

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ:

Директор института

энергетики и автоматизированных систем

С.И. Лукьянов

28 сентября 2016 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ПРОГРАММИРОВАНИЕ И ОСНОВЫ АЛГОРИТМИЗАЦИИ

Направление подготовки

27.03.04 Управление в технических системах

Профиль программы

Системы и средства автоматизации технологических процессов

Уровень высшего образования – бакалавриат

Программа подготовки – академический бакалавриат

Форма обучения

заочная

Институт
Кафедра
Курс

Энергетики и автоматизированных систем
Автоматизированных систем управления
3

Магнитогорск
2016 г.

1 Цели освоения дисциплины

Цель освоения дисциплины «Программирование и основы алгоритмизации» являются изучение приёмов создания алгоритмов программируемой системы и реализация их с помощью алгоритмического языка.

Для достижения поставленной цели в дисциплине решаются **задачи**:

- знакомство с интегрированной средой разработки программного продукта;
- изучение алгоритмического языка C/C++;
- знакомство с понятием типа данных “класс” языка C++;
- освоение основных концепций структурного и объектно-ориентированного программирования;
- знакомство с основными методами отладки программ и составлением проекта.

2 Место дисциплины в структуре образовательной программы подготовки бакалавра

Дисциплина Б1.В.14 «Программирование и основы алгоритмизации» в вариативную часть блока 1 образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин:

- Б1.Б.9 «Математика»;
- Б1.Б.13 «Информатика и информационные технологии».

Перед началом изучения дисциплины студент должен обладать следующими знаниями, умениями и навыками:

знать:

- число, переменная функция: переменные и постоянные величины, возрастающая и убывающая переменная величины, функция;
- основы из алгебры логики: логические функции, общие понятия логического переменного, логическое отрицание, умножение (конъюнкция), сложение (дизъюнкция), равнозначность двух высказываний;
- символы латинского (латиница) и русского (кириллица) и специальные символы, кодирование символов (таблица ASCII кодов) и чисел;
- операторы и выражения языка Visual Basic of Application (VBA);
- функции, процедуры-функции и нестандартные функции VBA;

уметь:

- работать в текстовом редакторе;
- применять основные законы алгебры логики и преобразования логических выражений: переместительный, сочетательный, распределительный законы, закон инверсий;

владеть:

- методами преобразования части числа из одной системы счисления в другую;
- способами задания основных элементарных функций;
- способами введения типов в языке VBA, переменных и массивов из элементов введённых типов;
- навыками принятия решений (условные операторы) и циклов в VBA.

Знания (умения, навыки), полученные при изучении данной дисциплины будут необходимы для изучения следующих дисциплин:

- Б1.В.15 «Теория автоматического управления»;
- Б1.В.12 «Самонастраивающиеся системы»;
- Б1.В.ДВ.05.01 «Интегрированные системы проектирования и управления» (Б1.В.ДВ.05.02 «Аппаратное и программное обеспечение открытых интегрированных систем»);
- Б1.В.ДВ.01.01 «Системы автоматизации и управления» (Б1.В.ДВ.01.02 «Автоматизированное управление в технических системах»);

- Б1.В.ДВ.02.01 «Автоматизация технологических процессов и производств» (Б1.В.ДВ.02.02 «Оптимизация управления технологическими процессами металлургического производства»);
- Б1.В.ДВ.04.01 «Базы данных в АСУ ТП» (Б1.В.ДВ.04.02 «Автоматизированные информационные системы»).

