|  |  |
| --- | --- |
| Министерство науки и высшего образования Российской Федерации | |
| Федеральное государственное бюджетное образовательное учреждение | |
| высшего образования | |
| «Российский государственный университет им. А.Н. Косыгина | |
| (Технологии. Дизайн. Искусство)» | |
|  | |
| Институт | Мехатроники и информационных технологий |
| Кафедра | Информационных технологий |

|  |  |  |
| --- | --- | --- |
| **РАБОЧАЯ ПРОГРАММА**  **УЧЕБНОЙ ДИСЦИПЛИНЫ** | | |
| **Технология программирования** | | |
| Уровень образования | бакалавриат | |
| Направление подготовки | 09.03.02 | Информационные системы и технологии |
| Профиль | Информационные системы и технологии в индустрии моды | |
| Срок освоения образовательной программы по очной форме обучения | 4 года | |
| Форма обучения | очная | |

|  |  |  |  |
| --- | --- | --- | --- |
| Рабочая программа учебной дисциплины «Технология программирования»основной профессиональной образовательной программы высшего образования*,* рассмотрена и одобрена на заседании кафедры, протокол № 14 от 05.07.2021 г. | | | |
| Разработчики рабочей программы «Технология программирования»: | | | |
|  | Канд. техн. наук, доцент | А. А. Семенов | |
|  | Старший преподаватель | А. М. Козлов | |
|  |  |  | |
|  |  |  | |
| Заведующий кафедрой: | | канд. техн. наук, доцент И. Б. Разин |

# ОБЩИЕ СВЕДЕНИЯ

* + - 1. Учебная дисциплина «Технология программирования» изучается во втором, третьем семестрах.
      2. Курсовая работа/Курсовой проект – не предусмотрены.

## Формы промежуточной аттестации:

|  |  |
| --- | --- |
| второй семестр | - экзамен |
| третий семестр | - зачет с оценкой |
|  |  |

## Место учебной дисциплины в структуре ОПОП

* + - 1. Учебная дисциплина «Технология программирования» относится к обязательной части.
      2. Основой для освоения дисциплины являются результаты обучения по предшествующим дисциплинам и практикам:
    - Введение в профессию;
    - Информатика.
      1. Результаты обучения по учебной дисциплине, используются при изучении следующих дисциплин:
    - Компьютерная графика;
    - Компьютерное проектирование;
    - Технологии обработки информации;
    - Инфокоммуникационные системы и сети.

# ЦЕЛИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

* + - 1. Целями изучения дисциплины «Технология программирования» являются:
* изучение способов представления и структурирования информации о явлениях и процессах в окружающем мире применительно к своей профессиональной деятельности;
* освоение методов ориентирования и взаимодействия с ресурсами информационной среды, осуществления выбора различных моделей использования информационных и коммуникационных технологий в индустрии моды:
* изучение методов построения алгоритмов и основных этапов разработки и создания современных программных продуктов;
* освоение подходов к построению рациональных диалоговых интерфейсов, ориентированных на пользователя;
* изучение базовых правил и принципов современного объектно-ориентированного и визуального программирования;
* формирование у обучающихся компетенций, установленных образовательной программой в соответствии с ФГОС ВО по данной дисциплине.
  + - 1. Результатом обучения по учебной дисциплине является овладение обучающимися знаниями, умениями, навыками и опытом деятельности, характеризующими процесс формирования компетенций и обеспечивающими достижение планируемых результатов освоения учебной дисциплины.

## Формируемые компетенции, индикаторы достижения компетенций, соотнесённые с планируемыми результатами обучения по дисциплине:

| **Код и наименование компетенции** | **Код и наименование индикатора**  **достижения компетенции** | **Планируемые результаты обучения**  **по дисциплине** |
| --- | --- | --- |
| ОПК-2  Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности | ИД-ОПК-2.1  Понимание базовых принципов современных информационных технологий, видов программных средств, в том числе отечественного производства | - Оценивает сущность и значение информационных технологий в развитии современного общества; владеет основными методами, способами и средствами получения, хранения, переработки информации;  - Использует навыки работы с программными средствами для управления информацией и коммуникации с соблюдением основных требований информационной безопасности.  - Применяет методики использования программных средств для решения практических задач в профессиональной деятельности;  - Умеет разрабатывать современные эффективные интерфейсы «человек - электронно-вычислительная машина»;  - Рационально оценивает и обосновывает принимаемые проектные решения, составляет блок-схемы алгоритмов и реализует их на языках программирования высокого уровня;  - Осуществляет практическую реализацию информационно-коммуникационных программ и компьютерных приложений, выполняет тестовые примеры для проверки их корректности и эффективности. |
| ОПК-3  Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности | ИД-ОПК-3.3  Соблюдение основных требований информационной безопасности |
| ОПК-6  Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий | ИД-ОПК-6.1  Понимание принципов программирования для практического применения в области информационных систем и технологий |

# СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

* + - 1. Общая трудоёмкость учебной дисциплины по учебному плану составляет:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| по очной форме обучения – | 8 | **з.е.** | 288 | **час.** |

## Структура учебной дисциплины для обучающихся по видам занятий (очная форма обучения)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Структура и объем дисциплины** | | | | | | | | | |
| **Объем дисциплины по семестрам** | **форма промежуточной аттестации** | **всего, час** | **Контактная аудиторная работа, час** | | | | **Самостоятельная работа обучающегося, час** | | |
| **лекции, час** | **практические занятия, час** | **лабораторные занятия, час** | **практическая подготовка, час** | ***курсовая работа/***  ***курсовой проект*** | **самостоятельная работа обучающегося, час** | **промежуточная аттестация, час** |
| 2 семестр | экзамен | 180 | 36 |  | 54 | 10 |  | 63 | 27 |
| 3 семестр | зачет с оценкой | 108 | 17 |  | 34 | 10 |  | 57 |  |
| Всего: |  | 288 | 53 |  | 88 | 20 |  | 120 | 27 |

## Структура учебной дисциплины для обучающихся по разделам и темам дисциплины: (очная форма обучения)

| **Планируемые (контролируемые) результаты освоения:**  **код(ы) формируемой(ых) компетенции(й) и индикаторов достижения компетенций** | **Наименование разделов, тем;**  **форма(ы) промежуточной аттестации** | **Виды учебной работы** | | | | **Самостоятельная работа, час** | **Виды и формы контрольных мероприятий, обеспечивающие по совокупности текущий контроль успеваемости;**  **формы промежуточного контроля успеваемости** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Контактная работа** | | | |
| **Лекции, час** | **Практические занятия, час** | ***Лабораторные работы/ индивидуальные занятия, час*** | **Практическая подготовка, час** |
|  | **Второйсеместр** | | | | | | |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел I. Основные понятия программирования.** | х | х | х | х | 10 |  |
| Лекция 1.1. Структура программы на языках высокого уровня. | 2 |  |  |  | х | Контроль посещаемости. |
| Лекция 1.2. Типы данных в языках высокого уровня. Скалярные числовые типы. | 2 |  |  |  | х | Контроль посещаемости. |
| Лабораторная работа № 1.1.Простые вычисления с числами целых и вещественных типов. |  |  | 4 | 1 | х | Выполнение лабораторной работы. |
| Лабораторная работа № 1.2. Тригонометрические вычисления и расчёты. |  |  | 4 | 1 | х | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел II. Структурные типы данных.** | х | х | х | х | 10 |  |
| Лекция 2.1. Символьные и строковые типы данных. | 2 |  |  |  | х | Контроль посещаемости. |
| Лекция 2.2. Тип-диапазон, перечисляемые типы, массивы. | 2 |  |  |  | х | Контроль посещаемости. |
| Лекция 2.3. Множества, записи. | 2 |  |  |  | х | Контроль посещаемости. |
| Лабораторная работа № 2.1.Работа с символьными данными. |  |  | 4 | 1 | х | Выполнение лабораторной работы. |
| Лабораторная работа № 2.2.Работа с матрицами (двумерными массивами). |  |  | 6 | 1 | х | Выполнение лабораторной работы. |
| Лабораторная работа № 2.3.Работа с записями. |  |  | 6 | 1 | х | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел III. Подпрограммы.** | х | х | х | х | 10 |  |
| Лекция 3.1. Описание подпрограмм. Параметры подпрограмм. | 2 |  |  |  | х | Контроль посещаемости. |
| Лекция 3.2. Процедурный тип данных. | 2 |  |  |  | х | Контроль посещаемости. |
| Лекция 3.3. Структура модулей. Интерфейсная и исполняемая часть модулей. | 2 |  |  |  | х | Контроль посещаемости. |
| Лекция 3.4. Создание и использование библиотек DLL. | 2 |  |  |  | х | Контроль посещаемости. |
| Лабораторная работа № 3.1.Использование подпрограмм и модулей. |  |  | 6 | 1 | х | Выполнение лабораторной работы. |
| Лабораторная работа № 3.2.Использование библиотек DLL. |  |  | 6 | 1 | х | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел IV. Файлы.** | х | х | х | х | 10 |  |
| Лекция 4.1. Способы доступа к различным типам файлов. | 2 |  |  |  |  | Контроль посещаемости. |
| Лекция 4.2. Основные операторы для работы с различными типами файлов. | 2 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 4.1.Поиск файлов. Работа с найденными текстовыми файлами. |  |  | 6 | 1 |  | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел V. Основы объектно-ориентированного программирования.** | х | х | х | х | 11 |  |
| Лекция 5.1. Принципы объектно-ориентированного программирования. | 2 |  |  |  | х | Контроль посещаемости. |
| Лекция 5.2. Общая характеристика объектов и классов. | 2 |  |  |  | х | Контроль посещаемости. |
| Лекция 5.3. Использование визуальных компонентов современных сред программирования. | 2 |  |  |  | х | Контроль посещаемости. |
| Лекция 5.4. События и их обработка. | 2 |  |  |  | х | Контроль посещаемости. |
| Лабораторная работа № 5.1.Статическая графика в окнах Windows. |  |  | 6 | 1 | х | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел VI. Использование динамической памяти.** | х | х | х | х | 12 |  |
| Лекция 6.1. Использование указателей. | 2 |  |  |  |  | Контроль посещаемости. |
| Лекция 6.2. Выделение и освобождение динамической памяти. | 2 |  |  |  |  | Контроль посещаемости. |
| Лекция 6.2. Жизненный цикл программы. | 2 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 6.1.Динамическая графика в окнах Windows. |  |  | 6 | 1 |  | Выполнение лабораторной работы. |
|  | Экзамен | х | х | х | х | 27 | Электронное тестирование. |
|  | **ИТОГО за второйсеместр** | **36** |  | **54** | **10** | **90** | **Экзамен** |
|  | **Третий семестр** | | | | | | |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел VII. Введение в C++** | 1 | х | 2 | х | 9 |  |
| Лекция 7.1. Характеристика языка программирования C++ и структура программы на С++ | 1 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 7.1. Линейное программирование |  |  | 2 |  |  | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел VIII. Переменные и функции стандартной библиотеки** | 2 | х | 2 | 1 | 6 |  |
| Лекция 8.1. Переменные и функции стандартной библиотеки С++ | 2 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 8.1. Расчетные задачи |  |  | 2 | 1 |  | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел IX. Базовые конструкции программирования и циклы** | 2 | х | 2 | 1 | 6 |  |
| Лекция 9.1. Базовые конструкции программирования и циклы | 2 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 9.1. Циклы |  |  | 2 | 1 |  | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел X. Ветвления и передача управления** | 2 | х | 4 | 1 | 6 |  |
| Лекция 10.1. Ветвления и передача управления | 2 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 10.1. Ветвления и циклы |  |  | 4 | 1 |  | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел XI. Массивы и строки** | 2 | х | 4 | 2 | 6 |  |
| Лекция 11.1. Массивы и строки | 2 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 11.1. Массивы |  |  | 2 | 1 |  | Выполнение лабораторной работы. |
| Лабораторная работа № 11.2. Строки типа char |  |  | 2 | 1 |  |  |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел XII. Функции** | 2 | х | 4 | 1 | 6 |  |
| Лекция 12.1. Функции | 2 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 12.1. Функции |  |  | 4 | 1 |  | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел XIII. Указатели и динамическое выделение памяти** | 2 | х | 4 | 1 | 6 |  |
| Лекция 13.1. Указатели и динамическое выделение памяти | 2 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 13.1. Указатели и функции. Динамическое выделение памяти |  |  | 4 | 1 |  | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел XIV. Потоки и файлы** | 2 | х | 4 | 1 | 6 |  |
| Лекция 14.1. Потоки и файлы | 2 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 14.1. Потоковые классы С++ и файлы. Текстовые и бинарные файлы |  |  | 4 | 1 |  | Выполнение лабораторной работы. |
| ОПК-2:  ИД-ОПК-2.1  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 | **Раздел XV. Технологии программирования. Структуры** | 2 | х | 8 | 2 | 6 |  |
| Лекция 15.1. Технологии программирования. Структуры | 2 |  |  |  |  | Контроль посещаемости. |
| Лабораторная работа № 15.1. Массивы и функции |  |  | 2 |  |  | Выполнение лабораторной работы. |
| Лабораторная работа № 15.2. Простая графика |  |  | 2 | 1 |  | Выполнение лабораторной работы. |
| Лабораторная работа № 15.3. Графика. Структура. Периметр и площадь |  |  | 4 | 1 |  | Выполнение лабораторной работы. |
|  | Зачет с оценкой | х | х | х | х | х | Зачет проводится в письменной форме по билетам согласно программе зачета. |
|  | **ИТОГО за третий семестр** | **17** |  | **34** | **10** | **57** | **Зачет с оценкой** |
|  | **ИТОГО за весь период** | **53** |  | **88** | **20** | **120** |  |

