Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 31.05.2024 13:44:14
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет машиностроения

УТВЕРЖДАЮ
Декан факультета машиностроения
_____ /Е.В. Сафонов/
«15» февраля 2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Программирование и основы алгоритмизации»

Направление подготовки
27.03.04.«Управление в технических системах»

Образовательная программа (профиль подготовки)
«Электронные системы управления»

Квалификация (степень) выпускника
Бакалавр

Форма обучения
Очная

Москва, 2024 г.

Разработчик(и):

ст. преподаватель_____ Н.В. Груненков

ст. преподаватель_____ Е.В. Пикалов

Согласовано:

Заведующий кафедрой «Автоматика и управление»,
д.т.н., проф. /А.А. Радионов/

Содержание

1. Цели, задачи и планируемые результаты обучения по дисциплине ... 4
2. Место дисциплины в структуре образовательной программы ... 4
3. Структура и содержание дисциплины ... 4
   - 3.1. Виды учебной работы и трудоемкость ... 4
   - 3.2. Тематический план изучения дисциплины ... 5
   - 3.3. Содержание дисциплины ... 6
   - 3.4. Тематика семинарских/практических и лабораторных занятий ... 7
   - 3.5. Тематика курсовых проектов (курсовых работ) ... 9
4. Учебно-методическое и информационное обеспечение ... 9
   - 4.1. Нормативные документы и ГОСТы ... 9
   - 4.2. Основная литература ... 9
   - 4.3. Дополнительная литература ... 10
   - 4.4. Электронные образовательные ресурсы ... 10
   - 4.5. Лицензионное и свободно распространяемое программное обеспечение ... 10
   - 4.6. Современные профессиональные базы данных и информационные справочные системы ... 11
5. Материально-техническое обеспечение ... 12
6. Методические рекомендации ... 12
   - 6.1. Методические рекомендации для преподавателя по организации обучения ... 12
   - 6.2. Методические указания для обучающихся по освоению дисциплины ... 13
7. Фонд оценочных средств ... 14
   - 7.1. Методы контроля и оценивания результатов обучения ... 15
   - 7.2. Шкала и критерии оценивания результатов обучения ... 15
   - 7.3. Оценочные средства ... 17

1. Цели, задачи и планируемые результаты обучения по дисциплине

Целью освоения дисциплины «Программирование и основы алгоритмизации» является формирование у студентов знаний в области разработки и проектирования программного обеспечения систем автоматизации и управления.

Задачи дисциплины: основной задачей материала является овладение научной терминологией в области проектирования и использования программного обеспечения в системах автоматизации и управления техническими объектами.

Обучение по дисциплине «Программирование и основы алгоритмизации» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
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	ИОПК -6.1. Знает структуры данных, используемые для представления типовых информационных объектов; типовые алгоритмы обработки данных. ИОПК -6.2. Умеет разрабатывать алгоритмы решения задач применительно к процессам и системам управления техническими (технологическими) объектами; использовать язык программирования для создания программы. ИОПК -6.3. Владеет основными технологиями программирования навыками чтения и составления технической документации на программный продукт.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование и основы алгоритмизации» относится к дисциплинам базовой части (Блока 1) Б1.1.14. основной образовательной программы бакалавриата.

Дисциплина базируется на следующих, пройденных дисциплинах:

- «Высшая математика»;
- «Компьютерные технологии в управлении техническими системами»;
- «Компьютерные системы обработки экспериментальных данных»;
- «Моделирование систем управления».

3. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 9 зачетных(е) единиц(ы) (324 часов).

3.1 Виды учебной работы и трудоемкость

3.1.1. Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			3 семестр	4 семестр
1	**Аудиторные занятия**	**144**	72	72
	В том числе:			
1.1	Лекции	36	18	18
1.2	Семинарские/практические занятия	36	18	18
1.3	Лабораторные занятия	72	36	36
2	**Самостоятельная работа**	**180**	90	90
	В том числе:			
2.1	Подготовка и защита лабораторных работ	72	36	36
2.2	Самостоятельное изучение	108	54	54
3	**Промежуточная аттестация**			
	Зачет/диф.зачет/экзамен		зачет	экзамен
	Итого			

3.2 Тематический план изучения дисциплины

(по формам обучения)

3.2.1. Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	**Раздел 1. Введение в алгоритмизацию**		**2**				**6**
2	**Раздел 2. Основные алгоритмические структуры**		**2**				**6**
3	**Раздел 3. Типовые приемы алгоритмизации**		**2**				**6**
4	**Раздел 4. Методы разработки алгоритмов**		**2**		**2**		**4**
5	**Раздел 5. Анализ сложности алгоритмов**		**2**	**4**	**6**		**12**
6	**Раздел 6. Введение в программирование**		**2**				**6**
7	**Раздел 7. Состав языка программирования**		**2**				**6**
8	**Раздел 8. Структуры данных**		**2**				**6**
9	**Раздел 9. Технология программирования**		**2**	2	**4**		**8**
10	**Раздел 10. Введение в язык программирования С++**		**2**		**2**		**4**

11	**Раздел 11. Типы данных и их особенности в С++**		**2**	4	**8**		**16**
12	**Раздел 12. Операторы в языке С++**		**2**	2	**4**		**8**
13	**Раздел 13. Функции в языке С++**		**2**	2	**4**		**8**
14	**Раздел 14. Стандартные управляющие конструкции языка С++**		**2**	2	**4**		**8**
15	**Раздел 15. Линейные структуры данных в языке С++**		**2**	4	**8**		**16**
16	**Раздел 16. Операторы цикла в языке С++**		**2**	4	**6**		**12**
17	**Раздел 17. Указатели, арифметика указателей в языке С++**		**2**	4	**8**		**16**
18	**Раздел 18. Введение в ООП**		**2**	8	**16**		**32**
Итого			**36**	**36**	**72**		**180**

3.3 Содержание дисциплины

Раздел 1. Введение в алгоритмизацию

В данном разделе раскрываются следующие темы: понятие алгоритма и его свойства; исполнители алгоритмов; способы описания алгоритмов; классификация алгоритмов.

