

МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное
учреждение высшего образования "Пермский
государственный национальный исследовательский
университет"**

Кафедра прикладной математики и информатики

**Авторы-составители: Перескокова Ольга Ивановна
Русакова Ольга Леонидовна
Шейна Татьяна Юрьевна**

Рабочая программа дисциплины

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ I

Код УМК 93609

**Утверждено
Протокол №9
от «21» мая 2019 г.**

Пермь, 2019

1. Наименование дисциплины

Алгоритмизация и программирование I

2. Место дисциплины в структуре образовательной программы

Дисциплина входит в обязательную часть Блока « С.1 » образовательной программы по направлениям подготовки (специальностям):

Специальность: **10.05.03** Информационная безопасность автоматизированных систем
специализация Безопасность открытых информационных систем

3. Планируемые результаты обучения по дисциплине

В результате освоения дисциплины **Алгоритмизация и программирование I** у обучающегося должны быть сформированы следующие компетенции:

10.05.03 Информационная безопасность автоматизированных систем (специализация : Безопасность открытых информационных систем)

ОПК.2 Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

Индикаторы

ОПК.2.1 Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности

ОПК.2.2 Анализирует типовые языки программирования, составляет программы

ОПК.2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения

4. Объем и содержание дисциплины

Направления подготовки	10.05.03 Информационная безопасность автоматизированных систем (направленность: Безопасность открытых информационных систем)
форма обучения	очная
№№ триместров, выделенных для изучения дисциплины	1
Объем дисциплины (з.е.)	4
Объем дисциплины (ак.час.)	144
Контактная работа с преподавателем (ак.час.), в том числе:	56
Проведение лекционных занятий	28
Проведение практических занятий, семинаров	14
Проведение лабораторных работ, занятий по иностранному языку	14
Самостоятельная работа (ак.час.)	88
Формы текущего контроля	Входное тестирование (1) Защищаемое контрольное мероприятие (1) Итоговое контрольное мероприятие (1) Письменное контрольное мероприятие (2)
Формы промежуточной аттестации	Экзамен (1 триместр)

5. Аннотированное описание содержания разделов и тем дисциплины

Входной контроль

- Понятие алгоритма. Свойства алгоритма.
- Понятие переменной и типов переменных
- Способы описания алгоритмов (алгоритмический язык, язык блок-схем, язык программирования Паскаль)
- Основные алгоритмические структуры (ветвление, цикл с предусловием, цикл с параметром)
- Понятие подпрограммы и использование подпрограмм
- Рекурсия
- Понятие одномерного массива и работа с массивами. Основные задачи по обработке массивов (поиск минимума и максимума, сумма элементов массива)

Простые типы данных. Основные алгоритмические структуры. Функции и рекурсия

Введение в понятие алгоритма. Свойства алгоритма. Языки программирования. Способы описания языков программирования. Этапы решения задачи с использованием компьютера. Принципы тестирования программ. Структура программы на языке C++. Скалярные типы данных. Организация ввода-вывода.

Понятия: оператор, операция и операнд. Приоритеты выполнения операций. Директивы препроцессора.

Операторы ветвления (if, switch). Простое и вложенное ветвление

Операторы циклов с условием (while, do-while). Оператор цикла с параметром. Обработка последовательностей. Формула общего члена последовательности. Рекуррентное соотношение.

Взаимозаменяемость трех видов циклов.

Досрочный выход из цикла. Вложенные циклы.

Принципы структурного программирования. Метод пошаговой детализации. Функции: описание.

Локальные и глобальные переменные. Структура программного модуля. Параметры функции: передача по значению и по ссылке

Понятие указателя и его использование. Указатели при передаче параметров функции.

Отличие ссылки от указателя. Перегрузка функций. Inline - функции.

Рекурсия. Виды рекурсии: прямая и косвенная рекурсия, выполнение действий на рекурсивном спуске и на рекурсивном возврате. Механизм реализации рекурсии при помощи стека. Реализация итеративных алгоритмов рекурсивным образом. Трассировка рекурсивных алгоритмов.

Программирование линейных и ветвящихся алгоритмов.

Этапы решения задачи путем составления алгоритма для ЭВМ. Математическая модель задачи.

Лексика, синтаксис и семантика языка программирования C++. Основные типы данных и операции в C++. Приоритет выполнения операций. Оператор ветвления. Полное и неполное ветвление. Вложенные ветвления. Оператор выбора (switch). Основы тестирования программ.

Программирование циклических алгоритмов. Обработка последовательностей.

Итерация как способ решения задач. Цикл с параметром. Цикл с пред-условием. Цикл с постусловием.