## Краткое содержание учебной дисциплины

|  |  |  |
| --- | --- | --- |
| **№ пп** | **Наименование раздела и темы дисциплины** | **Содержание раздела (темы)** |
| **Второй семестр** | | |
| **Раздел I** | **Основные понятия программирования** | |
| Лекция 1.1 | Структура программы на языках высокого уровня. | Введение. Основные понятия программирования.  Структура программы на языках-компиляторах в общем виде с модулями, функциями и процедурами. Основные операторы. Отладка и компиляция программ. Изучение программной оболочки Lazarus, компилятора и основных операторов языка Object Pascal. |
| Лекция 1.2 | Типы данных в языках высокого уровня. Скалярные числовые типы. | Типы данных и их представление в двоичном коде. Различие операций с разными типами данных. Основные операторы и операции для работы с разными типами. Важность использования функций преобразования типов и форматирования в операциях ввода-вывода. Целесообразность и эффективность при выборе типов данных. Порядковые типы данных. Логические типы данных. Целые типы данных. Вещественные типы данных. |
| Лабораторная работа № 1.1 | Простые вычисления с числами целых и вещественных типов. | Использование арифметических операций, процедур и функций для работы с целыми и вещественными типами данных согласно вариантам заданий. |
| Лабораторная работа № 1.2 | Тригонометрические вычисления и расчёты. | Ввод данных для тригонометрических вычислений согласно вариантам заданий. Расчёты по соответствующим формулам и вывод результатов. |
| **Раздел II** | **Структурные типы данных** | |
| Лекция 2.1 | Символьные и строковые типы данных. | Однобайтные и двухбайтные символьные данные. Операции с символьными данными. Связь символов и их кодов в различных кодовых таблицах. Строковые типы данных как массивы символов. Операторы для работы со строками. Преобразования строк разных типов. |
| Лекция 2.2 | Тип-диапазон, перечисляемые типы, массивы. | Использование типа-диапазона в операторах множественного выбора и при объявлении массивов. Применение перечисляемых типов данных для работы с ассоциативными массивами. Массивы как способ создания последовательных списков однородных данных. Диапазоны индексов массивов. Обращение к элементам массивов. Особенности использования одномерных и многомерных массивов. |
| Лекция 2.3 | Множества, записи. | Особенности операций с типом-множество. Количество возможных присваиваемых значений переменным типа множество. Примеры программ, использующих тип-множество. Записи как основной способ объединения и структурирования данных разных типов. Объявление записей и их использование в программах. Доступ к полям записей. Вариантные поля записей. |
| Лабораторная работа № 2.1 | Работа с символьными данными. | Обработка символьных строк в соответствии с вариантами заданий. Анализ и преобразования символьных данных. |
| Лабораторная работа № 2.2 | Работа с матрицами (двумерными массивами). | Ввод данных в двумерный массив (матрицу). Обработка данных (математические вычисления в строках и столбцах матрицы) в соответствии с вариантами заданий. |
| Лабораторная работа № 2.3 | Работа с записями. | Ввод структурированной информации в поля записей. Создание выборок информации по заданным полям в соответствии с вариантами заданий и вывод результатов вычислений. |
| **Раздел III** | **Подпрограммы** | |
| Лекция 3.1 | Описание подпрограмм. Параметры подпрограмм. | Описание подпрограмм в соответствующих разделах языка высокого уровня. Параметры подпрограмм: параметры-значения, параметры-переменные, параметры-константы. Выбор способа передачи параметров в подпрограмму и возвращения результатов. Различные способы вызова подпрограмм. Рекурсия и опережающее описание. |
| Лекция 3.2 | Процедурный тип данных. | Создание и использование процедурных типов данных. Тип-процедура и тип-функция. Передача подпрограммы в качестве параметра в другую подпрограмму. Вызов внешних подпрограмм из модулей и библиотек. Адрес подпрограммы и его получение для доступа и вызова подпрограммы. |
| Лекция 3.3 | Структура модулей. Интерфейсная и исполняемая часть модулей. | Модульная структура приложений. Понятие «проект» в современных средах разработки программ на языках высокого уровня. Создание файла-модуля. Структура модуля. Интерфейсная и исполняемая часть модулей. Доступ к объявленным в модуле подпрограммам. Подключение модулей к головной программе и вызов подпрограмм. |
| Лекция 3.4 | Создание и использование библиотек DLL | Способы создания файлов на языках высокого уровня для их компиляции в динамически подключаемые библиотеки. Особенности использования библиотек в различных операционных системах. Статический и динамический способы вызова подпрограмм из библиотек. Основные операторы для работы с DLL в операционных системах семейства Windows. |
| Лабораторная работа № 3.1 | Использование подпрограмм и модулей | Оформление подпрограммы в отдельном модуле. Подключение модуля к головной программе и использование подпрограммы.Тип подпрограммы и способ обработки данных указаны в вариантах заданий. |
| Лабораторная работа № 3.2 | Использование библиотек DLL. | Оформление подпрограммы с заданиями по вариантам в библиотеках DLL. Вызов подпрограмм из головной программы статическим и динамическим способами. |
| **Раздел IV** | **Файлы** | |
| Лекция 4.1 | Способы доступа к различным типам файлов. | Понятие о файловых системах как способах структурирования данных. Объявление и доступ к различным типам файлов. Связь понятий «файлы» и «потоки». Имена файлов и логические устройства. Поиск файлов на жестком диске по имени и расширению. Системные функции для работы с файлами. Иерархическая древовидная структура файловой системы в современных операционных системах. Свойства файлов: имя, тип, атрибуты, размер, дата создания/изменения. |
| Лекция 4.2 | Основные операторы для работы с различными типами файлов. | Текстовые, типизированные, нетипизированные файлы – различные способы инициирования для чтения и записи. Последовательный и прямой доступ к данным. Особенности работы с файлами разных типов. Различные операторы для чтения и записи данных. Определение признака конца строки в текстовых файлах последовательного доступа. Перемещение файлового указателя в файлах прямого доступа. Определение признака конца файла. |
| Лабораторная работа № 4.1 | Поиск файлов. Работа с найденными текстовыми файлами. | Ввод произвольного имени текстового файла. Поиск на жестком диске компьютера всех файлов с заданным именем (одноименные файлы могут находиться в разных папках), поочередный вывод на экран их содержимого. Выполнение обработки каждого найденного файла в соответствии с заданием в варианте. |
| **Раздел V** | **Основы объектно-ориентированного программирования** | |
| Лекция 5.1 | Принципы объектно-ориентированного программирования. | Основные понятия и принципы объектно-ориентированного программирования. Инкапсуляция: инфоpмация об объекте (его свойства) и функции объекта (методы) содеpжатся в описании этого объекта. Наследование: один объект может быть постpоен на базе описания дpугого объекта. Полиморфизм: способность многих объектов использовать один и тот же метод, пpи этом выполняемые действия зависят от объекта, котоpый его вызвал. Применение этих принципов при разработке современных программных продуктов. |
| Лекция 5.2 | Общая характеристика объектов и классов. | Структуры объектов и классов в объектно-ориентированных языках высокого уровня. Свойства и методы объектов и классов. Конструктор и деструктор объекта. Объявление объектов в разделе описаний и создание методов в виде подпрограмм. Инициализация полей объектов. Статическое и динамическое создание экземпляров объектов и классов. Использование объектов и классов в современных средах разработки приложений. Библиотеки объектов, классов. |
| Лекция 5.3 | Использование визуальных компонентов современных сред программирования. | Понятия форм и окон в графических операционных системах. Визуальные компоненты в современных средах разработки приложений с графическими интерфейсами. Интеpфейс с гpафическим устpойством – GDI. Основные изобразительные средства операционных ситем семейства Windows: перо, кисть, шрифт. Свойства основных изобразительных средств и их использование в прикладных программах. |
| Лекция 5.4 | События и их обработка. | Возникновение и классификация событий. События, генерируемые операционной системой. События, генерируемые программами и процессами. Воздействия пользователя на программу посредством генерирования событий текствовым устройством ввода (клавиатурой) и графическими указателями (мышь, джойстик, палец на интерактивном экране мобильного смартфона и т.п.). Особенности обpаботки различных типов событий в пpикладных пpогpаммах. |
| Лабораторная работа № 5.1 | Статическая графика в окнах Windows. | Создать окна в операционной системе Windows. Используя функции из библиотеки Lazarus и GDI API Windows, создать в окне Windows изображение по заданию в варианте. В изображении обязательно хотя бы один раз использовать линии, прямоугольники, окружности, многоточечные фигуры-полигоны. Обязательно использовать разные типы и цвета линий, разные типы заливки фигур. Обязательно использовать текстовые надписи. |
| **Раздел VI** | **Использование динамической памяти.** | |
| Лекция 6.1 | Использование указателей. | Понятия об указателях и ссылочной системе адресации данных. Различные типы данных имеют различные размеры областей хранения. Типизированные и нетипизированнные указатели. Определение адреса хранящейся в памяти области данных. Специфика обращения по адресам указателей и ссылкам к разным типам данных. Возможные операции с указателями. Операции с данными, расположенными по адресам указателей. |
| Лекция 6.2 | Выделение и освобождение динамической памяти. | Разные способы выделения и освобождения динамической памяти для типизированных и нетипизированнных указателей. Функции языков высокого уровня выделения и освобождения памяти для разных типов указателей. Определение количества резервируемой памяти для нетипизированных указателей. Создание и использование динамических структур данных в памяти: списков, стеков, очередей, деков. Битовые операции в языке Pascal. |
| Лекция 6.2 | Жизненный цикл программы. | Применение основ объектно-ориентированного программирования для создания многозадачной системы с интерактивным интерфейсом. Основные принципы создания современных графических интерфейсов для удобной и эффективной интерактивной работы пользователей. Usability-технологии при разработке интерфейсов. Методы и способы создания современных конкурентоспособных продуктов для развития отечественного рынка прикладных программ. |
| Лабораторная работа № 6.1 | Динамическая графика в окнах Windows. | Создание окна Windows. Разделение окно линиями на заданное в варианте количество частей. Линии должны изменяться в соответствии с изменениями размеров окна воздействиями пользователя. Написать процедуры для обработки нажатия на клавиши мыши по заданию в варианте. Геометрические фигуры должны иметь центр в точке нажатия «мыши», текстовые сообщения должны выводиться, начиная от точки нажатия «мыши». При перерисовке окна (например, во время изменения его размеров), созданные с помощью «мыши» изображения фигур и текстовые надписи должны сохраняться и обновляться. |
| **Третий семестр** | | |
| **Раздел VII** | **Введение в C++** | |
| Лекция 7.1 | Характеристика языка программирования C++ и структура программы на С++. | Характеристика языка С++. Состав языка и принцип работы компилятора языка высокого уровня. Идентификаторы. Типы данных в С++. Структура программы на языке С++. Создание консольных и визуальных приложений в ИСР C++Builder. Альтернативный ввод-вывод в стиле С. |
| Лабораторная работа № 7.1 | Линейное программирование. | Обзор сред разработки на C++. Изучение основных инструментов интегрированной среды разработки C++Builder, разработка консольных и визуальных проектов. Реализация задач линейного программирования согласно вариантам заданий. |
| **Раздел VIII** | **Переменные и функции стандартной библиотеки** | |
| Лекция 8.1 | Переменные и функции стандартной библиотеки С++ | Переменные в С++. Константы. Манипуляторы. Преобразования типов. Арифметические операции. Функции стандартной библиотеки. |
| Лабораторная работа № 8.1 | Расчетные задачи | Применение арифметических операций для работы с целыми и вещественными типами данных. Реализация расчетных задач согласно вариантам заданий. |
| **Раздел IX** | **Базовые конструкции программирования и циклы** | |
| Лекция 9.1 | Базовые конструкции программирования и циклы | Операции отношения и логические операции. Базовые конструкции структурного программирования. Циклы. Цикл с параметром (for). Цикл с предусловием (while). Цикл с постусловием (do while). Рекомендации по программированию циклов. |
| Лабораторная работа № 9.1 | Циклы | Применение циклов для реализации расчетных задач согласно вариантам заданий. |
| **Раздел X** | **Ветвления и передача управления** | |
| Лекция 10.1 | Ветвления и передача управления | Ветвления. Условный оператор if. Вложенные конструкции. Проблемы соответствия if и else во вложенных ветвлениях. Условная операция. Оператор switch. Операторы передачи управления. Приоритеты операций в С++. |
| Лабораторная работа № 10.1 | Ветвления и циклы | Циклические конструкции с условиями и оператором switch. Применение ветвлений и циклов для реализации расчетных задач согласно вариантам заданий. |
| **Раздел XI** | **Массивы и строки** | |
| Лекция 11.1 | Массивы и строки | Массивы. Сортировка массива методом выбора. Сортировка массива методом пузырька. Многомерные массивы. Строки типа char. Операции со строками. |
| Лабораторная работа № 11.1 | Массивы | Реализация операций с одномерными и двумерными массивами согласно вариантам заданий. |
| Лабораторная работа № 11.2 | Строки типа char | Реализация операций со строками типа char согласно вариантам заданий. |
| **Раздел XII** | **Функции** | |
| Лекция 12.1 | Функции | Виды функций в C++. Передача аргументов в функцию по значению. Возвращение функцией значения. Передача аргументов по ссылке. Возвращение значения по ссылке. Перегруженные функции. Переменные и функции. Передача массивов в функции. |
| Лабораторная работа № 12.1 | Функции | Процедурное программирование. Применение функций при разработке консольных и визуальных приложений согласно вариантам заданий. |
| **Раздел XIII** | **Указатели и динамическое выделение памяти** | |
| Лекция 13.1 | Указатели и динамическое выделение памяти | Характеристика и синтаксис указателей в С++. Указатели и массивы. Указатели и функции. Указатели и строки. Динамический массив и динамическое выделение памяти. |
| Лабораторная работа № 13.1 | Указатели и функции. Динамическое выделение памяти | Применение указателей при передаче аргументов в функции на примере обработки одномерных и двумерных динамических массивов при разработке консольных и визуальных приложений согласно вариантам заданий. |
| **Раздел XIV** | **Потоки и файлы** | |
| Лекция 14.1 | Потоки и файлы | Ввод/вывод. Потоки С++ и файлы. Форматированный ввод/вывод. Конструкторы и методы. Двоичный ввод/вывод. Бинарные файлы. Двунаправленный ввод/вывод. Функции стандартной библиотеки для работы с файлами или ввод/вывод в стиле С. |
| Лабораторная работа № 14.1 | Потоковые классы С++ и файлы. Текстовые и бинарные файлы | Работа с текстовыми и бинарными файлами при разработке консольных и визуальных приложений согласно вариантам заданий. |
| **Раздел XV** | **Технологии программирования. Структуры** | |
| Лекция 15.1 | Технологии программирования. Структуры | Технологии программирования. Правила кодирования и документирования программы. Структуры. Структуры и функции. Массивы структур. Поиск в массиве структур. Вложенность структур. Рекурсия. Алгоритм быстрой сортировки. Массивы структур и бинарные файлы. |
| Лабораторная работа № 15.1 | Массивы и функции | Обработка массивов посредством функций при разработке консольных и визуальных приложений согласно вариантам заданий. |
| Лабораторная работа № 15.2 | Простая графика | Работа с компонентом Image. Построение графических примитивов: отрезок, круг, сектор, эллипс, прямоугольник, полигон, полилиния. Рисование на холсте с динамической закраской. Выбор цвета, толщины и формы кисти. |
| Лабораторная работа № 15.3 | Графика. Структура. Периметр и площадь | Работа с графикой с применением структуры фигур:  struct Koordinata {  int x;  int y;  };.  Построение многоугольников по точкам, сохранение их в текстовый файл, очистка холста, загрузка данных из файла и восстановление нарисованных многоугольников. Расчет периметра и площади нарисованных и загруженных из файла многоугольников. |