Раздел 2. Основные алгоритмические структуры

Данный раздел содержит основные сведения об алгоритмах и подразделяется на следующие темы: понятие структурированного алгоритма; линейные алгоритмы; разветвляющаяся алгоритмическая конструкция; циклические алгоритмы.

Раздел 3. Типовые приемы алгоритмизации

В данном разделе изучаются стандартные методы решения алгоритмических задач, такие как: вычисление суммы и произведения; рекурсивная алгоритмическая конструкция; вычисление бесконечной суммы с заданной точностью; подсчет количества элементов; табулирование функций; вложенные циклы.

Раздел 4. Методы разработки алгоритмов

Четвертый раздел содержит информацию об основных методах, при помощи которых разрабатываются алгоритмы: критерии выбора алгоритма решения задачи; основные принципы и способы разработки алгоритмов; методы построения алгоритмов.

Раздел 5. Анализ сложности алгоритмов

Данный раздел содержит описание понятия, анализа сложности алгоритмов, и соответствующую классификацию: понятие сложности алгоритмов; асимптотический анализ алгоритмов; классы сложности алгоритмов; классификация вычислительных задач.

Раздел 6. Введение в программирование

В разделе изучаются современные технологии программирования: понятия «программирование», «программа», «язык программирования»; эволюция языков программирования; классификация языков программирования; парадигма программирования; интегрированная среда разработки.

Раздел 7. Состав языка программирования

В разделе затрагиваются вопросы, связанные с стандартным содержанием практически каждого языка программирования: описание состава языка программирования; алфавит языка; лексемы; выражения; операторы.

Раздел 8. Структуры данных

В данном разделе содержатся сведения о различных структурах данных в программировании: понятие «структура данных»; логическое и физическое представление данных; структуры хранения данных; операции над структурами данных.

Раздел 9. Технология программирования

В данном разделе содержатся сведения об основных технологиях, применяемых при программировании: понятие технологии программирования; основные этапы развития технологии программирования; стадии разработки программы.

Раздел 10. Введение в язык программирования C++

В данном разделе приводится краткая историческая справка касательно появления и развития этого языка; изучаются общие сведения о языке C++, необходимые для ознакомления с его структурой и свойствами.

Раздел 11. Типы данных и их особенности в C++

В этом разделе предлагается для изучения базовые типы данных, применяемые в языке, их особенности и возможности применения; также приводятся простые примеры их объявления, инициализации и использования; затрагиваются более сложные типы данных, такие как структуры и строки.

Раздел 12. Операторы в языке C++

Данный раздел посвящён изучению базовых операторов языка, принципам их вызова и применения; использования соответствующих операндов; рассматриваются способы избегания типичных ошибок на простых примерах.

Раздел 13. Функции в языке C++

Этот раздел включает в себя такие темы, как: определение и основные понятия касательно функции в языке C++; объявление и вызов функций; параметры и аргументы функций; преимущества и особенности использования функций.

Раздел 14. Стандартные управляющие конструкции языка C++

В данном разделе освещаются следующие темы: необходимость и особенности применения управляющих конструкций языка; виды управляющих конструкций; особенности и различия конструкций if-else и switch-case.

Раздел 15. Линейные структуры данных в языке C++

Раздел посвящён изучению особенностей хранения и организации структур данных при работе с языком C++; отдельное внимание уделяется линейным структурам данных, методам работы и обращения к ним.

Раздел 16. Операторы цикла в языке C++

В этом разделе приводится подробное описание циклов в C++, их различия и особенности; на примерах разбираются условия и возможности их применения в реальных задачах.

Раздел 17. Указатели в языке C++

Данный раздел раскрывает такие темы как: структура и принципы работы указателей, особенности их объявления; особенности работы с указателями, область их применения; арифметика указателей.

Раздел 18. Подключаемые файлы, организация структуры проекта

В этом разделе предлагаются для изучения основы объектно-ориентированного программирования. Рассматриваются такие темы как: классы, инкапсуляция, наследование, полиморфизм, абстракция, конструкторы, деструкторы, списки инициализации и другие, связанные с ООП.

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1. Семинарские/практические занятия

Практическая работа 1. Защита лабораторной работы №1,2
Практическая работа 2. Защита лабораторной работы №3,4
Практическая работа 3. Защита лабораторной работы №5,6
Практическая работа 4. Защита лабораторной работы №7,8
Практическая работа 5. Защита лабораторной работы №9,10

Практическая работа 6. Защита лабораторной работы №11,12
Практическая работа 7. Защита лабораторной работы №13,14
Практическая работа 8. Защита лабораторной работы №15,16
Практическая работа 9. Защита лабораторной работы №17,18
Практическая работа 10. Защита лабораторной работы №19,20
Практическая работа 11. Защита лабораторной работы №21,22
Практическая работа 12. Защита лабораторной работы №23,24
Практическая работа 13. Защита лабораторной работы №25,26
Практическая работа 14. Защита лабораторной работы №27,28
Практическая работа 15. Защита лабораторной работы №29,30
Практическая работа 16. Защита лабораторной работы №31,32
Практическая работа 17. Защита лабораторной работы №33,34
Практическая работа 18. Защита лабораторной работы №35,36