Операторы досрочного прерывания цикла. Вычисление суммы N первых членов ряда. Формула общего члена ряда. Программирование рекуррентных соотношений. Вычисление суммы ряда с заданной точностью.

Структурирование программ. Выделение подзадач. Программирование функций. Способы передачи параметров. Рекурсивные подпрограммы.

Принципы структурного программирования. Деление задачи на подзадачи. Восходящее и нисходящее проектирование. Метод пошаговой детализации. Описание функций на языке C++. Понятие указателя. Способы передачи параметров: по значению, по ссылке или через указатель. Локальные и глобальные

переменные. Рекурсия. Виды рекурсии. Прямая и косвенная рекурсия. Рекурсия с одним или несколькими рекурсивными вызовами. Механизм реализации рекурсии при помощи системного стека. Программирование итеративных алгоритмов рекурсивным образом. Программирование выполнения действий на рекурсивном спуске или на рекурсивном возврате, либо одновременно на рекурсивном спуске и на рекурсивном возврате.

Основные структуры данных. Методы сортировок

Файлы последовательного доступа: ввод-вывод данных при помощи текстовых файлов. Решение типовых задач теории чисел: определение простоты числа, разложение числа на простые сомножители, алгоритм Евклида для определения НОД, вычисление НОК.

Структура данных одномерный массив. Использование указателей при работе с массивами.

Стандартные алгоритмы обработки одномерных массивов (поиск минимума (максимума), определения наличия свойства элементов в массиве, подсчет количества и т.п.). Перестановки, вставка и удаление в массиве. Статические и динамические массивы.

Временная и емкостная сложность алгоритма. Определение порядка сложности алгоритма по времени исполнения на основании представленного кода. Алгоритм "Решето Эратосфена" для поиска всех простых чисел на заданно промежутке.

Методы поиска: линейный и бинарный поиск. Методы внутренних сортировок:

- простыми обменами;
- простым выбором;
- простыми вставками;
- бинарными вставками;
- Шейкер-сортировка;
- сортировка подсчетом;
- цифровая сортировка;
- сортировка Шелла;
- сортировка слияниями;
- быстрая сортировка Хоара
- пирамидальная сортировка.

Символьный тип данных: основные операции. Работа со строковыми значениями как с массивами символов (способ унаследованный от C). Строковый тип данных string: внутреннее представление и основные операции. (класс из библиотеки STL). Сравнение данных способов работы со строками.

Стандартные функции для поиска подстроки в строке, вставки, удаления и извлечения подстроки.

Реализация типовых алгоритмов обработки строк. Работа с числами, выходящими за пределы диапазонов значений скалярных типов данных: алгоритмы длинной арифметики.

Структура данных двумерный массив. Основные алгоритмы обработки двумерных массивов.

Динамическое выделение памяти для двумерного массива. Передача двумерного массива как параметра функции. Работа с диагоналями. Вставка и удаление в двумерном массиве.

Ввод-вывод данных из файлов последовательного доступа (текстовых файлов). Определение простоты числа. Поиск НОД и НОК.

Файловые потоки в C++. Чтение из текстового файла. Запись в текстовый файл. Работа с текстовым файлом на дополнение. Чтение из файла не-определенного числа элементов (чтение до конца файла).

Простые и составные числа. Эффективный алгоритм для определения простоты отдельного числа.

Определение наибольшего общего делителя двух чисел: эффективный алгоритм Евклида. Вычисление наименьшего общего кратного через наибольший общий делитель.

Одномерные массивы и базовые алгоритмы их обработки.

Одномерный массив. Статические и динамические массивы. Алгоритмы поиска минимум (максимума) их индексов и количества. Определение наличия свойства в массиве. Перестановки в массиве. Вставка и удаление в массиве. Поиск двух (трех) минимумов (максимумов), экстремума и его количества за один проход по массиву. Использование указателей при работе с одномерными массивами в C++.

Временная и емкостная сложность алгоритма. Оценка порядка сложности. Решето Эратосфена.

Сложность алгоритма по времени исполнения. Оценка порядка функции временной сложности. Емкостная сложность алгоритма. Определение порядка сложности по коду программы для циклического алгоритма. Оптимизация алгоритма. Решето Эратосфена для поиска простых чисел на промежутке.

Методы сортировки и поиска.

Линейный поиск. Бинарный поиск. Сортировка простыми обменами. Сортировка простым выбором. Сортировка простыми вставками. Сортировка бинарными вставками. Шейкер-сортировка. Сортировка Шелла. Алгоритм слияния двух упорядоченных массивов. Сортировка слияниями. Пирамидальная сортировка. Быстрая сортировка Хоара. Сортировка подсчетом. Цифровая сортировка. Оценка сложности алгоритмов сортировки и поиска.

