|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИфедеральное государственное АВТОНОМНОЕ образовательное учреждение высшего образования«Национальный исследовательский ядерный университет «МИФИ» |
| **Обнинский институт атомной энергетики –** филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»**(ИАТЭ НИЯУ МИФИ)** |

|  |
| --- |
| **УТВЕРЖДАЮ** |
| Начальник отделенияинтеллектуальных кибернетических систем\_\_\_\_\_\_\_\_\_\_\_\_ С.О.Старков |
| «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

|  |
| --- |
| Программирование |
| *название дисциплины* |
|  |
| для студентов направления подготовки |
|  |
| 09.03.02 Информационные системы и технологии  |
|  |
|  |
|  |
| профиль: |
| Информационные технологии |
|  |
|  |
|  |
| Форма обучения: очная |

**г. Обнинск 2021 г.**

Программа составлена в соответствии с образовательным стандартом высшего образования НИЯУ МИФИ по направлению 09.03.02 Информационные системы и технологии

Программу составил (а) (и):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.О. Перетятько, ассистент

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.О. Фамилия, должность, ученая степень, ученое звание

Рецензент:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.О. Фамилия, должность, ученая степень, ученое звание

Программа рассмотрена на заседании отделения интеллектуальных кибернетических систем (О)

(протокол № \_\_\_\_ от «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.)

Руководитель образовательной программы

09.03.02 Информационные системы и технологии

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.А. Мирзеабасов

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

|  |  |
| --- | --- |
|  |  |

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения ОПОП бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

|  |  |  |
| --- | --- | --- |
| Коды компетенций | **Результаты освоения ООП****Содержание компетенций** | **Перечень планируемых результатов обучения по дисциплине** |
| ОПК-2 | Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности | **Знать:**Порядок создания проектов в современных IDE**Уметь:**Использовать современные IDE**Владеть:**Средой программирования Qt Creator |
| ОПК-6 | Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий | **Знать:** основы программирования на языке программирования С и С++;Базовые конструкции и типовые алгоритмы программирования.**Уметь:** писать программы среднего уровня сложности;Использовать базовые конструкции и типовые алгоритмы при программировании;Выполнять отладку написанных программ.**Владеть:** библиотекой Qt и средой разработки Qt Creator.  |

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина реализуется в рамках обязательной части.

Для освоения дисциплины необходимы компетенции, сформированные в рамках изучения следующих дисциплин: ~~перечислить дисциплины~~.

Дисциплины и/или практики, для которых освоение данной дисциплины необходимо как предшествующее: «Объектно-ориентированное программирование», «Операционные системы», «Технологии программирования».

Дисциплина изучается на 1 курсе (ах) в 1,2 семестре(ах).

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

*Вариант № 2: при условии изучения дисциплины в нескольких семестрах*

|  |  |
| --- | --- |
| **Вид работы** | **Форма обучения** (вносятся данные по реализуемым формам) |
| **Очная** | **Заочная** |
| **Семестр** | **Курс**  |
| **№ 1** | **№ 2** | **Всего** |  |  |  |
| **Количество часов на вид работы:** |
| **Контактная работа обучающихся с преподавателем** |  |  |
| **Аудиторные занятия *(всего)*** | 64 | 64 | 128 |  |  |  |
| В том числе: |  |  |  |  |  |  |
| *лекции**(лекции в интерактивной форме)* | 0 | 0 | 0 |  |  |  |
| *практические занятия**(практические занятия в интерактивной форме)* | 32 | 32 | 64 |  |  |  |
| *лабораторные занятия* | 32 | 32 | 64 |  |  |  |
| **Промежуточная аттестация** |  |  |  |  |  |  |
| В том числе: |  |  |  |  |  |  |
| *зачет*  | **-** | **-** | **-** |  |  |  |
| *экзамен* | 54 | **-** | **54** |  |  |  |
| **Самостоятельная работа обучающихся** |  |  |  |  |  |  |
| **Самостоятельная работа обучающихся *(всего)*** | 170 | 116 | 286 |  |  |  |
| В том числе: |  |  |  |  |  |  |
| *ПРИМЕР: проработка учебного (теоретического) материала* | 26 | 20 | 46 |  |  |  |
| *ПРИМЕР: выполнение индивидуальных заданий (подготовка сообщений, презентаций)* | 120 | 80 | 200 |  |  |  |
| *ПРИМЕР: подготовка ко всем видам контрольных испытаний текущего контроля успеваемости (в течение семестра)* | 8 | 8 | 16 |  |  |  |
| *ПРИМЕР: подготовка ко всем видам контрольных испытаний промежуточной аттестации (по окончании семестра)* | 16 | 8 | 24 |  |  |  |
| **Всего (часы):** | **288** | **180** | **468** |  |  |  |
| **Всего (зачетные единицы):** | **8** | **5** | **13** |  |  |  |