3 Компетенции обучающегося, формируемые в результате освоения дисциплины и планируемые результаты обучения

В результате освоения дисциплины «Программирование и основы алгоритмизации» обучающийся должен обладать следующими компетенциями:

Структурный элемент компетенции	Планируемые результаты обучения
ДПК-3 способностью разрабатывать новое программное обеспечение, необходимое для управления техническими системами и решения практических задач профессиональной деятельности	
Знать	<ul style="list-style-type: none"> - принципы программного управления компьютером; - методы формального представления алгоритмов, основные (типовые) алгоритмы обработки данных; - принципы структурного и модульного программирования с использованием операторов языка C/C++
Уметь	<ul style="list-style-type: none"> - разрабатывать алгоритмы решения прикладных задач на основе типовых структур алгоритмов; - разрабатывать прикладные программные продукты с помощью современных средств и языков программирования с применением современных информационных технологий обработки данных (включая СУБД)
Владеть	<ul style="list-style-type: none"> - навыками работы в интегрированных средах разработки программного обеспечения (в т.ч. редактирования, компиляции, отладки программ); - навыками работы с современными инструментариями разработки прикладных программных продуктов на базе современных языков программирования; - методами построения современных проблемно-ориентированных прикладных программных средств
ПК-2 способностью проводить вычислительные эксперименты с использованием стандартных программных средств с целью получения математических моделей процессов и объектов автоматизации и управления	
Знать	<ul style="list-style-type: none"> - алгоритмы формирования выборки и обработки данных вычислительного эксперимента с целью создания на их основе модели технологического процесса; - особенности использования стандартных программных пакетов при создании моделей различных типов; - основные принципы и методологию разработки программного обеспечения, включая типовые способы организации данных и построения алгоритмов обработки данных с реальных объектов, синтаксис и семантику универсального алгоритмического языка программирования высокого уровня
Уметь	<ul style="list-style-type: none"> - использовать стандартные пакеты прикладных программ для решения практических задач на основе современных технологий программирования и алгоритмизации; - решать исследовательские и проектные задачи с использованием компьютеров и стандартных программных средств
Владеть	<ul style="list-style-type: none"> - навыками создания математических моделей процессов и объектов

	<p>автоматизации и управления с использованием стандартных программных средств;</p> <ul style="list-style-type: none">- навыками работы и организации практического функционирования программных средств и систем автоматизации и управления.
--	---

4 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 акад. часа, в том числе:

- контактная работа – 10,9 акад. часов;
- аудиторная – 8 акад. часов;
- внеаудиторная – 2,9 акад. часов
- самостоятельная работа – 124,4 акад. часов

Раздел/ тема дисциплины	Курс	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
Раздел 1. Основные понятия программирования	3							ДПК-3 - зув ПК-2 - зув
<i>1.1 Алгоритмы, определение, способы записи. Методы разработки</i>				-	10	Самостоятельное изучение учебной литературы	Выполнение контрольной работы Конспект лекций	
<i>1.2 Основные виды, этапы проектирования и жизненный цикл программных продуктов. Стандарты на разработку</i>				-	10	Самостоятельное изучение учебной литературы	Выполнение контрольной работы Конспект лекций	
Итого по разделу		0,5		-	20			
Раздел 2 Структурно-модульное программирование	3							ДПК-3 - зув ПК-2 - зув
<i>2.1 Программирование на языках C/C++. Алфавит, типы данных. Основные положения структурного программирования: операторы управления</i>		0,5		0,5	10	Самостоятельное изучение учебной литературы	Выполнение практической работы Конспект лекций	
<i>2.2 Массивы: одномерные и многомерные массивы. Структуры, синтаксис. Объявление типа: typedef. Перечисления. Объединения</i>		0,5		0,5	10	Самостоятельное изучение учебной литературы	Выполнение практической работы Конспект лекций	
<i>2.3. Функции. Объявления и определения</i>		1		1/1И ¹	10	Самостоятельное изучение учебной	Выполнение	

