ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ»

УТВЕРЖДАЮ
Ректор РГАИС
А.О. Аракелова
22 мая 2025 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

«ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ»

Направление подготовки: 09.04.02 «Информационные системы и технологии»
Профиль: «Информационные системы и технологии»
Квалификация (степень) выпускника: магистр
Форма обучения: очная

Москва – РГАИС – 2025

Разработчик: доцент кафедры Информационных технологий Трухманов В.Б. // Рабочая программа учебной дисциплины предназначена для обучающихся по направлению 09.04.02 «Информационные системы и технологии». — М.: Российская государственная академия интеллектуальной собственности (РГАИС), кафедра Информационных технологий, 2025.

© ФГБОУ ВО РГАИС, 2025

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1.1. Цель и задачи дисциплины

Изучение дисциплины «Технологии программирования» направлено на получение знаний в области современных средств, методов и технологий программирования, получение представления у обучающихся о возможностях современных сред программирования, формирование умений и навыков программирования на наиболее распространенных языках и средах программирования, развитие алгоритмического мышления. Изучение дисциплины «Технологии программирования» нацелено на понимание основных принципов разработки программного обеспечения, того, какие технологии программирования следует использовать для решения тех или иных задач в профессиональной деятельности и какие ресурсы для этого требуются.

Целью дисциплины «Технологии программирования» является формирование у обучающихся теоретических знаний, практических навыков и умений в области современных методов, средств и технологий программирования, необходимых с дальнейшей профессиональной деятельности.

Для достижения поставленной цели решаются следующие задачи:

– иметь навыки выявления наиболее популярных у профессиональных разработчиков базовых языков, методов и технологий программирования;

– иметь навыки определения Visual Studio, как наиболее перспективной среды программирования для обучения современным технологиям программирования;

– изучить язык C++ для работы в консоли Visual Studio 2022: его синтаксиса, основных типов данных, основных операторов, приемов программирования;

– изучить основы визуального программирования с использованием языка C#: использования конструктора форм, основных элементов, обработчиков событий на языке C#.

– иметь навыки использования проектной деятельности, как основной формы учебного процесса для формирования необходимых знаний, умений и навыков программирования в будущей профессиональной деятельности.

1.2. Место дисциплины в структуре образовательной программы

«Технология программирования» - дисциплина по выбору блока Б1.О.05 дисциплина реализуется на первом году обучения (1 семестр).

Место дисциплины «Технология программирования» определено, как одна из основных дисциплина, на базе которой выстраивается содержание других учебных дисциплин: базы и банки данных, проектирование и создание автоматизированных информационных систем, программирование на языках высокого уровня, основы мехатроники и робототехники, Web-программирование, разработка приложений для мобильных устройств. Проектная деятельность, заложенная в эту дисциплину, используется на протяжении всего обучения и находит свое воплощение в выпускной квалификационной работе.

2. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ) В ЗАЧЕТНЫХ ЕДИНИЦАХ С УКАЗАНИЕМ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ (АСТРОНОМИЧЕСКИХ) ЧАСОВ ПО ВИДАМ УЧЕБНЫХ ЗАНЯТИЙ

Виды занятий	Объем дисциплины		
	Форма обучения		
	Очная форма обучения	Очно-заочная форма обучения	Заочная форма обучения
Объем зачетных единиц	5	5	5
Общая трудоемкость в часах	180	180	180
Аудиторные занятия	68	68	28
Лекции	34	34	10
Практические занятия (семинары)	34	34	18
Самостоятельная работа	85	85	143
Контроль	27	27	9
Форма контроля	Экзамен	Экзамен	Экзамен

3. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ), СТРУКТУРИРОВАННОЕ ПО ТЕМАМ (РАЗДЕЛАМ) С УКАЗАНИЕМ КОМПЕТЕНЦИЙ, ФОРМИРУЕМЫХ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

3.1. Учебно-тематический план курса и распределение компетенций по темам занятий

Наименование темы	Формируемые компетенции (или их части)		
	УК-1	ОПК-2	ПК-2
Тема 1. Особенности среды программирования Visual Studio 2022. Возможности. Установка и настройка.	+		
Тема 2. Основные типы данных. Операторы ввода-вывода данных. Основные математические операторы.		+	
Тема 3. Условные операторы.		+	
Тема 4. Циклические операторы.		+	
Тема 5. Основы визуального программирования в Visual Studio.		+	
Тема 6. Разработка приложений с использованием строковых типов данных.		+	
Тема 7. Преобразование данных и проведение сложных вычислений на примере создания приложения «Калькулятор».		+	
Тема 8. Создание приложений с использованием динамических объектов.	+	+	+
Тема 9. Символьные типы данных. Управление объектами с помощью прерываний.	+	+	+
Тема 11. Массивы. Операторы работы с массивами. Создание приложений с использованием массивов.		+	+
Тема 12. Создание приложений с использованием анимации и мультипликации.	+	+	+

3.2. Содержание разделов дисциплины (модуля) и контрольные вопросы для самостоятельной работы (самоконтроля) обучающихся

Тема 1. Особенности среды программирования Visual Studio 2022. Возможности. Установка и настройка.

Сравнительный анализ современных методов, средств, технологий программирования. Обзор современных сред программирования.

Особенности среды программирования Visual Studio 2022. Установка среды программирования Visual Studio 2022, поиск в интернете, загрузка. Окно выбора версии: Community 2022, Professional 2022 или Enterprise 2022. Выбор версии Community - бесплатной версии для обучения. Загрузка и запуск установочного файла. Окно Visual Studio Installer. Особенности предлагаемых пакетов программ. Пакеты Разработка классических приложений NET и Разработка классических приложений на C++» Установка Visual Studio 2022. Запуск Visual Studio. Создание проекта. Выбор режимов Консольное приложение C++ и Мастер классических приложений Windows. Настройка нового проекта. Проверка Локального отладчика.

Контрольные вопросы:

1. Что представляет собой Среда программирования Visual Studio 2022? Каковы ее возможности?
2. Как осуществляется ее поиск в интернете и загрузка?
3. Чем отличаются версии Visual Studio 2022: Community 2019, Professional 2019 или Enterprise 2019?
4. Как осуществляется установка Visual Studio 2022 и ее запуск?
5. Как создается проект в режиме консольного приложения?
6. Как осуществляется настройка проекта и проверка локального отладчика?

Тема 2. Основные типы данных. Операторы ввода-вывода данных. Основные математические операторы.

Арифметические типы данных: int (целый), char (символьный), wchar (расширенный символьный), bool (логический), float (вещественный), double (вещественный с двойной точностью). Операторы ввода-вывода данных. Оператор присвоения «=». Стандартные потоки передачи данных от клавиатуры и на дисплей. Класс двунаправленных потоков «iostream», операторы «Cin» и «Cout», использование комбинации операторов. Особенности использования кодировочной таблицы Кириллицы, подключение кодировочной таблицы. Использование библиотеки «std» и оператора «setlocale» для подключения Кириллицы.