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование раздела /темы дисциплины  | Виды учебной работы в часах (вносятся данные по реализуемым формам) |
| Очная форма обучения |
| Лек | Пр | Лаб | Внеауд | СРО |
| 1. | Программирование на языке программирование на С |  |  |  |  |  |
| 1.1. | Основные понятия и простые программы | - | 2 | - | - | 6 |
| 1.2. | Среда программирования Qt – установка, настройка и создание программ-проектов | - | - | 2 | - | 6 |
| 1.3. | Системный подход: алгоритмы и блок-схемы | - | 2 | - | - | 6 |
| 1.4. | Разработка программ: базовые алгоритмы; Псевдокод и выбор решения | - | 2 | - | - | 6 |
| 1.5. | ЯП: алфавит, типы данных, выражения и операции | - | 2 | - | - | 6 |
| 1.6. | Операторы ЯП: присваивание, ввод-вывод, условный, case-switch, комментарий | - | 2 | - | - | 6 |
| 1.7. | Операторы ЯП: Циклы | - | 2 |  | - | 6 |
| 1.8. | АЛГОхитрости-1: решение и циклы | - | - | 2 | - | 6 |
| 1.9. | Структуры данных - Массивы | - | 2 |  | - | 6 |
| 1.10. | АЛГОхитрости-2: массивы | - | - | 2 | - | 6 |
| 1.11. | Функции и библиотеки Си | - | 2 | - | - | 6 |
| 1.12. | Потоковый ввод-вывод в С++; Рекурсия и рекуррентность. Алгоритм Евклида; Ряды; RANDOM. | - | 4 | - | - | 8 |
| 1.13. | Структуры данных – строки | - | 2 | - | - | 8 |
| 1.14. | АЛГОхитрости-3: строки | - | - | 2 | - | 6 |
| 1.15. | Сортировки: пузырёк, выборочная, QuickSort, Шелл; Алгоритмы поиска  | - | - | 4 | - | 6 |
| 1.16. | Передача массивов в подпрограммы + слияние двух массивов; Функции сортировки: выборочная и слияние  | - | - | 2 | - | 6 |
| 1.17. |  Условная трансляция; Макрокоманды Си | - | 2 | - | - | 6 |
| 1.18. |  Структурированные типы данных в Cи/C++: cтруктуры и др. | - | 2 | - | - | 6 |
| 1.19. |  Указатели и динамические переменные | - | 4 | - | - | 8 |
| 1.20. | Алгохитрости: указатели и динамические переменные | - | - | 4 | - | 6 |
| 1.21. |  Си файлы  | - | 2 | - | - | 6 |
| 1.22. |  Системы счисления | - | - | 2 | - | 6 |
| 1.23. |  Алгохитрости: системы счисления | - | - | 2 | - | 6 |
| 1.24. |  Машинное представление данных | - | - | 2 | - | 6 |
| 1.25 |  Распределение памяти | - | - | 4 | - | 8 |
| 1.26 |  Абстрактные структуры данных | - | - | 2 | - | 6 |
| 1.27. |  Структуры данных: стеки, очереди, списки, деревья | - | - | 2 | - | 6 |
|  | **Итого за 1 семестр:**  | **-** | **32** | **32** | **-** | **170** |
| 2. | Программирование на языке программирование на С++ |  |  |  |  |  |
| 2.1. | Системный анализ и основные понятия ООП  | - | 2 | - | - | 6 |
| 2.2. | Инкапсуляция; Классы и объекты | - | 2 | 2 | - | 6 |
| 2.3. | Классы: конструкторы и деструкторы | - | 2 | 2 | - | 6 |
| 2.4. | Некоторые отличия языков Си и С++  | - | - | 2 | - | 6 |
| 2.5. | Пространство имён | - | 2 | 2 | - | 6 |
| 2.6. | Наследование | **-** | 2 | - | **-** | 6 |
| 2.7. | Классы: массивы, указатели | - | 2 | - | - | 6 |
| 2.8. | Полиморфизм: перегрузка функций и операций | - | 2 | 4 | - | 6 |
| 2.9. | Структуры данных: стеки в классах С++ через массивы | - | - | 2 | - | 6 |
| 2.10. | Шаблоны функций и классов | - | 2 |  | - | 6 |
| 2.11. | Пузырьковая сортировка-шаблон | - |  | 2 | - | 6 |
| 2.12. | Виртуальные функции. Абстрактные классы | **-** | 4 | - | **-** | 6 |
| 2.13. | Обработка исключительных ситуаций | - | 2 | 2 | - | 6 |
| 2.14. | RTTI и приведение типов | - | 2 |  | - | 4 |
| 2.15. | Стандартные библиотеки: контейнеры, итераторы и алгоритмы | - | - | 4 | - | 6 |
| 2.16. | ПРГ в среде Qt: модули и объектная модель | - | - | 2 | - | 6 |
| 2.17. | ПРГ в среде Qt: контейнеры | - | - | 2 | - | 6 |
| 2.18. | ПРГ в среде Qt: элементы управления | **-** | - | 2 | **-** | 6 |
| 2.19. | ПРГ в среде Qt: события | - | - | 2 | - | 4 |
| 2.20. | ПРГ в среде Qt: создание приложений, меню, диалоговые окна. SDI и MDI приложения. | - | - | 2 | - | 6 |
|  | **Итого за 2 семестр:**  | - | **32** | **32** | - | **116** |
|  | **Всего:**  | **-** | **64** | **64** | - | **286** |

*Прим.: Лек – лекции, Пр – практические занятия / семинары, Лаб – лабораторные занятия, Внеауд – внеаудиторная работа, СРО – самостоятельная работа обучающихся*

4.2. Содержание дисциплины, структурированное по разделам (темам)

