Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 30.05.2024 11:45:14
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 алгоритмизация»

Направление подготовки
27.03.05 «Инноватика»

Образовательная программа (профиль подготовки)
«Аддитивные технологии»

Квалификация (степень) выпускника
Бакалавр

Форма обучения
Очная

Москва, 2024 г.

Разработчик(и):

Старший преподаватель_____Е.С. Березин

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
4. Учебно-методическое и информационное обеспечение 10
5. Материально-техническое обеспечение 11
6. Методические рекомендации 11
7. Фонд оценочных средств 13

1. Цели, задачи и планируемые результаты обучения по дисциплине

К основным целям освоения дисциплины «Программирование и алгоритмизация» следует отнести:

- формирование у студентов знаний в области проектирования и использования программного обеспечения;
- приобретение студентами знаний технологии программирования, умений и навыков разработки прикладных программ;
- развитие способности студентов к самостоятельному изучению и освоению новых перспективных технологий программирования;
- подготовка студентов к деятельности в соответствии с квалификационной характеристикой бакалавра по направлению, в том числе формирование умений использовать освоенные технологии программирования в профессиональной деятельности.

К основным задачам освоения дисциплины «Программирование и алгоритмизация» следует отнести:

- овладение научной терминологией в области проектирования и использования программного обеспечения в инновационной деятельности;
- овладение методологией проектирования и нормативной документацией для приобретения навыков разработки прикладных программ;
- практическое освоение технологии программирования;

изучение способов подготовки и принятия решений по оценке эффективности технологий программирования как на начальном этапе проектирования, так и конечном этапе прекращения сопровождения программ, находящихся в эксплуатации.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование и алгоритмизация» относится к числу учебных дисциплин базовой части блока 1 (Б.1.1.09) основной образовательной программы бакалавриата.

Дисциплина «Программирование и алгоритмизация» взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ООП:

- Цифровая грамотность;

- Основы баз данных и информационных систем;

- Алгоритмы решений нестандартных задач.

3. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетные единицы (216 часов).

Изучается на 2 и 3 семестрах. Формы промежуточной аттестации -зачет во втором семестре и экзамен в третьем семестре.

3.1 Виды учебной работы и трудоемкость

3.1.1. Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			2 семестр	3 семестр
1	**Аудиторные занятия**	**108**	**36**	**72**
	В том числе:			
1.1	Лекции	36	18	18

1.2	Семинарские/практические занятия	18	-	18
1.3	Лабораторные занятия	54	18	36
2	**Самостоятельная работа**	**108**	**36**	**72**
	В том числе:			
2.1	Подготовка и защита лабораторных работ	36	-	36
2.2	Самостоятельное изучение	72	36	36
3	**Промежуточная аттестация**			
	Зачет/диф.зачет/экзамен		зачет	экзамен
	Итого	**216**		

3.2 Тематический план изучения дисциплины

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	
	2 семестр						
1	**Программирование как составляющая процесса разработки программного обеспечения.** Общая характеристика процесса разработки программного обеспечения. Ключевые этапы разработки. Модели процесса разработки программного обеспечения. Методологии разработки программного обеспечения. Защита программного обеспечения.		2				2
2	**Алгоритмические основы программирования.** Понятие алгоритма. Исполнители алгоритмов. Свойства алгоритмов. Типы алгоритмов. Формы представления алгоритмов. Методы разработки и анализа алгоритмов		4				4
3	**Системы программирования.** Понятие системы программирования. Программные компоненты системы программирования. Редактор текста. Трансляторы. Компоновщик. Отладчик. Библиотеки подпрограмм. Языки программирования. Уровни языков. Поколения языков. Интегрированные среды разработки.		2				4

