**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ»**

УТВЕРЖДАЮ
Ректор РГАИС
А.О. Аракелова
25 мая 2026 г.

# РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

# «ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ»

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

**Москва – РГАИС – 2026**

**Разработчик:** доцент кафедры Информационных технологий В.Б. Трухманов «Программирование на языках высокого уровня» // 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.

---

**© ФГБОУ ВО РГАИС, 2026**

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

## 1.1. Цель и задачи дисциплины

Изучение дисциплины «Программирование на языках высокого уровня» 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 «Программирование на языках высокого уровня» 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 «Программирование на языках высокого уровня» 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:**

– анализировать возможности одной из наиболее популярных у профессиональных разработчиков интегрированной среды программирования Delphi 10.4 Community Edition;

– иметь навыки определения Delphi 10.4 Community Edition как одной из наиболее перспективной среды программирования для обучения современным технологиям программирования;

– изучить язык Pascal для работы в Console Application Delphi: его синтаксиса, основных типов данных, основных операторов, приемов программирования;

– изучить основы визуального программирования Windows VCL Application – Delphi: использования конструктора форм, основных элементов, обработчиков событий;

– изучить возможности разработки программных приложений в Windows VCL Application – Delphi: создание и обработка прерываний,

создание динамических объектов создание эффектов анимации, создание и использование баз данных;

– 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

«Программирование на языках высокого уровня» (ПЯВУ) - дисциплина части учебного плана, формируемой участниками образовательных отношений и реализуется на первом году обучения (2 семестр).

Место дисциплины «Программирование на языках высокого уровня» определено, как одна из основных дисциплина, которая дополняет содержание дисциплины «Технологии программирования» и на базе которой выстраивается содержание других учебных дисциплин: базы и банки данных, проектирование и создание автоматизированных информационных систем, 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 | 2 | 2 | 2 |
| Общая трудоемкость в часах | 72 | 72 | 72 |
| Аудиторные занятия | 34 | 34 | 14 |
| Лекции | 6 | 6 | 4 |
| Практические занятия (семинары) | 28 | 28 | 10 |
| Самостоятельная работа | 38 | 38 | 54 |
| Контроль | | | 4 |
| Форма контроля | Зачет | Зачет | Зачет |

# 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 | ПК-2 |
| Тема 1. Интегрированная среда программирования Delphi. Основы программирования в консоль на языке Pascal. | + | |
| Тема 2. Визуальное программирование в среде Delphi. | + | + |
| Тема 3. Разработка программных проектов в среде Delphi. | | + |

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

**Тема 1. Интегрированная среда программирования Delphi. Основы программирования в консоль на языке Pascal.**

Среда программирования Delphi 10.4 Community Edition. Программирование в консоль. Язык программирования Pascal и Console Application Delphi. Структура программ языка Pascal. Объявление глобальных переменных, констант, типов и меток. Основные типы данных. Порядковые и вещественные типы данных. Ввод вывод данных. Ветвления. Операторы повторений.

*Контрольные вопросы:*

1. Что представляет собой структура программ языка Pascal.
2. Как осуществляется описание глобальных переменных?
3. Какие основные типы используются в языке Pascal?
4. Как работает условный оператор «if»?
5. Как работают полная и укороченная конструкция оператора «if»?
6. Что такое составные условия?
7. Что такое составные условия со вложенными «if»?
8. Какие логические операторы используются в Pascal в составных условиях?
9. Что представляет собой оператор «case .. of»?

10. Что такое оператор повторения?
11. Что такое итерация?
12. Как работает циклический оператор «for»?
13. Как описываются начальное значение, конечное значение и шаг цикла оператора «for»?
14. Как работает циклический оператор «while»?
15. Как работает циклический оператор «repeat .. until»?
16. Что такое вложенные циклы и как они работают?

**Тема 2. Визуальное программирование в среде Delphi.**

Создание визуального проекта в Windows VCL Application – Delphi. Форма приложения. Инспектор свойств. Инспектор объектов. Инспектор форм. Палитра компонентов. Добавление компонентов на форму проекта. Свойства компонентов. Палитры компонентов Standard, Additional, Dialogs. Создание интерфейса приложения с помощью компонентов и изменения их свойств. Создание визуальных приложений в Windows VCL Application – Delphi. Разработка программного проекта с использованием сложных вычислений. Особенности использования переменных целого, вещественного и строкового типов в сложных вычислениях. Описание переменных. Функции конвертирования значений переменных одного типа в другой. Переменные индикаторы. Особенности использования переменных индикаторов в проектах Windows VCL Application – Delphi.