Раздел/ тема дисциплины	Курс	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
<i>функции. Структуры и массивы как параметры функции</i>						литературы	практической работы	
<i>2.4 Ввод/вывод в C/C++. Открытие и закрытие потоков. Файловый ввод/вывод</i>		-		-	14	Самостоятельное изучение учебной литературы	Конспект лекций	
Итого по разделу		2		2/1И¹	44			
Раздел 3 Объектно-ориентированное программирование (ООП)	3							ДПК-3 - зув ПК-2 - зув
<i>3.1 Основы ООП. Классы. Описание класса и определение объектов. Конструкторы и деструкторы</i>		0,5		0,5	10	Самостоятельное изучение учебной литературы	Выполнение практической работы Конспект лекций	
<i>3.2 Наследование. Виртуальные классы</i>		0,5		0,5	10	Самостоятельное изучение учебной литературы	Выполнение практической работы Конспект лекций	
<i>3.3 Перегрузка функций. Конструктор копий. Перегрузка оператор. Применение полиморфизма. Виртуальные функции</i>		-		1И ¹	10	Самостоятельное изучение учебной литературы	Выполнение практической работы Конспект лекций	
Итого по разделу		1		2/1И¹	30			
Раздел 4 Прикладное программирование	3							ДПК-3 - зув ПК-2 - зув
<i>4.1 Динамические структуры. Сортировка</i>				-	15	Самостоятельное изучение учебной литературы	Конспект лекций	
<i>4.2 Рекурсия и итерация. Рекурсия как метод вычислений. Графы. Поиск, постановка задачи, виды</i>				-	15,4	Самостоятельное изучение учебной литературы	Конспект лекций	
Итого по разделу		0,5		-	30,4			
Итого:		4		4/2И¹	124,4		Экзамен, контрольная работа	

5 Образовательные технологии

Для реализации предусмотренных видов учебной работы в качестве образовательных технологий в преподавании дисциплины «Программирование и основы алгоритмизации» используются традиционная и модульно-компетентностная технологии.

Передача необходимых теоретических знаний и формирование основных представлений по курсу «Программирование и основы алгоритмизации» происходит с использованием мультимедийного оборудования.

Теоретический курс включает: вводную лекцию, первое представление о предмете и знакомит студентов с назначением и задачами курса; проблемные лекции являются результатом усвоения полученной информации посредством постановки проблемного вопроса и поиска путей его решения; лекции – консультации, изложение нового материала сопровождается постановкой вопросов и дискуссией в поисках ответов на эти вопросы.

В ходе проведения лекционных занятий предусматривается:

- использование электронного демонстрационного материала по темам объектно-ориентированного программирования;
- использование электронных учебников по отдельным темам занятий;
- активные и интерактивные формы обучения: вариативный опрос, дискуссии, устный опрос, контрольная работа, тестовый опрос, индивидуальная «защита» практических работ и т.д.

Лекционный материал закрепляется в ходе практических работ, на которых выполняются групповые или индивидуальные задания по пройденной теме. Часть практических занятий проводится в интерактивной форме с использованием следующих методов:

- работа в команде, предусматривает совместную деятельность студентов в группе под руководством лидера, направленную на решение общей задачи с делением ответственности и полномочий;
- проблемное обучение, которое заключается в стимулировании студентов к самостоятельной «добыче» знаний, необходимых для решения конкретной проблемы;
- контекстное обучение, которое позволяет усвоить материал путем выявления связей между конкретным знанием и его применением;
- обучение на основе опыта, активизация познавательной деятельности студентов за счет ассоциации их собственного опыта с предметом изучения.

Самостоятельная работа стимулирует студентов к самостоятельной проработке в процессе выполнения домашних и контрольных работ, а также в процессе подготовки к устному опросу, тестированию и итоговой аттестации.

6 Учебно-методическое обеспечение самостоятельной работы студентов

По дисциплине «Программирование и основы алгоритмизации» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов на лабораторных занятиях происходит под контролем преподавателя в ходе выполнения практических работ, при решении задач и выполнении упражнений, которые для студентов определяет преподаватель. Внеаудиторная самостоятельная работа осуществляется в виде проработки материала практических занятий, выполнения домашних заданий и при консультациях с преподавателем.