3.4.2. Лабораторные занятия

Лабораторная работа 1. Составление блок-схемы алгоритма.
Лабораторная работа 2. Форматированный ввод-вывод в C++.
Лабораторная работа 3. Функции.
Лабораторная работа 4. Базовые типы данных.
Лабораторная работа 5. Введение в std::string.
Лабораторная работа 6. Перечисления.
Лабораторная работа 7. Оператор goto, бесконечные циклы, цикл while.
Лабораторная работа 8. Циклы for, do … while.
Лабораторная работа 9. Операторы break и continue.
Лабораторная работа 10. Генерация случайных чисел.
Лабораторная работа 11. Массивы.
Лабораторная работа 12. Массивы и циклы.
Лабораторная работа 13. Сортировка массива методом выбора.
Лабораторная работа 14. Многомерные массивы.
Лабораторная работа 15. Указатели.
Лабораторная работа 16. Взаимосвязь указателей и массивов.
Лабораторная работа 17. Динамическое выделение памяти.
Лабораторная работа 18. Динамические массивы.
Лабораторная работа 19. Ссылки.
Лабораторная работа 20. Цикл foreach.
Лабораторная работа 21. Указатели типа void.
Лабораторная работа 22. Введение в std::array.
Лабораторная работа 23. Введение в итераторы в C++.
Лабораторная работа 24. Введение в ООП.
Лабораторная работа 25. Спецификаторы доступа классов.
Лабораторная работа 26. Инкапсуляция, геттеры и сеттеры.
Лабораторная работа 27. Конструкторы классов.
Лабораторная работа 28. Списки инициализации экземпляров класса.
Лабораторная работа 29. Инициализация нестатических членов класса.
Лабораторная работа 30. Делегирующие конструкторы, деструкторы.
Лабораторная работа 31. Скрытый указатель *this.
Лабораторная работа 32. Оценка сложности программ на основе метрики Маккейба.
Лабораторная работа 33. Измерение характеристик программ системой метрик Холстеда.
Лабораторная работа 34. Анализ программы.
Лабораторная работа 35. Программные контексты: переменные.
Лабораторная работа 36. Программные контексты: элементы массива, индексы.

3.5 Тематика курсовых проектов (курсовых работ)

Курсовые работы/проекты присутствуют

Темы курсовых работ:

1. Алгоритмы поиска в тексте
2. Очередь и стек
3. Работа с файловой системой
4. Сортировка одномерных массивов
5. Обработка строк
6. Тамагочи
7. Бинарные деревья
8. Алгоритмы поиска в линейных структурах
9. Односвязные и двусвязные списки
10. Моделирование работы банка
11. Графы
12. Моделирование учёта товара в аптеке
13. Двумерная графика
14. Мессенджер через веб-сокеты
15. Уравновешивание химический уравнений
16. Криптография
17. Сжатие, хэширование и сверка ключей
18. Передача файлов через веб-сокеты
19. Булева алгебра
20. Многофункциональный калькулятор
21. Статистическая обработка данных
22. Аппроксимация, интерполяция, экстраполяция
23. Моделирование работы лифтов
24. Минимизация булевых функций
25. Калах (игра)

4. Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

Не предусмотрено

4.2 Основная литература

1. Конова, Е.А. Алгоритмы и программы. Язык C++. [Электронный ресурс] / Е.А. Конова, Г.А. Поллак. — Электрон. дан. — СПб.: Лань, 2017. — 384 с. — Режим доступа: http://e.lanbook.com/book/90158 — Загл. с экрана.

2. Окулов, С.М. Задачи по программированию. [Электронный ресурс] / С.М. Окулов, Т.В. Ашихмина, Н.А. Бушмелева, М.А. Корчёмкин. — Электрон. дан. — М.: Издательство "Лаборатория знаний", 2017. — 826 с. — Режим доступа: http://e.lanbook.com/book/94162 — Загл. с экрана.

3. Павловская Т.А. C/C++. Структурное и объектно-ориентированное программирование: практикум /Т.А.Павловская, Ю.А.Щупакт - СПб.: Питер, 2011.-352с.:ил. - (Учебное пособие)

4. Гагарина Л. Г., Дорогова Е. Г. Основы программирования на языке С. Учебное пособие. – М.: Издательство: ИНФРА-М, 2021 г.

4.3 Дополнительная литература

1. Алгоритмизация задач. Массивы. [Электронный ресурс] — Электрон. дан. — М.: МГТУ им. Н.Э. Баумана, 2009. — 24 с. — Режим доступа: http://e.lanbook.com/book/52572 — Загл. с экрана.

2. Долгов, А.И. Алгоритмизация прикладных задач. [Электронный ресурс] — Электрон. дан. — М.: ФЛИНТА, 2016. — 136 с. — Режим доступа: http://e.lanbook.com/book/85872 — Загл. с экрана.

4.4 Электронные образовательные ресурсы

Проведение занятий и аттестаций возможно в дистанционном формате с применением системы дистанционного обучения университета (СДО-LMS) на основе разработанных кафедрой электронных образовательных ресурсов (ЭОР) по всем разделам программы:

Название ЭОР	
Основы программирования и алгоритмизации в машиностроении (часть 2)	https://online.mospolytech.ru/course/view.php?id=3575
Основы программирования и алгоритмизации в машиностроении (часть 2)	https://online.mospolytech.ru/course/view.php?id=4165

Разработанные ЭОР включают тренировочные и итоговые тесты.

Порядок проведения работ в дистанционном формате устанавливается отдельными распоряжениями проректора по учебной работе и/или центром учебно-методической работы.