Строковый тип данных. Стандартные функции обработки строк. Длинная арифметика.

Символьный тип данных. Преобразование символов в верхнему (ниж-нему) регистру. Строковый тип данных. Стандартные функции поиска, вставки и удаления в строке. Алгоритмы обработки текстов: выделение, слов, предложений. Работа с числами в различных системах счисления в строковом формате. Алгоритмы длинной арифметики для вычислений с числами, выходящими за диапазоны стандартных типов данных.

Двумерные массивы, представление и основные операции.

Двумерные массивы. Статическое и динамическое выделение памяти под двумерный массив. Работа со строками, столбцами и диагоналями двумерного массива. Перестановки в двумерном массиве. Вставка и удаление в двумерном массиве.

Индивидуальная работа

Выполнение индивидуальной работы по одной из теме: "Обработка числовой и символьной информации с использованием массивов", Определение алгоритма и идеи решения поставленной задачи. Написание и структурирование программного кода решения задачи. Разработка полного набора тестов для тестирования созданной программы. Документирование полученного программного решения: оформление отчета, создание презентации и очная защита работы.

Выбор алгоритма и метода решения задачи.

Анализ постановки задачи. Определение подходящих алгоритмов для решения задачи. Анализ временной и емкостной сложности выбранного алгоритма. Работа со специальной литературой по выбору и анализу алгоритма. Оптимизация выбранного алгоритма. Описание идеи решения задачи и выбранного алгоритма решения задачи с обоснованием его эффективности.

Разработка программы.

Написание программного кода на языке C++. Структурирование программного кода с помощью выделения функций. Устранение повторов в ко-де. Оптимизация программного кода. Читабельность программного кода. Комментирование программы.

Разработка полного набора тестов для тестирования программы.

Принципы тестирования программ. Метод «черного ящика» и метод «белого ящика» при разработке набора тестов. Полнота набора тестов. Возможности автоматического прогона всего набора тестов в системах программирования. Документирование набора тестов.

Внешнее тестирование программы.

Пошаговое выполнение и отладка программы. Поиск и исправление ошибок в программе. Коррекция набора тестов. Поиск ошибок в программе в случае закрытых внешних тестов. Поиск ошибок в программе в случае, когда известен тест неверной работы программы.

Оформление документации (отчета) о выполнении индивидуальной работы.

Отчет о выполнении индивидуальной работы. Разделы отчета. Оформление отчета. Оформление списка литературы. Оформление титульного листа. Приложения к отчету.

Защита индивидуальной работы.

Презентация защиты. Встраивание мультимедиа в презентацию. Эффективная презентация. Подготовка выступления. Дискуссия. Ответы на вопросы. Оценивание выступлений других участников.

Экзамен

Контрольное мероприятие в письменной форме, включающее два теоретических вопроса (билеты) и практические задания (написание кода программ для поставленных задач)

Практическая часть экзамена.

Практическая часть экзамена выполняется на компьютере в течение 1,5 часов. Для решения предлагаются 5-6 задач типовых задач на алгоритмы, которые изучались в течение триместра. Задача считается решенной, если проходит все тесты. При решении задач можно пользоваться конспектами и другими источниками информации, напечатанными на бумаге. Пользоваться доступом в Интернет и электронными источниками информации (ноутбуками, мобильными телефонами и смартфонами, планшетами, электронными часами, CD и MP3 плеерами, компакт-дисками, модулями флеш-памяти, картами памяти) запрещается, в противном случае это приводит к удалению студента с экзамена. Двум сидящим рядом студентам должны быть выданы разные варианты. Помощь сокурсника расценивается как списывание и так же приводит к удалению с экзамена обоих студентов: и того, кто дал списать, и того, кому дали списать.

Теоретическая часть экзамена.

Теоретическая часть экзамена выполняется по билетам письменно на бумаге в течение 1,5 часов. При решении заданий пользоваться электронными устройствами (ноутбуками, мобильными телефонами и смартфонами, планшетами, электронными часами, CD и MP3 плеерами, компакт-дисками, модулями флеш-памяти, картами памяти), а так же письменными источниками (конспектами, книгами, учебниками, шпаргалками, любыми, подготовленными до начала экзамена, личными записями) запрещается, и попытка воспользоваться ими приводит к удалению студента с экзамена. Студентам заранее предоставляется список вопросов к экзамену. В билете три вопроса: первые два вопроса из предоставленного студентам списка. Третий вопрос билета содержит практическую задачу, направленную на анализ или модификацию представленного алгоритма или практическое применение теоретических знаний.