## Организация самостоятельной работы обучающихся

Самостоятельная работа студента – обязательная часть образовательного процесса, направленная на развитие готовности к профессиональному и личностному самообразованию, на проектирование дальнейшего образовательного маршрута и профессиональной карьеры.

Самостоятельная работа обучающихся по дисциплине организована как совокупность аудиторных и внеаудиторных занятий и работ, обеспечивающих успешное освоение дисциплины.

Аудиторная самостоятельная работа обучающихся по дисциплине выполняется на учебных занятиях под руководством преподавателя и по его заданию*.* Аудиторная самостоятельная работа обучающихся входит в общий объем времени, отведенного учебным планом на аудиторную работу, и регламентируется расписанием учебных занятий.

Внеаудиторная самостоятельная работа обучающихся – планируемая учебная, научно-исследовательская, практическая работа обучающихся, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия, расписанием учебных занятий не регламентируется.

Внеаудиторная самостоятельная работа обучающихся включает в себя:

подготовку к лекциям, практическим занятиям, лабораторным работам и экзамену;

изучение специальной рекомендованной литературы;

изучение разделов/тем, не выносимых на лекции и практические занятия самостоятельно;

подготовка к выполнению лабораторных работ;

подготовка к практическим занятиям*;*

подготовка к компьютерному тестированию на промежуточных аттестациях;

выполнение индивидуальных заданий;

подготовка к промежуточной аттестации в течение семестра.

Самостоятельная работа обучающихся с участием преподавателя в форме иной контактной работы предусматривает групповую и (или) индивидуальную работу с обучающимися и включает в себя:

проведение индивидуальных и групповых консультаций по отдельным темам/разделам дисциплины;

проведение консультаций перед экзаменом, перед зачетом с оценкой;

консультации по организации самостоятельного изучения отдельных разделов/тем, базовых понятий учебных дисциплин профильного/родственного бакалавриата, которые формировали ОПК и ПК, в целях обеспечения преемственности образования.

Перечень разделов/тем, полностью или частично отнесенных на самостоятельное изучение с последующим контролем:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ пп** | **Наименование раздела /темы *дисциплины,* выносимые на самостоятельное изучение** | **Задания для самостоятельной работы** | **Виды и формы контрольных мероприятий**  **(учитываются при проведении текущего контроля)** | **Трудоемкость, час** |
| **Раздел I** | **Основные понятия программирования** | | | |
| Лабораторная работа № 1.1 | Простые вычисления с числами целых и вещественных типов. | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 5 |
| Лабораторная работа № 1.2 | Тригонометрические вычисления и расчёты. | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 5 |
| **Раздел II** | **Структурные типы данных** | | | |
| Лабораторная работа № 2.1 | Работа с символьными данными. | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 3 |
| Лабораторная работа № 2.2 | Работа с матрицами (двумерными массивами). | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 3 |
| Лабораторная работа № 2.3 | Работа с записями. | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 4 |
| **Раздел III** | **Подпрограммы** | | | |
| Лабораторная работа № 3.1 | Использование подпрограмм и модулей | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 5 |
| Лабораторная работа № 3.2 | Использование библиотек DLL. | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 5 |
| **Раздел IV** | **Файлы** | | | |
| Лабораторная работа № 4.1 | Поиск файлов. Работа с найденными текстовыми файлами. | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 10 |
| **Раздел V** | **Основы объектно-ориентированного программирования** | | | |
| Лабораторная работа № 5.1 | Статическая графика в окнах Windows. | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 11 |
| **Раздел VI** | **Использование динамической памяти.** | | | |
| Лабораторная работа № 6.1 | Динамическая графика в окнах Windows. | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 12 |
| **Раздел VII** | **Введение в C++** | | | |
| Лабораторная работа № 7.1 | Линейное программирование. | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 9 |
| **Раздел VIII** | **Переменные и функции стандартной библиотеки** | | | |
| Лабораторная работа № 8.1 | Расчетные задачи. | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 6 |
| **Раздел IX** | **Базовые конструкции программирования и циклы** | | | |
| Лабораторная работа № 9.1 | Циклы | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 6 |
| **Раздел X** | **Ветвления и передача управления** | | | |
| Лабораторная работа № 10.1 | Ветвления и циклы | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 6 |
| **Раздел XI** | **Массивы и строки** | | | |
| Лабораторная работа № 11.1 | Массивы | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 3 |
| Лабораторная работа № 11.2 | Строки типа char | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 3 |
| **Раздел XII** | **Функции** | | | |
| Лабораторная работа № 12.1 | Функции | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 6 |
| **Раздел XIII** | **Указатели и динамическое выделение памяти** | | | |
| Лабораторная работа № 13.1 | Указатели и функции. Динамическое выделение памяти | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 6 |
| **Раздел XIV** | **Потоки и файлы** | | | |
| Лабораторная работа № 14.1 | Потоковые классы С++ и файлы. Текстовые и бинарные файлы | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 6 |
| **Раздел XV** | **Технологии программирования. Структуры** | | | |
| Лабораторная работа № 15.1 | Массивы и функции | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 2 |
| Лабораторная работа № 15.2 | Простая графика | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 2 |
| Лабораторная работа № 15.3 | Графика. Структура. Периметр и площадь | Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы. | Выполнение лабораторной работы. | 2 |

## Применение электронного обучения, дистанционных образовательных технологий

При реализации программы учебной дисциплины возможно применение электронного обучения и дистанционных образовательных технологий.

Реализация программы учебной дисциплины с применением электронного обучения и дистанционных образовательных технологий регламентируется действующими локальными актами университета.

Применяются следующие разновидности реализации программы с использованием ЭО и ДОТ.