4	**Язык программирования С/С++.** Элементы языка С. Базовые типы данных. Операции языка С. Управляющие операторы. Операторы форматного ввода и вывода. Структура программы. Математические функции. Форматы вывода данных. Особенности арифметических операций. Программирование ветвящихся и циклических алгоритмов.		2				4
5	**Язык программирования С/С++.** Массивы. Обработка массивов. Указатели. Функции. Файлы. Символьные переменные и строки. Преобразование строк. Форматирование строк. Структуры.		4				4
6	**Язык программирования С/С++ Объектно-ориентированное программирование.** Классы. Дружественные функции и классы. Статические элементы и функции. Константные функции. Перегрузка операций. Шаблоны		4				4
7	Лабораторная работа 1. Составление блок-схемы алгоритма.				2		2
8	Лабораторная работа 2. Форматированный ввод-вывод в С++.				2		2
9	Лабораторная работа 3. Функции.				2		2
10	Лабораторная работа 4. Базовые типы данных.				2		2
11	Лабораторная работа 5. Введение в std::string.				2		2
12	Лабораторная работа 6. Перечисления.				2		2
13	Лабораторная работа 7. Оператор goto, бесконечные циклы, цикл while.				2		2
14	Лабораторная работа 8. Циклы for, do … while.				2		2
15	Лабораторная работа 9. Операторы break и continue				2		2
	Итого по 2 семестру		18		18		36
	3 семестр						
16	**Структуры и алгоритмы обработки данных** Понятие о структурах данных. Простые структуры и типы данных.		2				4
17	**Структуры и алгоритмы обработки данных** Составные линейные типы данных. Массив. Очередь. Стек. Дек. Линейные списки.		2				4

18	**Структуры и алгоритмы обработки данных** Составные нелинейные типы данных. Древовидные структуры данных. Элементы теории графов.		2				4
19	**Структуры и алгоритмы обработки данных** Внешние структуры данных. Файлы. Базы данных.		2				4
20	**Структуры и алгоритмы обработки данных** Алгоритмы обработки данных. Поиск. Сортировка.		2				4
21	**Структуры и алгоритмы обработки данных** Параллельные алгоритмы. Параллельный поиск. Параллельная сортировка. Параллельные алгоритмы на графах. Современные алгоритмы обработки данных.		2				4
22	**Технологии программирования** Технология структурного программирования. Технология объектно-ориентированного программирования. Технология визуального программирования. Технологии для создания и работы интернет приложений.		2				4
23	**Оценка сложности алгоритмов и программ** Сравнение алгоритмов. Критерии оценки сложности. Виды сложностей. Пространственная сложность. Временная сложность. Асимптотическая оценка сложности		4				8
24	Лабораторная работа 10. Генерация случайных чисел				2		2
25	Лабораторная работа 11. Массивы.				2		2
26	Лабораторная работа 12. Массивы и циклы.				2		2
27	Лабораторная работа 13. Сортировка массива методом выбора.				2		2
28	Лабораторная работа 14. Многомерные массивы.				2		2
29	Лабораторная работа 15. Указатели.				2		2
30	Лабораторная работа 16. Взаимосвязь указателей и массивов.				2		2
31	Лабораторная работа 17. Динамическое выделение памяти.				2		2
32	Лабораторная работа 18. Динамические				2		2

	массивы.						
33	Лабораторная работа 19. Ссылки.				2		2
34	Лабораторная работа 20. Цикл foreach.				2		2
35	Лабораторная работа 21. Указатели типа void.				2		2
36	Лабораторная работа 22. Введение в std::array.				2		2
37	Лабораторная работа 23. Введение в итераторы в C++.				2		2
38	Лабораторная работа 24. Введение в ООП.				2		2
39	Лабораторная работа 25. Спецификаторы доступа классов.				2		2
40	Лабораторная работа 26. Инкапсуляция, геттеры и сеттеры.				2		2
41	Лабораторная работа 27. Конструкторы классов.				2		2
	Практическая работа 1. Защита лабораторной работы №10,11			2			
	Практическая работа 2. Защита лабораторной работы №12,13			2			
	Практическая работа 3. Защита лабораторной работы №14,15			2			
	Практическая работа 4. Защита лабораторной работы №16,17			2			
	Практическая работа 5. Защита лабораторной работы №18,19			2			
	Практическая работа 6. Защита лабораторной работы №20,21			2			
	Практическая работа 7. Защита лабораторной работы №22,23			2			
	Практическая работа 8. Защита лабораторной работы №24,25			2			
	Практическая работа 9. Защита лабораторной работы №26,27			2			
	Итого по 3 семестру		18	18	36		72
	Итого		**36**	18	**72**		**108**