*Контрольные вопросы:*

1. Что представляет собой «визуальное программирование»?
2. Как создать визуальный проект в Windows VCL Application – Delphi?
3. Что такое форма приложения?
4. Что такое инспектор свойств?
5. Что такое инспектор событий?
6. Что такое палитра компонентов?
7. Как создается интерфейс программного приложения?
8. Как изменять свойства компонентов с помощью обработчика событий?
9. Как происходит описание переменных в Windows VCL Application – Delphi?
10. Какие существуют функции конвертирования значений переменных одного типа в другой в Windows VCL Application – Delphi?
11. Что такое переменные индикаторы?

12. Как влияют переменные индикаторы на алгоритм обработки события?

**Тема 3. Разработка программных проектов в среде Delphi.**

Свойства компонентов, определяющие их положение на форме. Использование обработчика событий для изменения координат объектов на форме приложения. Использование генератора случайных чисел для изменения координат объекта. Символьные типы данных. Описание переменных символьного типа. Прерывания. Обработка прерываний. Обработка событий клавиатуры. Создание обработчика события OnKeyDown. Использование функции Key возвращения кода нажатой клавиши в обработчике события. Создание иллюзии управления объектом с помощью клавиатуры. Приемы создания эффекта анимации и мультипликации в визуальном программировании. Подготовка рисунков объектов анимации и мультипликации. Загрузка изображений в Image. 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. Создание приложений.

*Контрольные вопросы:*

1. Какие свойства компонентов определяют их положение на форме?
2. Как происходит изменение положения объектов на форме с помощью обработчика событий?
3. Как использовать генератор случайных чисел в обработчике событий для изменения положения объектов на форме?
4. Как создаются динамические объекты в Windows VCL Application – Delphi?
5. Как создается картинка динамического объекта и загружается в Image?
6. Для чего используется элемент Timer и как он работает?
7. Как создавать событие работы таймера в обработчике событий?
8. Что такое «прерывание» и как осуществляется обработка прерываний в Windows VCL Application – Delphi?
9. Как создать имитацию перемещения объекта по форме приложения с помощью клавиатуры?

10. Что такое анимация, что такое мультипликация? Какие существуют способы задания анимации и мультипликации?

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

12. Как подготовить картинки для мультипликации?

13. Как динамически загружать рисунки в элемент Image?

14. Как обеспечить очередность смены картинок в элементе Image?

15. Как задать динамику объектов мультипликации в Windows VCL Application – Delphi?

### 3.3. Активные и интерактивные формы проведения занятий

В качестве активных форм проведения занятий по дисциплине «Программирование на языках высокого уровня» 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.

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 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 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 должны 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 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 (то, что в данном труде относится непосредственно к изучаемой теме). Это важно в связи с тем, что к дополнительной литературе может быть отнесен широкий спектр текстов (учебных, научных, художественных, публицистических и т.д.), в которых исследуемый вопрос рассматривается либо частично, либо с какой-то одной точки зрения, порой нетрадиционной.

В своей совокупности изучение таких подходов существенно обогащает научный кругозор обучающихся. В данном контексте следует учесть, что дополнительную литературу целесообразно прорабатывать, во-первых, на базе уже освоенной основной литературы, и, во-вторых, изучать комплексно, всесторонне, не абсолютизируя чью-либо субъективную точку зрения.

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

Конспект – это краткое письменное изложение содержания источника, 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 информационно-правовых источников, что вполне достижимо при научной организации учебного труда.

Изучение дисциплины нужно начинать со знакомства с программой. По списку литературы требуется подобрать относящиеся к конкретной теме учебные материалы, дополнительные источники (книги, брошюры, журналы, Интернет-ресурсы и др.).

Среди учебной литературы, прежде всего, следует обратить внимание на учебники, а также на пособия, рекомендованные Министерством науки и высшего образования РФ или допущенные в качестве базовых. Это относится, в том числе и к учебно-методическим пособиям. Особо необходимо подчеркнуть, чтобы обучающиеся были осторожны при работе с интернет-источниками: иногда там встречаются откровенно ошибочные, недостоверные сведения. Следует использовать информацию только на знакомых, проверенных, солидных сайтах, сравнивая полученные результаты и используя здравый смысл.

При самостоятельном изучении основной рекомендованной литературы обучающимся необходимо обратить главное внимание на ключевые 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 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 обучающихся выделять в ней необходимый аспект изучаемой темы (то, что в данном труде относится непосредственно к изучаемой теме). Это важно в

связи с тем, что к дополнительной литературе может быть отнесен широкий спектр текстов (учебных, научных, художественных, публицистических и т. д.), в которых исследуемый вопрос рассматривается либо частично, либо с какой-то одной точки зрения, порой нетрадиционной.

Самостоятельная работа обучающихся будет эффективной и полезной, если она будет построена исходя из понимания обучающимися необходимости освоения информационных систем и технологий, приобретения навыков работы с документами при помощи офисных приложений общего назначения, обеспечения защиты информации, в том числе персональных данных.

