

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета радиотехники и
электроники 
В.А.Небольсин
«31» августа 2021 г.

РАБОЧАЯ ПРОГРАММА
дисциплины
«Программирование на ЭВМ»

Направление подготовки 11.03.01 Радиотехника

Профиль Радиотехнические средства передачи, приема и обработки сигналов

Квалификация выпускника бакалавр

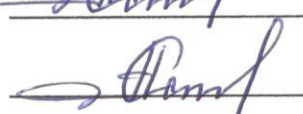
Нормативный период обучения 4года/4года и 11м.

Форма обучения очная/заочная

Год начала подготовки 2021

Автор программы  /Литвиненко Ю.В./

Заведующий кафедрой
Радиотехники  /Останков А.В./

Руководитель ОПОП  /Останков А.В./

Воронеж 2021

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

получение студентами знаний и практических навыков в области теории, методов, средств и технологий разработки программного обеспечения на языках высокого уровня

1.2. Задачи освоения дисциплины

- изучение основных этапов и принципов создания программного продукта, конструктивных компонентов и структуры компьютерных программ;

- изучение основ алгоритмизации и технологии программирования научно-технических задач;

- изучение языков программирования высокого уровня;

- практическое освоение интегрированной среды изучаемого алгоритмического языка высокого уровня.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программирование на ЭВМ» относится к дисциплинам части, формируемой участниками образовательных отношений блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Программирование на ЭВМ» направлен на формирование следующих компетенций:

ПК-4-Способен учитывать современные тенденции развития радиоэлектроники, измерительной и вычислительной техники в своей профессиональной деятельности

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-4	знать теоретические основы программирования на языке высокого уровня и технологию разработки программного обеспечения для решения задач в своей профессиональной деятельности
	уметь разрабатывать модели, алгоритмы и программное обеспечение на языке высокого уровня для решения задач в своей профессиональной деятельности
	владеть навыками работы в интегрированной

	программной среде изучаемого языка программирования.
--	--

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Программирование на ЭВМ» составляет 3з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		4
Аудиторные занятия (всего)	60	60
В том числе:		
Лекции	20	20
Лабораторные работы (ЛР)	40	40
Самостоятельная работа	48	48
Курсовая работа	+	+
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость академические часы з.е.	108 3	108 3

заочная форма обучения

Виды учебной работы	Всего часов	Семестры
		9
Аудиторные занятия (всего)	14	14
В том числе:		
Лекции	6	6
Лабораторные работы (ЛР)	8	8
Самостоятельная работа	90	90
Курсовая работа	+	+

Часы на контроль	4	4
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость академические часы з.е.	108 3	108 3

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Введение	История появления и развития языков программирования. Классификация языков программирования и их сравнительный анализ. Жизненный цикл программы. Интегрированные среды разработки, их сравнительный анализ.	4	-	8	12
2	Основы алгоритмизации	Понятие алгоритма. Свойства алгоритмов. Способы записи алгоритмов. Правила построения алгоритмов на языке блок-схем. Типы алгоритмов.	2	4	10	16
	Программирование на Си++	Начальные сведения о языке. Сферы применения языка Си++. Процесс написания программы. Синтаксис языка Си++. Операции и выражения. Операторы. Функции. Встроенные типы данных. Классы и объекты. Способы описания классов. Создание объектов. Производные типы данных. Создание и использование массивов, структур, объединений, указателей. Строки и литералы. Производные классы, наследование.	14	36	30	80
Итого			20	40	48	108

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Введение	История появления и развития языков программирования. Классификация языков программирования и их сравнительный анализ. Жизненный цикл программы. Интегрированные среды разработки, их сравнительный анализ.	2	-	10	12
2	Основы алгоритмизации	Понятие алгоритма. Свойства алгоритмов. Способы записи алгоритмов. Правила построения алгоритмов на языке блок-схем. Типы алгоритмов.	2	2	14	18
3	Программирование на Си++	Начальные сведения о языке. Сферы применения языка Си++. Процесс написания программы. Синтаксис языка Си++. Операции и выражения. Операторы. Функции. Встроенные типы данных.	2	6	66	74
зачет						4
Итого			6	8	90	108

5.2. Перечень лабораторных работ

- Программирование на языке Си++ в среде Code::Blocs. Программирование линейных алгоритмов.