В электронную образовательную среду, по необходимости, могут быть перенесены отдельные виды учебной деятельности:

|  |  |  |  |
| --- | --- | --- | --- |
| **использование**  **ЭО и ДОТ** | **использование ЭО и ДОТ** | **объем, час** | **включение в учебный процесс** |
| смешанное обучение | лекции | 53 | в соответствии с расписанием учебных занятий |
| лабораторные занятия | 88 |

# РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, КРИТЕРИИ ОЦЕНКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ, СИСТЕМА И ШКАЛА ОЦЕНИВАНИЯ

## Соотнесение планируемых результатов обучения с уровнями сформированности компетенций.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Уровни сформированности компетенции(-й)** | **Итоговое количество баллов**  **в 100-балльной системе**  **по результатам текущей и промежуточной аттестации** | **Оценка в пятибалльной системе**  **по результатам текущей и промежуточной аттестации** | **Показатели уровня сформированности** | | |
| **универсальной(-ых)**  **компетенции(-й)** | **общепрофессиональной(-ых) компетенций** | **профессиональной(-ых)**  **компетенции(-й)** |
|  | ОПК-2:  ИД-ОПК-2.3  ОПК-3:  ИД-ОПК-3.3  ОПК-6:  ИД-ОПК-6.1 |  |
| высокий |  | отлично/  зачтено (отлично)/  зачтено |  | Обучающийся:   * исчерпывающе и логически стройно излагает учебный материал, умеет связывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения; * способен уверенно использовать современные системы разработки прикладных программ с эффективными графическими интерфейсами и системы коммуникации в сети Internet; * показывает творческие способности в понимании и практическом использовании языков высокого уровня, использовании визуальных компонентов разработки приложений графических интерфейсов; * дополняет теоретическую информацию сведениями, самостоятельно полученными из источников научно-технической информации; * способен провести целостный анализ среды разработки современных программ на основе объектно-ориентированного и визуального программирования; * свободно ориентируется в учебной и профессиональной литературе; * дает развернутые, исчерпывающие, профессионально грамотные ответы на вопросы, в том числе, дополнительные. |  |
| повышенный |  | хорошо/  зачтено (хорошо)/  зачтено |  | Обучающийся:   * достаточно подробно, грамотно и по существу излагает изученный материал, приводит и раскрывает в тезисной форме основные понятия; * анализирует современные Технология программирования с незначительными пробелами; * способен использовать только основные функциональные возможности систем разработки программ и систем коммуникации в сети Internet; * способен провести анализ основных элементов разработки современных программ на основе объектно-ориентированного и визуального программирования; * допускает единичные негрубые ошибки; * достаточно хорошо ориентируется в учебной и профессиональной литературе; * ответ отражает знание теоретического и практического материала, не допуская существенных неточностей. |  |
| базовый |  | удовлетворительно/  зачтено (удовлетворительно)/  зачтено |  | Обучающийся:   * демонстрирует теоретические знания основного учебного материала дисциплины в объеме, необходимом для дальнейшего освоения ОПОП; * с неточностями излагает принципы и методы разработки современных программ на основе объектно-ориентированного и визуального программирования; * способен использовать отдельные элементы визуальной разработки прикладных программ; * анализирует современные технологии программирования с неточностями и ошибками; * демонстрирует фрагментарные знания основной учебной литературы по дисциплине; * ответ отражает знания на базовом уровне теоретического и практического материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профилю обучения. |  |
| низкий |  | неудовлетворительно/  не зачтено | Обучающийся:   * демонстрирует фрагментарные знания теоретического и практического материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации; * испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами; * не способен проанализировать учебно-методическую, техническую и научную литературу; * не владеет основными принципами и навыками работы в современных средах разработки прикладных программ, не умеет пользоваться системами коммуникации (Internet); * выполняет задания только по образцу и под руководством преподавателя; * ответ отражает отсутствие знаний на базовом уровне теоретического и практического материала в объеме, необходимом для дальнейшей учебы. | | |

# ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ, ВКЛЮЧАЯ САМОСТОЯТЕЛЬНУЮ РАБОТУ ОБУЧАЮЩИХСЯ

* + - 1. При проведении контроля самостоятельной работы обучающихся, текущего контроля и промежуточной аттестации по учебной дисциплине «Технология программирования»проверяется уровень сформированности у обучающихся компетенций и запланированных результатов обучения по дисциплине*,* указанных в разделе 2 настоящей программы.