4.5 Лицензионное и свободно распространяемое программное обеспечение

№	Наименование	Разработчик ПО (правообладатель)	Доступность (лицензионное, свободно распространяемое)	Ссылка на Единый реестр российских программ для ЭВМ и БД (при наличии)
1	Code::Blocks 20.03	The Code::Blocks team	Свободно распространяемое	

4.6 Современные профессиональные базы данных и информационные справочные системы

Перечень ресурсов сети Интернет, доступных для освоения дисциплины:

№	Наименование	Ссылка на ресурс	Доступность
Информационно-справочные системы			
	Stack Overflow	https://stackoverflow.com/	Доступна в сети Интернет без ограничений
	Онлайн-справочник по языку C++ и стандартным библиотекам	https://en.cppreference.com/w/	Доступна в сети Интернет без ограничений
	Онлайн-справочник по языку C++ и стандартным библиотекам Microsoft	https://learn.microsoft.com/en-us/cpp/?view=msvc-170	Доступна в сети Интернет без ограничений
Электронно-библиотечные системы			
	Лань	https://e.lanbook.com/	Доступна в сети Интернет без ограничений
	IPR Books	https://www.iprbookshop.ru/	Доступна в сети Интернет без ограничений
Профессиональные базы данных			
	База данных научной электронной библиотеки (eLIBRARY.RU)	http://www.elibrary.ru	Доступно
	Web of Science Core Collection – политематическая реферативно-библиографическая и наукометрическая (библиометрическая) база данных	http://webofscience.com	Доступно
	#SimpleCode	https://www.youtube.com/@SimpleCodeIT	Доступно
	Личный канал YouTube: Гоша Дударь	https://www.youtube.com/@gosha_dudar	Доступно

5. Материально-техническое обеспечение

Для проведения лекционных занятий необходимы аудитории, оснащенные мультимедийными проекторами и экранами. Для проведения лабораторных работ требуется компьютерный класс (АВ2507, АВ2614, АВ2618, АВ2619)

6. Методические рекомендации

Методика преподавания дисциплины «Программирование и основы алгоритмизации» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения аудиторных и внеаудиторных занятий:

– аудиторные занятия: лекции, практические работы, лабораторные работы, тестирование;

– внеаудиторные занятия: самостоятельное изучение отдельных вопросов, подготовка к лабораторным работам.

Образовательные технологии

Возможно проведение занятий и аттестаций в дистанционном формате с применением системы дистанционного обучения университета (СДО-LMS) на основе разработанных кафедрой «Автоматика и управление» электронных образовательных ресурсов (ЭОР) (см. п.4.4).

Порядок проведения работ в дистанционном формате устанавливается отдельными распоряжениями проректора по учебной работе и/или центром учебно-методической работы.

6.1 Методические рекомендации для преподавателя по организации обучения

6.1.1. Преподаватель организует преподавание дисциплины в соответствии с требованиями "Положения об организации образовательного процесса в московском политехническом университете и его филиалах", утверждённым ректором университета.

6.1.2. На первом занятии преподаватель доводит до сведения студентов содержание рабочей программы дисциплины (РПД) и предоставляет возможность ознакомления с программой.

6.1.3. Преподаватель особенно обращает внимание студентов на:

- виды и формы проведения занятий по дисциплине, включая порядок проведения занятий с применением технологий дистанционного обучения и системы дистанционного обучения университета (СДО мосполитеха);
- виды, содержание и порядок проведения текущего контроля успеваемости в соответствии с фондом оценочных средств;
- форму, содержание и порядок проведения промежуточной аттестации в соответствии с фондом оценочных средств, предусмотренным РПД.

6.1.4. Доводит до сведения студентов график выполнения учебных работ, предусмотренных РПД.

6.1.5. Необходимо с самого начала занятий рекомендовать студентам основную и дополнительную литературу и указать пути доступа к ней.

6.1.6. В начале или в конце семестра дать список вопросов для подготовки к промежуточной аттестации (экзамену или зачёту).

6.1.7. Рекомендуется факт ознакомления студентов с РПД и графиком работы письменно зафиксировать подписью студента в листе ознакомления с содержание РПД.

6.1.8. Преподаватели, ведущий лекционные и практические занятия, должны согласовывать тематический план практических занятий, использовать единую систему обозначений, терминов, основных понятий дисциплины.

6.1.9. При подготовке **к семинарскому занятию** по перечню объявленных тем преподавателю необходимо уточнить план их проведения, продумать формулировки и содержание учебных вопросов, выносимых на обсуждение, ознакомиться с перечнем вопросов по теме семинара.

В ходе семинара во вступительном слове раскрыть практическую значимость темы семинарского занятия, определить порядок его проведения, время на обсуждение каждого учебного вопроса. Применяя фронтальный опрос дать возможность выступить всем студентам, присутствующим на занятии.

В заключительной части семинарского занятия следует подвести его итоги: дать оценку выступлений каждого студента и учебной группы в целом. Раскрыть положительные стороны и недостатки проведенного семинарского занятия. Ответить на вопросы студентов. Выдать задания для самостоятельной работы по подготовке к следующему занятию.

6.1.10. Целесообразно в ходе защиты **лабораторных работ** задавать выступающим и аудитории дополнительные и уточняющие вопросы с целью выяснения их позиций по существу обсуждаемых проблем.

Возможно проведение занятий и аттестаций в дистанционном формате с применением системы дистанционного обучения университета (СДО-LMS). Порядок проведения работ в дистанционном формате устанавливается отдельными распоряжениями проректора по учебной работе и/или центром учебно-методической работы.

6.2 Методические указания для обучающихся по освоению дисциплины

1.2.1. Студент с самого начала освоения дисциплины должен внимательно ознакомится с рабочей программой дисциплины.

1.2.2. Студенту необходимо составить для себя график выполнения учебных работ, предусмотренных РПД с учётом требований других дисциплин, изучаемых в текущем семестре.

1.2.3. При проведении занятий и процедур текущей и промежуточной аттестации с использованием инструментов информационной образовательной среды дистанционного образования университета (LMS мосполитеха), как во время контактной работы с преподавателем так и во время самостоятельной работы студент должен обеспечить техническую возможность дистанционного подключения к системам дистанционного обучения. При отсутствии такой возможности обсудить ситуацию с преподавателем дисциплины.