- Программирование на языке Си++ в среде Code::Blocs. Программирование разветвляющихся алгоритмов. Логические операции и операции сравнения. Приоритет операций в С++. Оператор условной передачи управления if. Оператор множественного выбора switch.

- Программирование на языке Си++ в среде Code::Blocs. Программирование циклических алгоритмов. Оператор цикла for. Оператор цикла while. Оператор цикла do.
- Программирование на языке Си++ в среде Code::Blocs. Программирование с использованием одномерных массивов.
- Указатели. Программирование с использованием динамических двумерных массивов.
- Программирование с использованием строк.
- Программирование с использованием структур.
- Программирование с использованием функций.

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсовой работы в 4 семестре для очной формы обучения, в 6 семестре для заочной формы обучения.

Примерная тематика курсовой работы:

- разработка системы тестирования студентов по какой-либо дисциплине;
- разработка игры;
- разработка программы, вычисляющей день недели по введенной пользователем дате;
- разработка ежедневника;
- разработка математической системы построения графиков функций;
- разработка программы решения систем линейных уравнений.

Задачи, решаемые при выполнении курсовой работы:

- получение теоретических знаний и практических навыков по разработке приложений на языке высокого уровня Си++.

Пояснительная записка по курсовой работе состоит из трех частей. В первой, теоретической части, описываются современные методы разработки алгоритмов и проектирования программного обеспечения. Дается краткая сравнительная характеристика возможностей различных систем программирования. Вторая часть посвящена разработке алгоритма решения поставленной задачи. Третья разработка программы, описанию ее модульной структуры, входных и выходных данных, инструкции пользователя. В приложении приводится листинг разработанной программы и контрольный пример.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций

на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-4	знать теоретические основы программирования на языке высокого уровня и технологию разработки программного обеспечения для решения задач в своей профессиональной деятельности	Выполнение этапов разработки программного обеспечения	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь разрабатывать модели, алгоритмы и программное обеспечение на языке высокого уровня для решения задач в своей профессиональной деятельности	Разработка алгоритмов решения задач и построение блок-схем	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками работы в интегрированной программной среде изучаемого языка программирования.	Разработка программ на языке программирования Си++	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2. Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 4 семестре для очной формы обучения, 6 семестре для заочной формы обучения по двух балльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Незачтено
ПК-4	знать теоретические основы программирования на языке высокого уровня и технологию разработки программного обеспечения для решения задач в своей профессиональной деятельности	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь разрабатывать модели, алгоритмы и программное обеспечение на языке высокого уровня для решения задач в своей профессиональной деятельности	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

	владеть навыками работы в интегрированной программной среде изучаемого языка программирования.	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
--	--	--	--	------------------

7.2. Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1. Примерный перечень заданий для подготовки к тестированию

1. Строго определенная последовательность действий, необходимых для решения поставленной задачи, – это ... (в)
 - а) математическая модель задачи;
 - б) метод решения;
 - в) алгоритм;
 - г) блок-схема.
2. Что из перечисленного не относится к основным свойствам алгоритма: (б)
 - а) дискретность;
 - б) актуальность;
 - в) результативность
 - г) массовость
3. Свойство, означающее, что решение задачи, записанное в виде алгоритма, разбито на отдельные простейшие команды, которые расположены в порядке их выполнения, – это... (а)
 - а) дискретность;
 - б) определенность;
 - в) результативность;
 - г) детерминированность.
4. Что из перечисленного не относится к видам алгоритмов: (в)
 - а) линейные;
 - б) разветвляющиеся;
 - в) структурные;
 - г) циклические.
5. Что такое функция? (б)
 - а) Некоторая часть программы, содержащая описание переменных и констант основной программы
 - б) Некоторая часть программы, имеющая собственное имя и которая может вызываться из основной программы
 - в) Некоторая часть программы, содержащая вредоносный код, и блокирует определенные действия системы
 - г) Некоторая часть программы, в которой происходит начальная инициализация всех полей структур, массивов, переменных.
6. Что такое массив? (в)
 - а) Именованный набор переменных имеющих различные типы данных, и располагающихся в одной памяти