Наименование раздела дисциплины	Перечень практических работ
Основные понятия программирования	1. Разработать алгоритм по заданию

Наименование раздела дисциплины	Перечень практических работ
Структурно-модульное программирование	1. Операции и выражения 2. Условные операторы 3. Операторы циклов 4. Операторы циклов 5. Массивы 6. Структуры 7. Указатели 8. Функции
Объектно-ориентированное программирование	1. Создание объекта типа class 2. Конструкторы и деструкторы 3. Наследование 4. Множественное наследование 5. Виртуальные классы 6. Перегрузка функций 7. Перегрузка операторов
Прикладное программирование	1. Библиотека стандартных шаблонов (STL – Standard template library)

Пример экзаменационного билета

1. Алфавит языка C/C++
2. Понятие области видимости класс и прав доступа (public, private, protected)
3. Дан массив: а) вывести на экран сначала его неотрицательные элементы, затем отрицательные; б) верно ли, что сумма элементов, которые больше 20, превышает а (функция)

Пример варианта контрольной работы №1

1. Функция, определение
2. В каком случае не требуется прототип функции? Пример
3. Глобальные переменные, пример
4. Найти ошибку:


```
int f(int a, int b);
void main()
{.....}
int f(int a, int b)
{.....
return }
```
5. Решить с помощью функции:

$$z = \begin{cases} x - a, & \text{если } x \geq 0; \\ x / 2, & \text{если } x \leq 0; \end{cases}$$
 где а вводится с клавиатуры
6. Написать программу, в которой функция находит сумму элементов массива, имеющих нечетные значения

Пример варианта контрольной работы №2

1. Инкапсуляция –
2. Теряет ли оператор при перегрузке что-либо из своих функциональных исходных возможностей?
3. Правильен ли фрагмент (создание виртуальной функции)?


```
class B {
```

```

public: virtual int f (int a) = 0;
.....};
class D : public B {
public: int f (int a, int b) { return a*b; }
.....};

```

4. Перегрузить оператор +

Пример варианта домашней работы №1

Программирование циклов и условных операторов. Найти сумму ряда при различных и заданных значениях переменной ряда и заданном числе его членов:

$$S = \frac{(2 \cdot x)^2}{2} + \frac{(2 \cdot x)^4}{24} + \dots + (-1)^n \cdot \frac{(2 \cdot x)^{2n}}{(2n)!}; 0,1 \leq x \leq 1; n_{\max} = 15$$

Пример варианта домашней работы №2

Программа из двух функций; в первой функции (main) вводятся конкретные массивы, вызывается вторая функция. Вторая функция производит заданные операции над элементами массивов, переданными в функцию, и возвращает результат:

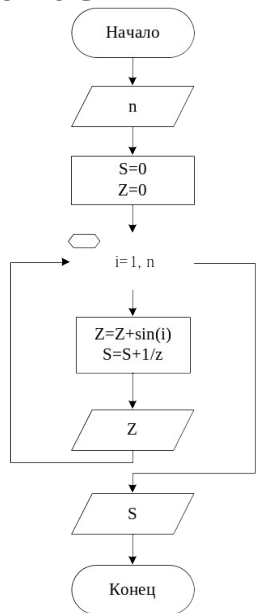
Для заданной матрицы размером 8 на 8 найти такое k, чтобы k-тая строка матрицы совпадала с k-тым столбцом; найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

7 Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
ДПК-3: способностью разрабатывать новое программное обеспечение, необходимое для управления техническими системами и решения практических задач профессиональной деятельности		
Знать	<ul style="list-style-type: none"> - принципы программного управления компьютером; - методы формального представления алгоритмов, основные (типовые) алгоритмы обработки данных; - принципы структурного и модульного программирования с использованием операторов языка C/C++ 	<p>Перечень теоретических вопросов к экзамену:</p> <ol style="list-style-type: none"> 1. Понятие алгоритма 2. Классификация алгоритмов 3. Язык программирования 4. Классификация языков 5. Способы записи алгоритмов 6. Алгоритм линейной структуры, пример 7. Алгоритм разветвляющейся структуры, пример 8. Алгоритм циклической структуры, пример 9. Принципы проектирования алгоритмов 10. Алфавит языка C/C++ 11. Идентификаторы и ключевые (служебные) слов 12. Константы языка C/C++, задание определение и использование 13. Типы данных 14. Спецификаторы класса памяти (auto, static, register, extern) и их влияние на время жизни переменной 15. Понятие указателя в C/C++: определение, инициализация, разыменование 16. Указатель на тип void, его использование с объектами разных типов 17. Перечислимый тип в C/C++: определение типа, переменных этого типа и их использование 18. Понятие массива, определение одномерного массива, обращение к отдельным элементам, инициализация 19. Многомерный массив (двух и трёхмерный), расположение элементов в памяти, инициализация при определении 20. Имя массива как указатель; доступ к элементам массива по указателю 21. Определение типа структуры и переменных типа структуры; инициализация