Практические/семинарские занятия

|  |  |  |
| --- | --- | --- |
| № | Наименование раздела /темы дисциплины | Содержание |
| 1. | Программирование на языке программирование на С |
| 1.1. | Основные понятия и простые программы | Базовые понятия и определения в программировании и информатике. Приведены примеры простых программ на ЯП С и Pascal. Приведён список рекомендуемой литературы. |
| 1.3. | Системный подход: алгоритмы и блок-схемы | Основные понятия и принципы системного подхода и анализа. Понятие алгоритма. Принципы построения блок-схем. |
| 1.4. | Разработка программ: базовые алгоритмы; Псевдокод и выбор решения | Описание основных этапов разработки программы. Представление алгоритмов и блок-схем с использованием псевдокода. |
| 1.5. | ЯП: алфавит, типы данных, выражения и операции | Алфавит, понятие данных, типы данных, выражения и операции в ЯП С. |
| 1.6. | Операторы ЯП: присваивание, ввод-вывод, условный, case-switch, комментарий | Оператор присваивания, операторы ввода/вывода данных, условных оператор и оператор множественного выбора в ЯП С с примерами и их описанием в блок-схемах. |
| 1.7. | Операторы ЯП: Циклы | Три типа операторов цикла в ЯП C, с примерами и их описанием в блок-схема. |
| 1.9. | Структуры данных - Массивы | Массивы как структуры данных в ЯП C, с описанием принципов обработки и примерами программ с их использованием. Примеры блок-схем, описывающих алгоритмы работы с массивами. |
| 1.11. | Функции и библиотеки Си | Понятие подпрограммы (функции) и библиотеки в ЯП С, с примерами. |
| 1.12. | Потоковый ввод-вывод в С++; Рекурсия и рекуррентность. Алгоритм Евклида; Ряды; RANDOM. | Операторы потокового ввода/вывода в ЯП С++. Понятие рекурсии и реккуретности. Реализования алгоритма Евклида с помощью рекурсии. Использование генератора случайных величин в С. |
| 1.13. | Структуры данных – строки | Символьные массивы и строки в ЯП С. Библиотеки работы с строками в ЯП С. |
| 1.17. |  Условная трансляция; Макрокоманды Си | Условная компиляция и использование макрокоманд в ЯП С. |
| 1.18. |  Структурированные типы данных в Cи/C++: cтруктуры и др. | Пользовательские, сложные типы данных: структуры, объединения, перечисления и битовые поля. Переименование типов с помощью оператора typedef. |
| 1.19. |  Указатели и динамические переменные | Адресация и распределение памяти для программ. Понятие указателя. Адресная арифметика. Работа с динамической памятью в ЯП С. |
| 1.21. |  Си файлы  | Понятие потока и файла в ЯП. Библиотеки и функции для работы с файлами в ЯП С. |
| 2. | Программирование на языке программирование на С++ |
| 2.1. | Системный анализ и основные понятия ООП  | Понятие ООП. Понятия теории систем и системного анализа (ТСиСА). Системный подход и программирование. Пример программы на ЯП С++. |
| 2.2. | Инкапсуляция; Классы и объекты | Свойство инкапсуляции. Понятие класса и объекта в ООП. Дружественные классы. Примеры классов на ЯП С++. |
| 2.3. | Классы: конструкторы и деструкторы | Понятие конструктора и деструктора в ООП. Их виды и назначение. Примеры конструкторов и деструкторов на ЯП С++. |
| 2.5. | Пространство имён | Понятие и назначение пространства имён. Примеры использования пространства имён в ЯП С++. |
| 2.6. | Наследование | Свойство наследования в ООП. Виды наследования. Примеры наследования на ЯП С++. |
| 2.7. | Классы: массивы, указатели | Массивы объектов и указателей на объекты, и особенности работы с ними. Примеры использование массивов объектов и указателей на объекты в ЯП С++. |
| 2.8. | Полиморфизм: перегрузка функций и операций | Полиморфизм ООП. Виды полиморфизма в ООП. Примеры перегрузки функций и операторов на ЯП С++. |
| 2.10. | Шаблоны функций и классов | Шаблоны как вид полиморфизма в ООП. Шаблоны функций и классов. Перегрузка шаблонной функции. Примеры различных видов шаблонов на ЯП С++.  |
| 2.12. | Виртуальные функции. Абстрактные классы | Виртуальные функции и абстрактные классы как вид полиморфизма в ООП. Примеры виртуальных функций и абстрактных классов на ЯП С++. |
| 2.13. | Обработка исключительных ситуаций | Понятие исключительных ситуаций. Виды исключительных ситуаций. Примеры обработки исключительных ситуация на ЯП С++. |
| 2.14. | RTTI и приведение типов | Понятие динамической идентификации типов (RTTI). |

Лабораторные занятия

|  |  |  |
| --- | --- | --- |
| № | Наименование раздела /темы дисциплины | Название лабораторной работы |
| 1. | Программирование на языке программирование на С |
| 1.1. | Среда программирования Qt – установка, настройка и создание программ-проектов | Краткое описание инструментария разработки Qt. Подробная инструкция установки среды программирования Qt Creator. |
| 1.2. | АЛГОхитрости-1: решение и циклы | Типовые алгоритмические приёмы для работы с циклами. |
| 1.3. | АЛГОхитрости-2: массивы | Типовые алгоритмические приёмы для работы с массивами. |
| 1.14. | АЛГОхитрости-3: строки | Типовые алгоритмические приёмы для работы со строками. |
| 1.15. | Сортировки: пузырёк, выборочная, QuickSort, Шелл; Алгоритмы поиска  | Алгоритм пузырьковой сортировки. Алгоритм выборочной сортировки. Алгоритм сортировки QuickSort. Алгоритм сортировки Шелла. Сравнение временной эффективности разных алгоритмов сортировки. Алгоритмы поиска.  |
| 1.16. | Передача массивов в подпрограммы + слияние двух массивов; Функции сортировки: выборочная и слияние  | Способы передачи параметров в подпрограммы в ЯП С. Примеры реализации выборочной сортировки и сортировки слиянием в виде функций. |
| 1.20. | Алгохитрости: указатели и динамические переменные | Типовые алгоритмические приёмы для работы с указателями и динамическими переменами.  |
| 1.22. |  Системы счисления | Понятие системы счисления и их виды. |
| 1.23. |  Алгохитрости: системы счисления | Типовые алгоритмические приёмы для работы с системами счисления. |
| 1.24. |  Машинное представление данных | Машинное представление разных типов данных в памяти компьютера. |
| 1.25 |  Распределение памяти | Краткий обзор структуры памяти программы. Сегменты памяти. |
| 1.26 |  Абстрактные структуры данных | Описание различных видов структур данных. |
| 1.27. |  Структуры данных: стеки, очереди, списки, деревья | Полустатические и некоторые динамические структуры данных с примерами их реализации на ЯП С. |
| 2. | Программирование на языке программирование на С++ |
| 2.4. | Некоторые отличия языков Си и С++  | Перечислены некоторые отличия языков Си и С++, с примерами. |
| 2.9. | Структуры данных: стеки в классах С++ через массивы | Пример реализации структуры "Стек" в виде класса. |
| 2.11. | Пузырьковая сортировка-шаблон | Пример шаблонной функции, реализующей сортировку методов пузырька. |
| 2.15. | Стандартные библиотеки: контейнеры, итераторы и алгоритмы | Библиотека STL. Библиотеки контейнеров в Qt. Понятие контейнера. Виды контейнеров. Возможные операции с контейнерами. Реализация контейнеров. Понятие алгоритма в котекте контейнеров. Понятие итератора.  |
| 2.16. | ПРГ в среде Qt: модули и объектная модель | Состав и описание основных модулей Qt5. Примеры использование описанных модулей. |
| 2.17. | ПРГ в среде Qt: контейнеры | Особенности библиотеки контейнеров в Qt5. Описание контейнеров в Qt5. Примеры использования описываемых контейнеров. |
| 2.18. | ПРГ в среде Qt: элементы управления | Состав и описание классов-элементов управления. Примеры использования представленных классов. |
| 2.19. | ПРГ в среде Qt: события | Понятие событий и их обработки в Qt. Описание классов, отвечающих для обработкуразличных событий. Примеры использование описанных классов. |
| 2.20. | ПРГ в среде Qt: создание приложений, меню, диалоговые окна. SDI и MDI приложения. | Пример создания приложения с графическим интерфейсом в Qt. Виды графический приложений в Qt – SDI и MDI приложения. Создание меню. Создание диалоговых окон. Создание главного окна. Создание панели инструментов.  |