3.3 Содержание дисциплины

Второй семестр

Программирование как составляющая процесса разработки программного обеспечения

Общая характеристика процесса разработки программного обеспечения. Ключевые этапы разработки программного обеспечения. Модели процесса разработки программного обеспечения. Методологии разработки программного обеспечения. Защита программного обеспечения.

Алгоритмические основы программирования

Понятие алгоритма. Исполнители алгоритмов. Свойства алгоритмов. Типы алгоритмов. Формы представления алгоритмов. Методы разработки и анализа алгоритмов.

Системы программирования

Понятие системы программирования. Программные компоненты системы программирования. Редактор текста. Трансляторы. Компоновщик. Отладчик. Библиотеки подпрограмм. Языки программирования. Уровни языков. Поколения языков. Интегрированные среды разработки.

Язык программирования C/C++

Элементы языка С. Базовые типы данных. Операции языка С. Управляющие операторы. Операторы форматного ввода и вывода. Структура программы. Математические функции. Форматы вывода данных. Особенности арифметических операций. Программирование ветвящихся и циклических алгоритмов. Массивы. Обработка массивов. Указатели. Функции. Файлы. Символьные переменные и строки. Преобразование строк. Форматирование строк. Структуры. Объектно-ориентированное программирование. Классы. Дружественные функции и классы. Статические элементы и функции. Константные функции. Перегрузка операций. Шаблоны.

Третий семестр

Структуры и алгоритмы обработки данных

Понятие о структурах данных. Простые структуры и типы данных. Составные линейные типы данных. Массив. Очередь. Стек. Дек. Линейные списки. Составные нелинейные типы данных. Древовидные структуры данных. Элементы теории графов. Внешние структуры данных. Файлы. Базы данных. Алгоритмы обработки данных. Поиск. Сортировка. Параллельные алгоритмы. Параллельный поиск. Параллельная сортировка. Параллельные алгоритмы на графах. Современные алгоритмы обработки данных.

Технологии программирования

Понятие технологии программирования. Понятие жизненного цикла ПС. Цели и структура современных моделей жизненного цикла ПС. Содержание отдельных этапов разработки ПС. Стандартизация жизненного цикла ПС. Планирование, управление и тестирование программного обеспечения. Отладка программ. Документирование, сопровождение, реинжиниринг и управление качеством. Задачи стандартизации программных средств. Стандарты ISO, SW-CMM. CASE-технологии

Современные технологии программирования. Технология визуального программирования. Технологии для создания и работы интернет приложений.

Оценка сложности алгоритмов и программ

Сравнение алгоритмов. Критерии оценки сложности. Виды сложностей. Пространственная сложность. Временная сложность. Асимптотическая оценка сложности.

3.4 Тематика семинарских/практических и лабораторных занятий

3.4.1. 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.

4. Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

Нормативные документы и ГОСТы отсутствуют.

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.Щупак .-СПб. :Питер , 2011.-352с.:ил. -(Учебное пособие)

4. Павловская Т.А. C/C++. Программирование на языке высокого уровня: для магистров и бакалавров : учеб. для вузов . Гриф МО /Т.А.Павловская .-СПб. : Питер , 2014.-461с.:ил. -(Учебник для вузов)

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.