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>структуры при определении</p> <p>22. Понятие объединения (union): определение объединения, инициализация объединения, обращение к элементам объединения</p> <p>23. Введение новых типов с помощью typedef</p> <p>24. Понятие выражения; первичные элементы выражения</p> <p>25. Операции инкремента и декремента (++ , --); префиксный и постфиксный инкремент</p> <p>26. Встроенная функция sizeof; её использование для определения размера переменной определённого типа</p> <p>27. Унарные операции(операторы) в C/C++. Порядок их выполнения в C/C++</p> <p>28. Бинарные операции в C/C++: арифметические операции</p> <p>29. Операции (операторы) побитового правого и левого сдвига операнда целого типа</p> <p>30. Операции (операторы) отношения в C/C++; порядок их выполнения. Понятие true и false в C/C++.</p> <p>31. Побитовые логические операции</p> <p>32. Логические операции в C/C++</p> <p>33. Тернарная операция ?: и её использование взамен оператора if</p> <p>34. Операция запятая и её использование в операторах (инструкциях) цикла</p> <p>35. Понятие функции как многократно используемого участка программы (подпрограммы). Выделение в стеке памяти для передачи фактических параметров</p> <p>36. Описание функции (прототип). Список формальных параметров, допустимые типы формальных параметров</p> <p>37. Определение функции. Тело функции использование оператора return</p> <p>38. Вызов функции. Механизм передачи фактических параметров по значению. Использование указателей для передачи параметров по ссылке</p> <p>39. Операторы выбора: условный оператор if</p> <p>40. Оператор выбора: переключатель switch</p> <p>41. Операторы цикла: for, while, do ... while</p> <p>42. Операторы передачи управления: return, continue</p> <p>43. Обращение к элементам массива по указателю</p> <p>44. Передача массива в функцию с помощью указателя. Обращение к элементу двумерного массива по указателю. Операторы new и delete</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
Уметь	<p>- разрабатывать алгоритмы решения прикладных задач на основе типовых структур алгоритмов;</p> <p>- разрабатывать прикладные программные продукты с помощью современных средств и языков программирования с применением современных информационных технологий обработки данных (включая СУБД)</p>	<p>45. Объявление переменных на внешнем уровне, их область видимости</p> <p>Примеры практических заданий для экзамена:</p> <ol style="list-style-type: none"> 1. Составить алгоритм вычисления по формуле $S = X \cdot Y^2$ 2. Составить алгоритм решения для функции $Z(X) = X$ при $X > 0$ и $Z(X) = X^2$ при $X \leq 0$ 3. Структура спецификация, поля структуры: позиция, наименование технического средства, количество. Программа выводит необходимое техническое средство, по выбранной позиции 4. Реализовать блок-схему на языке C++  <pre> graph TD Start([Начало]) --> Input[/n/] Input --> Init[S=0 Z=0] Init --> LoopStart{i=1, n} LoopStart --> Calc[Z=Z+sin(i) S=S+1/z] Calc --> OutZ[/Z/] OutZ --> OutS[/S/] OutS --> End([Конеч]) </pre>
Владеть	<p>- навыками работы в интегрированных средах разработки программного обеспечения (в т.ч. редактирования, компиляции, отладки программ);</p> <p>- навыками работы с современными инструментариями разработки прикладных</p>	<p>Перечень практических работ:</p> <ol style="list-style-type: none"> 1. Разработать алгоритм по заданию 2. Операции и выражения 3. Условные операторы 4. Операторы циклов 5. Операторы циклов