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

В качестве учебно-методических материалов используется рекомендованная литература.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

6.1. Паспорт фонда оценочных средств по дисциплине

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Контролируемые разделы (темы) дисциплины (результаты по разделам)** | **Код контролируемой компетенции (или её части) / и ее формулировка**  | **Наименование оценочного средства** |
| **Текущий контроль, 1 семестр** |
| 1. | Тема 1.1 – 1.12. | ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий. | Практическое задание №1 |
| 2. | Тема 1.13 – 1.27 | ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности. | Практическое задание №2 |
| **Промежуточный контроль, 1 семестр** |
|  | Экзамен | ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий. | Билеты к экзамену |
| Всего: |
| **Текущий контроль, 2 семестр** |
| 1. | - | - | - |
| 2. | - | - | - |
| **Промежуточный контроль, 2 семестр** |
|  | Зачёт | ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий. | Билеты к зачёту |
| Всего: |

6.2. ***Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы***

*6.2.1. Экзамен*

а) типовые вопросы (задания):

1. Алгоритмы: понятие, свойства, методы представления, управляющие структуры.
2. Процедуры и функции работы с символами и строками в С.
3. Поиск последовательный и бинарный.
4. Рекурсия и рекуррентность. Примеры рекуррентных алгоритмов.
5. Алгоритмы: циклы с предусловием, с постусловием, параметрические.
6. Подпрограммы в ЯП. Функции в С. Библиотеки подпрограмм.
7. Алфавит ЯП С. Лексемы, идентификаторы, структура программы. Знаки операций и разделители.
8. Последовательность разработки программ.
9. Сортировка: алгоритмы пузырьковой и выборочной сортировки.
10. Одномерные массивы в С.
11. Типы данных ЯП С. Приведение типов данных. Константы и переменные.
12. Операторы ЯП С. Выбор решения: бинарный и множественный выбор.
13. Подпрограммы в С. Рекурсивные функции.
14. Функции для работы с указателями и динамической памятью в С и С++.
15. Выражения: операнд, операция, выражение, приоритет операций.
16. Указатели и динамическая память в С. Понятие об адресах и указателях, объявление и использование указателей.
17. Выражения: определения, операнды и операции, правила записи выражений; арифметические и логические операции.
18. Условная трансляция. Препроцессор С. Макрокоманды.
19. Сортировка: алгоритм быстрой сортировки.
20. Файлы. Управление потоком. Функции для чтениязаписи.
21. Операторы ввода с клавиатуры и вывод на экран: форматный ввод/вывод в С.
22. Рекурсия и рекуррентность: числа Фибоначчи.
23. Операторы ввода с клавиатуры и вывод на экран: потоковый ввод/вывод в С.
24. Файлы. Функции для открытия/создания, закрытия файлов и работы с буфером.
25. Операторы С: переход в теле операторов выбора решение и циклов.
26. Функции для работы с указателями в С.
27. Свойства алгоритма. Методы разработки и способы представления алгоритмов.
28. Операторы цикла в С: цикл с параметром, вложенные циклы.
29. Файлы. Функции работы с файловой системой: создание, удаление, переименование файлов;
30. Операторы цикла в С: цикл с предварительным условием.
31. Основные понятия и утверждения целочисленной арифметики.
32. Операторы цикла в С: цикл с последующим условием.
33. Файлы. Перемещение указателя текущей позиции внутри файла
34. Системы счисления. Перевод из одной системы счисления в другую.
35. Сортировка: метод Шелла.
36. Размер и диапазоны типов данных С. Перечислимый тип данных.
37. Целочисленные алгоритмы: алгоритм Евклида (первая и вторая модификации).
38. Строки в С. Способы создания строки. Варианты задания условия цикла прохода до конца строки.
39. Целочисленные алгоритмы: алгоритм Евклида (первая модификация).
40. Правила описания символьных строк в С. Литералы.
41. Алгоритмы: блоксхемы (графические элементы) и псевдокод.
42. Выражения и операции: арифметические и логические.
43. Оператор присваивания. Простой, составной и пустой операторы.
44. Многомерные массивы в С.
45. Алгоритмы: элементарные базовые управляющие структуры.
46. Процедуры и функции работы с символами и строками в С.
47. Файлы. Стандартные потоки в С. Привязка другого физического файла к логическому.
48. Структуры в С. Принципы работы, разновидности полей структур.

б) критерии оценивания компетенций (результатов):

Оцениваются уровень освоения обучающимся материала, предусмотренного учебной программой, уровень владения профессиональными терминами, умение обучающегося применять теоретические знания при решении практических задач и обосновывать свои суждения.