1.2.4. Самостоятельная работа является одним из видов учебных занятий. Цель самостоятельной работы – практическое усвоение студентами вопросов, рассматриваемых в процессе изучения дисциплины.

Виды внеаудиторной самостоятельной работы:

- самостоятельное изучение отдельных тем дисциплины;
- подготовка к лекционным занятиям;
- подготовка к семинарам и практическим занятиям;
- оформление отчетов по выполненным лабораторным работам и подготовка к их защите.

Для выполнения любого вида самостоятельной работы необходимо пройти следующие этапы:

- определение цели самостоятельной работы;
- конкретизация познавательной задачи;

- самооценка готовности к самостоятельной работе;
- выбор адекватного способа действия, ведущего к решению задачи;
- планирование работы (самостоятельной или с помощью преподавателя) над заданием;
- осуществление в процессе выполнения самостоятельной работы самоконтроля (промежуточного и конечного) результатов работы и корректировка выполнения работы;
- рефлексия;
- презентация самостоятельной работы или защита лабораторной работы.

7. Фонд оценочных средств

Фонд оценочных средств представлен в Приложении 2 к рабочей программе и включает разделы:

7.1 Методы контроля и оценивания результатов обучения
7.2 Шкала и критерии оценивания результатов обучения
7.3 Оценочные средства
7.3.1 Текущий контроль
7.3.1.1 Тестовые вопросы
7.3.1.2 Вопросы к защите лабораторных работ
7.3.2 Типовые задания на курсовую работу
7.3.2.1 Вопросы для защиты курсовой работы
7.3.3 Промежуточная аттестация

7. Фонд оценочных средств

В процессе обучения в течение семестра используются оценочные средства текущего контроля успеваемости и промежуточных аттестаций. Применяются следующие оценочные средства: тест, защита практический работ, защита лабораторных работ, зачет.

Обучение по дисциплине «Программирование и основы алгоритмизации» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
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	ИОПК -6.1. Знает структуры данных, используемые для представления типовых информационных объектов; типовые алгоритмы обработки данных. ИОПК -6.2. Умеет разрабатывать алгоритмы решения задач применительно к процессам и системам управления техническими (технологическими) объектами; использовать язык программирования для создания программы. ИОПК -6.3. Владеет основными технологиями программирования навыками чтения и составления технической документации на программный продукт.

7.1 Методы контроля и оценивания результатов обучения

№ ОС	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Тест (Т)	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
2	ЗЛР	Средство проверки умений и навыков применять полученные знания для решения практических задач с помощью инструментальных средств.	Задания для защиты лабораторных работ
3	Курсовая работа (КР)	Средство проверки умений применять полученные знания по заранее определенной методике для решения задач или заданий по модулю или дисциплине в целом	Тематики заданий для выполнения курсовой работы

7.2 Шкала и критерии оценивания результатов обучения

Форма промежуточной аттестации: зачет.

Обязательными условиями подготовки студента к промежуточной аттестации является выполнение и защита студентом лабораторных работ, предусмотренных рабочей программой и прохождение всех промежуточных тестов не ниже, чем на 70% правильных

ответов. Промежуточные тестирования могут проводится как в аудитории Университета под контролем преподавателя, так и дистанционном формате на усмотрение преподавателя.

Шкала оценивания	**Описание**
Зачтено	Выполнены все виды учебной работы, предусмотренные рабочей программой. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Не зачтено	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Форма промежуточной аттестации: экзамен.

Обязательными условиями подготовки студента к промежуточной аттестации является выполнение и защита студентом лабораторных работ, предусмотренных рабочей программой и прохождение всех промежуточных тестов не ниже, чем на 70% правильных ответов. Промежуточные тестирования могут проводится как в аудитории Университета под контролем преподавателя, так и дистанционном формате на усмотрение преподавателя.

Шкала оценивания	**Описание**
Отлично	Выполнены все виды учебной работы, предусмотренные рабочей программой. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки,

	неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7.3 Оценочные средства

7.3.1 Текущий контроль

7.3.1.1 Тестовые вопросы

Текущий контроль выполняется с применением Банка тестовых вопросов (частично). Примеры тестов представлены ниже. Для подготовки к тестированию и защите лабораторных работ в разделе 3.7.1.1 приведён перечень контрольных вопросов. Результаты текущего контроля успешно зачитываются, если при тестировании набрано не менее 70 баллов из 100 возможных.

Установите соответствие между языками программирования и его характеристиками:			*MAT*
		Балл по умолчанию:	1
		Перемешать:	Да
		Показать количество правильных ответов после окончания:	Да
		Штраф за каждую неправильную попытку:	33.3
		ID-номер:	
#	**Вопрос**	**Ответ**	
1.	машинный язык	ориентирован на использование в конкретной ЭВМ	
2.	язык высокого уровня	используется для создания прикладных программ любого типа	
3.	специализированный язык программирования	оперирует понятиями своей предметной области	
4.	язык поддержки сложных структур данных	ориентирован на специализированные области применения	
	Общий отзыв к вопросу:		
	Для любого правильного ответа:	Ваш ответ верный.	
	Для любого неправильного ответа:	Ваш ответ неправильный.	
	Для любого частично правильного ответа:	Ваш ответ частично правильный.	
	Подсказка 1:		
	Показать количество правильных ответов (Подсказка 1):	Нет	
	Удалить некорректные ответы (Подсказка 1):	Нет	
	Теги:		
Вы должны указать по меньшей мере два вопроса и три ответа. Вы можете включить дополнительные неправильные ответы, создав ответ на пустой вопрос. Записи, где и вопрос и ответ пустые, будут игнорироваться			