- б) Именованный набор переменных и функций, которые располагаются в одной области памяти
 - в) Именованный набор переменных имеющий один тип данных, и располагающихся в одной области памяти
 - г) Именованный набор переменных имеющих символьный тип данных, и располагающихся в одной области памяти
7. Как написать следующее выражение на языке С «Переменной a присвоено значение b»? (б)
- а) `a==b`
 - б) `a=b`
 - в) `b=a`
 - г) `a:=b`
8. Как написать следующее выражение «Второму элементу массива Myarray присвоено значение пяти»? (б)
- а) `int [1] Myarray=«пять»`
 - б) `intMyarray [1] = 5`
 - в) `intMyarray [2] = «пять»`
 - г) `intMyarray [2] = 5`
9. В каких случаях необходимо использовать оператор return в теле функции?(б)
- а) Всегда
 - б) если необходимо, чтобы функция вернула значение
 - в) если необходимо обеспечить выход из функции в произвольном месте
 - г) если указан тип возвращаемого значения, в том числе и void
10. Каким способом можно задать многострочный комментарий в языке C++? (а)
- а) `/*комментарии к программе*/`
 - б) `//комментарии к программе//`
 - в) `//комментарии к программе`
 - г) `{комментарии к программе}`
11. Логическое выражение может возвращать результат типа: (б)
- а) integer
 - б) boolean
 - в) char
 - г) logical

7.2.2. Примерный перечень заданий для решения стандартных задач

1. Разработать математическую модель и алгоритм нахождения площади кольца по внутреннему и внешнему радиусам.
2. Разработать математическую модель и алгоритм нахождения периметра эллипса по двум радиусам.
3. Разработать математическую модель и алгоритм нахождения площади поверхности конуса по радиусу и высоте.

4. Разработать математическую модель и алгоритм решения следующей задачи. Два поезда одновременно выходят навстречу друг другу со станций, находящихся на расстоянии Y км, и движутся по закону $y = 4t^2$, (y – путь, км; t – время, ч). Определить время прохождения поездами любого из указанных в списке путей, расположенных через каждые $0,1 Y$ км.

5. Разработать математическую модель и алгоритм решения следующей задачи. Для выпуска продукции используется сырье, производимое тремя поставщиками А, В, С. Цена сырья А составляла в январе A_1 и ежемесячно возрастала на $\Delta A\%$, цена сырья В держалась на уровне B_1 в течение года, а цена сырья С от C_1 ежемесячно снижалась на $\Delta C\%$. Определить цены поставщиков в любом месяце года, введенного пользователем.

6. Разработать математическую модель и алгоритм вычисления расстояния между двумя параллельными прямыми, описываемыми уравнениями $A_1x + B_1y = C_1$ и $A_2x + B_2y = C_2$.

7. Разработать математическую модель и алгоритм нахождения суммы, разности, произведения и частного двух комплексных чисел в алгебраической форме: $Z_1 = A_1 + jB_1$ и $Z_2 = A_2 + jB_2$.

8. Разработать математическую модель и алгоритм нахождения координаты середины отрезка с координатами (X_1, Y_1, Z_1) и (X_2, Y_2, Z_2) .

9. Разработать математическую модель алгоритм нахождения расстояния от точки $(0, 0, 0)$ до плоскости, заданной уравнением $Ax + By + Cz + D = 0$.

10. Разработать математическую модель алгоритм, позволяющий определить, касается ли прямая $A_1x + B_1y + C_1 = 0$ параболы $y = Ax^2 + Bx + C$.

7.2.3. Примерный перечень заданий для решения прикладных задач

1. Реализовать программу нахождения площади кольца по внутреннему и внешнему радиусам.

2. Реализовать программу нахождения периметра эллипса по двум радиусам.

3. Реализовать программу нахождения площади поверхности конуса по радиусу и высоте.

4. Реализовать программу решения следующей задачи. Два поезда одновременно выходят навстречу друг другу со станций, находящихся на расстоянии Y км, и движутся по закону $y = 4t^2$, (y – путь, км; t – время, ч). Определить время прохождения поездами любого из указанных в списке путей, расположенных через каждые $0,1 Y$ км.