в) описание шкалы оценивания:

Ответ студента на зачёте оценивается согласно следующей таблице:

|  |  |
| --- | --- |
| **Оценка** | **Характеристики ответа студента** |
| 76 – 100% | – знания отличаются глубиной и содержательностью, даны логично построенные, полные, исчерпывающие ответы на вопросы; – обучающийся демонстрирует способность к анализу практических ситуаций, оперирует профессиональными терминами; – обучающийся умеет высказывать и обосновать свои суждения; |
| 51 – 75% | – знания имеют достаточный содержательный уровень, однако имеются определенные затруднения в ответе на уточняющие вопросы; – несущественные ошибки в определении понятий и категорий и т. п., кардинально не меняющие суть изложения; |
| 26 – 50% | – знания имеют фрагментарный характер, имеются определенные неточности и погрешности в формулировках, возникают затруднения при ответе на уточняющие вопросы; – при ответе на вопросы обучающийся не может обосновать закономерности, принципы, объяснить суть явления; допущены фактические ошибки; – обучающийся продемонстрировал слабое умение формулировать выводы и обобщения, |
| 0 – 25% | – обучающийся слабо ориентируется в теме задания, затрудняется дать ответ или дает неверные ответы на поставленные вопросы;– допущены существенные фактические ошибки при ответах на вопросы. |

*6.2.1. Зачёт*

а) типовые вопросы (задания):

1. Классы и объекты. Объявление и описание. Массивы объектов. Дружественные функции и классы.
2. Полиморфизм. Виды полиморфизма в С++. Шаблоны. Стандартные типы данных в шаблонах. Аргументы по умолчанию в шаблонных классах.
3. Обработка исключений в С++. Исключения в конструкторах и деструкторах.
4. Классы и объекты. Определение методов класса. Перегрузка методов. Вложенные классы.
5. Полиморфизм. Виртуальные функции. Статический и динамический полиморфизм. Механизмы связывания.
6. Пространство имён. Объявление пространства имён. Директриса using. Особенности пространства имен.
7. Наследование. Множественное наследование. Конструкторы и деструкторы при наследовании.
8. Полиморфизм. Перегрузка функций. Алгоритм выбора перегружаемой функции.
9. Обработка исключений в С++. Структура встроенного в С++ механизма обработки ошибок. Функции terminate() и unexpected().
10. Классы и объекты. Данные и функции класса. Объявление объектов класса. Порядок вызова конструкторов и деструкторов. Указатели на объекты, указатель this.
11. Полиморфизм. Шаблоны. Стандартные типы данных в шаблонах. Аргументы по умолчанию в шаблонных классах. Абстрактные классы. Чисто виртуальный метод.
12. Классы и объекты. Конструкторы. Инициализаторы в конструкторах. Конструктор по умолчанию. Конструктор с параметрами.
13. Полиморфизм. Виды полиморфизма в С++. Шаблоны функций и классов. Введение в стандартную библиотеку шаблонов.
14. Обработка исключений в С++. Исключения Обработка производных исключительных ситуаций.
15. Классы и объекты. Конструкторы и деструкторы. Описание и вызов. Порядок вызова конструкторов и деструкторов.
16. Контейнеры С++. Характеристики и реализация контейнеров. доступа к компонентам контейнеров.
17. Наследование. Открытое, защищённое и закрытое наследование. Указатели на производные типы. Указатели на члены класса.
18. Полиморфизм. Перегрузка операторов. Ограничения на перегруженные операторы
19. Контейнеры С++. Операции с контейнерами. Реализация контейнеров.
20. Классы и объекты. Массивы объектов. Указатели на объекты, указатель this. Вложенные классы.
21. Полиморфизм. Шаблоны. Ограничения на обобщенные функции. Свойства шаблонов классов. Ключевые слова typename и export.
22. Обработка исключений в С++. Исключения в конструкторах и деструкторах. Обработка производных исключительных ситуаций.
23. Инкапсуляция. Классы и объекты. Объявление и описание. Конструкторы. Инициализаторы в конструкторах. Конструктор по умолчанию. Конструктор с параметрами. Конструктор копирования. Деструкторы.
24. Полиморфизм. Виды полиморфизма в С++. Перегрузка функций. Алгоритм выбора перегружаемой функции. Перегрузка операторов. Ограничения на перегруженные операторы.
25. Строки в Qt. Регулярные выражения. Произвольный тип QVariant.
26. Классы и объекты. Определение методов класса. Перегрузка методов. Вложенные классы.
27. Пространство имён. Объявление пространства имён. Неименованные пространства имен. Пространства имен std.
28. Полиморфизм. Виды полиморфизма в С++. Шаблоны. Стандартные типы данных в шаблонах. Аргументы по умолчанию в шаблонных классах.

б) критерии оценивания компетенций (результатов):

Оцениваются уровень освоения обучающимся материала, предусмотренного учебной программой, уровень владения профессиональными терминами, умение обучающегося применять теоретические знания при решении практических задач и обосновывать свои суждения.

в) описание шкалы оценивания:

Ответ студента на зачёте оценивается согласно следующей таблице:

|  |  |
| --- | --- |
| **Оценка** | **Характеристики ответа студента** |
| 76 – 100% | – знания отличаются глубиной и содержательностью, даны логично построенные, полные, исчерпывающие ответы на вопросы; – обучающийся демонстрирует способность к анализу практических ситуаций, оперирует профессиональными терминами; – обучающийся умеет высказывать и обосновать свои суждения; |
| 51 – 75% | – знания имеют достаточный содержательный уровень, однако имеются определенные затруднения в ответе на уточняющие вопросы; – несущественные ошибки в определении понятий и категорий и т. п., кардинально не меняющие суть изложения; |
| 26 – 50% | – знания имеют фрагментарный характер, имеются определенные неточности и погрешности в формулировках, возникают затруднения при ответе на уточняющие вопросы; – при ответе на вопросы обучающийся не может обосновать закономерности, принципы, объяснить суть явления; допущены фактические ошибки; – обучающийся продемонстрировал слабое умение формулировать выводы и обобщения, |
| 0 – 25% | – обучающийся слабо ориентируется в теме задания, затрудняется дать ответ или дает неверные ответы на поставленные вопросы;– допущены существенные фактические ошибки при ответах на вопросы. |

*6.2.3. Практическое задание №1 : Рекурсия и рекуррентность*

а) типовые задания - образец:

**Recur1.** Описать рекурсивную функцию Fact(N) вещественного типа, вычисляющую значение факториала

N! = 1 \* 2 \*... N (N > 0 — параметр целого типа).