Отметьте свойства, присущие алгоритму:			*MA*
		Балл по умолчанию:	1
		Случайный порядок ответов	Да
		Нумеровать варианты ответов?	a
		Показать количество правильных ответов после окончания:	Да
		Штраф за каждую неправильную попытку:	33.3
		ID-номер:	
#	**Ответы**	**Отзыв**	**Оценка**
A.	дискретность		*20*
B.	аморфность		*0*
C.	детерминированность		*20*
D.	понятность		*20*
E.	целенаправленность		*0*
F.	результативность		*20*
G.	массовость		*20*
H.	многозначность		*0*
	Общий отзыв к вопросу:		
	Для любого правильного ответа:	Ваш ответ верный.	
	Для любого неправильного ответа:	Ваш ответ неправильный.	
	Для любого частично правильного ответа:	Ваш ответ частично правильный.	
	Подсказка 1:		
	Показать количество правильных ответов (Подсказка 1):	Нет	
	Удалить некорректные ответы (Подсказка 1):	Нет	
	Теги:		
Позволяет выбирать один или несколько правильных ответов из заданного списка. (MC/MA)			

Алгоритмом называется			*MC*
		Балл по умолчанию:	1
		Случайный порядок ответов	Да
		Нумеровать варианты ответов?	0
		Штраф за каждую неправильную попытку:	33.3
		ID-номер:	
#	**Ответы**	**Отзыв**	**Оценка**
A.	подробный перечень правил выполнения определенных действий		*0*
B.	ориентированный граф, указывающий порядок исполнения некоторого набора команд		*0*
C.	последовательность команд для компьютера		*0*
D.	описание последовательности действий в виде геометрических фигур, соединенных линиями и стрелками		*0*
E.	конечный набор предписаний для получения решения задачи посредством конечного количества операций		*100*
	Общий отзыв к вопросу:		
	Для любого правильного ответа:	Ваш ответ верный.	
	Для любого неправильного ответа:	Ваш ответ неправильный.	
	Подсказка 1:		
	Показать количество правильных ответов (Подсказка 1):	Нет	
	Удалить некорректные ответы (Подсказка 1):	Нет	
	Теги:		
Позволяет выбирать один или несколько правильных ответов из заданного списка. (MC/MA)			

Алгоритм должен состоять из отдельных шагов. Это свойство называется …			*МС*
		Балл по умолчанию:	1
		Случайный порядок ответов	Да
		Нумеровать варианты ответов?	0
		Штраф за каждую неправильную попытку:	33.3
		ID-номер:	
#	**Ответы**	**Отзыв**	**Оценка**
A.	понятность		*0*
B.	массовость		*0*
C.	однозначность		*0*
D.	дискретность		*100*
	Общий отзыв к вопросу:		
	Для любого правильного ответа:	Ваш ответ верный.	
	Для любого неправильного ответа:	Ваш ответ неправильный.	
	Подсказка 1:		
	Показать количество правильных ответов (Подсказка 1):	Нет	
	Удалить некорректные ответы (Подсказка 1):	Нет	
	Теги:		
Позволяет выбирать один или несколько правильных ответов из заданного списка. (МС/МА)			

Детерминированный алгоритм – это алгоритм, который …			*MC*
		Балл по умолчанию:	1
		Случайный порядок ответов	Да
		Нумеровать варианты ответов?	0
		Штраф за каждую неправильную попытку:	33.3
		ID-номер:	
#	**Ответы**	**Отзыв**	**Оценка**
A.	всегда обеспечивает регулярное решение и характеризуется отсутствием элементов, вносящих неопределенность, невозможностью произвольности в выборе решений, определяющих последовательность действий, недопустимостью применения методов проб и ошибок		*100*
B.	описывает систематическую процедуру поиска нужного решения среди всех возможных и реализован на методе проб и ошибок, повторов или случайного выбора		*0*
C.	позволяет проверить полученное решение легко на правильность, но поиск решения представляет трудности		*0*
	Общий отзыв к вопросу:		
	Для любого правильного ответа:	Ваш ответ верный.	
	Для любого неправильного ответа:	Ваш ответ неправильный.	
	Подсказка 1:		
	Показать количество правильных ответов (Подсказка 1):	Нет	
	Удалить некорректные ответы (Подсказка 1):	Нет	
	Теги:		
Позволяет выбирать один или несколько правильных ответов из заданного списка. (MC/MA)			

Метод отрабатывания назад			*MC*
		Балл по умолчанию:	1
		Случайный порядок ответов	Да
		Нумеровать варианты ответов?	0
		Штраф за каждую неправильную попытку:	33.3
		ID-номер:	
#	**Ответы**	**Отзыв**	**Оценка**
A.	предполагает принятие начального предположения или вычисление начального решения задачи		*0*
B.	предполагает разбиение исходной большой сложной задачи на более мелкие простые задачи		*0*
C.	позволяет решать задачи, комбинируя решения вспомогательных подзадач		*0*
D.	основан на предположении, что задача уже решена и состоит в том, что последовательно определяются условия, при которых это решение может быть получено		*100*
	Общий отзыв к вопросу:		
	Для любого правильного ответа:	Ваш ответ верный.	
	Для любого неправильного ответа:	Ваш ответ неправильный.	
	Подсказка 1:		
	Показать количество правильных ответов (Подсказка 1):	Нет	
	Удалить некорректные ответы (Подсказка 1):	Нет	
	Теги:		
Позволяет выбирать один или несколько правильных ответов из заданного списка. (MC/MA)			