5. Реализовать программу решения следующей задачи. Для выпуска продукции используется сырье, производимое тремя поставщиками А, В, С. Цена сырья А составляла в январе A_1 и ежемесячно возрастала на $\Delta A\%$, цена сырья В держалась на уровне B_1 в течение года, а цена сырья С от C_1 ежемесячно снижалась на $\Delta C\%$. Определить цены поставщиков в любом месяце года, введенного пользователем.

6. Реализовать программу вычисления расстояния между двумя параллельными прямыми, описываемыми уравнениями $A_1x + B_1y = C_1$ и $A_2x + B_2y = C_2$.

7. Реализовать программу нахождения суммы, разности, произведения и частного двух комплексных чисел в алгебраической форме: $Z_1 = A_1 + jB_1$ и $Z_2 = A_2 + jB_2$.

8. Реализовать программу нахождения координаты середины отрезка с координатами (X_1, Y_1, Z_1) и (X_2, Y_2, Z_2) .

9. Реализовать программу нахождения расстояния от точки $(0, 0, 0)$ до плоскости, заданной уравнением $Ax + By + Cz + D = 0$.

10. Реализовать программу, позволяющую определить, касается ли прямая $A_1x + B_1y + C_1 = 0$ параболы $y = Ax^2 + Bx + C$.

7.2.4. Примерный перечень вопросов для подготовки к зачету

1. Понятие алгоритма.
2. Методы формального описания алгоритмов.
3. Основные характеристики алгоритмов и этапы их разработки.
4. Разветвленные и циклические алгоритмы.
5. Сложные циклы. Алгоритмы с массивами.
6. Императивный и декларативный подходы к программированию. Примеры языков программирования. Понятие и назначение подпрограммы.
7. Языки высокого и низкого уровня. Примеры языков. Назначение и области применения. Процедурные языки программирования. Три типа базовых конструкций.
8. Объектно-ориентированные языки программирования. Понятия: метод, абстрагирование, инкапсуляция, класс, наследование, объект, полиморфизм, прототип. Примеры языков.
9. Логическое программирование. Системное программирование. Языки сценариев. Языки разметки данных. Примеры. Трансляторы (компиляторы и интерпретаторы).
10. Система программирования (назначение и состав). Этапы решения задачи на ЭВМ. Синтаксические и логические ошибки.
11. Язык Си. Идентификаторы, ключевые слова, комментарии. Основные типы данных. Основные арифметические операции (сокращенные операции).
12. Язык Си++. Операторы ввода-вывода `printf` и `scanf`, `getch`. Спецификации для различных типов переменных. Табулирование (фрагмент таблицы умножения).
13. Язык Си++. Математическая библиотека и математические функции (`abs`, `fabs`, `acos`, `asin`, `atan`, `cos`, `sin`, `tan`, `exp`, `pow`, `sqrt`, `M_PI`). Пример использования для вычисления.
14. Язык Си++. Логические операции (сравнения). Условный оператор `if`. Полная форма (`ifelse`) и сокращенная, операторы `break`, `continue`. Пример выбора большего из трех чисел.
15. Язык Си++. Оператор выбора `switch`. Пример программы – простейший

калькулятор (+, -, *, /).

16. Язык Си++. Условное выражение. Макроподстановки: определение констант, определение макрофункций выбора минимального из двух, максимального из трех, модуля, куба

17. Язык Си++. Оператор цикла while. Пример программы вычисления суммы двузначных чисел.

18. Язык Си++. Оператор цикла while. Пример программы вычисления суммы цифр числа.

19. Язык Си++. Алгоритм определения НОД. Программа определения НОД с циклом while.

20. Язык Си++. Алгоритм быстрого определения НОД при помощи операции остатка от деления. Программа определения НОД с циклом while.

21. Язык Си++. Алгоритм определения НОК. Программа определения НОК с циклом while.

22. Язык Си++. Оператор цикла for. Пример программы вычисления суммы чисел от a до b (вводятся).

23. Язык Си++. Понятие и вычисление факториала при помощи оператора for. Пример программы.