6. Методические указания для обучающихся по освоению дисциплины

Освоение дисциплины требует систематического изучения всех тем в той последовательности, в какой они указаны в рабочей программе.

Основными видами учебной работы являются аудиторные занятия. Их цель - расширить базовые знания обучающихся по осваиваемой дисциплине и систему теоретических ориентиров для последующего более глубокого освоения программного материала в ходе самостоятельной работы. Обучающемуся важно помнить, что контактная работа с преподавателем эффективно помогает ему овладеть программным материалом благодаря расстановке необходимых акцентов и удержанию внимания интонационными модуляциями голоса, а также подключением аудио-визуального механизма восприятия информации.

Самостоятельная работа преследует следующие цели:

- закрепление и совершенствование теоретических знаний, полученных на лекционных занятиях;
- формирование навыков подготовки текстовой составляющей информации учебного и научного назначения для размещения в различных информационных системах;
- совершенствование навыков поиска научных публикаций и образовательных ресурсов, размещенных в сети Интернет;
- самоконтроль освоения программного материала.

Обучающемуся необходимо помнить, что результаты самостоятельной работы контролируются преподавателем во время проведения мероприятий текущего контроля и учитываются при промежуточной аттестации.

Обучающимся с ОВЗ и инвалидов предоставляется возможность выбора форм проведения мероприятий текущего контроля, альтернативных формам, предусмотренным рабочей программой дисциплины. Предусматривается возможность увеличения в пределах 1 академического часа времени, отводимого на выполнение контрольных мероприятий.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации.

При проведении текущего контроля применяются оценочные средства, обеспечивающие передачу информации, от обучающегося к преподавателю, с учетом психофизиологических особенностей здоровья обучающихся.

7. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

При самостоятельной работе обучающимся следует использовать:

- конспекты лекций;
- литературу из перечня основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля);
- текст лекций на электронных носителях;
- ресурсы информационно-телекоммуникационной сети "Интернет", необходимые для освоения дисциплины;
- лицензионное и свободно распространяемое программное обеспечение из перечня информационных технологий, используемых при осуществлении образовательного процесса по дисциплине;
- методические указания для обучающихся по освоению дисциплины.

8. Перечень основной и дополнительной учебной литературы

Основная:

1. Павловская Т. А. С/С++. Программирование на языке высокого уровня:[учебник для студентов высших учебных заведений, обучающихся по направлению "Информатика и вычислительная техника"]/Т. А. Павловская.-Санкт-Петербург [и др.]:Питер,2010, ISBN 978-5-94723-568-5.-460.- Библиогр.: с. 383
2. Павловская Т. А.,Щупак Ю. А. С/С++. Структурное программирование:практикум/Т. А. Павловская, Ю. А. Щупак.-СПб.:Питер,2004, ISBN 5-94723-967-1.-239.
3. Страуструп, Б. Язык программирования С++ для профессионалов : учебник / Б. Страуструп. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 670 с. — ISBN 978-5-4497-0922-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/102077>
4. Павловская Т. А. С/С++. Программирование на языке высокого уровня:учеб. для студентов вузов, обучающихся по напр. "Информатика и вычислит. техника"/Т. А. Павловская.-СПб.:Питер,2005, ISBN 5-94723-568-4.-461.
5. Тяпичев, Г. А. Быстрое программирование на С++ / Г. А. Тяпичев. — Москва : СОЛОН-Пресс, 2017. — 373 с. — ISBN 5-98003-162-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/90357.html?replacement=1>

Дополнительная:

1. Самуйлов, С. В. Алгоритмы и структуры обработки данных : учебное пособие / С. В. Самуйлов. — Саратов : Вузовское образование, 2016. — 132 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/47275.html>
2. Петров, В. Ю. Информатика. Алгоритмизация и программирование. Часть 1 : учебное пособие / В. Ю. Петров. — Санкт-Петербург : Университет ИТМО, 2016. — 93 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/66473.html>
3. Фридман, А. Л. Язык программирования С++ : учебное пособие / А. Л. Фридман. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 217 с. — ISBN 978-5-4497-0920-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. <http://www.iprbookshop.ru/102076>
4. Анисимов А. Е.,Пупышев В. В. Сборник заданий по основаниям программирования:учеб. пособие/А. Е. Анисимов, В. В. Пупышев.-Москва:Интернет-Университет информационных технологий,2006, ISBN 5-9556-0059-0.-348.-Библиогр.: с. 347-348
5. Кнут Д. Э.Искусство программирования.[в 3 т. : учебное пособие : пер. с англ.] Т. 1.Основные алгоритмы/Д. Э. Кнут ; под общ. ред. д.ф.-м.н., проф. Ю. В. Козаченко.-3-е изд., [испр. и доп.].-М.; СПб.; Киев:Вильямс,2005, ISBN 5-8459-0080-8.-720
6. Задачи по программированию/С. А. Абрамов, Г. Г. Гнездилова, Е. Н. Капустина.Вып. 56.- М.:Наука,1988.-223