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	<p>программных продуктов на базе современных языков программирования; методами построения современных проблемно-ориентированных прикладных программных средств</p>	<p>6. Массивы 7. Структуры 8. Указатели 9. Функции 10. Создание объекта типа class 11. Конструкторы и деструкторы 12. Наследование 13. Множественное наследование 14. Виртуальные классы 15. Перегрузка функций 16. Перегрузка операторов</p>
<p>ПК-2 способностью проводить вычислительные эксперименты с использованием стандартных программных средств с целью получения математических моделей процессов и объектов автоматизации и управления</p>		
<p>Знать</p>	<p>алгоритмы формирования выборки и обработки данных вычислительного эксперимента с целью создания на их основе модели технологического процесса; особенности использования стандартных программных пакетов при создании моделей различных типов; основные принципы и методологию разработки программного обеспечения, включая типовые способы организации данных и построения алгоритмов обработки данных с реальными объектами, синтаксис и семантику универсального алгоритмического языка программирования высокого уровня</p>	<p>Перечень теоретических вопросов к экзамену: 1. Основные этапы алгоритмизации. Постановка задачи. Построение математической модели. Разработка алгоритма решения задачи. Программирование 2. Этапы работ по созданию программных продуктов 3. Составление технического задания на программирование 4. Технический проект по созданию программных продуктов 5. Рабочая документация (рабочий проект). Основные виды 6. Жизненный цикл программных продуктов 7. Маркетинг и спецификация программного продукта 8. Проектирование структуры программного продукта 9. Программирование, тестирование и отладка программ 10. Документирование программного продукта 11. Выход программного продукта на рынок программных средств 12. Эксплуатация и сопровождение программного продукта 13. Снятие программного продукта с продажи и отказ от сопровождения 14. Основные виды, этапы проектирования и жизненный цикл программных продуктов 15. Стандарты на разработку. Стандарты на разработку прикладных программных средств. Документирование, сопровождение и эксплуатация программных средств</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		16. От C к C++. Понятие объектно-ориентированного программирования 17. Перегрузка функций (статическая) 18. Понятие конструктора. Использование конструкторов для инициализации вновь созданной переменной типа класс 19. Понятие деструктора. Использование деструктора 20. Понятие о перегрузках операторов. Пример перегрузки оператора + 21. Перегрузка функций 22. Понятие области видимости класс и прав доступа (public, private, protected) 23. Механизм наследования. 24. Виртуальные базовые классы 25. Понятие полиморфизма, механизм. Примеры 26. Виртуальные функции
Уметь	- использовать стандартные пакеты прикладных программ для решения практических задач на основе современных технологий программирования и алгоритмизации; - решать исследовательские и проектные задачи с использованием компьютеров и стандартных программных средств	Перечень практических работ: 1. Библиотека стандартных шаблонов (STL – Standard template library): распределители памяти, предикаты, функции сравнения и объекты-функции 2. Библиотека стандартных шаблонов (STL – Standard template library): строковый класс 3. Библиотека стандартных шаблонов (STL – Standard template library): класс vector 4. Библиотека стандартных шаблонов (STL – Standard template library): класс list 5. Динамические структуры. Сортировка 6. Рекурсия и итерация. Рекурсия как метод вычислений 7. Графы. Поиск, постановка задачи, виды
Владеть	- навыками создания математических моделей процессов и объектов автоматизации и управления с использованием программных средств; - навыками работы и организации практического функционирования программных средств и систем автоматизации и управления	Перечень практических заданий на экзамен: 1. Определить выходной сигнал терморезистора в заданном температурном диапазоне. Вывести в два столбца, начальное сопротивление и температурный коэффициент задать как именованные константы 2. Структура спецификации, поля структуры: позиция, наименование технического средства, количество. Программа выводит необходимое техническое средство, по выбранной позиции 3. Рассчитать и вывести относительную погрешность n измерений тока и определить укладывается ли данная погрешность в класс точности прибора 4. Оценить n количество измерений температуры, на наличие грубой погрешности

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>5. Рассчитать выходной сигнал заданного регулятора, расчет выполняет функция</p> <p>6. Определить выходной сигнал нормирующего преобразователя (на основе неинвертирующего операционного усилителя), работающего совместно с термоэлектрическим преобразователем (считать, что термopара инерционное звено 1-го порядка, с заданной постоянной времени)</p>

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Программирование и основы алгоритмизации» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен проводится в устной форме по теоретическим вопросам и задачам.

