



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ
Директор ИЭиАС
В.Р. Храмшин

26.01.2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ПРОГРАММИРОВАНИЕ И ОСНОВЫ АЛГОРИТМИЗАЦИИ

Направление подготовки (специальность)
27.03.04 Управление в технических системах

Направленность (профиль/специализация) программы
Системы и средства автоматизации технологических процессов

Уровень высшего образования - бакалавриат

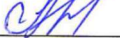
Форма обучения
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Автоматизированных систем управления
Курс	2
Семестр	4

Магнитогорск
2022 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 27.03.04 Управление в технических системах (приказ Минобрнауки России от 31.07.2020 г. № 871)

Рабочая программа рассмотрена и одобрена на заседании кафедры Автоматизированных систем управления
26.01.2022, протокол № 6

Зав. кафедрой  С.М. Андреев

Рабочая программа одобрена методической комиссией ИЭиАС
26.01.2022 г. протокол № 5

Председатель  В.Р. Храмшин

Рабочая программа составлена:
ст. преподаватель кафедры АСУ,  И.Г. Самарина

Рецензент:

зам.  директора ЗАО "КонсОМ СКС" , канд. техн. наук
Ю.Н. Волшуков



Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Автоматизированных систем управления

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.М. Андреев

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Автоматизированных систем управления

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.М. Андреев

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2025 - 2026 учебном году на заседании кафедры Автоматизированных систем управления

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.М. Андреев

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2026 - 2027 учебном году на заседании кафедры Автоматизированных систем управления

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.М. Андреев

1 Цели освоения дисциплины (модуля)

изучение приёмов создания алгоритмов программируемой системы и реализация их с помощью алгоритмического языка

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Программирование и основы алгоритмизации входит в обязательную часть учебного плана образовательной программы.

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

Метрология и средства измерений

Основы объектно-ориентированного программирования

Философия

Информатика

Учебная - практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности

Учебная - ознакомительная практика

Цифровые технологии обработки информации в автоматизированных системах управления

Начертательная геометрия и компьютерная графика

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Продвижение научной продукции

Производственная - технологическая (производственно-технологическая) практика

Электрические измерения

Технические измерения и приборы

Подготовка к сдаче и сдача государственного экзамена

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

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

Код индикатора	Индикатор достижения компетенции
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК-1.1	Анализирует задачу, выделяя ее базовые составляющие, рассматривает и предлагает возможные варианты решения поставленной задачи, оценивая их достоинства и недостатки
УК-1.2	Определяет, интерпретирует и ранжирует информацию, требуемую для решения поставленной задачи; осуществляет поиск информации по различным типам запросов
УК-1.3	При обработке информации отличает факты от мнений, интерпретаций, оценок, формирует собственные мнения и суждения, аргументирует свои выводы и точку зрения
ОПК-6	Способен разрабатывать и использовать алгоритмы и программы, современные информационные технологии, методы и средства контроля, диагностики и управления, пригодные для практического применения в сфере своей профессиональной деятельности
ОПК-6.1	Разрабатывает алгоритмы и программы для практического

	применения в сфере своей профессиональной деятельности
ОПК-6.2	Использует современные методы и средства контроля, диагностики и управления для практического применения в сфере своей профессиональной деятельности
ОПК-6.3	Использует современные информационные технологии в сфере своей профессиональной деятельности

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 зачетных единиц 144 академических часов, в том числе:

- контактная работа – 72 академических часов;
- аудиторная – 68 академических часов;
- внеаудиторная – 4 академических часов;
- самостоятельная работа – 36,3 академических часов;
- в форме практической подготовки – 0 академических часов;
- подготовка к экзамену – 35,7 академических часов