24. Язык Си++. Понятие и вычисление заданного числа последовательности Фибоначчи при помощи оператора for. Пример программы.

25. Язык Си++. Процедуры, понятие, описание, области применения. Формальные и фактические параметры. Параметры – переменные. Локальные переменные. Пример описания процедуры обмена значений двух переменных целого типа (перестановка значений).

26. Язык Си++. Функции, понятие, описание, области применения. Формальные и фактические параметры. Параметры – переменные. Локальные переменные. Пример описания функции выбора максимального из двух вещественных чисел.

27. Язык Си++. Символьные строки: объявление, инициализация, ввод/вывод. Примеры.

28. Язык Си++. Символьные строки: функции strlen, strcmp, strcpy, strncpy, strcat, strstr. Примеры.

29. Язык Си++. Обработка строк: пример подсчета количества слов в предложении.

30. Язык Си++. Символьные строки: указатели на строки, описание, применение. Пример поиска и подсчета количества вхождений подстроки в строку.

31. Язык Си++. Понятие и описание массива. Пример ввода и вывода массива. Заполнение случайными числами.

31. Язык Си++. Поиск максимального и минимального элементов массива. Пример программы.

33. Язык Си++. Сложение, вычитание, умножение векторов. Пример программы.

34. Язык Си++. Сортировка массива методом выбора максимального. Пример программы сортировки (фрагмент).

35. Язык Си++. Сортировка массива методом Пузырька. Пример программы сортировки (фрагмент).
36. Язык Си++. Описание матриц (двумерных массивов). Ввод и вывод матрицы на экран. Обработка матрицы (на примере удваивания элементов).
37. Язык Си++. Матрицы (двумерные массивы). Сложение матриц.
38. Язык Си++. Матрицы (двумерные массивы). Произведение матриц.
39. Язык Си++. Обработка массивов. Реверс массива (перестановка элементов в обратном порядке). Циклический сдвиг элементов массива вперед/назад.

7.2.5. Примерный перечень вопросов для подготовки к экзамену

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет проводится по результатам выполнения лабораторных работ, ответов на тестовые вопросы, решения стандартных и прикладных задач и ответов на вопросы.

Зачет ставится при правильном решении задачи и ответов на 50% вопросов.

7.2.7. Паспорт оценочных материалов

№п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Введение	ПК-4	Тест
2	Основы алгоритмизации	ПК-4	Тест, защита лабораторных работ, требования к курсовой работе
3	Программирование на Си++	ПК-4	Тест, защита лабораторных работ, требования к курсовой работе

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20мин.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1. Перечень учебной литературы, необходимой для освоения дисциплины

Воробьев Э.И.	Программирование на С++ : Учеб. пособие. - Воронеж : ГОУВПО "Воронежский государственный технический университет", 2008. - 104 с.
Холопкина Л.В.	Программирование на языке С++ : Учеб. пособие. - Воронеж : ГОУВПО "Воронежский государственный технический университет", 2011. - 184 с.
Корчагин Ю.Э.	Программирование на языках С и С++ : Лабораторный практикум: Учеб. пособие. - Воронеж : ГОУВПО "Воронежский государственный технический университет", 2009. - 251 с.
Минаева Ю.В.	Технологии разработки приложений в средах DELPHI и С++ Builder [Электронный ресурс] : Учеб. пособие. - Электрон. текстовые, граф. дан. (912Кб). - Воронеж : ФГБОУ ВПО "Воронежский государственный технический университет", 2011.

8.2. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных информационных справочных систем:

-интегрированная среда разработки Code::Blocs

Офисный пакет приложений MicroSoftOffice, Веб-браузер Internet Explorer; Open Office Text; Open Office Calc. Свободно распространяемое ПО. Научная электронная библиотека elibrary ([www. elibrary.ru](http://www.elibrary.ru))

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Дисплейный класс кафедры радиотехники.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО

ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Программирование на ЭВМ».

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсовой работы изложена в методических материалах. Выполнять этапы курсовой работы должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсовой работы, защитой курсовой работы.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоения учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом три дня эффективнее всего использовать для повторения и систематизации материала.