Показатели и критерии оценивания экзамена:

– на оценку «отлично» (5 баллов) – обучающийся должен полно раскрыть содержание материала в объеме программы дисциплины, чётко и правильно дать определения, привести доказательства на основе математических и логических выкладок, показать навыки исследовательской деятельности. Ответ должен быть самостоятельным, при ответе использованы знания, приобретённые ранее;

– на оценку «хорошо» (4 балла) – обучающийся должен раскрыть содержание материала в объеме программы дисциплины, в основном правильно дать основные определения и понятия предмета. При ответе допущены неточности, нарушена последовательность изложения, допущены небольшие неточности при выводах и использовании терминов, практические навыки нетвёрдые;

– на оценку «удовлетворительно» (3 балла) – обучающийся должен усвоить основное содержание материала. При ответе определения и понятия даны не чётко, допущены ошибки при промежуточных математических выкладках в выводах, практические навыки слабые;

– на оценку «неудовлетворительно» (2 баллов) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач. При ответе допущены грубые ошибки в определениях, доказательства теорем не проведено, не даны ответы на дополнительные вопросы преподавателя, отсутствуют навыки исследовательской деятельности;

– на оценку «неудовлетворительно» (1 балл) – не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач, основное содержание учебного материала не раскрыто.

8 Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература:

1. Самарина, И. Г. Программирование и основы алгоритмизации : учебное пособие. Ч. 1. Курс лекций / И. Г. Самарина. - Магнитогорск : МГТУ, 2013. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=908.pdf&show=dcatalogues/1/1118881/908.pdf&view=true> (дата обращения: 14.05.2020). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM.

2. Давыдова, Н. А. Программирование / Давыдова Н.А., Боровская Е.В., - 3-е изд., (эл.) - Москва : БИНОМ. ЛЗ, 2015. - 241 с.: ISBN 978-5-9963-2647-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/544438> (дата обращения: 18.09.2020). – Режим доступа: по подписке.

б) Дополнительная литература:

1. Полубенцева М. С/С++. Процедурное программирование / М. Полубенцева. - Санкт-Петербург : БХВ-Петербург, 2010. - 448 с. - ISBN 978-5-9775-0145-3. - URL: <https://ibooks.ru/reading.php?productid=18410> (дата обращения: 18.09.2020). - Текст: электронный

2. Павловская Т. С/С++. Структурное и объектно-ориентированное программирование. Практикум / Т. Павловская, Ю. Щупак. - Санкт-Петербург : Питер, 2011. - 352 с. - ISBN 978-5-459-00613-1. - URL: <https://ibooks.ru/reading.php?productid=21762> (дата обращения: 18.09.2020). - Текст: электронный.

3. Ашарина И. В. Объектно-ориентированное программирование в С++: лекции и упражнения. Учебное пособие для вузов / И.В. Ашарина. - Москва: Горячая Линия–Телеком, 2012. - 320 с. - ISBN 978-5-9912-7001-4. - URL: <https://ibooks.ru/reading.php?productid=333353> (дата обращения: 18.09.2020). - Текст: электронный

4. Хабибуллин И. Программирование на языке высокого уровня. С/С++ / И. Хабибуллин. - Санкт-Петербург : БХВ-Петербург, 2010. - 512 с. - ISBN 5-94157-559-9. - URL: <https://ibooks.ru/reading.php?productid=18532> (дата обращения: 18.09.2020). - Текст: электронный