Форма аттестации - экзамен

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в академических часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Основные понятия программирования								
1.1 Алгоритмы, определение, способы записи. Методы разработки	4	2		2	1,3	Самостоятельное изучение учебной литературы	Выполнение практической работы	УК-1.1, ОПК-6.1
1.2 Основные виды, этапы проектирования и жизненный цикл программных продуктов. Стандарты на разработку		2		2	3	Самостоятельное изучение учебной литературы	Выполнение практической работы, КР	УК-1.1, ОПК-6.1
Итого по разделу		4		4	4,3			
2. Структурно-модульное программирование								
2.1 Программирование на языках C/C++. Алфавит, типы данных. Основные положения структурного программирования: операторы управления	4	2		4/2И	6	Самостоятельное изучение учебной литературы	Выполнение практической работы	УК-1.1, ОПК-6.1
2.2 Массивы: одномерные и многомерные массивы. Структуры, синтаксис. Объявление типа: typedef. Перечисления. Объединения		4		6/2И	3	Самостоятельное изучение учебной литературы	Выполнение практической работы	УК-1.1, ОПК-6.1
2.3 Функции. Объявления и определения функции. Структуры и массивы как параметры функции		4		6/2И	4	Самостоятельное изучение учебной литературы	Выполнение практической работы	УК-1.1, ОПК-6.1
2.4 Ввод/вывод в C/C++. Открытие и закрытие потоков. Файловый ввод/вывод		2		4/1,9И	4	Самостоятельное изучение учебной литературы	Выполнение практической работы, КР	
Итого по разделу		12		20/7,9И	17			

3. Объектно-ориентированное программирование (ООП)								
3.1 Основы ООП. Классы. Описание класса и определение объектов. Конструкторы и деструкторы	4	2		2/2И	3	Самостоятельное изучение учебной литературы	Выполнение практической работы	УК-1.1, ОПК-6.1
3.2 Наследование. Виртуальные классы		4		2/2И	3	Самостоятельное изучение учебной литературы	Выполнение практической работы	УК-1.1, ОПК-6.1
3.3 Перегрузка функций. Конструктор копий. Перегрузка оператор. Применение полиморфизма. Виртуальные функции		4		4	3	Самостоятельное изучение учебной литературы	Выполнение практической работы, КР	УК-1.1, ОПК-6.1
Итого по разделу		10		8/4И	9			
4. Прикладное программирование								
4.1 Динамические структуры. Сортировка	4	4		1	3	Самостоятельное изучение учебной литературы	Выполнение практической работы	УК-1.1, ОПК-6.1
4.2 Рекурсия и итерация. Рекурсия как метод вычислений. Графы. Поиск, постановка задачи, виды		4		1	3	Самостоятельное изучение учебной литературы	Выполнение практической работы	УК-1.1, ОПК-6.1
Итого по разделу		8		2	6			
Итого за семестр		34		34/11,9И	36,3		экзамен	
Итого по дисциплине		34		34/11,9 И	36,3		экзамен	

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;
- встречи с представителями проектных и обслуживающих предприятий: встречи с представителями проектных и обслуживающих предприятий: ООО «ОСК», ООО «Информсервис ММК», ЗАО «КонсОМ»; предполагаемые темы встреч «Применение технологии объектно-ориентированного программирования для разработки обучающих программ-тренажеров», «Программное обеспечение современной системы управления»;
- 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 обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

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: 14.09.2022). - 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: 18.09.2022). - 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: 18.09.2022). - 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: 18.09.2022). - 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: 18.09.2022). - 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: 18.09.2022). - 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: 14.09.2022). - 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

3. Самарина, И. Г. Программирование и основы алгоритмизации : практикум / И. Г. Самарина, Е.Ю. Мухина, А. Р. Бондарева; МГТУ. - Магнитогорск : МГТУ, 2021. - 47 с. : табл., схемы, диагр. - Текст : непосредственный.

алгоритмизация

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно
7Zip	свободно	бессрочно
FAR Manager	свободно	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Национальная информационно-аналитическая система – Российский индекс научного	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	URL: http://window.edu.ru/
Электронные ресурсы библиотеки МГТУ им. Г.И. Носова	https://magtu.informsystema.ru/Marc.html?locale=ru
Международная реферативная и полнотекстовая справочная база	http://scopus.com
Международная база полнотекстовых журналов Springer Journals	http://link.springer.com/
Международная коллекция научных протоколов по различным отраслям	http://www.springerprotocols.com/

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Учебные аудитории для проведения занятий лекционного типа - Мультимедийные средства хранения, передачи и представления информации

Учебная аудитория для проведения практических занятий (компьютерный класс ауд. 448) - 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

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
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:

$$z = \begin{cases} x - a, & \text{если } x \geq 0; \\ x / 2, & \text{если } x \leq 0; \end{cases} \text{ , где } a \text{ вводится с клавиатуры}$$

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:

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

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.