Контрольные вопросы:

1. Что такое «данные» в программировании?
2. Что такое «типы данных»?
3. Какие типы данных относятся к арифметическим типам данных?
4. Что представляют собой операторы ввода-вывода данных?
5. Как работает оператор присваивания «=»?

6. Что представляют собой стандартные потоки передачи данных от клавиатуры и на дисплей?
7. Что такое потоковая обработка данных и как она работает в C++?
8. Как работают операторы «Cin» и «Cout»?
9. Что представляет собой кодировочная таблица Кириллица и как она подключается?
10. Как используется библиотека «std» и оператор «setlocale» для подключения Кириллицы?

Тема 3. Условные операторы.

Условный оператор «if». Блок-схема оператора. Полная и укороченная конструкция оператора «if». Составные условия. Составные условия со вложенными «if». Составные условия с логическими операторами. Оператор «Swich». Примеры программ.

Контрольные вопросы:

1. Как работает условный оператор «if»?
2. Как работают полная и укороченная конструкция оператора «if»?
3. Что такое составные условия?
4. Что такое составные условия со вложенными «if»?
5. Какие логические операторы используются в C++?
6. Как работают составные условия с логическими операторами?
7. Как работает оператор «Swich»?

Тема 4. Циклические операторы.

Цикл с параметром «for». Блок схема циклического оператора «for», описание синтаксиса. Параметр цикла, условия цикла, шаг цикла. Прямой и обратный циклы. Использование в качестве счетчика цикла символьных переменных. Цикл с предусловием «while». Цикл с постусловием «do while». Вложенные циклы. Примеры программ.

Контрольные вопросы:

1. Что такое циклический оператор?
2. Что такое итерация?
3. Как работает циклический оператор «for»?
4. Как описываются начальное значение, конечное значение и шаг цикла оператора «for»?
5. Как работает циклический оператор «while»?
6. Как задается начальное значение, конечное значение и шаг цикла оператора «while»?
7. Как работает циклический оператор «do while»?

8. Как задается начальное значение, конечное значение и шаг цикла оператора «do while»?
9. Что такое вложенные циклы и как они работают?

Тема 5. Основы визуального программирования.

Создание визуального проекта. Выбор языка C# и Приложение Windows Forms. Особенности интерфейса визуального проекта. Окно конструктора формы. Окно свойств. Свойства формы проекта. Изменение интерфейса формы с помощью свойств. Панель элементов. Добавление элементов на форму проекта. Свойства Элементов. Изменение интерфейса проекта с помощью свойств элементов. Элементы «TextBox» и «Button». Пример создания приложения.

Контрольные вопросы:

1. Что представляет собой «визуальное программирование»?
2. Как создается проект визуального программирования на языке C#?
3. Как использовать элементы в визуальном программировании?
4. Что такое «свойства» и как их задавать в визуальном программировании?
5. Какие основные свойства формы?
6. Как добавить элемент на форму и задать его свойства?
7. Что такое «событие» и как его создавать?
8. Что такое обработчик событий?

Тема 6. Основы визуального программирования в Visual Studio. Разработка приложений с использованием строковых типов данных.

Строковые типы данных. Объявление строковых типов данных в C++ и C#. Ввод данных в строковую переменную, вывод данных из строковой переменной. Особенности использования элемента TextBox для работы со строковыми данными в Visual Studio. Строка и подстрока. Склеивание строк. Использованием метода Substring для обработки строк. Примеры создания приложений.

Контрольные вопросы:

1. Какой тип данных служит для обработки строк в C++ и C#?
2. Каким может быть максимальный размер строки?
3. Как происходит обращение к элементу строки?
4. Как осуществляется соединение строк?
5. Для чего используется метод Substring и как он работает?

6. Что означает команда «TextBox1.Text = TextBox1.Text + button1.Text;»?

Тема 7. Преобразование данных и проведение сложных вычислений на примере создания приложения «Калькулятор».

Создание интерфейса калькулятора на форме приложения с использованием элементов TextBox и Button. Изменение свойств элементов. Обработчик событий. Создание событий нажатия клавиш цифр и действий. Код записи числа в TextBox. Код очистки экрана. Особенности использования переменных целого типа и вещественных типов в сложных вычислениях. Описание переменных. Использование переменных индикаторов для задания арифметических операций калькуляторов. Использование переменной индикатора для ввода только одной запятой в число. Использование переменных индикаторов первого действия, первого числа для выполнения последовательности вычислений.

Контрольные вопросы:

1. Как удалить все символы в строке?
2. Что представляет собой индикатор действия в приложении «Калькулятор» и как он работает?
3. Что означает команда «a = Convert.ToDouble(textBox1.Text)»?
4. Что означает команда «textBox1.Text = Convert.ToString(c)»?
5. Что представляет собой индикатор первого действия в приложении «Калькулятор» и как он работает?
6. Что представляет собой индикатор первого числа в приложении «Калькулятор» и как он работает?
7. Как работает индикатор запятой в приложении «Калькулятор»?
8. Как идет обработка событий нажатия клавиш действий в приложении «Калькулятор»?
9. Как идет обработка события нажатия клавиши сброса «C» в приложении «Калькулятор»?

Тема 8. Создание приложений с использованием динамических объектов.

Свойства объектов, определяющие их координаты на форме. Использование обработчика событий для изменения координат объектов на форме приложения. Использование генератора случайных чисел для изменения координат объекта. Создание приложения, в котором

динамический объект в виде мячика движется под углом 45^0, при столкновении с границами окна отскакивает по закону отражения.

Контрольные вопросы:

1. Как происходит изменение положения объектов на форме с помощью обработчика событий?
2. Как использовать генератор случайных чисел в обработчике событий для изменения положения объектов на форме?
3. Как создаются динамические объекты в Visual Studio?
4. Как создается картинка динамического объекта и загружается в Picture Box?
5. Для чего используется элемент Timer и как он работает?
6. Как создавать событие работы таймера в обработчике событий?
7. Что означает команда «pictureBox1.Left = pictureBox1.Left + 10»?
8. Как осуществляется «отскок» объекта мячик в программе Арканоид?

Тема 9. Символьные типы данных. Управление объектами с помощью прерываний клавиатуры.

Символьные типы данных. Описание переменных символьного типа. Ввод - вывод значений переменных. Таблица ASC II код. Использование символьных переменных для организации циклов. Примеры программ в консоли. Прерывания. Обработка прерываний. Обработка событий клавиатуры. Создание обработчика события KeyPress. Использование функции возвращения кода нажатой клавиши в обработчике события. Создание иллюзии управления объектом с помощью клавиатуры. Создание приложения управления объектами с помощью клавиатуры.

Контрольные вопросы:

1. Что представляет собой символьный тип данных «char»?
2. Что такое таблица ASC II кода?
3. Как задавать событие «KeyPress» в обработчике событий?
4. Что такое «прерывание» и как осуществляется обработка прерываний в Visual Studio?
5. Как установить фон звездного неба на форме?
6. Как сделать фон картинки динамического объекта прозрачным?
7. Как создать имитацию перемещения объекта, например звездолета по форме приложения с помощью клавиатуры?
8. Что такое «видимые объекты» и «невидимые объекты», какое свойство элемента отвечает за это?