С помощью этой функции вычислить факториалы пяти данных чисел.

**Recur2.** Описать рекурсивную функцию Fact2(N) вещественного типа, вычисляющую значение двойного факториала

N!! = N \* (N-2) \* (N-4) \*...

(N > 0 — параметр целого типа; последний сомножитель в произведении равен 2, если N - четное число, и 1, если N - нечетное).

С помощью этой функции вычислить двойные факториалы пяти данных чисел.

**Recur3.** Описать рекурсивную функцию PowerN(X, N) вещественного типа, находящую значение N-й степени числа X по формулам:

X0 = 1,

X N = (X N/2)2 при четных N > 0,

X N = X \* X N—1 при нечетных N > 0,

XN = 1/X —N при N < 0

(X ≠ 0 - вещественное число, N - целое; в формуле для четных N должна использоваться операция целочисленного деления).

С помощью этой функции найти значения XN для данного X при пяти данных значениях N.

**Recur4.** Описать рекурсивную функцию Fib1(N) целого типа, вычисляющую N-й элемент последовательности чисел Фибоначчи (N - целое число):

F1 = F2 = 1, Fk = Fk-2 + Fk-1, K = 3,4, ... .

С помощью этой функции найти пять чисел Фибоначчи с данными номерами, и вывести эти числа вместе с количеством рекурсивных вызовов функции Fib1, потребовавшихся для их нахождения.

**Recur5.** Описать рекурсивную функцию Fib2(N) целого типа, вычисляющую N-й элемент последовательности чисел Фибоначчи (N - целое число):

F1 = F2 = 1, Fk = Fk-2 + Fk-1, K = 3,4, ... .

Считать, что номер N не превосходит 20. Для уменьшения количества рекурсивных вызовов создать вспомогательный массив для хранения уже вычисленных чисел Фибоначчи и обращаться к нему при выполнении функции Fib2.

С помощью функции Fib2 найти пять чисел Фибоначчи с данными номерами.

**Recur6.** Описать рекурсивную функцию Fib3(N,K) целого типа, вычисляющую сумму N членов последовательности чисел Фибоначчи, начиная с члена номер К (N и К - целые числа):

F1 = F2 = 1, Fk = Fk-2 + Fk-1, K = 3,4, ... .

С помощью функции Fib3 найти сумму 19 чисел Фибоначчи начиная с 4-го члена.

**Recur7.** Описать рекурсивную функцию Combin1(N, K) целого типа, находящую C(N, K) - число сочетаний из N элементов по K - с помощью рекуррентного соотношения: C (N,0) = C(N, N ) = 1,

C(N, K) = C(N - 1, K) + C(N - 1, K - 1) при 0 < K < N.

Параметры функции - целые числа; N > 0, 0 < K < N. Дано число N и пять различных значений K.

Вывести числа C(N, K) вместе с количеством рекурсивных вызовов функции Combin1, потребовавшихся для их нахождения.

**Recur8.** Описать рекурсивную функцию Combin2(N, K) целого типа, находящую

 C(N, K) - число сочетаний из N элементов по K - с помощью рекуррентного соотношения:

C(N, 0) = C(N, N) = 1,

C(N, K) = C(N - 1, K) + C(N - 1, K - 1) при 0 < K < N.

Параметры функции - целые числа; N > 0, 0 < K < N. Считать, что параметр N не превосходит 20.

С помощью функции Combin2 найти числа C(N, K) для данного значения N и пяти различных значений K.

**Recur9.** Описать рекурсивную функцию RootK(X, K, N) вещественного типа, находящую приближенное значение корня K-й степени из числа X по формуле:

Y0 = 1,

YN+1= Yn - (Yn - X/(Yn )K-1)/K,

где Yn обозначает RootK(X, K, N) при фиксированных X и K.

Параметры функции: X (> 0) - вещественное число, K (> 1) и N (> 0) - целые.

С помощью функции RootK найти для данного числа *X* приближенные значения его корня K-й степени при шести данных значениях N.

**Recur10.**Описать рекурсивную функцию NOD(A, B) целого типа, находящую *наибольший общий делитель* (НОД) двух целых положительных чисел Аи B, используя *алгоритм Евклида:*

НОД(А, B) = НОД(В, Amod B), если B ≠ 0; НОД(А, 0) = A.

С помощью этой функции найти НОД(А, B), НОД(А, C), НОД(А, D), если даны числа А, B, C, D.

**Recur11.** Описать рекурсивную функцию DigitSum(K) целого типа, которая находит сумму цифр целого числа *K,* не используя оператор цикла.

С помощью этой функции найти суммы цифр для пяти данных целых чисел.

**Recur12.** Описать рекурсивную функцию МахЕlem(А, N) целого типа, которая находит максимальный элемент целочисленного массива А размера N (1 < N < 10), не используя оператор цикла.

С помощью этой функции найти максимальные элементы массивов А, B, C размера NA, NB, NC соответственно.

**Recur13.** Описать рекурсивную функцию DigitCount(S) целого типа, которая находит количество цифр в строке S, не используя оператор цикла.

С помощью этой функции найти количество цифр в каждой из пяти данных строк.

**Recur14.** В рекуррентном ряду арифметической прогрессии найдите **аn**с помощью рекурсивной функции, если известны:

1. Первый член ряда – **a1** = -2.5,
2. Шаг прогрессии – **d**=0.4,

не используя формулу n-го члена прогрессии.