7. Острейковский Владислав Алексеевич Информатика: Учеб. для вузов/Владислав Алексеевич Острейковский.-М.:Высш. шк.,2001, ISBN 5-06-003533-6.-511.-Библиогр.:с.508
8. Окулов С. М. Основы программирования: учебное пособие/С. М. Окулов.-М.: БИНОМ. Лаборатория знаний,2013, ISBN 978-5-9963-0482-0.-336.
9. Кнут Д. Э. Искусство программирования Т. 3. Сортировка и поиск/Д. Э. Кнут ; ред. Ю. В. Козаченко ; transl.: В. Т. Тертышный, И. В. Красиков.-2-е изд..-М.; СПб.; Киев: Вильямс,2004, ISBN 5-8459-0082-4.-832
10. Брукшир Дж. Гленн Введение в компьютерные науки: Пер. с англ./Под ред. В. Н. Штонды.-М.; СПб.; Киев: Изд. дом "Вильямс",2001, ISBN 5-8459-0179-0.-688.
11. Подбельский В. В. Язык Си++: Учеб. пособие для вузов/В. В. Подбельский.-М.: Финансы и статистика,2000, ISBN 5-279-02204-7.-560.-Библиогр.: с.538-539
12. Королев Л. Н., Миков А. И. Информатика. Введение в компьютерные науки: Учеб. пособие для вузов/Л. Н. Королев, А. И. Миков.-М.: Высш. шк.,2003, ISBN 5-06-004272-3.-341.-Библиогр.: с. 320-321

9. Перечень ресурсов сети Интернет, необходимых для освоения дисциплины

<https://code-live.ru/tag/cpp-manual/> Уроки программирования
<http://cppstudio.com/cat/274/> Язык программирования C++
<http://ptaskbook.com> Электронный задачник Programming Taskbook
<https://studfiles.net/preview/4229380/> Алгоритмизация и программирования
<http://www.psu.ru/elektronnye-resursy-dlya-psu> Электронные ресурсы для ПГНИУ
<http://window.edu.ru/> Единое окно доступа к образовательным ресурсам

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Образовательный процесс по дисциплине **Алгоритмизация и программирование I** предполагает использование следующего программного обеспечения и информационных справочных систем: Образовательный процесс по дисциплине предполагает использование следующего программного обеспечения и информационных справочных систем:

- доступ в режиме on-line в Электронную библиотечную систему (ЭБС);
- доступ в электронную информационно-образовательной среду университета.

Необходимое лицензионное и (или) свободно распространяемое программное обеспечение:

- приложение позволяющее просматривать и воспроизводить медиаконтент PDF-файлов «Adobe Acrobat Reader DC»;
- офисный пакет приложений «LibreOffice», Alt Linux.

Специализированное программное обеспечение:

- среда разработки на языке программирования высокого уровня C++;
- задачник Ptaskbook (свободно-распространяемый).

При освоении материала и выполнения заданий по дисциплине рекомендуется использование материалов, размещенных в Личных кабинетах обучающихся ЕТИС ПГНИУ (**student.psu.ru**).

При организации дистанционной работы и проведении занятий в режиме онлайн могут использоваться:

- система видеоконференцсвязи на основе платформы BigBlueButton (<https://bigbluebutton.org/>).
- система LMS Moodle (<http://e-learn.psu.ru/>), которая поддерживает возможность использования текстовых материалов и презентаций, аудио- и видеоконтент, а так же тесты, проверяемые задания, задания для совместной работы.
- система тестирования Indigo (<https://indigotech.ru/>).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Для лекционных занятий требуется аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) с соответствующим программным обеспечением, меловой (и) или маркерной доской.

Для проведения практических занятий - аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) с соответствующим программным обеспечением, меловой (и) или маркерной доской.

Для проведения лабораторных занятий - компьютерный класс. Состав оборудования определен в Паспорте компьютерного класса.

Для групповых (индивидуальных) консультаций - аудитория, оснащенная презентационной техникой

(проектор, экран, компьютер/ноутбук) с соответствующим программным обеспечением, меловой (и) или маркерной доской.

Для проведения текущего контроля - аудитория, оснащенная меловой (и) или маркерной доской.

Самостоятельная работа студентов: аудитория, оснащенная компьютерной техникой с возможностью подключения к сети «Интернет», с обеспеченным доступом в электронную информационно-образовательную среду университета, помещения Научной библиотеки ПГНИУ.

Помещения научной библиотеки ПГНИУ для обеспечения самостоятельной работы обучающихся:

1. Научно-библиографический отдел, корп.1, ауд. 142. Оборудован 3 персональными компьютера с доступом к локальной и глобальной компьютерным сетям.

2. Читальный зал гуманитарной литературы, корп. 2, ауд. 418. Оборудован 7 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

3. Читальный зал естественной литературы, корп.6, ауд. 107а. Оборудован 5 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

4. Отдел иностранной литературы, корп.2 ауд. 207. Оборудован 1 персональным компьютером с доступом к локальной и глобальной компьютерным сетям.

5. Библиотека юридического факультета, корп.9, ауд. 4. Оборудована 11 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

6. Читальный зал географического факультета, корп.8, ауд. 419. Оборудован 6 персональными компьютерами с доступом к локальной и глобальной компьютерным сетям.

Все компьютеры, установленные в помещениях научной библиотеки, оснащены следующим программным обеспечением:

Операционная система ALT Linux;

Офисный пакет Libreoffice.

Справочно-правовая система «КонсультантПлюс»

**Фонды оценочных средств для аттестации по дисциплине
Алгоритмизация и программирование I**

**Планируемые результаты обучения по дисциплине для формирования компетенции.
Индикаторы и критерии их оценивания**

ОПК.2

Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
<p>ОПК.2.2 Анализирует типовые языки программирования, составляет программы</p>	<p>Знание этапов разработки программ, представлений о различных технологиях программирования. Умение - создавать проекты в пакетах Visual Studio C++ или Code::Blocks; - выполнять компиляцию, отладку и тестирование программ; - использовать скалярные типы данных при решении задач; - определять результат работы программы, используя трассировочные таблицы. Владеет основными алгоритмическими структурами.</p>	<p style="text-align: center;">Неудовлетворител Не имеет представление о различных технологиях программирования, этапах разработки программ. Не умеет самостоятельно создавать проекты, выполнять отладку и тестирование программ. Не умеет правильно использовать скалярные типы данных при решении задач. Не умеет правильно использовать основные алгоритмические структуры. Не умеет определять результат работы программы.</p> <p style="text-align: center;">Удовлетворительн Имеет общее представление о различных технологиях программирования, этапах разработки программ. Умеет создавать проекты в пакетах Visual Studio C++ или Code::Blocks. В основном умеет выполнять компиляцию и отладку программ, но не умеет их самостоятельно тестировать. Умеет использовать скалярные типы данных при решении задач. В основном владеет основными алгоритмическими структурами, но не всегда применяет их верно. Плохо умеет определять результат работы программы.</p> <p style="text-align: center;">Хорошо Имеет четкое представление о различных технологиях программирования, этапах разработки программ. Умеет создавать проекты в пакетах Visual Studio C++ или Code::Blocks. Умеет выполнять компиляцию и отладку и тестирование программ. Умеет использовать скалярные типы данных при решении задач. В основном владеет основными алгоритмическими структурами,</p>

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
		<p>Хорошо но не всегда использует их эффективно. Достаточно хорошо умеет определять результат работы программы.</p> <p>Отлично Имеет четкое представление о различных технологиях программирования, этапах разработки программ. Умеет создавать проекты в пакетах Visual Studio C++ или Code::Blocks. Умеет выполнять компиляцию и отладку и тестирование программ. Умеет использовать скалярные типы данных при решении задач. Хорошо владеет основными алгоритмическими структурами и эффективно их применяет. Хорошо умеет определять результат работы программы.</p>
<p>ОПК.2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p>	<p>Владение - методами декомпозиции при написании программ; - разными способами передачи параметров в функцию. Умение - использовать рекурсивные функции; - использовать одномерные и двумерные массивы при решении задач; - выполнять обработку строк. Знание основных алгоритмов сортировки.</p>	<p>Неудовлетворител Не владеет методами декомпозиции при написании программ. Не владеет разными способами передачи параметров в функцию или плохо понимает их отличие. Не умеет использовать рекурсивные функции. Не умеет использовать одномерные и двумерные массивы при решении задач. Не знает основные алгоритмы сортировки. Не умеет выполнять обработку строк с использованием стандартных методов.</p> <p>Удовлетворительн Плохо владеет методами декомпозиции при написании программ. Не всегда верно использует разные способы передачи параметров в функцию. Умеет использовать только самые простые рекурсивные функции. Умеет использовать только статические одномерные и двумерные массивы при решении задач. Знает не более 3 основных алгоритмов сортировки. Не умеет выполнять обработку строк с использованием всех стандартных методов (владеет только 3-4 методами).</p> <p>Хорошо В достаточной степени владеет методами декомпозиции при написании программ. В основном верно использует разные способы передачи параметров в функцию. Умеет</p>