## Формы текущего контроля успеваемости, примеры типовых заданий:

| **№ пп** | **Формы текущего контроля** | * + - 1. **Примеры типовых заданий** |
| --- | --- | --- |
| Лабораторная работа № 1.1 | Выполнение лабораторной работы. | Простые вычисления с числами целых и вещественных типов.  Для всех вариантов и заданий:  - В заголовок окна программы записать свои данные: имя, фамилию и номер варианта.  - Ввод и вывод данных осуществлять в числовых форматах. Ввод всех исходных данных производить с клавиатуры. Вывод чисел вещественных типов производить в форматированном виде с точностью до 2-х знаков после точки.  - Все вычисления производить только посредством арифметических операций, процедур и функций для работы с целыми и вещественными типами данных.  - Не использовать в этой работе процедуры и функции для работы с символьными и строковыми типами данных.  Варианты:  1. Возвести целое число в квадрат без операции умножения, используя свойство: квадрат числа N равен сумме N первых нечетных чисел. Найти на интервале от 1 до N все целые числа, являющиеся квадратами целых чисел.  2. Разложить введенное с клавиатуры произвольное целое число на все возможные целые простые множители. (Например: 20 = 2\*2\*5; 21 = 3\*7; 22 = 2\*11; 23 = 1\*23).  3. Для введенного с клавиатуры произвольного целого числа определить все возможные целые делители, кроме 1 и самого числа. Если число не имеет таких делителей, сообщить об этом. (Например: для 20 – 2,4,5,10; для 21 - 3,7; для 23 – «нет делителей»).  4. Дано целое число N (N<=100), определяющее возраст человека (в годах). Дать для этого числа наименования "год", "года", "лет".  5. Дано целое число N. Посчитать количество и сумму его цифр.  … |
| Лабораторная работа № 1.2 | Выполнение лабораторной работы. | Тригонометрические вычисления и расчёты.  Для всех вариантов и заданий:  - В заголовок окна программы записать свои данные: имя, фамилию и номер варианта.  - Ввод и вывод данных осуществлять в числовых форматах. Ввод всех исходных данных производить с клавиатуры. Вывод чисел вещественных типов производить в форматированном виде с точностью до 2-х знаков после точки.  Варианты:  1. Даны три вещественных числа x,y,z. Определить, существует ли треугольник с длинами сторон x,y,z. Если треугольник существует, определить его тип (равносторонний, равнобедренный, прямоугольный) и вычислить все углы в градусах. Напечатать длины сторон с противолежащими углами.  2. Произвольный треугольник задан декартовыми координатами своих вершин (X1,Y1),(X2,Y2),(X3,Y3) – целыми числами. Определить, можно ли в него поместить окружность радиуса R(вещественное число).  3. Даны два отрезка А и В, заданные координатами их концов (АX1,АY1),(АX2,АY2) и (ВX1,ВY1),(ВX2,ВY2) – целыми числами. Определить, пересекаются ли они, в этом случае вычислить координаты (вещественные числа) точки их пересечения.  4. Отрезок, заданный координатами концов (X1,Y1),(X2,Y2) – целыми числами, определяет прямую, делящую пространство на две полуплоскости. Даны две точки А и В, заданные координатами (АX,АY) и (ВX,ВY) – целыми числами. Определить, расположены ли точки в одной полуплоскости, т.е. по одну сторону от прямой.  5. Луч, заданный координатами точки (X,Y) и углом ALPHA в градусах – целыми числами, определяет прямую, делящую пространство на две полуплоскости. Даны две точки А и В, заданные координатами (АX,АY) и (ВX,ВY) – целыми числами. Определить, расположены ли точки в одной полуплоскости, т.е. по одну сторону от прямой.  … |
| Лабораторная работа № 2.1 | Выполнение лабораторной работы. | Работа с символьными данными.  Для всех вариантов:  - Написать программу, которая читает с клавиатуры произвольную строку символов, преобразует ее и выводит результат на дисплей в виде строки.  - Длина строки заранее не вводится, а определяется после окончания ввода с клавиатуры.  - В строке группы символов разделяются одним или несколькими пробелами. Пробелы могут находиться в начале и в конце строки. Группа символов - последовательность букв, цифр или знаков, отделенная пробелами от других групп или находящаяся в начале/конце строки.  - Все буквы считать вводимыми только в нижнем регистре, Заглавные буквы не рассматривать.  - В заголовок окна программы записать свои данные: имя, фамилию и номер варианта.  Примеры возможных строк: ‘ 12.3 tryi yu ()?\*‘, ‘ 6е-1 swty +12\_3‘  Напечатать по вариантам:  1. Симметричные группы символов из строки, если они состоят только из цифр.  Например, строка: ‘ !2! 12321 reter 123 508805’ ; результат: ‘12321 508805’  2. Симметричные группы символов из строки, если они состоят только из латинских букв.  Например, строка: ‘ 12 abba 787 typepyt (\*)’ ; результат: ‘abba typepyt’;  3. Группы символов, состоящие только из латинских букв, если символы расположены в алфавитном порядке по возрастанию.  Например, строка: ‘ 12\*er faw acf ab2c fghjk’ ; результат: ‘acf fghjk’;  4. Группы символов, состоящие только из латинских букв, если символы расположены в алфавитном порядке по убыванию.  5. Группы символов, состоящие только из цифр, если цифры расположены по возрастанию.  Например, строка: ‘ 1276 357 12e6 0279 ’ ; результат: ‘357 0279’;  … |
| Лабораторная работа № 2.2 | Выполнение лабораторной работы. | Работа с матрицами (двумерными массивами).  Для всех вариантов:  - Ввести произвольную размерность квадратной матрицы в интервале 1..10.  - Предложить ввод данных (чисел целого типа) двумя способами:  1. Через датчик случайных чисел на интервале -100..100.  2. С клавиатуры в виде «Х[i],[j]=», где i и j – числа, соответствующие номеру строки и столбца.  - После окончания ввода данных вывести на экран матрицу в отформатированном виде по строкам и столбцам.  - Обработать данные согласно заданию в варианте и вывести результат.  - Если результатами вычислений являются вещественные числа, то выводить их с точностью до 2-х знаков после точки.  - Вектор (одномерный массив) можно не создавать, а сразу выводить результаты на печать.  Варианты:  1. Найти среднее арифметическое элементов каждого из четных столбцов этой матрицы.  2. Поменять местами строку, содержащую максимальный элемент, со строкой, содержащей минимальный элемент. Если оба таких элемента находятся в одной строке, то вывести сообщение и номер строки.  3. Получить новую матрицу вещественного типа путем деления всех элементов исходной матрицы на ее наименьший по модулю элемент (если он не равен 0).  4. Получить новую матрицу вещественного типа путем умножения всех элементов исходной матрицы на ее наибольший по модулю элемент (если он не равен 0).  5. Найти среднее арифметическое модулей элементов каждого из нечетных столбцов матрицы.  … |
| Лабораторная работа № 2.3 | Выполнение лабораторной работы. | Работа с записями.  Для всех вариантов и заданий:  - В заголовок окна программы записать свои данные: имя, фамилию и номер варианта.  - Для всех вариантов количество записей в массиве ограничить интервалом [1..10].  - Перед вводом данных предложить пользователю ввести количество записей от 1 до 10.  - После окончания ввода последней записи вывести все введённые значения на экран, а затем вывести результат их обработки.  Варианты:  1. Создать запись «Покупатель» с полями: «Имя» (строка), «Пол» (символ), «Цена товара» (вещественное число), «Количество товаров» (целое число). Имена покупателей могут повторяться, т.е. один покупатель может покупать разные товары в разном количестве.  Вывести информацию о покупателе, купившем больше всего товаров.  2. Создать запись «Студент» с полями: «Имя» (строка), «Предмет» (строка), «Оценка» (целое число). Имена студентов и названия предметов могут повторяться, т.е. один студент может получить несколько оценок по одному предмету.  Вывести информацию о студенте с наивысшим средним баллом.  3. Создать запись «Пассажир» с полями: «Имя» (строка), «Пол» (символ), «Вес одного предмета багажа» (вещественное число). Имена пассажиров могут повторяться, т.е. один пассажир может перевозить несколько предметов багажа.  Вывести информацию о суммарном весе багажа, перевозимом женщинами.  4. Создать запись «Кинофильм» с полями: «Название» (строка), «Страна» (строка), «Стоимость производства» (целое число), «Доход от проката» (целое число). Названия кинофильмов не повторяются.  Вывести информацию о самом прибыльном фильме (доход с вычетом производства).  5. Создать запись «Сотрудник» с полями: «Имя» (строка), «Пол» (символ), «Отдел» (строка), «Заработная плата» (вещественное число). Имена сотрудников не повторяются.  Посчитать среднюю заработную плату сотрудников-мужчин в каждом отделе.  … |
| Лабораторная работа № 3.1 | Выполнение лабораторной работы. | Использование подпрограмм и модулей.  Для всех вариантов:  - Ввести произвольную размерность одномерного массива в интервале 1..10.  - Ввести данные (тип данных указан в вариантах) с клавиатуры в виде «Х[i]=», где i – индекс элемента.  - После окончания ввода для контроля данных вывести на экран массив.  - Обработать данные в подпрограмме (тип подпрограммы и способ обработки указан в вариантах) и вывести результат.  - Ввод данных и вывод результатов на экран должны производиться в головной программе, все вычисления – в подпрограммах. Оформить подпрограммы в отдельном модуле. Подключить модуль к головной программе.  - Все символьные элементы считать вводимыми только в нижнем регистре.  - Если результатами вычислений являются вещественные числа, то выводить их с точностью до 1-го знака после точки.  Варианты:  1. Массив целых чисел. В процедуре отсортировать массив по возрастанию.  2. Массив вещественных чисел. В процедуре отсортировать массив по убыванию.  3. Массив символов. В процедуре отсортировать массив по алфавиту в обратном порядке.  4. Массив целых чисел. В процедуре удалить из массива нечетные элементы. Размерность итогового массива должна уменьшиться.  5. Массив символов (буквы). В функции найти в исходном массиве наименьшую по алфавиту букву, удалить этот элемент из массива (размерность итогового массива уменьшится на 1). Найденный элемент передать в головную программу в качестве возвращаемого значения функции.  … |
| Лабораторная работа № 3.2 | Выполнение лабораторной работы. | Использование библиотек DLL.  Для всех вариантов:  - Ввести два целых числа a и b.  - Произвести вычисления в подпрограммах. Оформить подпрограммы в библиотеке DLL.  1) функция - результат Х (целое число) передать в вызывающую программу в качестве возвращаемого значения. Использовать статический вызов.  2) процедура - результат Х (целое число) передать в вызывающую программу в виде параметра-переменной. Использовать динамический вызов.  - Ввод данных и вывод результатов на экран должны производиться в головной программе, все вычисления – в подпрограммах.  Варианты: 1) функции 2) процедуры  1. 1)  2)  2. 1)  2)  3. 1)  2)  4. 1)  2)  5. 1)  2)  … |
| Лабораторная работа № 4.1 | Выполнение лабораторной работы. | Поиск файлов. Работа с найденными текстовыми файлами.  Для всех вариантов:  - Ввести с клавиатуры произвольное имя текстового файла.  - Найти на жестком диске компьютера все файлы с заданным именем (одноименные файлы могут находиться в разных папках), поочередно вывести на экран их содержимое. Если ни одного файла не найдено, выдать сообщение.  - Выполнить обработку каждого найденного файла в соответствии с заданием в варианте.  - Создать текстовый файл с именем программы, в который записать результаты выполнения программы, т.е. продублировать вывод информации на экран и в текстовый файл. Этот файл перезаписывать при каждом запуске программы.  Варианты:  1. Подсчитать количество строк в файле.  2. Определить максимальную длину строки в файле.  3. Определить минимальную длину строки в файле.  4. Подсчитать в файле количество символов «.» (точек).  5. Определить, присутствует ли в начале файла слово ‘Program’. Слово не зависит от регистра букв.  ... |
| Лабораторная работа № 5.1 | Выполнение лабораторной работы. | Статическая графика в окнах Windows.  Для всех вариантов:  - Создать окно Windows. В заголовок окна записать свои данные: имя, фамилию и название рисунка.  - Используя функции ObjectPascal, визуальные компоненты среды разработки приложений Lazarus и GDI API Windows, создать в окне Windows изображение по заданию в варианте.  - В изображении обязательно хотя бы один раз использовать линии, прямоугольники, окружности, многоточечные фигуры-полигоны.  - Обязательно использовать разные типы и цвета линий, разные типы заливки фигур.  - Обязательно использовать текстовые надписи.  Варианты:  1. Стоящий на столе компьютер.  2. Загородный одноэтажный дом.  3. Магазин с вывеской.  4. Воздушный шар с корзиной.  5. Телефон с кнопочным набором.  ... |
| Лабораторная работа № 6.1 | Выполнение лабораторной работы. | Динамическая графика в окнах Windows.  Для всех вариантов:  - Создать окно Windows. В заголовок окна записать свои данные: имя, фамилию и номер варианта.  - Используя функции ObjectPascal, визуальные компоненты среды разработки приложений Lazarus и GDI API Windows, разделить окно линиями на заданное в варианте количество частей. Линии должны изменяться в соответствии с изменениями размеров окна.  - Написать процедуры для обработки нажатия на клавиши мыши по заданию в варианте. Геометрические фигуры должны иметь центр в точке нажатия «мыши», текстовые сообщения должны выводиться, начиная от точки нажатия «мыши».  - При перерисовке окна (например, во время изменения его размеров), созданные с помощью «мыши» изображения фигур и текстовые надписи должны сохраняться и обновляться.  **Варианты:**  **1**. Разделить окно двумя линиями: вертикальной и горизонтальной, проходящими по центру. Должны получиться 4 сектора окна:  Primer-1  Если левая кнопка «мыши» нажата в разных секторах, то выводить квадрат со стороной 30pix разными типами заливки для каждого сектора.  **2**. Разделить окно диагональной линией проходящей от левого верхнего угла до правого нижнего угла окна. Должны получиться 2 сектора окна: I и II.  Primer-2  Если левая кнопка «мыши» нажата в секторе I, то выводить квадрат со стороной 30pix сплошной заливкой. Если левая кнопка «мыши» нажата в секторе II, то выводить круг с радиусом 30pix сплошной заливкой.  **3**. Разделить окно диагональной линией проходящей от правого верхнего угла до левого нижнего угла окна. Должны получиться 2 сектора окна: I и II.  Primer-3  Если левая кнопка «мыши» нажата в секторе I, то выводить квадрат со стороной 30pix сплошной заливкой. Если левая кнопка «мыши» нажата в секторе II, то выводить круг с радиусом 30pix сплошной заливкой.  **4**. Разделить окно двумя линиями: вертикальной и горизонтальной, проходящими по центру. Должны получиться 4 сектора окна:  Primer-1  Если левая кнопка «мыши» нажата в разных секторах, то выводить текст «Test Font» разными типами шрифта для каждого сектора.  **11-17**. Разделить окно диагональной линией проходящей от левого верхнего угла до правого нижнего угла окна. Должны получиться 2 сектора окна: I и II.  Primer-2  Если левая кнопка «мыши» нажата в секторе I, то выводить текст «Sector I» одного типа и цвета шрифта. Если левая кнопка «мыши» нажата в секторе II, то выводить текст «Sector II» другого типа и цвета шрифта.  ... |
| Лабораторная работа № 7.1 | Выполнение лабораторной работы. | Линейное программирование.  **Вариант 1**   1. Написать программу пересчёта расстояния из вёрст в километры (1 верста = 1066.8 м). Программа должна обеспечивать ввод расстояния в верстах и вывод расстояния в километрах. 2. Написать программу для расчёта по двум формулам (результат вычисления по первой формуле должен совпадать со второй).   ;  **Вариант 2**   1. Написать программу пересчёта веса из фунтов в килограммы (1 фунт = 405.9 грамм). Программа должна обеспечивать ввод веса в фунтах и вывод веса в килограммах. 2. Написать программу для расчёта по двум формулам (результат вычисления по первой формуле должен совпадать со второй).   ;  **Вариант 3**   1. Написать программу вычисления стоимости поездки на автомобиле на дачу (туда и обратно). Программа должна позволять вводить следующие исходные данные: расстояние до дачи (км); расход бензина (количество литров на 100 км пробега); цена одного литра бензина (руб.). 2. Написать программу для расчёта по двум формулам (результат вычисления по первой формуле должен совпадать со второй).   ;  **Вариант 4**   1. Написать программу вычисления силы тока в электрической цепи. Программа должна позволять вводить величину напряжения (В) и сопротивления (Ом). 2. Написать программу для расчёта по двум формулам (результат вычисления по первой формуле должен совпадать со второй).   ;  **Вариант 5**   1. Написать программу пересчёта расстояния из дюймов в сантиметры (1 дюйм = 0.0254 м). Программа должна обеспечивать ввод расстояния в дюймах и вывод расстояния в сантиметрах. 2. Написать программу для расчёта по двум формулам (результат вычисления по первой формуле должен совпадать со второй).   ;  … |
| Лабораторная работа № 8.1 | Выполнение лабораторной работы. | Расчетные задачи.  **Вариант 1.** Подсчитать k-количество цифр в десятичной записи целого неотрицательного числа n.  **Вариант 2.** Дано n вещественных чисел. Вычислить разность между максимальным и минимальным из них.  **Вариант 3.** Дана непустая последовательность различных натуральных чисел, за которой следует 0. Определить порядковый номер наименьшего из них.  **Вариант 4.** Даны целое n>0 и последовательность из n вещественных чисел, среди которых есть хотя бы одно отрицательное число. Найти величину наибольшего среди отрицательных чисел этой последовательности.  **Вариант 5.** Дано n вещественных чисел. Определить, образуют ли они возрастающую последовательность.  … |
| Лабораторная работа № 9.1 | Выполнение лабораторной работы. | Циклы.  Часть 1. Единое задание для закрепления пройденного материала. Приложение реализовать в консольном и в визуальном исполнении.   1. Используя цикл for, написать программу, которая выводит таблицу значений функции y=2x2-5x-8. При этом пользователь должен ввести диапазон и шаг изменения аргумента х. 2. Используя цикл while, написать программу, которая определяет произведение цифр введённого пользователем натурального числа типа int. Предусмотреть обработку ошибок.   Часть 2. Индивидуальное задание по вариантам.  **Вариант 1.** Вычислить дробную часть среднего геометрического трех заданных положительных чисел.  **Вариант 2.** Вычислить длину окружности, площадь круга и объем шара одного и того же радиуса.  **Вариант 3.** Вычислить периметр и площадь прямоугольного треугольника по длинам двух катетов.  **Вариант 4.** По координатам трех вершин некоторого треугольника найти его площадь и периметр.  **Вариант 5.** По длинам двух сторон некоторого треугольника и углу между ними найти длину третьей стороны и вычислить площадь этого треугольника.  … |
| Лабораторная работа № 10.1 | Выполнение лабораторной работы. | Ветвления и циклы.  Часть 1. Единое задание для закрепления пройденного материала. Приложения реализовать в консольном и в визуальном исполнении.   1. Дана заштрихованная область (см. рис.). Написать программу, которая позволяет вводить с клавиатуры точку с координатами (х; у) и определять попадает ли данная точка в область или нет.   **Х**  **У**  -1  1 2   1. Написать программу, которая выводит в виде таблицы значения функции y=*f(x)* на интервале  с шагом :   , *если* *х=5 или х=7*  *f(х) =* ,  *если х=4*  ,  *в остальных случаях*  Необходимо реализовать программу в двух версиях. В одной версии следует применить циклическую конструкцию с условиями, а в другой версии – циклическую конструкцию со switch.  Часть 2. Индивидуальное задание по вариантам. Приложение реализовать в консольном и в визуальном исполнении.  **Вариант 1.** Вычислить значение функции    A  Написать два варианта программы - с использованием оператора if и с использованием оператора switch. Результат представить в виде таблицы.  **Вариант 2.** Вычислить значение функции    B  Написать два варианта программы - с использованием оператора if и с использованием оператора switch. Результат представить в виде таблицы.  **Вариант 3.** Вычислить значение функции    C  Написать два варианта программы - с использованием оператора if и с использованием оператора switch. Результат представить в виде таблицы.  **Вариант 4.** Вычислить значение функции    Написать два варианта программы - с использованием оператора if и с использованием оператора switch. Результат представить в виде таблицы.  **Вариант 5.** Вычислить значение функции    E  Написать два варианта программы - с использованием оператора if и с использованием оператора switch. Результат представить в виде таблицы.  … |
| Лабораторная работа № 11.1 | Выполнение лабораторной работы. | Массивы.  Часть 1. Единое задание для закрепления пройденного материала. Приложения реализовать в консольном исполнении.   1. Написать программу, которая позволяет вводить одномерный целочисленный массив из 9 элементов и вычислять: а) сумму положительных элементов массива с чётными номерами; б) количество элементов массива, равных нулю; в) вещественный результат деления максимального элемента массива на минимальный. В случае возникновения исключительных ситуаций программа должна выводить соответствующие сообщения. 2. Написать программу, которая позволяет вводить целочисленную матрицу 3×4, выводить её в наглядном формате и определять номер столбца, сумма элементов которого минимальна. Программа должна позволять производить вычисления столько раз, сколько угодно пользователю.   Часть 2. Индивидуальное задание по вариантам. Приложение реализовать в консольном исполнении.  **Вариант 1.** Дан массив целых чисел. Написать программу для сортировки массива по возрастанию.  **Вариант 2.** Дан массив целых чисел. Написать программу для определения максимальной и минимальной суммы двух соседних элементов массива.  **Вариант 3.** Дан массив целых чисел. Написать программу для определения максимальной и минимальной разницы между двумя соседними элементами массива.  **Вариант 4**. Дан массив вещественных чисел. Написать программу для сортировки массива по возрастанию целой части его элементов.  **Вариант 5.** Дан массив целых чисел. Написать программу для сортировки заданного массива по убыванию суммы цифр элементов.  … |
| Лабораторная работа № 11.2 | Выполнение лабораторной работы. | Строки типа char.  Часть 1. Единое задание для закрепления пройденного материала. Приложения реализовать в консольном и в визуальном исполнении.   1. Написать программу, которая позволяет вводить с клавиатуры строку, содержащую более одного слова, и определять: а) количество символов введённой строки; б) содержит ли строка слова одинаковой длины; в) количество запятых. Необходимо учесть, что при вводе строки, могут использоваться не только пробелы, но и знаки пунктуации. 2. Напишите программу, которая позволяет вводить с клавиатуры строку, содержащую более одного слова, и определять количество слов, заканчивающихся на заданную пользователем букву. Необходимо учесть, что при вводе строки, могут использоваться не только пробелы, но и знаки пунктуации   Часть 2. Индивидуальное задание по вариантам. Приложение реализовать в консольном и/или в визуальном исполнении.  **Вариант 1.** В заданном тексте удалить слова, начинающиеся с заданной буквы. При написании программы нельзя пользоваться стандартными функциями обработки строк.  **Вариант 2.** Вывести на экран заданный текст, удалив из него лишние пробелы, т.е. из нескольких подряд идущих пробелов оставить только один. При написании программы нельзя пользоваться стандартными функциями обработки строк.  **Вариант 3.** Задан текст, заканчивающийся точкой. Вывести на экран сначала все цифры, входящие в него, а затем все остальные литеры. При написании программы нельзя пользоваться стандартными функциями обработки строк.  **Вариант 4.** Определить, сколько различных литер входит в заданный текст, заканчивающийся точкой. При написании программы нельзя пользоваться стандартными функциями обработки строк.  **Вариант 5.** Заданный текст вывести на экран по строкам, понимая под строкой либо очередные 60 символов, если среди них нет запятой, либо текст до запятой включительно. При написании программы нельзя пользоваться стандартными функциями обработки строк.  … |
| Лабораторная работа № 12.1 | Выполнение лабораторной работы. | Функции.  Часть 1. Единое задание для закрепления пройденного материала. Приложения реализовать в визуальном исполнении.   1. Написать функцию, которая возвращает возведённое в степень число, и программу, использующую эту функцию для возведения введённого пользователем числа в заданную им степень. 2. Написать программу, позволяющую сортировать введенный пользователем целочисленный одномерный массив по убыванию или по возрастанию (в зависимости от выбора пользователя). Сортировку по убыванию необходимо реализовать методом пузырька, а сортировку по возрастанию – методом выбора. Указанные виды сортировок нужно оформить в виде отдельных функций. 3. Напишите функцию, которая переворачивает строку (массив типа char). Используйте цикл for, который меняет местами первый и последний символы, затем следующие и т.д. Строка должна передаваться в функцию как аргумент. Напишите программу, которая должна принимать от пользователя строку (содержащую более одного слова), вызывать функцию, а затем выводить полученный результат. 4. Напишите функцию вычисления корней квадратного уравнения. Программа должна принимать от пользователя коэффициенты, вызывать функцию, а затем выводить полученный результат. Параметрами функции должны быть коэффициенты и корни уравнения. Такие аргументы как корни уравнения необходимо передавать по ссылке. Значение, возвращаемое функцией, должно передавать в вызывающую программу информацию о наличии корней: 2 – два разных корня; 1 – корни одинаковые; 0 – уравнение не имеет решения. Кроме того, функция должна проверять корректность исходных данных. Если исходные данные неверные, то функция должна возвращать -1.   Часть 2. Индивидуальное задание по вариантам. Приложение реализовать в консольном и/или в визуальном исполнении.  **Вариант 1.** Дана строка. Написать функцию для определения, входит ли в строку заданное слово, или нет. При написании программы необходимо использовать стандартные функции обработки строк.  **Вариант 2.** Дана строка символов. Написать функцию для определения одинаковы или нет второе и последнее слово в этой строке. При написании программы необходимо использовать стандартные функции обработки строк.  **Вариант 3.** Дана строка символов. Написать функцию для определения слов, которые начинаются и заканчиваются на одинаковые буквы. При написании программы необходимо использовать стандартные функции обработки строк.  **Вариант 4.** ДанDа строка символов. Написать функцию для определения, входит ли в эту строку заданное слово. Если входит, то его нужно удалить. При написании программы необходимо использовать стандартные функции обработки строк.  **Вариант 5.** Дана строка символов. Написать функцию для формирования строки с обратным порядком слов. При написании программы необходимо использовать стандартные функции обработки строк.  … |
| Лабораторная работа № 13.1 | Выполнение лабораторной работы. | Указатели и функции. Динамическое выделение памяти.  Часть 1. Единое задание для закрепления пройденного материала. Приложения реализовать в консольном исполнении.   1. Напишите программу, которая позволяет вводить динамический одномерный целочисленный массив и посредством вызова функции определять минимальную сумму между двумя соседними элементами массива. При передаче массива в функцию используйте указатели, а размерность передавайте по ссылке. Применение глобальных переменных категорически запрещено. 2. Напишите программу, позволяющую вводить динамический многомерный целочисленный массив и посредством функций осуществлять: 1) вывод введённого массива на экран в наглядном формате; 2) вычисление суммы элементов в тех строках, которые содержат хотя бы один отрицательный элемент; 3) определение минимального из чисел, встречающихся в заданной матрице более одного раза. При передаче аргументов в функции старайтесь использовать указатели. Применение глобальных переменных категорически запрещено   Часть 2. Индивидуальное задание по вариантам. Приложение реализовать в консольном исполнении. При передаче массива в функцию используйте указатели, а размерность передавайте по ссылке. Применение глобальных переменных категорически запрещено.  **Вариант 1**. Пусть пользователь вводит динамическую матрицу размером NxM. Написать функцию для вычисления суммы диагональных элементов.  **Вариант 2.** Пусть пользователь вводит динамическую матрицу размером NxM. Написать функцию для вычисления максимальных и минимальных элементов в столбцах.  **Вариант 3.** Пусть пользователь вводит динамическую матрицу размером NxM. Написать функцию для вычисления максимальной и минимальной суммы элементов в строках.  **Вариант 4.** Пусть пользователь вводит динамическую матрицу размером NxM. Написать функцию для сортировки строк матрицы по убыванию.  **Вариант 5.** Пусть пользователь вводит динамическую матрицу размером NxM. С помощью функции определить максимальный элемент.  … |
| Лабораторная работа № 14.1 | Выполнение лабораторной работы. | Потоковые классы С++ и файлы. Текстовые и бинарные файлы.  Часть 1. Единое задание для закрепления пройденного материала. Приложение 1 реализовать в консольном и в визуальном исполнении, приложение 2 – в консольном исполнении.   1. Напишите программу, которая позволяет: а) выводить на экран содержимое указанного текстового файла; б) записывать в другой текстовый файл только те предложения, которые содержат введённое с клавиатуры слово. Если указанное слово отсутствует в тексте, программа должна выводить сообщение и не должна создавать выходной файл. По возможности старайтесь использовать динамическое выделение памяти. 2. Напишите программу, которая позволяет создавать список данных железнодорожных поездов и хранить его в бинарном файле (например, …\railwaybase.dat). Каждая запись бинарного файла содержит: номер поезда (sizeof(int) байт); пункт отправления-прибытия (30 байт); количество проданных билетов (sizeof(int) байт); цена одного билета (sizeof(float) байт). Программа должна позволять: а) добавлять в двоичный файл записи, причём столько раз, сколько угодно пользователю; б) показывать все записи в наглядном формате, причём стоимость одного билета должна выводиться с двумя знаками после запятой; в) подсчитывать выручку указанного номера поезда. Пользователь должен иметь возможность в любой момент выбрать необходимое ему действие с помощью меню, например следующего вида:   ===================RailwayBase=====================  1 - dobavit zapis; 2 - pokazat vse zapisi;  3 - pokazat vyruchku poezda; 4 - vyhod;  Vash vybor:  Меню реализуйте с помощью функции, возвращающей значение. Предусмотрите обработку исключительных ситуаций.  Часть 2. Индивидуальное задание по вариантам. Приложение реализовать в консольном исполнении.  **Вариант 1.** Дан текстовый файл, состоящий из 5 строк. Написать функцию для сравнения первого слова второй строки и последнего слова пятой строки.  **Вариант 2.** Дан текстовый файл, состоящий из некоторого количества строк. Написать функцию для сравнения первой и последней строки.  **Вариант 3.** Дан текстовый файл, состоящий из некоторого количества строк. Написать функцию для создания нового файла с обратным порядком строк.  **Вариант 4.** Дан текстовый файл, состоящий из некоторого количества строк. Написать функцию для создания нового файла, содержащего вторые слова строк первого файла.  **Вариант 5.** Даны два текстовых файла, состоящие из некоторого количества строк. Написать функцию для сравнения этих файлов.  … |
| Лабораторная работа № 15.1 | Выполнение лабораторной работы. | Массивы и функции.  **Вариант 1.** Преобразовать одномерный целочисленный массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные (элементы, равные 0, считать положительными).  **Вариант 2.** Сжать одномерный вещественный массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.  **Вариант 3.** Сжать одномерный вещественный массив, удалив из него все элементы, модуль которых находится в интервале [a,b]. Освободившиеся в конце массива элементы заполнить нулями.  **Вариант 4.** Преобразовать одномерный вещественный массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом – все остальные.  **Вариант 5.** Преобразовать одномерный целочисленный массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине – элементы, стоявшие в четных позициях.  … |
| Лабораторная работа № 15.2 | Выполнение лабораторной работы. | Простая графика. Работа с компонентом Image. Построение графических примитивов: отрезок, круг, сектор, эллипс, прямоугольник, полигон, полилиния. Рисование на холсте с динамической закраской. Выбор цвета, толщины и формы кисти.    Вариант 1. Воспроизвести, реализовав ластик по нажатию правой кнопки мышки.  Вариант 2. Воспроизвести, добавив форму кисти «ромб».  Вариант 3. Воспроизвести, добавив форму кисти «пентагон».  Вариант 4. Воспроизвести, добавив форму кисти «треугольник».  Вариант 5. Воспроизвести, добавив форму кисти «эллипс».  … |
| Лабораторная работа № 15.3 | Выполнение лабораторной работы. | Часть 1. Единое задание для закрепления пройденного материала.  Работа с графикой с применением структуры фигур:  struct Koordinata {  int x;  int y;  };  Построение многоугольников по точкам, сохранение их в текстовый файл, очистка холста, загрузка данных из файла и восстановление нарисованных многоугольников. Расчет периметра и площади нарисованных и загруженных из файла многоугольников. Воспроизвести приложение.    Часть 2. Индивидуальное задание по вариантам. Приложение реализовать в консольном и/или в визуальном исполнении.  **Вариант 1**. Нарисовать график функции f(X)= 2\*.  **Вариант 2**. Нарисовать график функции f(X)=.  **Вариант 3**. Нарисовать график функции f(X)= .  **Вариант 4**. Нарисовать график функции f(X)=sin(X).  **Вариант 5**. Нарисовать график функции f(X)=cos(X).  … |