**Recur15.** Написать рекурсивную функцию вычисления количества цифр в заданном натуральном числе a.

**Recur16.** Разработать программу для расчета числа сочетаний из n элементов по m (обозначается), используя следующее рекурсивное описание:



(Сочетанием из n элементов по m*,* называется любая комбинация, состоящая из n элементов, взятых из данного множества, которые отличаются, по крайней мере, одним элементом. Под сочетанием можно понимать любое подмножество, содержащее n элементов, взятых из данного множества, состоящего из m элементов.)

**Recur17.** Разработать рекурсивную программу расчета суммы двух натуральных чисел, используя только прибавление единицы.

**Recur18.** Разработать рекурсивную программу вывода на печать цифр десятичного числа в обратном порядке (для числа 492 вывести на печать 294).

**Recur19.** Разработать рекурсивную программу, выводящую на печать квадраты натуральных чисел от 1 до **n**.

**Recur20.** Разработать рекурсивную функцию, которая определяет, является ли заданное натуральное число простым. Используя эту функцию вывести все простые числа на заданном интервале [**a**, **b**].

**Recur21.** Ввести с клавиатуры последовательность из **n** десятичных цифр. Описать рекурсивную функцию, которая превращает последовательность символов в десятичное целое число.

**Recur22.** Описать рекурсивную функцию вычисления значения по формуле:



Используя эту функцию, вычислить значения для различных **n** > 3.

**Recur23.** Написать рекурсивную функцию, определяющуюна какое наименьшее число нужно умножить целое число N, чтобы получить число, состоящее из одних пятерок?

Проверить работу функции с N = 12345679.

**Recur24.** Описать рекурсивную логическую функцию simm (s, i, j), проверяющую, является ли симметричной часть строки s, начинающаяся i-м и кончающаяся j-м ее элементом.

**Recur25.** Найти все трехзначные числа, представимые в виде сумм факториалов своих цифр. Использовать рекурсивную функцию вычисления **n!**.

**Recur26.** Написать программу вычисления суммы факториалов всех нечётных чисел от 1 до 9.

**Recur27.** Ввести Дано натуральное число **n**. Найти **(2n)!** и **2n!**.

**Recur28.** Описать функцию, которая удаляет из строки все лишние пробелы.

Пробелы считаются лишними, если их подряд идет более двух, если они стоят в конце строки после последней точки, если они стоят в начале строки до первого символа не пробела.

**Recur29.** Определите **n-**й член последовательности, в которой каждый следующий член равен сумме квадратов всех предыдущих.

б) критерии оценивания компетенций (результатов):

Оцениваются уровень освоения обучающимся материала, предусмотренного учебной программой, умение обучающегося применять теоретические знания при решении практических задач.

в) описание шкалы оценивания:

Выполнение практических работ оценивается от 0 до 30 баллов. Баллы выставляются пропорционально выполненному заданию согласно следующей таблице:

|  |  |
| --- | --- |
| **Оценка** | **Характеристики ответа студента** |
| 76 – 100% | Студент полностью и правильно выполнил задание. |
| 51 – 75% | Студент выполнил задание с небольшими недочётами. |
| 26 – 50% | Студент выполнил задание, допустим несколько фактических ошибок. |
| 0 – 25% | Стдуент не выполнил задание или допустил существенные ошибки при выполнении задания. |

*6.2.4. Практическое задание №2:* Машинное представление данных и распределение памяти.

а) типовые задания - образец:

**Задание 1.** Машинное представление данных

1. Исследовать размеры памяти для всех типов данных С++ в Qt 5 (целые, символьные, логические, перечисляемые, вещественные, строки) и нарисовать схемы машинного представления данных.
2. Вывести в файл datatype.dtс заданные преподавателем данные и просматривая их на экране в 16-ричном виде определить где какое число и символ (смотреть 16-ричное значение в браузере FAR: F3-Просмотр, F4-Код).
3. Вывести в программе адреса введённых переменных и определить последовательность размещения переменных в оперативной памяти и сколько памяти отводится под каждое значение

**Задание 2.** Распределение памяти

Исследовать:

* Размещение в оперативной памяти ПЭВМ переменных разного типа (в соответствии с их машинным представлением); в каких адресах оперативной памяти (ОП) размещаются глобальные и локальные переменные; определить динамику распределения памяти (порядок следования переменных в ОП);
* Видимость одноимённых переменных, которые объявлены в подпрограммах и способ использования этих переменных (обращения к ним);
* Размещение в памяти подпрограмм (функций): параллельных и вложенных (в т.ч., многоуровневой вложенности); определить в каких адресах ОП размещаются подпрограммы, какова динамика распределения памяти (порядок следования подпрограмм в ОП).

б) критерии оценивания компетенций (результатов):

Оцениваются уровень освоения обучающимся материала, предусмотренного учебной программой, уровень владения профессиональными терминами, умение обучающегося применять теоретические знания при решении практических задач и обосновывать свои суждения.

в) описание шкалы оценивания:

Выполнение практических работ оценивается от 0 до 30 баллов. Баллы выставляются пропорционально выполненному заданию согласно следующей таблице:

|  |  |
| --- | --- |
| **Оценка** | **Характеристики ответа студента** |
| 76 – 100% | Студент ответил на все вопросы, ответы полностью отражают суть вопроса и свидетельствуют о понимании студентом изучаемого материала, в ответах на вопросы используется грамотная терминология. |
| 51 – 75% | Студент ответил на 75% вопросов, ответы в целом отражают суть вопроса и свидетельствуют о понимании студентом изучаемого материала, в ответах на вопросы используется грамотная терминология. Присутствуют несущественные ошибки в определении понятий и категорий и т. п., кардинально не меняющие суть изложения. |
| 26 – 50% | Студент ответил на 50% вопросов, имеются определенные неточности и погрешности в формулировках, возникают затруднения при ответе на уточняющие вопросы. Допущены фактические ошибки. |
| 0 – 25% | Студент, не ответил на более половины вопросов, ответы не отражают суть вопроса и свидетельствуют о непонимании студентом изучаемого материала. Допущены существенные фактические ошибки при ответах на вопросы. |

6.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

 Рейтинговая оценка знаний является интегральным показателем качества теоретических и практических знаний и навыков студентов по дисциплине и складывается из оценок, полученных в ходе текущего контроля и промежуточной аттестации.

 Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы студентов.

 Промежуточная аттестация предназначена для объективного подтверждения и оценивания достигнутых результатов обучения после завершения изучения дисциплины.

 Текущий контроль осуществляется два раза в семестр: контрольная точка № 1 (КТ № 1) и контрольная точка № 2 (КТ № 2).

 Результаты текущего контроля и промежуточной аттестации подводятся по шкале балльно-рейтинговой системы.

|  |  |  |
| --- | --- | --- |
| **Вид контроля** | **Этап рейтинговой системы Оценочное средство** | **Балл** |
| Минимум  | Максимум |
| **Текущий**  | **Контрольная точка № 1** |  |  |
| Практическое задание №1 | 0 | 30 |
| **Контрольная точка № 2** |  |  |
| Практическое задание №2 | 0 | 30 |
| **Промежуточный**  | **Зачет/Экзамен** |  |  |
|  | Экзамен | 0 | 40 |
|  | Зачет | 0 | 40 |
| **ИТОГО по дисциплине** | 0 | 100 |

Методические материалы, определяющие процедуры оценивания:

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная учебная литература:

– Qt 5.10. Профессиональное программирование на С++. - СПб.: БХВ-Петербург, 2018. - 1072 с.: ил.

– C/C++. Программирование на языке высокого уровня / Т.П. Павловская. – СПб.: Питер, 2003. – 461. c.: ил.

б) дополнительная учебная литература:

– Информатика: учебник / Б.В. Соболь [и др.].-Изд. 3-е, дополн. и перераб. –

Ростов н/Д: Феникс, 2007. – 446 [1] с.-(Высшее образование).

– Вирт. Н. Алгоритмы и структуры данных: Пер. с англ. – М.: мир, 1989. – 360 с.: ил.

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее - сеть «Интернет»), необходимых для освоения дисциплины

1. Ресурсы электронно-библиотечной системы Центра информационно-библиотечного обеспечения учебно-научной деятельности НИЯУ МИФИ // URL: [www.library.mephi.ru](http://www.library.mephi.ru)
2. Документация библиотеки Qt // URL: <https://doc.qt.io/all-topics.html>

9. Методические указания для обучающихся по освоению дисциплины

Дисциплина практические занятия и лабораторные работы. Изучение курса завершается зачетом с оценкой.

Успешное изучение курса требует посещения, активной работы на практических занятиях и лабораторных работах, выполнения всех учебных заданий преподавателя, ознакомления с основной и дополнительной литературой.

|  |  |
| --- | --- |
| **Вид учебного занятия** | **Организация деятельности студента** |
| **Практические занятия** |  Практические служат для закрепления изученного материала, развития умений и навыков решения задач. Основная цель проведения практических (семинарских) занятий - формирование у студентов аналитического, творческого мышления путем приобретения практических навыков. Важнейшей составляющей любой формы практических занятий являются упражнения (задания). Основа в упражнении - пример, который разбирается с позиций теории, развитой в лекции. Как правило, основное внимание уделяется формированию конкретных умений, навыков, что и определяет содержание деятельности студентов - решение задач, графические работы, уточнение категорий и понятий науки, являющихся предпосылкой правильного мышления и речи. |
| **Контрольная работа** | Для подготовки к контрольной работе необходима работа с конспектами лекций и семинаров. Студент должен уметь решать задачи, аналогичные задачам, входящим в его домашнее задание. Решение задач нужно излагать подробно и аккуратно, объясняя все действия и указывая правила и формулы, использованные при решении каждой задачи. |
| **Подготовка к экзамену** | При подготовке к экзамену в дополнение к изучению конспектов лекций, учебных пособий и слайдов, необходимо пользоваться учебной литературой, рекомендованной к настоящей программе. При подготовке к экзамену нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить по нескольку типовых задач из каждой темы. При решении задач всегда необходимо уметь качественно интерпретировать итог решения. |

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

***10.1. Перечень информационных технологий***

– Проверка домашних заданий и консультирование посредством электронной почты.

– Использование электронных презентаций при проведении практических занятий.

***10.2. Перечень программного обеспечения***

– Программы, демонстрации видео материалов (проигрыватель «Windows Media Player»).

– Фреймворк для разработки программ на языке программирования C++ («Qt»).

– Среда разработки программ («Qt Creator»).

– Программы для демонстрации и создания презентаций («Microsoft Power Point»).

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

При изучении дисциплины по данному курсу используются учебные мультимедийные аудитории для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенные специализированной мебелью и техническими средствами обучения (переносной экран, переносной проектор, ноутбук). Есть доступ к WI-FI.

12. Иные сведения и (или) материалы

12.1. Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине

При проведении лекционных и практических занятий предусматривается использование систем мультимедиа, компьютерных учебников, тестовых форм опросов.

Лекционные занятия проходят с обсуждением учебного материала, демонстрируемого в форме презентаций на экране с использованием мультимедиа-проектора.

Практические занятия включают в себя решение типовых задач. При проведении практических занятий используются активные и интерактивные формы проведения занятий.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№****пп** | **Наименование темы дисциплины** | **Вид занятий (лекция, семинары, практические занятия)** | **Количество ак. ч.** | **Наименование активных и интерактивных форм проведения занятий** |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| …. |  |  |  |  |

12.2. **Формы организации самостоятельной работы обучающихся (темы, выносимые для самостоятельного изучения; вопросы для самоконтроля; типовые задания для самопроверки**

Представлены в фонде оценочных средств по дисциплине.

12.3. Краткий терминологический словарь

ЯП – язык программирования

ЯПВУ – язык программирования высокого уровня

ООП – объектно-ориентированное программирование