9. Как осуществить имитацию запуска ракеты со звездолета в программе «Звездные войны»?

10. Как осуществить имитацию столкновения объектов с последующим взрывом в программе «Звездные войны»?

Тема 10. Массивы. Операторы работы с массивами. Создание приложений с использованием массивов.

Массивы. Виды массивов. Объявление массивов. Доступ к элементам массивов. Ввод-вывод данных в массивах. Двумерные массивы. Особенности доступа к элементам массива и ввода-вывода данных. Примеры программ в консоли. Разработка приложений с использованием с использованием массивов в качестве памяти. Разработка приложений с использованием с использованием массива в качестве цифрового следа.

Контрольные вопросы:

1. Что такое «массивы»?
2. Как осуществляется объявление массивов?
3. Как происходит обращение к элементам массива?
4. Как происходит ввод и вывод данных в массиве?
5. Что такое «поточная обработка элементов массива»?
6. Что такое двумерный массив, как происходит обработка данных двумерного массива?
7. Что такое «цифровой след» в массиве?
8. Как происходит связь массива с объектами приложения?
9. Как осуществляется управление поведением объектов с помощью данных массива?

Тема 11. Создание приложений с использованием анимации и мультипликации.

Приемы создания эффекта анимации и мультипликации в визуальном программировании. Подготовка рисунков объектов анимации и мультипликации. Загрузка изображений в Select Resourse. Установка и настройка таймеров. Создания кода смены изображений по таймеру. Согласование смены элементов изображений с темпом движения объектов. Использование генераторов случайных чисел для создания эффекта непредсказуемости поведения объектов. Создание приложения движения объектов с использованием анимации и мультипликации.

Контрольные вопросы:

1. Что такое анимация, что такое мультипликация?

2. Какие существуют способы задания анимации и мультипликации?

3. Как создавать эффект движения с помощью смены картинки объекта?

4. Как подготовить картинки для мультипликации?

5. Как динамически загружать рисунки в элемент pictureBox?

6. Как обеспечить очередность смены картинок в элементе pictureBox?

7. Как задать динамику объектов мультипликации в Visual Studio?

8. Как использовать генератор случайных чисел для создания эффекта неожиданности поведения объектов мультипликации в Visual Studio?

3.3. Активные и интерактивные формы проведения занятий

В качестве активных форм проведения занятий по дисциплине «Технологии программирования» предлагаются четыре формы проведения занятий: лекция-беседа, консультационная работа, практическое занятие и проектная деятельность. Выбор интерактивной формы предоставляется непосредственно преподавателю.

Лекция-беседа предполагает непосредственный контакт преподавателя с аудиторией. В начале занятия обучаемые получают материалы лекции в электронном виде.

Во время занятия преподаватель знакомит обучаемых с учебным материалом, акцентирую внимание на разборе примеров программ. Обучаемые имеют возможность воспроизвести программы в визуальной среде программирования на компьютерах. В процессе рассмотрения учебного материала они могут задавать преподавателю уточняющие вопросы. В свою очередь, преподаватель может вносить добавления, расширяющие и углубляющие содержание учебного материала, а также задавать вопросы. Вопросы преподаватель может адресовать как всей аудитории, так и кому-то конкретно. Они могут быть как простые, способные сосредоточить внимание на отдельных важнейших элементах темы, так и проблемные. Обучающиеся, продумывая ответ на заданный вопрос, получают возможность самостоятельно прийти к тем выводам и обобщениям, которые преподаватель должен был сообщить им в качестве новых знаний, либо понять глубину и важность обсуждаемой проблемы, что повышает интерес и степень восприятия материала.

Консультационная работа преподавателя предполагает два вида консультаций: групповые и индивидуальные. Групповые консультации представляют собой своеобразную форму проведения лекционных занятий, основным содержанием которых является разъяснение отдельных, часто наиболее сложных или практически значимых вопросов изучаемой программы. Групповые консультации проводятся в случаях, когда необходимо подробно рассмотреть практические вопросы, недостаточно или совсем не освещенные в лекциях, или при проведении других видов занятий, а также с целью оказания помощи в самостоятельной работе, в подготовке к выполнению лабораторных и практических занятий, в написании рефератов или выпускных работ, сдаче экзаменов и зачетов. Проведение индивидуальных консультаций проводится преподавателем в специально отведенное время. В этом случае к нему за помощью могут обратиться как те, кто испытывает трудности в изучении данной темы, так и обучающиеся, которые хотели бы более глубоко разобраться в вопросах семинара.

Практическое занятие представляет собой разработку компьютерных программ в профессиональной среде программирования. Главная цель практического занятия - закрепление учебного материала, полученных во время лекционных занятий, формирование умений применять полученные знания на практике в будущей профессиональной деятельности.

Проектная деятельность является формой организации учебного процесса, основной задачей которого является разработка учебного программного проекта и самостоятельного доведение его до конечного результата - готового проекта, например, программного приложения. Главная цель проектной деятельности — это закрепление полученных знаний умений и навыков в области программирования в процессе самостоятельной разработки программного приложения в соответствии с техническим заданием. В процессе выполнения проекта на занятии возникает атмосфера творчества, повышающая интерес к учебной дисциплине. На определенной стадии выполнения проекта обучающиеся стремятся расширить свои знания о предметной области изучаемой дисциплины либо в виде консультаций с преподавателем, либо самостоятельно. В проектной деятельности допускается и даже приветствуется усложнения исходного технического задания самими обучающимися в сторону создания более совершенного программного приложения.

4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

4.1. Методические рекомендации по самостоятельному изучению курса (дисциплины)

Самостоятельная работа обучающихся – это индивидуальная или коллективная учебная деятельность, осуществляемая без непосредственного руководства преподавателя. Самостоятельная работа есть особо организованный вид учебной деятельности, проводимый с целью повышения эффективности подготовки обучающихся к последующим занятиям, формирования у них навыков самостоятельной отработки учебных заданий, а также овладения методикой организации своего самостоятельного труда в целом.

Являясь необходимым элементом дидактической связи различных методов обучения между собой, самостоятельная работа обучающихся призвана обеспечить более глубокое, творческое усвоение понятийного аппарата дисциплины, знаний возможностей и особенностей современных технологий программирования.

Во время лекций обучающимся необходимо сосредоточить внимание на её прослушивание, уловить то главное, что скажет лектор. Основные положения лекции, отдельные важные факты и выводы из рассматриваемых вопросов обучающиеся получают в электронном виде, отдельные положения важные для обучающихся нужно записывать. Записи следует делать кратко.

Главным определяющим фактором успешной работы обучающихся является его самостоятельная работа.

Следует обратить особое внимание на самостоятельное изучение предоставленных учебных материалов и рекомендованной учебной литературы. В процессе изучения учебных материалов необходимо самостоятельно разобрать теоретический материал, разобрать примеры в указанной среде программирования и выполнить задания для самостоятельной работы.

Успеха в заочном обучении можно добиться только при правильной организации регулярных занятий. Поэтому обучающимся необходимо систематически заниматься.