## 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 разработки 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 языке Pascal используются: конъюнкция (and), дизъюнкция (or), отрицание (not). Они позволяют проверить сразу несколько условий за раз. Их используют для создания сложных условий, например в циклах.

**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 Delphi 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.

**Отладчик** - инструмент для поиска и устранения ошибок в разрабатываемых приложениях в Delphi. 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.

**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 Delphi, 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.

**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.

**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 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 зачету

1. Что представляет собой структура программ языка Pascal.
2. Как осуществляется описание глобальных переменных?
3. Какие основные типы используются в языке Pascal?
4. Как работает условный оператор «if»? Как работают полная и укороченная конструкция оператора «if»?
5. Что такое составные условия? Что такое составные условия со вложенными «if»?
6. Какие логические операторы используются в Pascal в составных условиях?
7. Что представляет собой оператор «case .. of»?
8. Что такое оператор повторения?
9. Как работает оператор повторений «for»?
10. Как работает циклический оператор «while»?
11. Как работает циклический оператор «repeat .. until»?
12. Что такое вложенные циклы и как они работают?
13. Что представляет собой «визуальное программирование»?
14. Что такое форма приложения?
15. Что такое инспектор свойств?
16. Что такое инспектор событий?
17. Что такое палитра компонентов?
18. Как создается интерфейс программного приложения?
19. Как изменять свойства компонентов с помощью обработчика событий?
20. Как происходит описание переменных в Windows VCL Application – Delphi?
21. Какие существуют функции конвертирования значений переменных одного типа в другой в Windows VCL Application – Delphi?
22. Какие свойства компонентов определяют их положение на форме?
23. Как использовать генератор случайных чисел в обработчике событий для изменения положения объектов на форме?
24. Как создаются динамические объекты в Windows VCL Application – Delphi?

25. Как создается картинка динамического объекта и загружается в Image?

26. Для чего используется элемент Timer и как он работает?

27. Как создавать событие работы таймера в обработчике событий?

28. Что такое «прерывание» и как осуществляется обработка прерываний в Windows VCL Application – Delphi?

29. Как создать имитацию перемещения объекта по форме приложения с помощью клавиатуры?

30. Что такое анимация, что такое мультипликация? Какие существуют способы задания анимации и мультипликации?

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

## 5.2. Тестовые задания

**1. Основной блок программы в Pascal расположен между:**

a) Begin … End;
b) Program … End.;
c) Program … Begin.

**2. Подраздел объявления глобальных переменных объявляется зарезервированным словом:**

a) Begin;
b) Var;
c) Array.

**3. Заголовок процедуры объявляется зарезервированным словом:**

a) Procedure;
b) Label;
c) Type.

**4. Оператор Uses предназначен для:**

a) объявления процедур и функций;
b) создания нового раздела программы;
c) подключения к программе системных модулей и библиотек.

**5. К целым типам данных относится:**

a) integer;
b) real;

c) char.

**6. К вещественным типам относится:**

a) integer;

b) real;

c) char.

**7. К строковым типам относятся:**

a) integer;

b) char;

c) string.

**8. Оператор присваивания в языке Pascal:**

a) := ;

b) = ;

c) <?> .

**9. Для вывода на экран значения переменной «x» служит процедура:**

a) Read(x);

b) Write(x);

c) Cin(x).

**10. Для ввода значений переменных с клавиатуры служит процедура:**

a) Read(x);

b) Write(x);

c) Cin(x).

**11. Структура условного оператора в языке Pascal имеет следующий вид:**

a) cin <условие> cout <оператор1> else <оператор2>;

b) then <условие> if <оператор1> else <оператор2>;

c) if <условие> then <оператор1> else <оператор2>.

**12. Конструкция укороченного составного условного оператора с конъюнкцией в языке Pascal имеет вид:**

a) Cin (x>y) && ((x>z) then … ;

b) If (x>y) and (x>z) then … ;

c) If (x>y) || (x>z) then … .

**13. В языке Pascal счетным оператором цикла является:**

a) repeat <операторы> until <условие>;

b) for <парам.цик.>:=<нач.знач.> to <кон.знач> do <оператор>;

c) while <условие> do <оператор>.

**14. В языке Pascal оператором цикла с постпроверкой условия является:**

a) repeat <операторы> until <условие>;

b) for <парам.цик.>:=<нач.знач.> to <кон.знач> do <оператор>;

c) while <условие> do <оператор>.

**15. В языке Pascal оператором цикла с предпроверкой условия является:**

a) repeat <операторы> until <условие>;

b) for <парам.цик.>:=<нач.знач.> to <кон.знач> do <оператор>;

c) while <условие> do <оператор>.