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
		<p>Хорошо использовать рекурсивные функции. Умеет использовать статические и динамические одномерные и двумерные массивы при решении задач. Знает не более 6 основных алгоритмов сортировки. Не умеет выполнять обработку строк с использованием всех стандартных методов (владеет только 5-6 методами).</p> <p>Отлично Хорошо владеет методами декомпозиции при написании программ. Верно использует разные способы передачи параметров в функцию. Умеет использовать рекурсивные функции. Умеет использовать статические и динамические одномерные и двумерные массивы при решении задач. Знает не менее 10 основных алгоритмов сортировки. Умеет выполнять обработку строк с использованием всех стандартных методов..</p>
<p>ОПК.2.1 Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности</p>	<p>Умение - эффективно применять полученные знаний в области программирования при разработке сложной программы; - доказать правильность его работы; - последовательно изложить применяемый в задаче алгоритм решения. Владение навыками разработки "дружественного" интерфейса для пользователя с защитой от неверного ввода данных.</p>	<p>Неудовлетворител Не умеет применять полученные знаний в области программирования при разработке сложной программы (индивидуального проекта), не владеет навыками разработки "дружественного" интерфейса для пользователя с защитой от неверного ввода данных, не умеет последовательно изложить применяемый в задаче алгоритм решения и доказать правильность его работы.</p> <p>Удовлетворительн Умеет применять полученные знаний в области программирования при разработке сложной программы (индивидуального проекта), но использует при этом неэффективные алгоритмы. Не владеет или плохо владеет навыками разработки "дружественного" интерфейса для пользователя с защитой от неверного ввода данных, плохо умеет последовательно изложить применяемый в задаче алгоритм решения и доказать правильность его работы (не умеет правильно выбирать тесты для отладки программы).</p> <p>Хорошо</p>

Компетенция (индикатор)	Планируемые результаты обучения	Критерии оценивания результатов обучения
		<p style="text-align: center;">Хорошо</p> <p>Умеет эффективно применять полученные знаний в области программирования при разработке сложной программы (индивидуального проекта), в достаточной степени владеет навыками разработки "дружественного" интерфейса для пользователя с защитой от неверного ввода данных, умеет в достаточной степени изложить применяемый в задаче алгоритм решения, но не всегда может доказать правильность его работы (подобрать необходимые тесты).</p> <p style="text-align: center;">Отлично</p> <p>Умеет эффективно применять полученные знаний в области программирования при разработке сложной программы (индивидуального проекта), хорошо владеет навыками разработки "дружественного" интерфейса для пользователя с защитой от неверного ввода данных, умеет хорошо и последовательно изложить применяемый в задаче алгоритм решения, может доказать правильность его работы (подобрать необходимые тесты).</p>

Оценочные средства текущего контроля и промежуточной аттестации

Схема доставки : Базовая

Вид мероприятия промежуточной аттестации : Экзамен

Способ проведения мероприятия промежуточной аттестации : Оценка по дисциплине в рамках промежуточной аттестации определяется на основе баллов, набранных обучающимся на контрольных мероприятиях, проводимых в течение учебного периода.

Максимальное количество баллов : 100

Конвертация баллов в отметки

«отлично» - от 81 до 100

«хорошо» - от 61 до 80

«удовлетворительно» - от 41 до 60

«неудовлетворительно» / «незачтено» менее 41 балла

Компетенция (индикатор)	Мероприятие текущего контроля	Контролируемые элементы результатов обучения
Входной контроль	Входной контроль Входное тестирование	Уметь анализировать программу, содержащую конструкции ветвления, циклы и подпрограммы. Знать основные типы данных (числовые, символьный и логический) и уметь работать с ними. Уметь использовать массивы. Уметь составлять простейшие программы.
ОПК.2.2 Анализирует типовые языки программирования, составляет программы	Простые типы данных. Основные алгоритмические структуры. Функции и рекурсия Письменное контрольное мероприятие	Уметь выводить формулу общего члена ряда. Уметь выводить рекуррентное соотношение для заданного ряда. Уметь писать программу для вычисления суммы ряда, используя формулу общего члена ряда. Умение писать программу для вычисления суммы ряда, используя рекуррентное соотношение. Знать основные алгоритмические структуры и уметь их использовать при написании программ.

Компетенция (индикатор)	Мероприятие текущего контроля	Контролируемые элементы результатов обучения
<p>ОПК.2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p> <p>ОПК.2.2 Анализирует типовые языки программирования, составляет программы</p>	<p>Основные структуры данных. Методы сортировок</p> <p>Письменное контрольное мероприятие</p>	<p>Знание методов сортировок и умение их использовать при написании программ. Умение использовать основные структурные типы данных при написании программ. Уметь определять, какую задачу решает программа. Уметь выполнять трассировку программы. Уметь писать эффективные по времени выполнения программы</p>
<p>ОПК.2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения</p> <p>ОПК.2.1 Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности</p>	<p>Индивидуальная работа</p> <p>Защищаемое контрольное мероприятие</p>	<p>Уметь решать задачи "длинной арифметики" на языке C++. Уметь использовать одномерные массивы. Знать основные алгоритмические структуры. Уметь использовать функции. Владеть навыками отладки и тестирования программ.</p>

Компетенция (индикатор)	Мероприятие текущего контроля	Контролируемые элементы результатов обучения
ОПК.2.1 Применяет знания основных положений и концепций в области программирования, архитектуру языков программирования, основную терминологию и базовые алгоритмы, основные требования информационной безопасности ОПК.2.3 Применяет на практике опыт решения задач с использованием базовых алгоритмов, анализа типов коммуникаций и интеграции различных типов программного обеспечения ОПК.2.2 Анализирует типовые языки программирования, составляет программы	Экзамен Итоговое контрольное мероприятие	Знание основных алгоритмических структур и умение применять их при написании программ. Умение использовать функции и рекурсию. Знание основных структурных типов данных и умение их использования. Знание механизма передачи параметров. Умение использовать динамическую память. Знание основных типов сортировок.

Спецификация мероприятий текущего контроля

Входной контроль

Продолжительность проведения мероприятия промежуточной аттестации: **2 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставляемый за мероприятие промежуточной аттестации: **0**

Проходной балл: **0**

Показатели оценивания	Баллы
Предлагается 6 заданий на написание программы	60
Предлагается тест из 10 вопросов с кратким ответом	40

Простые типы данных. Основные алгоритмические структуры. Функции и рекурсия

Продолжительность проведения мероприятия промежуточной аттестации: **6 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставляемый за мероприятие промежуточной аттестации: **30**

Проходной балл: **12.3**

Показатели оценивания	Баллы
Уметь выводить рекуррентное соотношение для заданного ряда.	10
Уметь писать программу для вычисления суммы ряда, используя рекуррентное соотношение.	10

Уметь писать программу для вычисления суммы ряда, используя формулу общего члена ряда.	5
Уметь выводить формулу общего члена ряда.	5

Основные структуры данных. Методы сортировок

Продолжительность проведения мероприятия промежуточной аттестации: **8 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставаемый за мероприятие промежуточной аттестации: **30**

Проходной балл: **12.3**

Показатели оценивания	Баллы
Знание методов сортировок и умение их использовать при написании программ.	10
Уметь писать эффективные по времени выполнения программы	10
Уметь определять, какую задачу решает программа. Уметь выполнять трассировку программы.	5
Умение использовать основные структурные типы данных при написании программ.	5

Индивидуальная работа

Продолжительность проведения мероприятия промежуточной аттестации: **8 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставаемый за мероприятие промежуточной аттестации: **10**

Проходной балл: **4.1**

Показатели оценивания	Баллы
Уметь использовать одномерные и двумерные массивы при создании программного продукта	2
Уметь использовать функции.	2
Уметь создавать пользовательский интерфейс и осуществлять корректность пользовательского ввода.	2
Уметь создавать документацию к программному продукту.	2
Владеть навыками отладки и тестирования программ.	2

Экзамен

Продолжительность проведения мероприятия промежуточной аттестации: **4 часа**

Условия проведения мероприятия: **в часы самостоятельной работы**

Максимальный балл, выставаемый за мероприятие промежуточной аттестации: **30**

Проходной балл: **12.3**

Показатели оценивания	Баллы
Ответ на второй вопрос билета	10
Ответ на первый вопрос билета	10
Решение практических задач	10