Организация самостоятельной работы обучающихся должна строиться по системе поэтапного освоения материала. Метод поэтапного изучения включает в себя предварительную подготовку, непосредственное изучение теоретического содержания источника, обобщение полученных знаний.

Предварительная подготовка включает в себя уяснение цели изучения материала, оценку широты информационной базы анализируемого вопроса, выяснение его научной и практической актуальности. Изучение теоретического содержания заключается в выделении и уяснении ключевых понятий и положений, выявлении их взаимосвязи и систематизации. Обобщение полученных знаний подразумевает широкое осмысление теоретических положений через определение их места в общей структуре изучаемой дисциплины и их значимости для практической деятельности.

Методические рекомендации по проектной деятельности.

Проектная деятельность работа обучающихся является одним из видов учебной деятельности, которая призвана, прежде всего, сформировать навыки разработки программных приложений в соответствии с техническим заданием. Основной целью проектной деятельности дисциплины «Технологии программирования» является закрепление полученных знаний умений и навыков в области программирования в процессе самостоятельной разработки программного приложения.

Ключевым моментом проектной деятельности является разработка технического задания. Проектная деятельность осуществляется в рамках практических занятий, а также самостоятельной работы дома. При разработке технического задания следует ориентироваться на содержание теоретического материала учебной дисциплины и практических занятий. Особое внимание следует уделять разработке структурной схеме программного проекта и взаимосвязи объектов и компонентов. В техническом задании должна быть указаны требования к главной форме проекта и ее интерфейсу, которые позволят определить оптимальный состав элементов интерфейса, события, запрограммировать обработку событий. От того насколько точно составлено техническое задание зависит успешность всей проектной деятельности.

Проектная деятельность должна быть построена таким образом, чтобы обучающиеся имели возможность не только довести проект до готового программного приложения, но и усложнить техническое задание в сторону создания более совершенного программного приложения

Методические рекомендации по работе с литературой.

При самостоятельном изучении основной рекомендованной литературы обучающимся необходимо обратить главное внимание на узловые положения, излагаемые в изучаемом тексте.

Необходимо внимательно ознакомиться с содержанием соответствующего блока информации, структурировать его и выделить в нем центральное звено. Обычно это бывает ключевое определение или совокупность сущностных характеристик рассматриваемого объекта. Для того, чтобы убедиться, насколько глубоко усвоено содержание темы, в конце соответствующих глав и параграфов учебных пособий обычно дается перечень контрольных вопросов, на которые обучающийся должен уметь дать четкие и конкретные ответы.

Работа с дополнительной литературой предполагает умение выделять в ней необходимый аспект изучаемой темы.

В своей совокупности изучение таких подходов существенно обогащает научный кругозор обучающихся. В данном контексте следует учесть, что дополнительную литературу целесообразно прорабатывать на базе уже освоенной основной литературы, изучать комплексно и всесторонне.

Обязательный элемент самостоятельной работы обучающихся с источниками и литературой – ведение необходимых записей. Основными общепринятыми формами записей являются конспект, выписки, тезисы, аннотации, резюме, план.

Конспект – это краткое письменное изложение содержания статьи, доклада, лекции, включающее в сжатой форме основные положения и их обоснование.

Выписки – это краткие записи в форме цитат (дословное воспроизведение отрывков источника, произведения, статьи, содержащих существенные положения, мысли автора), либо лаконичное, близкое к тексту изложение основного содержания.

Тезисы – это сжатое изложение ключевых идей прочитанного источника или произведения.

Аннотации, резюме – это соответственно предельно краткое обобщающее изложение содержания текста, критическая оценка прочитанного документа или произведения.

В целях структурирования содержания изучаемой работы целесообразно составлять ее план, который должен раскрывать логику построения текста, а также способствовать лучшей ориентации обучающегося в содержании произведения.

Самостоятельная работа обучающегося будет эффективной и полезной в том случае, если она будет построена исходя из понимания обучающимися необходимости обеспечения максимально широкого охвата информационных источников, что вполне достижимо при научной организации учебного труда.

4.2. Глоссарий

Embarcadero Technologies - американская компания, занимающаяся разработкой программного обеспечения для создания средств управления базами данных и самих баз данных и администрирования баз данных для платформ Oracle, Microsoft SQL Server, MySQL, Sybase.

Embarcadero Delphi - интегрированная среда разработки ПО для Microsoft Windows, macOS, iOS и Android на языке Delphi (ранее носившем название Object Pascal), созданная первоначально фирмой Borland и на данный момент принадлежащая и разрабатываемая Embarcadero Technologies. Embarcadero Delphi является частью пакета Embarcadero RAD Studio и поставляется в четырёх редакциях: Community (распространяется бесплатно и имеет ограниченную лицензию на использование в коммерческих целях), Professional, Enterprise и Architect.

Embarcadero C++ Builder - современная визуальная среда программирования в основе которой лежит идеология визуального конструирования программ с использованием синтаксиса языка C++. Классифицируется, как средство быстрой разработки приложений.

Microsoft Visual Studio 2022 линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, UWP а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом для всех платформ, поддерживаемых Windows.

Анимация - придание изображению подвижности, мультяшно-двигательных функций. Мультипликационные фильмы по-английски называются «анимационными фильмами».

Визуальное программирование - способ создания компьютерной программы путём манипулирования графическими объектами вместо написания её текста. Интерфейс программы в визуальном программировании создается с помощью элементов и изменения их свойств в панели свойств.

Для изменения свойств объектов в процессе выполнения программы используется обработчик событий со стандартным набором событий. Для создания кода обработчика событий используется объектный язык программирования, в нашем случае C#.

Визуальная среда программирования - современная среда программирования, основанная на парадигме визуального программирования или по-другому визуального конструирования программ. Наиболее популярными средами визуального программирования являются Visual Studio, Delphi, C++ Builder.

Вложенные циклы - циклы, размещенные внутри других циклов. На первом проходе внешний цикл вызывает внутренний, который исполняется до своего завершения, после чего управление передается в тело внешнего цикла. На втором проходе внешний цикл опять вызывает внутренний и так далее до завершения внешнего цикла.

Константа - способ адресации данных, изменение которых программой не предполагается или запрещается.

Конструктор форм - инструмент создания или редактирования формы путем перетаскивания на форму элементов из набора элементов и изменения их свойств.

Консоль отладки - специальное окно в среде программирования предназначенное для ввода информации с клавиатуры и вывода ее на экран, а также для вывода сообщений об ошибках в ходе отладки программы.

Логические операторы - операторы логического (булевского) типа, соответствующие операциям высказываниями в алгебре логики. Как и высказывания, логические выражения могут принимать одно из двух истинностных значений — «истинно» или «ложно». В программировании на языках C++ и C # используются: конъюнкция (&&), дизъюнкция (||), отрицание (!). Они позволяют проверить сразу несколько условий за раз. Их используют для создания сложных условий, например в циклах.