Исполнитель алгоритма способен исполнять некоторый класс алгоритмов			*МС*
		Балл по умолчанию:	1
		Случайный порядок ответов	Да
		Нумеровать варианты ответов?	0
		Штраф за каждую неправильную попытку:	33.3
		ID-номер:	
#	**Ответы**	**Отзыв**	**Оценка**
A.	покомандно или программно		*100*
B.	только покомандно		*0*
C.	только программно		*0*
	Общий отзыв к вопросу:		
	Для любого правильного ответа:	Ваш ответ верный.	
	Для любого неправильного ответа:	Ваш ответ неправильный.	
	Подсказка 1:		
	Показать количество правильных ответов (Подсказка 1):	Нет	
	Удалить некорректные ответы (Подсказка 1):	Нет	
	Теги:		
Позволяет выбирать один или несколько правильных ответов из заданного списка. (МС/МА)			

	Свойство алгоритма «дискретность» означает		*MC*
		Балл по умолчанию:	1
		Случайный порядок ответов	Да
		Нумеровать варианты ответов?	0
		Штраф за каждую неправильную попытку:	33.3
		ID-номер:	
#	**Ответы**	**Отзыв**	**Оценка**
A.	алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа		*0*
B.	при точном исполнении всех команд алгоритма процесс должен приводить к определенному результату		*0*
C.	алгоритм должен состоять из последовательности конечного числа шагов		*100*
D.	алгоритм должен быть ориентирован на конкретного исполнителя и содержать команды, входящие в систему его команд		*0*
E.	исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма		*0*
	Общий отзыв к вопросу:		
	Для любого правильного ответа:	Ваш ответ верный.	
	Для любого неправильного ответа:	Ваш ответ неправильный.	
	Подсказка 1:		
	Показать количество правильных ответов (Подсказка 1):	Нет	
	Удалить некорректные ответы (Подсказка 1):	Нет	
	Теги:		

Свойство алгоритма «дискретность» означает			*МС*
		Балл по умолчанию:	1
		Случайный порядок ответов	Да
		Нумеровать варианты ответов?	0
		Штраф за каждую неправильную попытку:	33.3
		ID-номер:	
#	**Ответы**	**Отзыв**	**Оценка**
Позволяет выбирать один или несколько правильных ответов из заданного списка. (МС/МА)			

7.3.1.2 Вопросы для защиты лабораторных работ

1. Дайте определение понятию «алгоритм».
2. Дайте определение понятию «блок-схема».
3. Назовите свойства алгоритма.
4. Какие способы представления алгоритмов вы знаете?
5. Какие уровни языков программирования вы знаете?
6. Какие базовые типы данных вы знаете?
7. Какие данные может хранить переменная типа string?
8. Сколько занимает в памяти переменная типа bool?
9. Назовите особенности линейных алгоритмов.
10. В чём состоит особенность циклических алгоритмов?
11. Опишите принцип работы цикла while.
12. Опишите принцип работы цикла do … while.
13. Опишите принцип работы цикла for.
14. В чём заключается особенность цикла foreach?
15. Что называется оператором?
16. Назовите отличие унарных операций от бинарных.
17. Назовите пример унарной операции?
18. Назовите пример бинарной операции?
19. Что называется операндом?
20. Что такое компиляция программы?
21. Что такое трансляция программы?
22. Что такое линковка программы?
23. На каком этапе к программе добавляются библиотечные функции?
24. Что такое ветвящийся алгоритм?
25. Дайте определение понятию «массив».
26. Является ли массив линейной структурой данных?
27. Каким образом адресуются элементы в массивах?
28. Дайте определение понятию «указатель».
29. Сколько байт в памяти компьютера занимает указатель?
30. В чём сходство между указателями и массивами?
31. Чем полезны вложенные циклы?
32. Дайте определение понятию «функция»?
33. Что называется параметром функции?
34. Что такое аргумент функции?

35. Что называется «висячим» указателем?
36. Дайте определение понятию «класс»?
37. В чём заключается разница между структурой и классом?
38. Как называется метод, автоматически вызываемый при создании экземпляра класса?
39. Что такое заголовочный файл?
40. Назовите преимущества инкапсуляции.

7.3.2 Типовые задания на курсовую работу

Курсовая работа как элемент учебной дисциплины должна способствовать формированию компетенции ОПК-6.

Целью выполнения курсовой работы является закрепление теоретических сведений по программированию и алгоритмизации, обеспечивающее на должном уровне понимание обучающимся принципов построения алгоритмов и написания программ общего назначения.

Исходными данными для выполнения курсовой работы являются задания для написания консольных приложений на языке программирования C++.

Задание на курсовую работу формулируется следующим образом:

1) Подготовить список выполняемых программой команд, реализуемых функций, и необходимых для их корректной работы параметров, переменных и иных структур данных.

2) Разработать блок-схемы алгоритмов программы и всех её реализуемых подфункций, необходимых для её корректной работы в соответствии с заданием.

3) Написать программный код на языке C++ согласно разработанным ранее блок-схемами, а также нормам разработки программ, которые были изучены на лабораторных занятиях.

4) Провести тестирование и отладку разработанной программы, последовательно проведя позитивное и негативные тестирования.

Отчёт по курсовой работе должен содержать:

- титульный лист
- содержание
- введение, кратко описывающее область поставленной задачи, а также цель и саму задачу
- теоретическую часть, содержащую полную информацию о решаемой задаче, её предметной области и существующих методах её решения.
- практическую часть, содержащую описание процесса разработки программы, а также все блок-схемы, точно описывающие принцип работы программы.
- заключение, содержащее вывод о проделанной работе и полученных навыках
- список литературы
- приложения, содержащие листинг программы и дополнительные блок-схемы

Тематики курсовых работ приведены в разделе 3.5 рабочей программы дисциплины.

7.3.2.1 Вопросы для защиты курсовой работы

1. Какие существуют алгоритмы поиска в тексте?
2. Что такое очередь и стек?