**16. В языке Pascal вложенные циклы – это:**

a) циклы, объединенные с помощью логических операторов and, or, xor;

b) циклы, размещенные внутри других циклов;

c) циклы, содержащие гиперссылки.

**17. В языке Pascal к строковым типам относятся:**

a) string;

b) char;

c) array.

**18. В языке Pascal к символьным типам относятся:**

a) string;

b) char;

c) array.

**19. В языке Pascal для описания массивов используется:**

a) string;

b) char;

c) array.

**20. В языке Pascal массив – это:**

a) тип данных, в котором хранится упорядоченный набор однотипных элементов;

b) тип данных, в котором хранятся упорядоченные наборы элементов разного типа;

c) тип данных, в котором хранятся случайные наборы элементов разного типа.

**21. В Delphi компонент представляет собой:**

a) функциональный элемент интерфейса, обладающий определенными свойствами;

b) набор свойств, событий и методов для решения какой-то задачи программирования;

c) аллюр для отображения информации о процессах, идущих в ходе выполнения программы.

**22. Свойство Name в большинстве компонентов означает:**

a) начало программы;

b) тело программы;

c) имя компонента.

**23. Изменить размеры формы можно в инспекторе объектов с помощью свойств:**

a) Height и Width;

b) Длина и Ширина;

c) Left и Wright.

**24. Изменить форму фигуры компонента Shape можно с помощью свойства:**

a) color;

b) shape;

c) length.

**25. Изменить цвет компонента Shape можно с помощью свойства:**

a) Properties – Events;

b) Enabled - True;

c) Brush – Color.

**26. Изменить место расположения компонента на форме можно с помощью свойств:**

a) Shape и Height;

b) Left и Top;

c) Align - None.

**27. Чтобы объект стал невидимый на форме нужно:**

a) свойство Visible сделать false;

b) свойство True сделать Unvisible;

c) удалить свойство Visible.

**28. Компонент Timer предназначен для:**

a) установки времени создания программы;

b) выполнения последовательности действий через заданный промежуток времени;

c) ускорение выполнения программы.

**29. Компонент Button – это:**

a) установочный ключ загрузки системы;

b) устройство прерывания;

c) кнопка.

**30. Чтобы ввести текс в компонент Edit1 нужно выполнить:**

a) Edit1.Text:=’Текст’;

b) Edit1.Text==’Текст’;

c) Text.Edit1.Pole=Текст.

**Ключ**
**к версии теста по дисциплине**
**«Программирование на языках высокого уровня»**

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

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

## Основная литература

1. Кононова, З. А. Программирование в Delphi : разработка приложений : учебное пособие : [16+] / З. А. Кононова, С. О. Алтухова ; Липецкий государственный педагогический университет им. П. П. Семенова-Тян-Шанского. – Липецк : Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского, 2017. – Часть 1. – 110 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=577073. – Библиогр. в кн. – ISBN 978-5-88526-908-7. – Текст : электронный.

2. Кононова, З. А. Программирование в Delphi : создание приложений : учебное пособие / З. А. Кононова, С. О. Алтухова ; Липецкий государственный педагогический университет им. П. П. Семенова-Тян-Шанского. – Липецк : Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского, 2020. – Часть 2. – 87 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=619369. – Библиогр. в кн. – ISBN 978-5-907168-97-8 (Ч. 2). - ISBN 978-5-88526-907-0. – Текст : электронный.

3. Кононова, З. А. Программирование в Delphi : создание приложений : учебное пособие ] / З. А. Кононова, С. О. Алтухова ; Липецкий государственный педагогический университет им. П. П. Семенова-Тян-Шанского. – Липецк : Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского, 2020. – Часть 3. – 79 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=619370 . – Библиогр. в кн. – ISBN 978-5-907335-42-4 (Ч. 3). - ISBN 978-5-88526-907-0. – Текст : электронный.

## Дополнительная литература

1. Воробьев, Г. А. Основы программирования на Python : учебно-методическое пособие / Г. А. Воробьев ; Липецкий государственный педагогический университет им. П. П. Семенова-Тян-Шанского. – Липецк : Липецкий государственный педагогический университет им. П.П. Семенова-

Тян-Шанского, 2022. – 89 с. : ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=700515. – Библиогр. в кн. – ISBN 978-5-907461-84-0. – Текст : электронный.

2. Кангин, В. В. Разработка SCADA-систем : учебное пособие / В. В. Кангин, М. В. Кангин, Д. Н. Ямолдинов. – Москва ; Вологда : Инфра-Инженерия, 2019. – 565 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=564221. – Библиогр.: с. 551. – ISBN 978-5-9729-0319-1. – Текст : электронный..

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, мультимедиа-проекторы, 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.