Локальный отладчик - инструмент для поиска и устранения ошибок в разрабатываемых приложениях в Visual Studio. Осуществляет пошаговое выполнение кода программы в поисках точки, в которой была допущена ошибка. Таким образом пользователю становится понятно, в каком месте нужно внести исправления в код. Отладчик также позволяет вносить временные изменения, чтобы продолжить работу с программой.

Массив - тип данных, в котором хранится упорядоченный набор однотипных элементов. Массивы нужны для удобного хранения большого числа значений и быстрого и удобного доступа к ними. Структуру данных можно представить по аналогии с набором пронумерованных коробок, в

каждой из которых находится какой-то предмет. Этот предмет — элемент массива, а номер на коробке — индекс элемента, порядковый номер, по которому его можно найти. У массивов есть альтернативные названия: матрица, вектор, ряд.

Межпарадигмальный подход в программировании - объединение в одной среде программирования нескольких парадигм программирования. Например, в Visual Studio в языке C# удачно сочетаются парадигмы структурного, модульного и визуального программирования.

Мультипликация - технические приёмы создания иллюзии движущихся изображений с помощью последовательности неподвижных изображений (кадров), сменяющих друг друга с большой частотой (от 12 кадров в секунду для рисованной мультипликации до 30 кадров в секунду для компьютерной анимации).

Обработчик событий - программа, которая выполняются в случае наступления определенного события (нажатия на кнопку, изменения содержимого текстового поля, щелчка мышью элементе и т. д.).

Оператор ввода-вывода данных - операторы, которые позволяют ввести в программу данные во время выполнения программы и осуществить вывод рассчитанных данных в понятном человеку виде.

Отладка программы - поиск (локализация), анализ и устранение ошибок в программном приложении, которые были найдены во время тестирования.

Парадигма программирования - совокупность идей и понятий, определяющих стиль разработки компьютерных программ и реализованный на языке программирования. Различают парадигмы: линейное программирование, структурное программирование, модульное или функциональное программирование, объектно-ориентированное программирование и визуальное программирование. Парадигмы программирования связывают с эволюционным развитием методов и средств программирования. В современных средах программирования, например в Visual Studio, используется межпарадигмальный подход.

Переменные - поименованная либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным. Переменную можно представить по аналогии с коробочкой, в которую можно что-то положить. То, что лежит в коробочке, является значением переменной.

Потоки ввода-вывода данных - упорядоченные последовательности данных, которым соответствует определенный источник для потоков

ввода или получатель для потоков вывода. В C++ для связи с клавиатурой используется поток cin, для связи с экраном cout.

Преобразование данных - перевод данных из одного типа в другой, например из строки символов в число и наоборот, из вещественного типа в целый путем округления и т. д.

Прерывания клавиатуры - событие нажатия на какую-либо клавишу клавиатуры в процессе выполнения программы. Обработка событий нажатия клавиши осуществляется по значению вводимой символьной переменной либо в виде символа, либо кода ASC II. Обычно в программе по прерыванию выполняется какая-либо команда или запускается подпрограмма.

Программирование – процесс создания компьютерных программ на одном из языков программирования.

Программное приложение - отлаженная программа и комплекс программ ориентированных на решение конкретных задач и рассчитанная на взаимодействие с пользователем.

Свойства формы и элементов - способ доступа к параметрам объекта, через которые задаются его свойства. Обращение к свойству элемента можно рассматривать как обращение к полям объекта в объектном программировании.

События в программировании - действиями пользователя с клавиатурой, мышью, сенсорным экраном, сообщениями других программ и потоков.

Символьные типы данных - представляет собой тип данных, предназначенный для хранения одного символа (буквы, знака или кода). В переменную этого типа может быть помещен любой из 256 символов расширенного кода ASC II. Это буквы ['A'…'Z', 'a'…'z'], ['А'…'Я', 'а'…'я'], цифры ['0'…'9'], знаки препинания и специальные символы.

Строковые типы данных - тип данных значениями которого является произвольная последовательность (строка) символов алфавита. Переменная строкового типа может иметь длину до 255 символов. Обрабатывать строку можно целиком с помощью специальных процедур или функций, либо поэлементно, обращаясь к каждому элементу по его порядковому номеру.

Технология программирования - совокупность методов и средств, используемых в процессе разработки программного обеспечения.

Типы данных - фундаментальное понятие языка программирования, которое определяет, что именно представляют собой данные, как они хранятся в памяти компьютера, как осуществляется доступ к ним, какие действия с ними можно осуществлять и в какой последовательности.

Условные операторы - представляет собой оператор ветвления и используется для разветвления процесса вычислений на два направления. Сначала проверяется условие. Если условие выполняется, то выполняется Оператор 1, если не выполняется, то выполняется Оператор 2. После этого управление передается на оператор, следующий за условным.

Циклические операторы - используются для организации многократно повторяющихся вычислений. Любой цикл состоит из тела цикла, то есть тех операторов, которые выполняются несколько раз, начальных установок, модификации параметра цикла и проверки условия продолжения выполнения цикла.

Цифровой след (или цифровая тень) - прием в программировании, когда поведение динамического объекта определяется значениями элементов массива, определяемых программным кодом. Изменение положения объекта или изменение каких-то других его свойств отражается в соответствующих элементах массива.

Язык программирования - формальный язык, для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических, семантических правил, определяющих внешний вид программы и действия, которые выполнит компьютер под её управлением.

5. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Оценка качества освоения обучающимися образовательных программ включает в себя порядок, периодичность, систему оценок и формы проведения текущего контроля успеваемости и промежуточной аттестации обучающихся.

Нормативно-методическое обеспечение текущего контроля успеваемости и промежуточной аттестации обучающихся осуществляется в соответствии с положением ФГБОУ ВО РГАИС «Об осуществлении текущего контроля успеваемости и промежуточной аттестации обучающихся».

Основными задачами текущего контроля успеваемости является систематический мониторинг за формированием компетенций, предусмотренных ФГОС ВО и ООП, повышение качества знаний обучающихся, приобретение и развитие навыков самостоятельной работы, повышение академической активности обучающихся.

Критерии оценки обучающихся

Текущая аттестация (текущий контроль) уровня усвоения содержания дисциплины возможно проводить в ходе всех видов учебных занятий методами устного и письменного опроса (работ), в процессе выступлений обучающихся на практических занятиях, защиты рефератов, а также посредством тестирования.

Качество письменных работ оценивается исходя из того, что обучающиеся:

- выбрали и использовали форму и стиль изложения, соответствующие целям и содержанию дисциплины;
- применили связанную с темой информацию, используя при этом понятийный аппарат специалиста в данной области;
- представили структурированный и грамотно написанный текст, имеющий связное содержание.

Тестовые материалы оцениваются по процентному соотношению правильных вариантов. Количество правильных ответов в пределах от 90 до 100 % - «отлично»; в пределах от 75 до 89 % - «хорошо»; в пределах от 50 до 74 % - «удовлетворительно»; менее 50 % - «неудовлетворительно».

Сдача зачета происходит в устной форме по билетам. В ходе зачета студент должен продемонстрировать знания и умения по предмету учебного курса. Качество ответов студентов и выполнение заданий оценивается: «зачтено», «зачтено с оценкой» и/или «не зачтено», «не зачтено с оценкой».