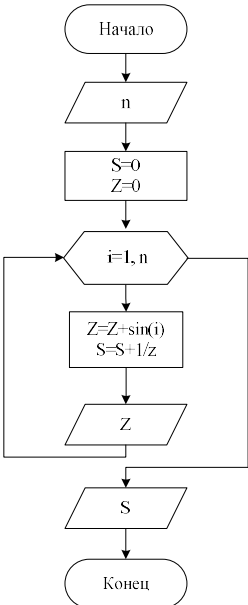
7 Оценочные средства для проведения промежуточной аттестации
а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
УК-1: Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач		
УК-1.2	Определяет, интерпретирует и ранжирует информацию, требуемую для решения поставленной задачи; осуществляет поиск информации по различным типам запросов	<p>Перечень теоретических вопросов к экзамену:</p> <ol style="list-style-type: none"> 1. Основные этапы алгоритмизации. Постановка задачи. Построение математической модели. Разработка алгоритма решения зада. 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 16. Принципы проектирования алгоритмов <p>Перечень практических работ:</p> <ol style="list-style-type: none"> 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

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		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
ОПК-6: Способен разрабатывать и использовать алгоритмы и программы, современные информационные технологии, методы и средства контроля, диагностики и управления, пригодные для практического применения в сфере своей профессиональной деятельности		
ОПК-6.1	Разрабатывает алгоритмы и программы для практического применения в сфере своей профессиональной деятельности	<i>Перечень теоретических вопросов к экзамену:</i> 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

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>20. Имя массива как указатель; доступ к элементам массива по указателю</p> <p>21. Определение типа структуры и переменных типа структуры; инициализация 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) побитого 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>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>44. Передача массива в функцию с помощью указателя. Обращение к элементу двумерного массива по указателю. Операторы new и delete</p> <p>45. Объявление переменных на внешнем уровне, их область видимости</p> <p>46. От C к C++. Понятие объектно-ориентированного программирования</p> <p>47. Перегрузка функций (статическая)</p> <p>48. Понятие конструктора. Использование конструкторов для инициализации вновь созданной переменной типа класс</p> <p>49. Понятие деструктора. Использование деструктора</p> <p>50. Понятие о перегрузках операторов. Пример перегрузки оператора +</p> <p>51. Перегрузка функций</p> <p>52. Понятие области видимости класс и прав доступа (public, private, protected)</p> <p>53. Механизм наследования.</p> <p>54. Виртуальные базовые классы</p> <p>55. Понятие полиморфизма, механизм. Примеры</p> <p>56. Виртуальные функции</p> <p>Примеры практических заданий для экзамена: Перечень практических заданий на экзамен:</p> <ol style="list-style-type: none"> 1. Определить выходной сигнал терморезистора в заданном температурном диапазоне. Вывести в два столбца, начальное сопротивление и температурный коэффициент задать как именованные константы 2. Структура спецификация, 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 5. Рассчитать выходной сигнал заданного регулятора, расчет выполняет функция. 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) 6. Составить алгоритм вычисления по формуле $S = X \cdot Y^2$

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>7. Составить алгоритм решения для функции $Z(X) = X$ при $X > 0$ и $Z(X) = X^2$ при $X \leq 0$</p> <p>8. Структура спецификация, поля структуры: позиция, наименование технического средства, количество. Программа выводит необходимое техническое средство, по выбранной позиции</p> <p>9. Реализовать блок-схему на языке C++</p>  <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 --> OutputZ[/Z/] OutputZ --> LoopStart LoopStart --> OutputS[/S/] OutputS --> End([Конец]) </pre>

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.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ
КОНТРОЛЬНОЙ РАБОТЫ**

по дисциплине «Программирование и основы алгоритмизации»

Оформление контрольной работы должно содержать следующие пункты:

- a) Титульный лист
- b) Лист с заданием
- c) Непосредственно сам листинг программы и окно результатов работы программы.
- d) Реферат
- e) Список использованных источников

Задачи для контрольной работы

1 Задача: Вычислить сумму ряда (по вариантам). Не использовать **стандартные** функции возведения в степень и факториал, а написать свои функции для вычисления числителя и знаменателя.

№ варианта
1. $S = \frac{(2 \cdot x)^2}{2} + \frac{(2 \cdot x)^4}{24} + \dots + (-1)^n \cdot \frac{(2 \cdot x)^{2n}}{(2n)!}$; $0,1 \leq x \leq 1; n_{\max} = 15$
2. $S = -(1+x)^2 + \frac{(1+x)^4}{2!} + \dots + (-1)^n \cdot \frac{(1+x)^{2n}}{n!}$; $-2 \leq x \leq 0,1; n_{\max} = 40$
3. $S = x + \frac{x^3}{3!} + \dots + \frac{x^{2n+1}}{(2n+1)!}$; $0,1 \leq x \leq 1; n_{\max} = 20$
4. $S = \frac{x}{3!} + \frac{x^2}{5!} + \dots + \frac{(x)^n}{(2n+1)!}$; $0,2 \leq x \leq 0,8; n_{\max} = 20$
5. $S = x \cos \frac{\pi}{4} + x^2 \cos 2 \frac{\pi}{4} + \dots + x^n \cos n \frac{\pi}{4}$; $0,1 \leq x \leq 0,8; n_{\max} = 40$
6. $S = 3 \cdot x + 8 \cdot x^2 + \dots + n \cdot (n+2) \cdot x^n$; $0,1 \leq x \leq 0,8; n_{\max} = 40$
7. $S = \cos x + \frac{\cos 3x}{3^2} + \dots + \frac{\cos(2n-1) \cdot x}{(2n-1)^2}$; $\pi/5 \leq x \leq \pi; n_{\max} = 40$
8. $S = \frac{x^2}{2} - \frac{x^4}{12} + \dots + (-1)^{n+1} \cdot \frac{x^{2n}}{2n \cdot (2n-1)}$; $0,1 \leq x \leq 0,8; n_{\max} = 10$
9. $S = \frac{\cos x^2}{2} + \frac{\cos 2x^2}{3} + \dots + \frac{\cos nx^2}{n+1}$; $\pi/5 \leq x \leq 9\pi/5; n_{\max} = 40$
10. $S = \sin \frac{\pi}{4} + x \cdot \sin \frac{2\pi}{4} + \dots + x^{n-1} \cdot \sin \frac{n\pi}{4}$; $0,1 \leq x \leq 0,8; n_{\max} = 40$

№ варианта	
11.	$S = \frac{x^2}{3} - \frac{x^4}{15} + \dots + (-1)^{n+1} \cdot \frac{x^{2n}}{4n^2 - 1}; \quad 0,1 \leq x \leq 1; n_{\max} = 20$
12.	$S = 1 + \frac{(2x)^2}{1!} + \dots + \frac{(2x)^{2n}}{n!}; \quad 0,1 \leq x \leq 1; n_{\max} = 20$
13.	$S = 1 - \frac{3x^2}{3!} + \frac{5x^4}{5!} + \dots + (-1)^{n-1} \cdot \frac{(2n-1) \cdot x^{2n-2}}{(2n-1)!}; \quad 0,1 \leq x \leq 0,9; n_{\max} = 20$
14.	$S = \sin x + \frac{\sin 2^2 x}{2^2} + \dots + \frac{\sin n^2 x}{n^2}; \quad \pi/4 \leq x \leq 3\pi/4; n_{\max} = 20$
15.	$S = \sin x + 2 \sin x + \dots + n! \sin x; \quad 0,2 \leq x \leq 0,8; n_{\max} = 20$
16.	$S = x \sin \frac{\pi}{4} + x^2 \sin 2 \frac{\pi}{4} + \dots + x^n \sin(n \frac{\pi}{4}); \quad 0,1 \leq x \leq 0,8; n_{\max} = 40$
17.	$S = \cos x + \frac{\cos x}{2} + \dots + \frac{\cos xn}{n^2}; \quad 0,2 \leq x \leq 0,8; n_{\max} = 20$
18.	$S = (x+2) + 4(x+2)^2 + \dots + n^2(x+2)^n; \quad 0,2 \leq x \leq 0,8; n_{\max} = 20$
19.	$S = \sin x + \frac{\sin 2x}{4} + \dots + \frac{\sin nx}{n^2}; \quad 0,2 \leq x \leq 0,8; n_{\max} = 20$
20.	$S = \operatorname{sh} x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2n+1}}{(2n+1)!}; \quad -2 \leq x \leq 0,1; n_{\max} = 40$
21.	$S = \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{(-1)^{n-1} x^n}{n}; \quad -2 \leq x \leq 0,1; n_{\max} = 30$
22.	$S = \frac{x}{2} + \frac{x^2}{4} + \dots + \left(\frac{x}{2}\right)^n; \quad -2 \leq x \leq 2; n_{\max} = 20$
23.	$S = \sin x - \frac{\sin 2x}{2} + \dots + (-1)^{n+1} \frac{\sin nx}{n}, \quad \frac{\pi}{5} \leq x \leq \frac{4\pi}{5}; n_{\max} = 40$
24.	$S = x \cos \frac{\pi}{3} + \frac{x^2 \cos 2 \frac{\pi}{3}}{2} + \dots + \frac{x^n \cos n \frac{\pi}{3}}{n}; \quad 0,1 \leq x \leq 0,8; n_{\max} = 30$
25.	$S = 1 + 3x^2 + \dots + \frac{2n+1}{n!} x^{2n}; \quad 0,1 \leq x \leq 1; n_{\max} = 10$
26.	$S = 1 + \frac{\cos \frac{\pi}{4}}{1!} x + \dots + \frac{\cos n \frac{\pi}{4}}{n!} x^n; \quad 0,1 \leq x \leq 1; n_{\max} = 25$

