2 Дополнительная литература (Имеется в библиотеке ВлГУ)

1. Андреева Е.В. Программирование -- это так просто, программирование -- это так сложно. Современный учебник программирования. [Электронный ресурс] / Андреева Е.В. - М.: МЦНМО, 2009. - <http://www.studentlibrary.ru/book/ISBN9785940575344.html>
2. Шень А. Программирование: теоремы и задачи [Электронный ресурс] / Шень А. - 4-е изд., стереотипы. - М.: МЦНМО, 2011. - <http://www.studentlibrary.ru/book/ISBN9785940576969.html>
3. Аблязов Р.З. Программирование на ассемблере на платформе x86-64 [Электронный ресурс] / Аблязов Р.З. - М. : ДМК Пресс, 2011. - <http://www.studentlibrary.ru/book/ISBN9785940746768.html>
4. В.Б. Иванов Прикладное программирование на C/C++: с нуля до мультимедийных и сетевых приложений [Электронный ресурс] / В.Б. Иванов - М. : СОЛОН-ПРЕСС, 2008. - <http://www.studentlibrary.ru/book/ISBN5980032797.html>
5. Введение в теорию языков программирования [Электронный ресурс] / Довек Жиль, Леви Жан-Жак - М. : ДМК Пресс, 2013. - <http://www.studentlibrary.ru/book/ISBN9785940749134.html>

7.3. Программное и коммуникационное обеспечение

Операционная система Windows, стандартные офисные программы MS Office, среда разработки Visual Studio, программный пакет TASM для разработки программ на языке ассемблера, Интернет-ресурсы.

7.4. Электронные средства обучения

Набор слайдов, методические указания к выполнению лабораторных и практических работ, к курсовому проекту, учебная цифровая вычислительная машина, контрольные тесты.

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Лабораторное оборудование


Лабораторные занятия проводятся в компьютерной лаборатории с использованием персональных компьютеров с установленной лицензионной средой разработки Visual Studio и программным пакетом TASM.

При проведении лабораторных работ используется мультимедиа проектор и интерактивная доска.

8.2. Средства вычислительной техники и демонстрационное оборудование


Лекции читаются в аудитории кафедры ВТ (404-2), оснащенной мультимедиа проектором. Практические и лабораторные занятия проводятся в аудиториях 411-2 или 401-2, все аудитории оснащены мультимедиа проекторами. При выполнении самостоятельной работы по освоению дисциплины студенты имеют возможность работать в компьютерном классе кафедры ВТ (401-2 или 406-2) с выходом в сеть Интернет, используя лицензионное прикладное и системное программное обеспечение, а также электронные методические материалы.

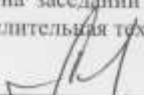
Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 09.03.01 – «Информатика и вычислительная техника»

Рабочую программу составил старший преп. кафедры ВТ  А.А. Сушинина
(ФИО, подпись)

Рецензент:

ООО "АСТЭК", инженер-программист отдела Встроенного ПО  А.В. Егорычева

Программа рассмотрена и одобрена на заседании кафедры Вычислительная техника
Протокол № 6 от 15.02.2016 года
Заведующий кафедрой  В.Н. Ланцов

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии
направления 09.03.01 – «Информатика и вычислительная техника»
Протокол № 1 от 15.02.16 года
Председатель комиссии  В.Н. Ланцов