«зачтено», «зачтено с оценкой»:

- полные, осознанные знания в рамках курса лекций и дополнительной литературы, логичное и грамотное изложение материала.

«не зачтено» «не зачтено с оценкой»:

- допускаются существенные ошибки в знании курса лекций, при ответе вскрывается ошибочное понимание основных понятий курса.

Сдача экзамена происходит в устной форме по билетам.

Качество ответов на экзамене оцениваются на «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

Оценка «отлично» выставляется обучающемуся, если:

- даны исчерпывающие и обоснованные ответы на все поставленные вопросы, правильно решены практические задачи;
- ответы были четкими и краткими, основные мысли излагались в строгой логической последовательности;
- обучающийся продемонстрировал умение самостоятельно анализировать факты, события, явления, процессы в их взаимосвязи и диалектическом развитии.

Оценка «хорошо» выставляется обучающемуся, если:

- даны полные, достаточно обоснованные ответы на поставленные вопросы, правильно решены практические задания;
- в ответах не всегда выделялось главное, при решении практических задач не всегда использовались рациональные методики расчётов;
- ответы в основном были краткими, но не всегда четкими.

Оценка «удовлетворительно» выставляется обучающемуся, если:

- даны в основном правильные ответы на все поставленные вопросы, но без должной глубины и обоснования, при решении практических задач студент использовал прежний опыт и не применял новые методики выполнения расчётов, однако на уточняющие вопросы даны в целом правильные ответы;
- при ответах не выделялось главное;
- ответы были многословными, нечеткими и без должной логической последовательности;
- на отдельные дополнительные вопросы не даны положительные ответы.

Оценка «неудовлетворительно» выставляется обучающемуся, если не выполнены требования, соответствующие оценке «удовлетворительно».

Обучающиеся, пропустившие свыше 75% учебного времени, не аттестуются по итогам семестра. Вопрос об аттестации таких обучающихся решается в индивидуальном порядке.

5.1. Список вопросов к экзамену

1. Что представляет собой Среда программирования Visual Studio 2022? Каковы ее возможности? Как создается проект в режиме консольного приложения?
2. Что такое «данные» и «типы данных» в программировании? Какие типы данных относятся к арифметическим типам данных?
3. Что представляют собой операторы ввода-вывода данных? Как работает оператор присваивания «=»?
4. Что такое потоковая обработка данных и как она работает в C++? Как работают операторы «Cin» и «Cout»?
5. Что представляет собой кодировочная таблица Кириллица и как она подключается? Как используется библиотека «std» и оператор «setlocale» для подключения Кириллицы?
6. Как работает условный оператор «if»? Как работают полная и укороченная конструкция оператора «if»? Что такое составные условия? Что такое составные условия со вложенными «if»?
7. Какие логические операторы используются в C++? Как работают составные условия с логическими операторами? Как работает оператор «Swich»?
8. Что такое циклический оператор? Как работает циклический оператор «for»?
9. Как работает циклический оператор «while»?
10. Как работает циклический оператор «do while»?
11. Что такое вложенные циклы и как они работают?
12. Что представляет собой «визуальное программирование»? Как создается проект визуального программирования на языке C#? Как использовать элементы и свойства в визуальном программировании?
13. Как добавить элемент на форму и задать его свойства? Что такое «событие» и как его создавать? Что такое обработчик событий?
14. Какой тип данных служит для обработки строк в C++ и C#? Как происходит обращение к элементу строки? Как осуществляется соединение строк?

15. Для чего используется метод Substring и как он работает? Как удалить все символы в строке?

16. Что такое переменная индикатор и как работают переменные индикаторы в в приложении «Калькулятор»?

17. Как работают команды «a = Convert.ToDouble(textBox1.Text)» и «textBox1.Text = Convert.ToString(c)»?

18. Что такое динамические объекты? Как создаются динамические объекты в Visual Studio? Как создается картинка динамического объекта и загружается в Picture Box?

19. Как создаются динамические объекты в Visual Studio? Для чего используется элемент Timer и как он работает? Как создавать событие работы таймера в обработчике событий?

20. Что представляет собой символьный тип данных «char»? Что такое таблица ASC II кода? Как задавать событие «KeyPress» в обработчике событий?

21. Что такое «прерывание» и как осуществляется обработка прерываний в Visual Studio?

22. Как установить фон звездного неба на форме? Как сделать фон картинки динамического объекта прозрачным?

23. Как создать имитацию перемещения объекта, например звездолета по форме приложения с помощью клавиатуры?

24. Что такое «видимые объекты» и «невидимые объекты», какое свойство элемента отвечает за это? Как осуществить имитацию запуска ракеты со звездолета в программе «Звездные войны»?

25. Что такое «массивы»? Как осуществляется объявление массивов? Как происходит обращение к элементам массива? Как происходит ввод и вывод данных в массиве?

26. Что такое «поточная обработка элементов массива»? Что такое двумерный массив, как происходит обработка данных двумерного массива?

27. Что такое «цифровой след» в массиве? Как происходит связь массива с объектами приложения? Как осуществляется управление поведением объектов с помощью данных массива?

28. Что такое анимация, что такое мультипликация? Какие существуют способы задания анимации и мультипликации? Как создавать эффект движения с помощью смены картинки объекта?

29. Как подготовить картинки для мультипликации? Как динамически загружать рисунки в элемент pictureBox? Как обеспечить очередность смены картинок в элементе pictureBox?

30. Как задать динамику объектов мультипликации в Visual Studio? Как использовать генератор случайных чисел для создания эффекта неожиданности поведения объектов мультипликации в Visual Studio?

5.2. Тестовые задания

1. Как подключить стандартную библиотеку iostream?
a) #include <iostream>;
b) #include <iostream.h>;
c) #include “iostream”;
d) #include ‘iostream.h’.

2. Как правильно подключить русский язык?
a) # Setlocale<LC_All, “Russian”>;
b) # Setlocale<“Russian”>;
c) Setlocale(LC_All,“Russian”);
d) Setlocale(“Russian”).

3. Где правильного инициализирована переменная целого типа?
a) int a4
b) int a=5;
c) float a;
d) char a=3.

4. Что выведет на экран данный фрагмент кода программы?
const int x=22;
x++;
cout <<x<< endl;

a) 22;
b) 23;
c) Ошибка компиляции;
d) NULL.

5. Где правильно указан комментарий?
a) # здесь комментарий;
b) /* здесь комментарий/*;
c) /# здесь комментарий;
d) // здесь комментарий.

6. Что выведет на экран данный фрагмент кода программы?

```
int a=5;
int b;
b=++a;
cout << a << b << endl;
```

a) 66;
b) 55;
c) 56;
d) 65.

7. Что выведет на экран данный фрагмент кода программы?

```
int a=5;
int b=6;
int c=a + 5 * --b;
cout <<  c << endl;
```

a) 40;
b) 30;
c) 35;
d) 11.

8. Что выведет на экран данный фрагмент кода программы?

```
int a=9;
int b=6;
int c=4;
if ((a>b) && (c>a-b)) cout << a << b << c << endl;
else cout << c << b << a << endl;
```

a) 964;
b) 469;
c) 496;
d) 947.

9. Укажите оператор выбора в языке С ++.

a) case;
b) choice;
c) switch … case …;
d) default.

10. Что выведет на экран данный фрагмент кода программы?

```
int i, a=5, c=0;
for (i=1; i<=10; i++) c=c+a;
cout << c << endl;
```

a) 45;
b) 50;
c) 55;
d) 0.

11. Что выведет на экран данный фрагмент кода программы?

```
float x, x1, dx;
x = 3;
x1 = 8;
dx = 0.2;
while (x <= x1) {
      cout << x << " ";
      x = x + dx; }
```

a) последовательность цифр от 3 до 10 с шагом 0,2,;
b) последовательность цифр от 10 до 3 с шагом 0,2,;
c) последовательность цифр от 8 до 3 с шагом 2,;
d) последовательность цифр от 3 до 8 с шагом 0,2.

12. Что выведет на экран данный фрагмент кода программы?

```
float x, x1, dx, sum;
x = 3;
x1 = 5;
dx = 0.5;
sum = 0;
do {sum = sum + x; x = x + dx;} while (x <=  x1);
cout << sum;
```

a) 20;
b) 15;
c) 25;

d) 0.

13. Что выведет на экран данный фрагмент кода программы?

```
for (int y =1; y <= 10; y++) {
        for (int x =1; x <= 10; x++)
cout << 0 << " ";
        cout << endl; }
```

a) Сообщение об ошибке;
b) Квадрат 10 на 10 состоящий из нулей;
c) Квадрат 9 на 9 состоящий из нулей;
d) Строку из 100 нулей.

14. Что выведет на экран данный фрагмент кода программы?

```
for (int y =1; y <= 10; y++) {
        for (int x =1; x <= 10; x++) {
if ((x = =y) || (y = = 11 – x)) cout << 1 << " ";
else cout << 0 << " "; }
        cout << endl; }
}
```

a) Квадрат 9 на 9 состоящий из нулей;
b) Квадрат 9 на 9 состоящий из единиц;
c) Квадрат 10 на 10 состоящий из нулей с диагоналями из единиц;
d) Квадрат 10 на 10 состоящий из нулей с крестом посередине из единиц.

15. Что выведет на экран данный фрагмент кода программы?

```
string st, st1="Здаравстуйте, товарищи!";
for (int i =1; i <= 1; i++)  st = st + st1[i];
cout << st << endl;
```

a) Здравствуйте;
b) иЗдравствуй;
c) товарищи;
d) , товарищи!.

16. Что такое массив?

a) Таблица, хранящая различные значения;

b) Строка или таблица, в которой могут храниться значения одного типа;

c) Структура данных, хранящая набор значений одного типа, объединенных под одним единым именем и идентифицируемых по индексу;

d) Ячейка в памяти компьютера, где может находиться одно значение.

17. Дан массив int array[5] = { 3, 10, 7, 9, 2}. Как обратиться к числу 7?

a) array[7];
b) array[2];
c) array[3];
d) array[2+].

18. Что делает фрагмент кода программы?

```
int b=0;
for (int i = 0; i < 10; i++)
  b+=array[i];
```

a) Определяет индекс максимального элемента массива array;
b) Подсчитывает количество элементов массива array;
c) Вычисляет сумму индексов массива array;
d) Посчитывает сумму первых 10 элементов массива array.

19. Что выведет на экран данный фрагмент кода программы?

```
int array [3][6] = {{1, 2,- 3, 4, 5,- 6},
{-9, 8, 7, 6, -5, 4},
{1, 9, 2, -8, 3, 7}};
int b = 0;
for (int i = 0; i < 3; i++)
   if (array [1][i] > 0) b + = array [1][i];
cout << b << endl;
```

a) 15;
b) 6;
c) 3;
d) 12.

20. Что выведет на экран данный фрагмент кода программы?

```
int x, y;
```

```
int mass [9][9];
for (y = 1; y <= 8; y++)
for (x = 1; x <= 8; x++)
mass[y][x] = 0;
for (y = 1; y <= 8; y++)
{for (x = 1; x <= 8; x++)
cout << mass[y][x] << " ";
cout << endl;}
```

a) Квадрат 9 на 9, состоящий из нулей;
b) Квадрат 8 на 8, состоящий из нулей;
c) 0;
d) 99.

21. Какое действие выполнит данный фрагмент кода программы?

```
button1.Left = button1.Left + 150;
```

a) Переместит кнопку на 150 пикселей вправо;
b) Переместит кнопку на 150 пикселей влево;
c) Переместит кнопку на 150 пикселей вверх.

22. Какое действие выполнит данный фрагмент кода программы?

```
button1.Top = button1.Top - 150;
```

a) Переместит кнопку на 150 пикселей вправо;
b) Переместит кнопку на 150 пикселей влево;
c) Переместит кнопку на 150 пикселей вверх.

23. Какое действие выполнит данный фрагмент кода программы?

```
Random rnd = new Random()
```

a) Выберет случайное действие;
b) Инициирует генератор случайных чисел rnd;
c) Выберет случайную комбинацию клавиш.

24. Что выведет на экран данный фрагмент кода программы?

```
char ch;
for (ch ='a'; ch<=z; ch++)
{
```

```
cout << ch << " ";
}
```

a) Последовательность букв русского алфавита от "а" до "я"
b) Последовательность букв латинского алфавита от "z" до "a";
c) Последовательность букв латинского алфавита от "a" до "z".

25. Для чего предназначено событие KeyPress?

a) Обработки прерываний клавиатуры;
b) Разметки поля формы;
c) Отмены последнего действия выполнения программы.

26. Какое действие выполнит данный фрагмент кода программы

```
ch = e.KeyChar;
if (ch == 'd') pictureBox1.Left + 10;
```

a)Удалит рисунок из pictureBox1
b) При нажатии на клавишу d переместит картинку на 10 пикселей вправо;
c) Заблокирует клавишу d.

27.Какое действие выполнит данный фрагмент кода программы

```
Timer1.Enabled = true;
```

a) Включит Timer1;
b) Отключит Timer1;
c) Инициирует генератор случайных чисел.

28. Какое событие таймера Timer1 задает последовательность команд, которые он будет выполнять?

a) Do;
b) Tick;
c) Work.

29.Какое действие выполнит данный фрагмент кода программы

```
pictureBox1.Visible = False;
```

a) Запустит генератор случайных чисел;
b) Удалит рисунок из инспектора объектов;
c) Сделает рисунок невидимым.

30.Какое действие выполнит данный фрагмент кода программы

```
pictureBox1.Image = Properties.Resources.juk;
```

a) Загрузит рисунок в контейнер из файла Juk;
b) Изменит свойство Image контейнера рисунков;
c) Удалит контейнер рисунков из инспектора ресурсов.

Ключ
к демоверсии теста по дисциплине
«Технологии программирования»

1	2	3	4	5
a	c	b	c	d
6	7	8	9	10
a	b	a	c	b
11	12	13	14	15
d	a	b	c	a
16	17	18	19	20
c	b	d	a	b
21	22	23	24	25
a	c	b	c	a
26	27	28	29	30
b	a	b	c	a

6. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Основная и дополнительная учебная литература

Основная литература

1. Горелов, С. В. Современные технологии программирования: разработка Windows-приложений на языке C# : учебник для студентов, обучающихся по дисциплине «Современные технологии программирования», направление «Прикладная информатика» (09.03.03 — для бакалавров, 09.04.03 — для магистров) : в 2 томах / С. В. Горелов ; под науч. ред. П. Б. Лукьянова ; Финансовый университет при Правительстве Российской Федерации. – Москва : Прометей, 2019. – Том 2. – 379 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=576036 . – Библиогр. в кн. – ISBN 978-5-907100-18-3. – Текст : электронный.
2. Горелов, С. В. Современные технологии программирования: разработка Windows-приложений на языке C# : учебник для студентов, обучающихся по дисциплине «Современные технологии программирования», направление «Прикладная информатика» (09.03.03 — для бакалавров, 09.04.03 — для магистров) : в 2 томах / С. В. Горелов ; под науч. ред. П. Б. Лукьянова ; Финансовый университет при Правительстве Российской Федерации. – Москва : Прометей, 2019. – Том 1. – 363 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=576037 . – Библиогр. в кн. – ISBN 978-5-907100-09-1. – Текст : электронный.

Дополнительная литература

1. Лебеденко, Л. Ф. Основы программирования на C++ : учебное пособие / Л. Ф. Лебеденко, О. И. Моренкова. – 2-е изд., перераб. и доп. – Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2021. – 200 с. : ил., табл., схем. – Режим доступа: по подписке.

– URL: https://biblioclub.ru/index.php?page=book&id=694769 . – Библиогр. в кн. – Текст : электронный.

2. Лафоре Р. Объектно-ориентированное программирование в C++ / Р.Лафоре. – 4-е изд. – Спб.: Питер, 2022. – 928с.

Библиотечный фонд Академии укомплектован печатной или электронной основной учебной литературой по дисциплинам базовой части всех циклов, изданными за последние 5 лет.

Фонд дополнительной литературы включает в себя официальные справочно-библиографические и периодические издания в расчете не менее одного экземпляра на каждые 100 обучающихся. Каждому обучающемуся обеспечен доступ к комплектам библиотечного фонда и периодическое издание из следующего перечня: Копирайт; wipo magazine; Библиотековедение; Биржа интеллектуальной собственности (БИС); Бюллетень Министерства юстиции Российской Федерации; Вестник гражданского права; Государство и право; Инновации; Интеллектуальная собственность. Авторское право и смежные права; Интеллектуальная собственность. Промышленная собственность; Международное публичное и частное право; Общество: социология, психология, педагогика; Патентный поверенный; Патенты и лицензии. Интеллектуальные права; Уголовное право; Управление проектами и программами; Хозяйство право; Экономическая политика.

7. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ИНФОРМАЦИОННО-СПРАВОЧНЫХ СИСТЕМ И РЕСУРСОВ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

В процессе реализации образовательной программы в вузе применяются современные интерактивные и мультимедийные средства обучения (компьютеры, мультимедиапроекторы, интерактивные доски и др.), тематические стенды и плакаты, а также электронные информационные образовательные ресурсы.

На основе аппаратно-программного комплекса в РГАИС функционирует и постоянно совершенствуется портал электронного обучения и дистанционных образовательных технологий (ЭОиДОТ), обеспечиваемый преимущественно авторским учебным контентом и методическими разработками профессорско-преподавательского состава Академии.

В РГАИС функционируют читальный зал и электронная библиотека. Сотрудникам и обучающимся обеспечен доступ к электронной библиотечной системе «Университетская библиотека онлайн», насчитывающей более 100 тысяч наименований изданий с доступом в режиме онлайн, а также к объектам Национальной электронной библиотеки (в соответствии с договором с ФГБУ «Российская государственная библиотека»).

Имеется компьютерный класс, возможности которого позволяют каждому из обучающихся работать на компьютере с установленным комплектом лицензионного программного обеспечения не менее 20 часов в год. Академия обеспечена необходимым комплектом лицензионного программного обеспечения

Электронная информационно-образовательная среда Академии обеспечивает:

- доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем и электронным образовательным ресурсам, указанным в рабочих программах;
- фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения программы;
- формирование электронного портфолио обучающегося, в том числе сохранение его работ и оценок за эти работы.
- доступ к современным профессиональным базам данных, информационным справочным и поисковым системам, в том числе:

справочно-правовой системе «Гарант»: www.garant.ru; справочно-правовой системе «Консультант плюс»: www.consultant.ru; библиотеке «Книгофонд»: www.knigafund.ru; Университетской библиотеке www.biblioclub.ru.

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Для ведения образовательной деятельности по данной дисциплине Академия располагает материально-технической базой, обеспечивающей проведение всех видов лабораторной, практической и научно-исследовательской работы обучающихся, предусмотренных учебным планом РГАИС, и соответствующей действующим санитарным и противопожарным правилам и нормам.

Для организации и ведения учебного процесса Академия располагает зданием общей площадью 5936,2 кв.м, учебная и учебно-лабораторная площадь составляет 1249,6 кв.м. Для питания сотрудников и обучающихся имеется столовая площадью 130,1 кв.м.

Аудиторные занятия проводятся в специальных помещениях, представляющих собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также в помещениях для самостоятельной работы. Имеются помещения для хранения и профилактического обслуживания учебного оборудования. Специальные помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа имеются наборы демонстрационного оборудования и учебно-наглядных пособий, обеспечивающие тематические иллюстрации, соответствующие примерным программам дисциплин (модулей), рабочим учебным программам дисциплин (модулей).

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

9. ОСОБЕННОСТИ ОБУЧЕНИЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Организация образовательного процесса для лиц с ограниченными возможностями здоровья осуществляется в соответствии с приказом Минобрнауки России от 9 июня 2016 г. № 694 «О внесении изменений в административные регламенты предоставления государственных услуг в части обеспечения условий доступности государственных услуг для инвалидов», «Методическими рекомендациями по организации образовательного процесса для инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса» Министерства образования и науки РФ от 08.04.2014 г. № АК-44/05вн.

Академия предоставляет инвалидам и лицам с ограниченными возможностями здоровья (по их заявлению) возможность обучения по образовательной программе, учитывающей особенности их психофизического развития, индивидуальных возможностей и при необходимости, обеспечивающей коррекцию нарушений развития и социальную адаптацию указанных лиц. Для инвалидов и лиц с ограниченными возможностями здоровья Академия устанавливает особый порядок освоения дисциплин (модулей).

Подбор и разработка учебных материалов для обучающихся с ограниченными возможностями здоровья производится с учетом их индивидуальных особенностей.

Предусмотрена возможность обучения по индивидуальному графику.