3. Что такое дек?
4. Какие необходимы библиотеки для осуществления работы с файловой системой?
5. Какие существуют методы сортировки линейных структур?
6. В чём особенность строкового типа данных?
7. Как реализуется программный «демон»?
8. Что такое бинарное дерево?
9. Перечислите алгоритмы поиска в сортированных структурах данных.
10. Перечислите сходства и различия односвязных и двусвязных списков.
11. Назовите преимущества использования классов.
12. Назовите области применения графов.
13. Назовите преимущества инкапсуляции.
14. Перечислите методы построения двумерной графики.
15. Назовите особенности организации сетевого взаимодействия посредством веб-сокетов.
16. Опишите методы обработки строк.
17. Криптография
18. Назовите применяемые методы хэширования и сжатия данных.
19. Назовите преимущества многопоточности при реализации сетевого взаимодействия.
20. Перечислите как минимум семь операций над булевыми числами.
21. Назовите особенности реализации пользовательского интерфейса.
22. Назовите основные математические операции языка C++.
23. Назовите различия между потоком и процессом.
24. Назовите преимущества использования рекурсивных функций.
25. Опишите особенности организации многопоточной программы.

7.3.3 Промежуточная аттестация

Промежуточная аттестация (зачёт) проводится на 3 семестре обучения в форме зачета

Зачет проводится по билетам, ответы предоставляются письменно с последующим устным собеседованием. Билеты формируются из вопросов представленного ниже перечня.

Регламент проведения зачета:

1. В билет включается (2) вопроса из разных разделов.

2. Перечень вопросов содержит 25 вопроса по изученным темам на лекционных и лабораторных занятиях (прилагается).

3. Время на подготовку письменных ответов - до 40 мин, устное собеседование - до 10 минут.

4. Проведение аттестации (зачета) с использованием средств электронного обучения и дистанционных образовательных технологий выполняется в соответствии с утверждённым в университете "Порядком проведения промежуточной аттестации с использованием средств электронного обучения и дистанционных образовательных технологий"

Перечень вопросов для подготовки к зачету и составления зачетных билетов для (3 семестр) (ОПК-6)

1. Этапы решения задач на ЭВМ.
2. Определение алгоритма. Свойства алгоритма.
3. Способы представления алгоритмов. Примеры.
4. Запись алгоритмов блок-схемами. Основные элементы блок-схем.
5. Стандартизация алгоритмов.
6. Основные алгоритмические структуры. Примеры.

7. Алгоритмы с ветвлением. Пример алгоритма.
8. Алгоритм цикла с предусловием. Пример алгоритма.
9. Алгоритм цикла с постусловием. Пример алгоритма.
10. Алгоритм цикла с управляющей переменной. Пример алгоритма.
11. Методы разработки и анализа алгоритмов.
12. Языки программирования: классификация и основные характеристики.
13. Состав алгоритмического языка (символы, лексемы, выражения, операторы, взаимосвязь элементов).
14. Системы программирования. Понятие, классификация, примеры.
15. Схема обработки прикладных программ в среде абстрактной системы программирования компилируемого типа (ввод, трансляция, компоновка, выполнение).
16. Процесс трансляции и компиляции.
17. Интегрированная среда программирования (понятие, структура, примеры).
18. Структура программы.
19. Обработка массивов.
20. Программирование ветвящихся алгоритмов.
21. Программирование циклических алгоритмов.
22. Базовые типы данных языка C/C++.
23. Форматы вывода данных языка C/C++.
24. Указатели в C/C++.
25. Структуры в C/C++.

Промежуточная аттестация (экзамен) проводится на 4 семестре обучения в форме экзамена

Экзамен проводится по билетам, ответы предоставляются письменно с последующим устным собеседованием. Билеты формируются из вопросов представленного ниже перечня.

Регламент проведения экзамена:

1. В билет включается (3) вопроса из разных разделов.

2. Перечень вопросов содержит 35 вопроса по изученным темам на лекционных и лабораторных занятиях (прилагается).

3. Время на подготовку письменных ответов - до 40 мин, устное собеседование - до 10 минут.

4. Проведение аттестации (экзамена) с использованием средств электронного обучения и дистанционных образовательных технологий выполняется в соответствии с утверждённым в университете "Порядком проведения промежуточной аттестации с использованием средств электронного обучения и дистанционных образовательных технологий"

Перечень вопросов для подготовки к экзамену и составления экзаменационных билетов для (4 семестр) (ОПК-6)

1. Простые структуры и типы данных.
2. Линейные статические структуры данных.
3. Временные структуры данных
4. Составные линейные типы данных.
5. Составные нелинейные типы данных.
6. Внешние структуры данных.
7. Методы сортировки данных.
8. Методы сортировки массивов.
9. Одномерные массивы: задачи поиска, замены и перестановок элементов массива.

10. Двумерные массивы: задачи поиска, замены и суммирования элементов двумерного массива.
11. Алгоритмы поиска в линейных структурах.
12. Алгоритм перебора с возвратом.
13. Рекурсивные алгоритмы.
14. Защита программного обеспечения.
15. Современные алгоритмы обработки данных.
16. Параллельные алгоритмы.
17. Параллельный поиск.
18. Параллельные алгоритмы на графах.
19. Параллельная сортировка.
20. Оценка сложности программ. Критерии оценки сложности программ
21. Отладка программ.
22. Оценка алгоритмов сортировки.
23. Оценка сложности программ. Пространственная сложность.
24. Оценка сложности программ. Временная сложность.
25. Оценка сложности программ. Асимптотическая оценка сложности.
26. Классификация ошибок по этапу обработки программ.
27. Тестирование программ.
28. Нисходящая и восходящая разработка программного обеспечения.
29. Технология визуального программирования.
30. Уровни языков программирования.
31. Процедурная парадигма программирования.
32. Объектно-ориентированная парадигма программирования
33. Технология визуального программирования.
34. Технологии для создания и работы интернет-приложений.
35. Технология структурного программирован