№ варианта
27. $S = 1 - \frac{x^2}{2!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}; 0,1 \leq x \leq 1; n_{\max} = 10$
28. $S = \frac{x^3}{3} - \frac{x^5}{15} + \dots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2 - 1}; 0,1 \leq x \leq 1; n_{\max} = 30$

2 Задача:

1. Напишите программу, которая: а) вводит с клавиатуры два непустых массива целых чисел в диапазоне от нуля до девяти и печатает их разность; б) заменить элементы массива средним арифметическим соседних элементов (крайние элементы не трогать); в) каждый элемент, кратный трем, увеличить в два раза

2. Одномерный массив А длиной N, определить: а) первый минимальный элемент массива; б) все элементы массива, предшествующие первому по порядку наименьшему числу, помножить на этот наименьший элемент; в) подсчитать количество таких элементов и определить, на каких позициях находятся эти элементы

3. Двумерный массив размером M на M: а) отразить относительно главной диагонали (левый верхний угол становится правым нижним, а правый нижний - левым верхним); б) сумму положительных элементов на главной диагонали; в) индекс первого нуля

4. Одномерный массив B[n] и преобразовать его следующим образом: а) увеличить каждый элемент массива на максимальный элемент; б) найти индекс и значение последнего минимального элемента массива; в) отсортировать элементы массива по убыванию

5. Двумерный массив размером M на N: а) заменить нулями все ее элементы, расположенные на главной диагонали; б) найти произведение элементов, расположенных на побочной диагонали; в) максимальный из элементов, расположенных на главной диагонали

6. В массиве целых чисел определить: а) количество четных и нечетных чисел; б) количество элементов вне задаваемого с клавиатуры диапазона значений; в) произведение всех положительных элементов массива

7. Двумерный массив: а) поменять местами строку, в которой расположен элемент с наибольшим значением, со строкой, содержащей элемент с наименьшим значением (предполагается, что такие элементы единственны); б) указать индексы всех элементов с наибольшим значением; в) произведение элементов, расположенных на побочной диагонали

8. Одномерный массив заполнен целыми числами. Переместить: а) нулевые элементы массива в конец, сдвинув остальные элементы влево; б) количество максимальных элементов; в) среднее арифметическое значение этих элементов

9. Даны два массива равной длины. Подсчитать количество элементов с одинаковым местоположением, которые а) равны, б) элемент первого массива больше, в) элемент второго массива больше

10. Дан массив некоторых числовых данных: а) определить, сколько раз данное число встречается в массиве; б) удалить из него все повторяющиеся элементы; в) отсортировать его (любым методом)

11. Даны одномерные массивы А и В длиной N: а) упорядоченные по возрастанию элементы массивов; б) сформировать из них новый одномерный массив С; в) найти сумму элементов массива С, имеющих нечетные номера

12. Заполнить массив числами Фибоначчи (каждое из значений получается путем сложения двух предыдущих, первый элемент равен 1, второй – 2, третий – 3, четвертый – 5,