## Критерии, шкалы оценивания текущего контроля успеваемости:

| **Наименование оценочного средства (контрольно-оценочного мероприятия)** | **Критерии оценивания** | **Шкалы оценивания** | |
| --- | --- | --- | --- |
| **100-балльная система** | **Пятибалльная система** |
| Лабораторная работа | Работа выполнена полностью. Нет ошибок в логических рассуждениях и в реализации задания в виде файла или выполняемой программы. Возможно наличие одной неточности или описки, не являющиеся следствием незнания или непонимания учебного материала и не влияющей на функциональные качества программы. Обучающийся показал полный объем знаний, умений в освоении пройденных тем и применение их на практике. Работа зачтена. |  | 5 |
| Работа выполнена полностью, но выбран неэффективный алгоритм или метод реализации, обоснований шагов решения недостаточно. Допущена одна ошибка или два-три недочета, которые незначительно влияют на качество представленной работы. Работа зачтена. |  | 4 |
| Допущены более одной ошибки или более двух-трех недочетов, которые оказывают значительное влияние на представляемый файл или компьютерную программу, ухудшают их информативность и функциональные возможности. Работа зачтена. |  | 3 |
| Работа выполнена не полностью. Допущены грубые ошибки. Файлы не содержат необходимой информации, компьютерная программа выдаёт неправильные результаты при вычислении тестовых примеров. Работа не зачтена. |  | 2 |
| Работа не выполнена. |  |