3. Иванов, И.П. Сборник задач по курсу «Алгоритмы и структуры данных». [Электронный ресурс] / И.П. Иванов, А.Ю. Голубков, С.Ю. Скоробогатов. — Электрон. дан. — М. : МГТУ им. Н.Э. Баумана, 2013. — 32 с. — Режим доступа: http://e.lanbook.com/book/52435 — Загл. с экрана.

4. Иванова Г.С. Технология программирования. – М.: Изд-во КноРус, 2011.

5. Керниган, Б. Язык программирования С / Бриан В. Керниган, Деннис Ритчи .-2-е изд., перераб. и доп .-М. : Вильямс ,2012.-304с.:ил. -(Серия книг по программированию от Prentice Hall)

6. Москвитина, О.А. Сборник примеров и задач по программированию. [Электронный ресурс] / О.А. Москвитина, В.С. Новичков, А.Н. Пылькин. — Электрон. дан. — М. : Горячая линия-Телеком, 2014. — 245 с. — Режим доступа: http://e.lanbook.com/book/64090 — Загл. с экрана.

7. Окулов, С.М. Алгоритмы обработки строк. [Электронный ресурс] — Электрон. дан. — М. : Издательство "Лаборатория знаний", 2015. — 258 с. — Режим доступа: http://e.lanbook.com/book/66113 — Загл. с экрана.

8. Шилдт, Герберт C++. Базовый курс : пер. с англ. /Герберт Шилдт .-3-е изд .-М. :Вильямс , 2014.-624с.:ил.

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	Ссылка
Основы программирования и алгоритмизации в машиностроении (часть 1)	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

- Интегрированная среда разработки Microsoft Visual Studio 2013 и выше.
- https://app.diagrams.net/

Современные профессиональные базы данных и информационные справочные системы

Не предусмотрено.

5. Материально-техническое обеспечение

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

6. Методические рекомендации

Методика преподавания дисциплины «Основы программирования и алгоритмизации в

машиностроении» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения аудиторных и внеаудиторных занятий:

– аудиторные занятия: лекции, лабораторные работы, тестирование;

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

6.1 Методические рекомендации для преподавателя по организации обучения

На первом занятии по дисциплине необходимо ознакомить студентов с порядком ее изучения (темами курса, формами занятий, текущего и промежуточного контроля), раскрыть место и роль дисциплины в системе наук, ее практическое значение, довести до студентов требования к форме отчетности и применения видов контроля. Выдаются задания для подготовки к семинарским занятиям.

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

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

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Задачи самостоятельной работы студента:

- развитие навыков самостоятельной учебной работы;
- освоение содержания дисциплины;
- углубление содержания и осознание основных понятий дисциплины;
- использование материала, собранного и полученного в ходе самостоятельных занятий для эффективной подготовки к зачету.

Виды внеаудиторной самостоятельной работы:

- самостоятельное изучение отдельных тем дисциплины;
- подготовка к лекционным 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 работы или защита лабораторной работы.

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 лабораторных работ, экзамен.

Обучение по дисциплине «Основы программирования и алгоритмизации в машиностроении» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	**Индикаторы достижения компетенции**
ОПК-7. Способен использовать информационно-коммуникационные компьютерные технологии, базы данных, пакеты прикладных программ для решения инженерно-технических и технико-экономических задач планирования и управления работами по инновационным проектам	ИОПК-7.1 Демонстрирует знание принципов современных информационных технологий; ИОПК-7.3 Использует современные пакеты прикладных программ для решения инженерно-технических и технико-экономических задач управления работами по инновационным проектам в области аддитивных технологий;
ОПК-10. Способен разрабатывать и применять алгоритмы и программные приложения для решения практических задач цифровизации в области профессиональной деятельности	ИОПК-10.1 Способен разрабатывать алгоритмы и программные приложения в области профессиональной деятельности ИОПК-10.2 Применяет прикладное программное обеспечение для решения технико-экономических задач, планирования и проведения работ по проекту

7.1 Методы контроля и оценивания результатов обучения

№ ОС	Наименование оценочногосредства	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

7.2 Шкала и критерии оценивания результатов обучения

Форма промежуточной аттестации: зачет.

Промежуточная аттестация обучающихся в форме зачёта проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в

течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «зачтено» или «не зачтено».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Основы программирования и алгоритмизации в машиностроении» (выполнили лабораторные работы, прошли промежуточный контроль в виде компьютерного тестирования).

Шкала оценивания	**Описание**
Зачтено	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателям, 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 показателям, 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.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Программирование и алгоритмизация» (прошли промежуточный контроль, выполнили и защитили лабораторные работы).

Шкала оценивания	**Описание**
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков

	приведенным в таблицах показателям, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности, не испытывает затруднений при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует частичное соответствие знаний, умений, навыков приведенным в таблицах показателям, 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.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателям, 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.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует полное отсутствие или недостаточное соответствие знаний, умений, навыков приведенным в таблицах показателям, допускаются значительные ошибки,проявляется отсутствие знаний, умений, навыков по ряду показателей, студент не может оперировать знаниями и умениями при их переносе на новые ситуации.

7.3 Оценочные средства

7.3.1. Текущий контроль

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 75 баллов из 100 возможных.

1. **Совокупность правил образования цепочек символов, образующих идентификаторы, операторы, операции и другие лексические компоненты языка – это**

   А. Синтаксис;

B. Лексика;
C. Алфавит;
D. Семантика.

2.Любая информация, представленная в формализованном виде и пригодная для обработки алгоритмом – это

A. Константы;
B. Данные;
C. Переменные.

3.Какой диапазон значений имеет тип int для 32-разрядных вычислительных систем:

A. от 0 до 255
B. от -32768 до 32767
C. от 0 до 65535
D. от 0 до 4 294 967 295

4.Какой размер в байтах имеет переменная вещественного типа float

A. 2
B. 4
C. 8
D. 10

5.Дан массив int L[3][3] = { { 2, 3, 4 }, { 3, 4, 8 }, { 1, 0, 9 } };. Чему будет равно значение элемента этого массива L[1][2]

A. 2
B. 3
C. 4
D. 8

6.Для объявления размера массива должна использоваться_____, потому что она делает программу более масштабируемой

A. переменная
B. константа
C. именованная константа
D. символ

7.Процесс упорядоченного размещения элементов в массиве называется_____

A. Сортировка
B. Поиск
C. Проверка
D. Изменение

8.Какой тип данных отсутствует в Си в отличие от большинства других языков:

A. Real
B. Integer
C. String
D. Char

9.Идентификатор - это:

A. последовательность латинских букв, цифр и символа «_», начинающаяся с буквы или символа «_»
B. неизменяемые объекты языка (константы)
C. последовательность латинских и русских букв
D. способ кодирования, допустимые преобразования над значением данной переменной

10. Этап, занимающий наибольшее время, при разработке программы:

A. тестирование
B. сопровождение
C. проектирование
D. программирование
E. формулировка требований

11. Оценка сложности линейного алгоритма равна

A. O(1)
B. O(n)
C. $O(N^2)$
D. O(n!)

12. Авторское право на программный продукт действует в течение

A. всей жизни автора и 50 лет после его смерти
B. всей жизни автора и 100 лет после его смерти
C. всей жизни автора

13. Что выполняется раньше:

A. компиляция
B. отладка
C. компоновка
D. тестирование

14. Способы оценки качества:

A. сравнение с аналогами
B. наличие документации
C. оптимизация программы
D. структурирование алгоритма

15. Процесс упорядоченного размещения элементов в массиве называется_____

A. Сортировка
B. Поиск
C. Проверка
D. Изменение

16. Выражения - это:

A. конструкции, включающие константы (литералы), переменные, знаки операций, скобки для управления порядком выполнения операций, обращения к функциям
B. основные строительные блоки программы; в языке Си++ указанием на наличие выражения служит символ «точка с запятой», стоящий в конце него
C. набор символов и операций
D. операторы, выполняющие определенные действия с переменными

17. Какой тип данных отсутствует в Си в отличие от большинства других языков:

A. Real
B. Integer
C. String
D. Char

18. Идентификатор - это:

A. последовательность латинских букв, цифр и символа «_», начинающаяся с буквы или символа «_»
B. неизменяемые объекты языка (константы)
C. последовательность латинских и русских букв
D. способ кодирования, допустимые преобразования над значением данной переменной

19. Интерфейс

A. представляет собой набор функций, которые реализуют класс
B. представляет собой описание набора функций
C. представляет собой описание набора функций, которые реализуют класс
D. представляет собой описание набора функций

20. Этап, занимающий наибольшее время, при разработке программы:

A. тестирование
B. сопровождение
C. проектирование
D. программирование

Е. формулировка требований

7.3.1.1 Вопросы для защиты лабораторных работ

1. Как создать новое консольное приложение?
2. Какое расширение имеют программы на C++?
3. Что такое компиляция?
4. Как скомпилировать программу?
5. Почему среда разработки Microsoft Visual Studio называется интегрированной?
6. Что такое отладка?
7. Для чего применяется ключевое слово void в C++?
8. Что такое main() в C++?
9. С помощью чего показывается начало и конец тела функции или класса в C++?
10. Какова роль комментариев?
11. Какие два вида комментариев существуют?
12. Чувствителен ли к регистру язык программирования C++?
13. Можно ли выводить на устройство вывода текст на кириллице при написании программ на языке программирования C++ в среде Microsoft Visual Studio?
14. Что такое блок комментариев и когда он применяется?
15. Какова структура программ на C++ и что такое хороший стиль?
16. Что такое директива препроцессора и каково ее функциональное предназначение?
17. Что такое условие?
18. К чему прибегают, когда возникает необходимость повторного использования операторов?
19. Какие типы циклов существуют?
20. Как работает цикл типа while?
21. В чем отличие принципа работы цикла типа do…while от while?
22. В каких случаях применяют цикл типа do…while?
23. Что такое одномерный массив?
24. Как нумеруются элементы массива?
25. Как размещается массив в пямяти компьютера?
26. С помощью чего осуществляется доступ к каждому элементу массива?
27. Как задается размерность массива?
28. Как задается размерность массива при помощи константы?
29. Как производится вычисление суммы элементов массива?
30. Как работает сортировка массива простым выбором?
31. Как работает метод пузырьковой сортировки?
32. Что такое двумерный массив?
33. Что такое указатель?
34. Как происходит побайтовое смещение при использовании указателей?
35. Что такое функция?
36. Как называется информация, передаваемая в функцию для обработки?
37. Обязательно ли описывать (объявлять) функцию перед её вызовом?
38. Что такое хороший стиль программирования?
39. Как осуществляется передача параметров в функцию?
40. Как выполняется передача параметров по значению?
41. Как выполняется передача параметров по адресу?
42. Какими двумя способами выполняются операции распределения памяти в C и C++?
43. Какие существуют два способа хранения информации в оперативной памяти ПК?
44. Как осуществляется работа с динамической памятью в C++?
45. Как осуществляется проверка на выделение памяти?
46. Каким образом реализуется работа с двумерными динамическими массивами в C++?

7.3.2. Промежуточная аттестация

Промежуточная аттестация проводится во 2 и 3 семестрах обучения в форме зачета во втором семестре и экзамена в третьем семестре.

Зачет проводится по билетам или с применением LMS системы в виде компьютерного тестирования в стенах университета в присутствии преподавателя, ответы предоставляются письменно с последующим устным собеседованием. Билеты формируются из вопросов представленного ниже перечня.

Перечень вопросов для подготовки к зачету.

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++.

Регламент проведения экзамена:

Перечень вопросов для подготовки к экзамену и составления экзаменационных билетов.

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.
    Технология структурного программирования.