5. Кубенский А. Структуры и алгоритмы обработки данных: объектно ориентированный подход и реализация на С++ / А. Кубенский. - Санкт-Петербург : БХВ-Петербург, 2010. - 464 с. - ISBN 5-94157-506-8. - URL: <https://ibooks.ru/reading.php?productid=18563> (дата обращения: 18.09.2020). - Текст: электронный

в) Методические указания:

1. Самарина, И. Г. Программирование и основы алгоритмизации : учебное пособие. Ч. 3 / И. Г. Самарина, А. Р. Бондарева ; МГТУ. - Магнитогорск : МГТУ, 2015. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=1485.pdf&show=dcatalogues/1/1124014/1485.pdf&view=true> (дата обращения: 14.05.2020). - Макрообъект. - Текст: электронный. - Сведения доступны также на CD-ROM
2. Самарина, И. Г. Программирование и основы алгоритмизации: учебное пособие / И. Г. Самарина, А. Р. Бондарева; МГТУ. - Магнитогорск: МГТУ, 2018. - 71 с.: табл., схемы, диагр. - Текст : непосредственный.

г) Программное обеспечение и Интернет-ресурсы:

Наименование ПО	№ договора	Срок действия лицензии
Наименование ПО	№ договора	Срок действия лицензии
MS Windows XP Professional(для классов)	Д-1227-18 от 08.10.2018	11.10.2021
MS Office 2003 Professional	№ 135 от 17.09.2007	бессрочно
7Zip	свободно распространяемое ПО	бессрочно
FAR Manager	свободно распространяемое ПО	бессрочно

Название курса	Ссылка
Электронная база периодических изданий East View Information Services, ООО «ИВИС»	https://dlib.eastview.com/
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: https://elibrary.ru/project_risc.asp
Поисковая система Академия Google (Google Scholar)	URL: https://scholar.google.ru/
Информационная система - Единое окно доступа к информационным ресурсам	URL: http://window.edu.ru/

Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»	URL: http://www1.fips.ru/
Российская Государственная библиотека. Каталоги	https://www.rsl.ru/ru/4readers/catalogues/
Электронные ресурсы библиотеки МГТУ им. Г.И. Носова	http://magtu.ru:8085/marcweb2/Default.asp
Федеральный образовательный портал – Экономика. Социология. Менеджмент	http://ecsocman.hse.ru/
Университетская информационная система РОССИЯ	https://uisrussia.msu.ru
Международная наукометрическая реферативная и полнотекстовая база данных научных изданий «Web of science»	http://webofscience.com
Международная реферативная и полнотекстовая справочная база данных научных изданий «Scopus»	http://scopus.com
Международная база полнотекстовых журналов Springer Journals	http://link.springer.com/
Международная коллекция научных протоколов по различным отраслям знаний Springer Protocols	http://www.springerprotocols.com/
Международная база научных материалов в области физических наук и инжиниринга SpringerMaterials	http://materials.springer.com/
Международная база справочных изданий по всем отраслям знаний SpringerReference	http://www.springer.com/references
Международная реферативная база данных по чистой и прикладной математике zbMATH	http://zbmath.org/
Международная реферативная и полнотекстовая справочная база данных научных изданий «Springer Nature»	https://www.nature.com/siteindex
Архив научных журналов «Национальный электронно-информационный конкорциум» (НП НЭИКОН)	https://archive.neicon.ru/xmlui/

9 Материально-техническое обеспечение дисциплины:

Материально-техническое обеспечение дисциплины включает:

Тип и название аудитории	Оснащение аудитории
Учебные аудитории для проведения занятий лекционного типа	Мультимедийные средства хранения, передачи и представления информации
Учебная аудитория для проведения практических занятий: компьютерный класс	Персональные компьютеры с пакетом MS Office, выходом в Интернет и с доступом в электронную информационно-образовательную

	среду университета
Помещения для самостоятельной работы обучающихся	Персональные компьютеры с пакетом MS Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета
Учебные аудитории для групповых и индивидуальных консультаций, текущего контроля и промежуточных консультаций	Доска, мультимедийный проектор, экран
Помещение для хранения и профилактического обслуживания учебного оборудования	Стеллажи для хранения учебно-методической документации