и т.д.). Добавить к элементам массива на нечетных индексах заданное число m (с клавиатуры)

13. Дан массив: а) вывести его на экран; б) инвертировать (поменять местами 1-ий элемент с последним, 2-ой с предпоследним и т.д.) и вывести; в) подсчитать количество элементов массива, превышающих первый элемент

14. Дана матрица M : а) удалить из массива строку и столбец, на пересечении которых, расположен максимальный элемент; б) определить количество элементов, значения которых лежат в диапазоне $[y1..y2]$; в) найти их сумму

15. Целочисленный одномерный массив: а) циклически (последний элемент массива становится 1 элементом) сдвинуть элементы массива вправо на k позиций (k с клавиатуры); б) удвоить все нечетные элементы; в) заменить половинными значениями все четные элементы.

16. Дан массив положительных чисел, составить другой массив, каждый элемент которого равен сумме или произведению соседних элементов исходного массива (сумму вычислять, если она больше 15, в противном случае – произведение). Упорядочить по убыванию

17. Дан одномерный массив D и число x : а) печатает “ДА”, если x совпадает с одним из элементов D , в противном случае “НЕТ”; б) проверить, есть ли в массиве отрицательные элементы; в) поменять местами минимальный и максимальный элементы массива

18. Двумерный массив: а) отсортировать четные строки по возрастанию, а нечетные – по убыванию; б) найти, сколько раз в массиве повторяется самое частое число; в) заменить это число на a

19. Дан массив M на N : а) уменьшить все положительные элементы массива на 1, а отрицательные – увеличить на b ; б) вычислить номер максимального элемента массива; в) заменить отрицательные элементы массива их квадратами

20. Дан массив M : а) подсчитать количество элементов массива, = минимальному и максимальному; б) проверить, есть ли в массиве 0 и вывести номер первого из найденных; в) вычислить среднее арифметическое элементов массива.

21. В массиве целых чисел определить: а) номер 1 максимального элемента; б) количество элементов вне задаваемого с клавиатуры диапазона значений; в) сумму элементов превышающих число k

22. Массив M на N заполнен целыми числами: а) в строке содержащей не менее 3-х отрицательных чисел, знаки элементов на противоположные; б) отсортировать четные строки по возрастанию, а нечетные – по убыванию; в) найти разность между максимальным и минимальным элементом массива

23. Дан массив M на N : а) заполнить его произвольными числами в диапазоне до 15; б) перенести содержимое двумерного массива в одномерный массив; в) упорядочить его по возрастанию

24. Два одномерных массива: а) найти максимальный элемент из двух массивов; б) сложить элементы массивов с одинаковыми индексами; в) найти среднее арифметическое значений элементов получившегося массива

25. Дана матрица чисел, определить: а) минимальные элементы в главной диагонали; б) какая из диагоналей матрицы содержит большую сумму элементов; в) элемент в центре матрицы заменить на 0

26. Дан одномерный массив, найти а) наименьшее количество элементов, которые нужно удалить из массива, чтобы он был упорядоченным; б) количество положительных и отрицательных чисел; в) произведение первого и последнего элементов массива

27. Заполнить двумерный массив числами Фибоначчи (каждое из значений получается путем сложения двух предыдущих). Элементы главной диагонали разделить на корень числа b

28. Дан массив, найти: а) элементы массива, значение которых совпадает с их индексом; б) сколько раз в массиве повторяется самое частое число; в) сумму элементов лежащих в диапазоне [a, b].

Темы для второго вопроса контрольной работы (реферат) по вариантам

1. От C к C++. Понятие объектно-ориентированного программирования
2. Перегрузка функций (статическая)
3. Дружественные функции
4. Понятие конструктора. Использование конструкторов для инициализации вновь созданной переменной типа класс
5. Понятие деструктора. Использование деструктора
6. Понятие о перегрузках операторов. Пример перегрузки оператора
7. Конструктор копий
8. Перегрузка конструктора
9. Понятие области видимости класс и прав доступа (public, private, protected)
10. Указатель this
11. Механизм наследования. Примеры
12. Множественное наследование
13. Виртуальные базовые классы
14. Понятие полиморфизма, механизм. Примеры
15. Виртуальные функции
16. Чистые виртуальные функции
17. Перегрузка операторов (+, -, [])
18. Перегрузка операторов (++, --, [])