## Промежуточная аттестация:

|  |  |
| --- | --- |
| **Форма промежуточной аттестации** | **Типовые контрольные задания и иные материалы**  **для проведения промежуточной аттестации:** |
| Экзамен:  Компьютерное тестирование | Задание 01 – Структура программы на языке Pascal  Вопрос 1.1. Текст программы на языке Pascal состоит из   1. структурированных блоков 2. пронумерованных операторов 3. бинарных кодов 4. команд центрального процессора   Вопрос 1.2. Раздел описаний программы на языке Pascal не может содержать   1. объявления операторов operate 2. объявления типов type 3. объявления переменных var 4. объявления констант const   Вопрос 1.3. Структурный блок программного кода на языке Pascal обозначается ключевыми словами...   1. begin и end 2. begin и stop 3. start и end 4. start и stop   Задание 02 – Типы данных  Вопрос 2.1. Укажите неправильное определение:   1. Тип данных определяет права доступа пользователей и администраторов. 2. Тип данных определяет возможные значения переменных, констант, функций, выражений, принадлежащих к данному типу. 3. Тип данных определяет внутреннюю форму представления данных в ЭВМ (количество байт памяти). 4. Тип данных определяет операции и функции, которые могут выполняться над значениями, принадлежащими к данному типу.   Вопрос 2.2. Укажите неправильное определение:   1. Тип данных определяет возможность или невозможность передачи информации по сети Интернет. 2. Тип данных определяет возможные значения переменных, констант, функций, выражений, принадлежащих к данному типу. 3. Тип данных определяет внутреннюю форму представления данных в ЭВМ (количество байт памяти). 4. Тип данных определяет операции и функции, которые могут выполняться над значениями, принадлежащими к данному типу.   Вопрос 2.3. На языке Pascal тип переменной или тип возвращаемого значения функции должен быть явно задан при объявлении...   1. которое предшествует их использованию 2. которое следует после их использования 3. которое происходит в момент их использования 4. которое можно не использовать   Задание 14 – Тип множество  Вопрос 14.1. Если в разделе описаний объявлено множество Mn:  Var  Mn: Set of '1'..'9';  то какое присваивание значения множеству будет неправильным:   1. Mn:=[1,3]; 2. Mn:=['1','3']; 3. Mn:=[]; 4. Mn:=['1']; 5. Mn:=['1'..'3'];   Вопрос 14.2. Если в разделе описаний объявлено множество Mn:  Var  Mn: Set of 0..9;  то какие присваивания значений множеству будут правильными (укажите все правильные присваивания):   1. Mn:=[0..9]; 2. Mn:=[0]+[1]; 3. Mn:=[0,1,2]; 4. Mn:=[-1,0]; 5. Mn:=[0-9]; 6. Mn:=['0','1','2'];   Вопрос 14.3. Проверка принадлежности элемента базового типа множеству осуществляется конструкцией [[1]].   * if <элемент> in <множество> * if <элемент> for <множество> * if <элемент> from <множество> * if <элемент> to <множество>   Задание 15 – Использование подпрограмм  Вопрос 15.1. Какое объявление подпрограммы не имеет ошибок?   1. procedure P1 (var i:integer); 2. function F1 (var i:integer):forward; 3. procedure P1 (var i:integer):integer; 4. function F1 (var i:integer);   Вопрос 15.2. Для передачи возвращаемого значения из функции в вызывающую программу надо...   1. присвоить это значение имени функции в процессе её выполнения 2. создать глобальную переменную, совпадающую с именем функции 3. возвращение значения производится автоматически после завершения работы функции, ничего делать не надо 4. один из параметров-переменных должен совпадать с именем функции   Вопрос 15.3. Рекурсия – это такой способ организации алгоритма, при котором подпрограмма вызывает...   1. саму себя 2. API Windows 3. администратора 4. короткое замыкание в блоке питания   Задание 21 - Объектно-оpиентиpованное пpогpаммиpование  Вопрос 21.1. Какого принципа объектно-оpиентиpованного пpогpаммиpования не существует?   1. **аддитивность** 2. инкапсуляция 3. наследование 4. полимоpфизм   Вопрос 21.2. Принцип объектно-оpиентиpованного пpогpаммиpования, который определяет, что инфоpмация об объекте (его свойства) и функции объекта (методы) содеpжатся в описании этого объекта, называется...   1. **инкапсуляция** 2. наследование 3. полимоpфизм 4. аддитивность   Вопрос 21.3. Принцип объектно-оpиентиpованного пpогpаммиpования, который определяет, что один объект может быть постpоен на базе описания дpугого объекта, называется...   1. **наследование** 2. полимоpфизм 3. аддитивность 4. инкапсуляция   Вопрос 21.4. Принцип объектно-оpиентиpованного пpогpаммиpования, который определяет способность многих объектов использовать один и тот же метод, пpи этом выполняемые действия зависят от объекта, котоpый его вызвал, называется...   1. **полимоpфизм** 2. аддитивность 3. инкапсуляция 4. наследование   ... |
| Зачет с оценкой:  в устной форме по билетам | Билет № 1.   1. Характеристика языка С++. Принцип работы компилятора языка высокого уровня. 2. Написать программу на С++ в визуальном исполнении. В заданном пользователем тексте посредством функции удалить слова, начинающиеся с указанной буквы. При написании программы нельзя пользоваться стандартными функциями обработки строк.   Билет № 2.   1. Идентификаторы. Комментарии. Типы данных в С++ и их спецификаторы. 2. Написать программу на С++ в консольном исполнении. В заданном пользователем тексте посредством функции удалить слова, заканчивающиеся на указанную букву. При написании программы можно пользоваться стандартными функциями обработки строк.   Билет № 3.   1. Структура программы на языке С++. Директива препроцессора и заголовочные файлы. 2. Разработать визуальное приложение на С++, которое позволяет вводить строку и с помощью функции определять количество согласных букв в строке.   Билет № 4.   1. Главная функция и её разновидности. Консольный ввод/вывод в стиле С++. Предназначение функций стандартной библиотеки. 2. Написать консольную программу на С++, которая позволяет вводить строку и с помощью функции определять количество гласных букв в строке.   Билет № 5.   1. Переменные в С++. Описание переменных. Разновидности переменных, область действия и область видимости. 2. Создать визуальное приложение на С++, которое проверяет отсортирован ли считанный из текстового файла массив по возрастанию. Проверку организовать посредством функции.   … |

## Критерии, шкалы оценивания промежуточной аттестации учебной дисциплины:

| **Форма промежуточной аттестации** | **Критерии оценивания** | **Шкалы оценивания** | | |
| --- | --- | --- | --- | --- |
| **Наименование оценочного средства** | **100-балльная система** | **Пятибалльная система** | |
| Экзамен:  компьютерное тестирование | За выполнение каждого тестового задания испытуемому выставляются баллы. За полностью правильный ответ к каждому заданию с выбором одного правильного варианта выставляется один балл, за неправильный — ноль. За задания с выбором нескольких правильных ответов или в заданиях с сопоставлениями испытуемый может получить менее 1 балла. Например, если правильных ответов в задании два, то за каждый он получает 0,5 балла, если правильных ответов три, то за каждый он получает 0,333 балла и т.п.  Правила оценки всего теста:  вне зависимости от количества заданий в тесте общая сумма баллов за все правильные ответы пересчитывается тестирующей компьютерной системой в итоговые баллы. 10 итоговых баллов эквивалентны 100% правильных ответов. Для того, чтобы получить отличную, хорошую, удовлетворительную или неудовлетворительную оценки, итоговые баллы за промежуточные аттестации каждого семестра складываются с баллами за выполненные лабораторные работы. |  | *5* | *85% - 100%* |
|  | *4* | *65% - 84%* |
|  | *3* | *41% - 64%* |
|  | *2* | *40% и менее 40%* |
| Зачет с оценкой:  в устной форме по билетам | Обучающийся: демонстрирует знания, отличающиеся глубиной и содержательностью, дает полный исчерпывающий ответ на все вопросы билета; свободно владеет научными понятиями, ведет диалог и вступает в научную дискуссию; способен к интеграции знаний по определенной теме, структурированию ответа, к анализу положений существующих теорий и направлений по вопросу билета; свободно выполняет практическое задание, предусмотренное программой, демонстрирует системную работу с основной и дополнительной литературой.  Ответ не содержит фактических ошибок и характеризуется глубиной, полнотой, уверенностью суждений, иллюстрируется примерами, в том числе из собственной практики.  Практическое задание выполнено полностью. Нет ошибок в логических рассуждениях и в реализации задания в виде файла или выполняемой программы. Возможно наличие одной неточности или описки, не являющиеся следствием незнания или непонимания учебного материала и не влияющей на функциональные качества программы. Обучающийся показал полный объем знаний, умений в освоении пройденных тем и применение их на практике. |  | *5* | |
| Обучающийся: показывает достаточное знание учебного материала, но допускает несущественные фактические ошибки, которые способен исправить самостоятельно, благодаря наводящему вопросу; успешно выполняет предусмотренные в программе практические задания средней сложности, активно работает с основной литературой, демонстрирует, в целом, системный подход к решению практических задач, к самостоятельному пополнению и обновлению знаний в ходе дальнейшей учебной работы и профессиональной деятельности.  Недостаточно раскрыта проблема по одному из вопросов билета; недостаточно логично построено изложение вопроса. В ответе раскрыто, в основном, содержание билета, имеются неточности при ответе на дополнительные вопросы.  Практическое задание выполнено полностью, но выбран неэффективный алгоритм или метод реализации, обоснований шагов решения недостаточно. Допущена одна ошибка или два-три недочета, которые незначительно влияют на качество представленной работы. |  | *4* | |
| Обучающийся: показывает знания фрагментарного характера, которые отличаются поверхностностью и малой содержательностью, допускает фактические грубые ошибки; не может обосновать закономерности и принципы, объяснить факты, нарушена логика изложения, отсутствует осмысленность представляемого материала, представления о межпредметных связях слабые; справляется с выполнением практических заданий, предусмотренных программой, знаком с основной литературой, рекомендованной программой, допускает погрешности и ошибки при теоретических ответах и в ходе практической работы. Содержание билета раскрыто слабо, имеются неточности при ответе на основные и дополнительные вопросы билета, ответ носит репродуктивный характер. Неуверенно, с большими затруднениями решает практические задачи или не справляется с ними самостоятельно.  В практическом задании допущены более одной ошибки или более двух-трех недочетов, которые оказывают значительное влияние на представляемый файл или компьютерную программу, ухудшают их информативность и функциональные возможности. |  | *3* | |
| Обучающийся обнаруживает существенные пробелы в знаниях основного учебного материала, допускает принципиальные ошибки в выполнении предусмотренных программой практических заданий.  На большую часть дополнительных вопросов по содержанию курса затрудняется дать ответ или не дает верных ответов.  Практическое задание выполнено не полностью или не выполнено вообще. Допущены грубые ошибки. Файлы не содержат необходимой информации, компьютерная программа выдаёт неправильные результаты при вычислении тестовых примеров. |  | *2* | |

## Система оценивания результатов текущего контроля и промежуточной аттестации.

Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.

|  |  |  |
| --- | --- | --- |
| **Форма контроля** | **100-балльная система** | **Пятибалльная система** |
| Текущий контроль: |  |  |
| Выполнение лабораторной работы |  | зачтено/не зачтено |
| Промежуточная аттестация  экзамен |  | отлично  хорошо  удовлетворительно  неудовлетворительно |
| **Итого за первый семестр** (дисциплину)  экзамен |  |
| Промежуточная аттестация  зачёт с оценкой |  | отлично  хорошо  удовлетворительно  неудовлетворительно |
| **Итого за второй семестр** (дисциплину)  зачёт с оценкой |  |

* + - 1. Полученный совокупный результат конвертируется в пятибалльную систему оценок в соответствии с таблицей:

|  |  |  |
| --- | --- | --- |
| **100-балльная система** | **пятибалльная система** | |
| **зачет с оценкой/экзамен** | **зачет** |
|  | отлично  зачтено (отлично) |  |
|  | хорошо  зачтено (хорошо) |
|  | удовлетворительно  зачтено (удовлетворительно) |
|  | неудовлетворительно |  |

# ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

* + - 1. Реализация программы предусматривает использование в процессе обучения следующих образовательных технологий:
    - проблемная лекция;
    - проектная деятельность;
    - групповые дискуссии;
    - анализ ситуаций и имитационных моделей;
    - преподавание дисциплины на основе результатов научных исследований;
    - поиск и обработка информации с использованием сети Интернет;
    - дистанционные образовательные технологии;
    - использование на лекционных занятиях видеоматериалов и наглядных пособий;
    - самостоятельная работа в системе компьютерного тестирования.

# ПРАКТИЧЕСКАЯ ПОДГОТОВКА

* + - 1. Практическая подготовка в рамках учебной дисциплины реализуется при проведении лабораторных работ, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

# ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

* + - 1. При обучении лиц с ограниченными возможностями здоровья и инвалидовиспользуются подходы, способствующие созданию безбарьерной образовательной среды: технологии дифференциации и индивидуального обучения, применение соответствующих методик по работе с инвалидами, использование средств дистанционного общения, проведение дополнительных индивидуальных консультаций по изучаемым теоретическим вопросам и практическим занятиям, оказание помощи при подготовке к промежуточной аттестации.
      2. При необходимости рабочая программа дисциплины может быть адаптирована для обеспечения образовательного процесса лицам с ограниченными возможностями здоровья, в том числе для дистанционного обучения.
      3. Учебные и контрольно-измерительные материалы представляются в формах, доступных для изучения студентами с особыми образовательными потребностями с учетом нозологических групп инвалидов:
      4. Для подготовки к ответу на практическом занятии, студентам с ограниченными возможностями здоровья среднее время увеличивается по сравнению со средним временем подготовки обычного студента.
      5. Для студентов с инвалидностью или с ограниченными возможностями здоровья форма проведения текущей и промежуточной аттестации устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.).
      6. Промежуточная аттестация по дисциплине может проводиться в несколько этапов в форме рубежного контроля по завершению изучения отдельных тем дисциплины. При необходимости студенту предоставляется дополнительное время для подготовки ответа на зачете или экзамене.
      7. Для осуществления процедур текущего контроля успеваемости и промежуточной аттестации обучающихся создаются, при необходимости, фонды оценочных средств, адаптированные для лиц с ограниченными возможностями здоровья и позволяющие оценить достижение ими запланированных в основной образовательной программе результатов обучения и уровень сформированности всех компетенций, заявленных в образовательной программе.

# МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

* + - 1. Характеристика материально-технического обеспечения дисциплины соответствует требованиями ФГОС ВО.
      2. Материально-техническое обеспечение дисциплины при обучении с использованием традиционных технологий обучения.

| **Наименование учебных аудиторий, лабораторий, мастерских, библиотек, спортзалов, помещений для хранения и профилактического обслуживания учебного оборудования и т.п.** | **Оснащенность учебных аудиторий, лабораторий, мастерских, библиотек, спортивных залов, помещений для хранения и профилактического обслуживания учебного оборудования и т.п.** |
| --- | --- |
| **119071, г. Москва, Малый Калужский переулок, дом 1, строение 3** | |
| аудитории для проведения занятий лекционного типа | комплект учебной мебели,  технические средства обучения, служащие для представления учебной информации большой аудитории:   * компьютерная техника (ноутбук/компьютер); * проектор; * экран. |
| аудитории для проведения практических занятий, выполнения лабораторных работ, занятий по практической подготовке, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации | комплект учебной мебели,  технические средства обучения, служащие для представления учебной информации большой аудитории:   * компьютерная техника (ноутбук/компьютер); * проектор; * экран; * персональные компьютеры, подключенные к сети Интернет. |
| **Помещения для самостоятельной работы обучающихся** | **Оснащенность помещений для самостоятельной работы обучающихся** |
| читальный зал библиотеки: | * компьютерная техника, подключение к сети «Интернет» |

* + - 1. Материально-техническое обеспечение учебной дисциплины при обучении с использованием электронного обучения и дистанционных образовательных технологий.

|  |  |  |
| --- | --- | --- |
| **Необходимое оборудование** | **Параметры** | **Технические требования** |
| Персональный компьютер/ ноутбук/планшет,  камера,  микрофон,  динамики,  доступ в сеть Интернет | Веб-браузер | Версия программного обеспечения не ниже: Chrome 72, Opera 59, Firefox 66, Edge 79, Яндекс.Браузер 19.3 |
| Операционная система | Версия программного обеспечения не ниже: Windows 7, macOS 10.12 «Sierra», Linux |
| Веб-камера | 640х480, 15 кадров/с |
| Микрофон | любой |
| Динамики (колонки или наушники) | любые |
| Сеть (интернет) | Постоянная скорость не менее 192 кБит/с |

Технологическое обеспечение реализации программы осуществляется с использованием элементов электронной информационно-образовательной среды университета.

# УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Автор(ы)** | **Наименование издания** | **Вид издания (учебник, УП, МП и др.)** | **Издательство** | **Год**  **издания** | **Адрес сайта ЭБС**  **или электронного ресурса *(заполняется для изданий в электронном виде)*** | **Количество экземпляров в библиотеке Университета** |
| 10.1 Основная литература, в том числе электронные издания | | | | | | | |
| 1 | Синаторов С.В. | Информационные технологии | Учебное пособие | М.: Флинта | 2021 | <https://znanium.com/catalog/document?id=374932> | - |
| 2 | Шитов В.Н. | Информатика и информационно-коммуникационные технологии в профессиональной деятельности | Учебное пособие | М: НИЦ ИНФРА-М | 2022 | <https://znanium.com/catalog/document?id=388696> | - |
| 3 | Немцова Т.И. и др.; под ред. Гагариной Л.Г. | Программирование на языке высокого уровня. Программирование на языке Object Pascal | Учебное пособие | М.: ИД ФОРУМ: НИЦ ИНФРА-М | 2020 | <https://znanium.com/catalog/document?id=362746> | - |
| 4 | Шуляк О.А. | Основы программирования | Учебно-методическая литература | М.: Флинта | 2021 | <https://znanium.com/catalog/document?id=390158> | - |
| 5 | Немцова Т.И., Голова С.Ю., Терентьев А.И.; под ред. Л.Г. Гагариной. | Программирование на языке высокого уровня. Программирование на языке C++ | Учебное пособие | М.: ИД ФОРУМ: ИНФРА-М | 2021 | <https://znanium.com/catalog/document?id=363426> | - |
| 10.2 Дополнительная литература, в том числе электронные издания | | | | | | | |
| 1 | Алексеев Е.Р., Чеснокова О.В., Кучер Т.В. | Free Pascal и Lazarus | Учебник | М: ДМК Пресс | 2010 | <https://znanium.com/catalog/document?id=147218> | - |
| 2 | Плотникова Н.Г. | Информатика и информационно-коммуникационные технологии (ИКТ) | Учебное пособие | М.: РИОР | 2021 | <https://znanium.com/catalog/document?id=370445> | - |
| 3 | Горбатов С.М., Тарасов Ю.С., Наумова М.Г. | Информационные технологии | Учебное пособие | М.: МИСиС | 2016 | <https://znanium.com/catalog/document?id=371025> | - |
| 4 | Федотова Е.Л. | Информационные технологии и системы | Учебное пособие | М.: Издательский Дом ФОРУМ | 2022 | <https://znanium.com/catalog/document?id=386738> | - |
| 5 | М. В. Огнева, Е. В. Кудрина | Программирование на языке С++: практический курс | Учебное пособие | М.: Издательство Юрайт | 2022 | <https://urait.ru/bcode/492984> | - |
| 10.3 Методические материалы (указания, рекомендации по освоению дисциплины (модуля) авторов РГУ им. А. Н. Косыгина) | | | | | | | |
| 1 | Козлов А.М. | Технология объектно-ориентированного пpогpаммиpования на языке Pascal | Учебно-методическое пособие. | М.: ФГБОУ ВО «PГУ им. А.Н. Косыгина» | 2020 | локальная сеть университета | 5 |
| 2 | Козлов А.М. | Технология программирования на языке Pascal | Методические указания | М.: ФГБОУ ВО «PГУ им. А.Н. Косыгина» | 2019 | локальная сеть университета | 5 |
| 3 | Семенов А.А. | Основы объектно-ориентированного программирования в среде С++Builder | Методическое пособие | М.: ИИЦ МГУДТ | 2010 | локальная сеть университета | 5 |

# ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

## Ресурсы электронной библиотеки, информационно-справочные системы и профессиональные базы данных:

|  |  |
| --- | --- |
| **№ пп** | **Электронные учебные издания, электронные образовательные ресурсы** |
|  | *ЭБС «Лань»* [*http://www.e.lanbook.com/*](http://www.e.lanbook.com/) |
|  | *«Znanium.com» научно-издательского центра «Инфра-М»*  [*http://znanium.com/*](http://znanium.com/) |
|  | *Электронные издания «РГУ им. А.Н. Косыгина» на платформе ЭБС «Znanium.com»* [*http://znanium.com/*](http://znanium.com/) |
|  | ЭБС «ИВИС» <http://dlib.eastview.com/> |
|  | Образовательная платформа «ЮРАЙТ»  <https://urait.ru/> |
|  | **Профессиональные базы данных, информационные справочные системы** |
|  | Scopus https://www.scopus.com (международная универсальная реферативная база данных, индексирующая более 21 тыс. наименований научно-технических, гуманитарных и медицинских журналов, материалов конференций примерно 5000 международных издательств); |
|  | Научная электронная библиотека еLIBRARY.RU https://elibrary.ru (крупнейший российский информационный портал в области науки, технологии, медицины и образования); |
|  | База данных в мире Academic Search Complete - обширная полнотекстовая научно-исследовательская. Содержит полные тексты тысяч рецензируемых научных журналов по химии, машиностроению, физике, биологии. <http://search.ebscohost.com> |

## Перечень программного обеспечения

|  |  |  |
| --- | --- | --- |
| **№п/п** | **Программное обеспечение** | **Реквизиты подтверждающего документа/ Свободно распространяемое** |
|  | Windows 10 Pro, MS Office 2019 | контракт № 18-ЭА-44-19 от 20.05.2019 |
|  | Lazarus — открытая среда разработки программного обеспечения на языке Object Pascal для компилятора Free Pascal. | Свободно распространяемое на условиях GNU General Public License. |
|  | Embarcadero C++Builder RAD Studio Professional Academic Concurrent License | № 15-02.01-2459 от 21.12.2021  Embarcadero License Certificate: #546431, #546432, #546433, #546434, #546435 |
|  | Code::Blocks — свободная кроссплатформенная среда разработки на C++. | Свободно распространяемое на условиях GNU General Public License v.3. |

### ЛИСТ УЧЕТА ОБНОВЛЕНИЙ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

В рабочую программу учебной дисциплины внесены изменения/обновления и утверждены на заседании кафедры:

|  |  |  |  |
| --- | --- | --- | --- |
| **№ пп** | **год обновления РПД** | **характер изменений/обновлений**  **с указанием раздела** | **номер протокола и дата заседания**  **кафедры** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |