

Муниципальное бюджетное общеобразовательное учреждение  
«Основная общеобразовательная школа №269  
Закрытого административно-территориального образования  
Александровск Мурманской области»

«Рассмотрено»  
Школьным методическим  
объединением учителей  
протокол № 1  
от «30» августа 2023 г.  
Руководитель ШМО  
Рау С.М. /Рау С.М./

«Согласовано и одобрено»  
на заседании  
педагогического совета  
Протокол № 1  
от «30» августа 2023г.  
Киселёва А.В. /Киселёва А.В./

«Утверждаю»  
Директор МБОУ ООШ №269  
Граб В.П. /Граб В.П./  
Приказ № 210 о.д.  
от «31» августа 2023 г.

Дополнительная общеобразовательная общеразвивающая программа  
технической направленности  
«Нейро-РУ - Искусственный интеллект с Python»  
8-9 классы (14-16 лет)  
(возраст обучающихся)  
1 год  
(срок реализации)

Учитель информатики  
:  
Татаршвили Л.И.

ЗАО Александровск  
г. Снежногорск  
2023 год

## Оглавление

1. Пояснительная записка .....	3
Направленность программы.....	3
Уровень .....	3
Актуальность .....	3
Описание программы .....	4
Формы реализации.....	4
Аннотация, адресат программы .....	5
Цель программы.....	6
Задачи обучения, развития, воспитания: .....	6
Задачи обучения .....	6
Задачи воспитания:.....	6
Задачи развития .....	6
Формы и режим занятий.....	6
Планируемые результаты обучения .....	6
Предметные результаты .....	6
Личностные результаты: .....	7
Метапредметные результаты: .....	8
2. Содержание учебного плана по компетенциям .....	8
3. Учебный план .....	12
4. Календарно-тематическое планирование .....	14
5. Рабочая программа в таблице .....	16
Формы аттестации и оценочные материалы, включая примеры контрольных заданий .....	27
Организационно-педагогические условия .....	32
6. Перечень источников информационного сопровождения (учебная литература и др.).....	36

## **Дополнительная общеобразовательная общеразвивающая программа технической направленности «Нейро-РУ - Искусственный интеллект с Python»**

### **1. Пояснительная записка**

Документы, на основании которых разработана программа:

- Федеральным законом от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Постановлением Главного государственного санитарного врача РФ от 28.09.2020 №28 «Об утверждении санитарных правил СП 2.4.3648-20 (Санитарно-эпидемиологические требования к организациям воспитания, обучения, отдыха и оздоровления детей и молодежи);
- Постановлением Главного государственного санитарного врача Российской Федерации от 28.01.2021 №2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;
- Приказом Министерства просвещения Российской Федерации от 27.07.2022 №629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Письмом Министерства образования и науки России от 18.11.2015 №09-3242 «О направлении информации» вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»;
- Распоряжение Правительства Российской Федерации от 29.05.2015 года №996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года»;
- Распоряжение Правительства Российской Федерации от 31.03.2022 года № 678-р «Концепция развития дополнительного образования детей».

### **Направленность программы**

Программа направлена на получение знаний, умений и навыков в области создания нейронных сетей на Python, повышения мотивации на продолжение обучения по ИТ- специальностям, может выступать в качестве дополнительной базовой подготовки по математике и информатике в школе.

Содержание и структура программы организованы по принципу дифференциации в соответствии с начальным уровнем сложности и предусматривает использование и реализацию общедоступных и универсальных форм организации материала, минимальную сложность предлагаемого для освоения содержания программы, а также предполагает знание обучающимися базовой терминологии.

Дополнительная общеобразовательная программа носит практико-ориентированный характер, заключающийся в формировании практических навыков в области создания нейронных сетей на Python на уровне, соответствующем уровню образования и возрасту обучающихся, носит профориентационный характер.

50% трудоёмкости дополнительной общеобразовательной программы отведено практическим занятиям. Не менее 50% проходит очно (при непосредственном контакте обучающихся и педагогического работника).

### **Уровень:**

стартовый.

Срок освоения образовательной программы: 1 год.

Минимальное количество человек в одной группе: 10-15 человек.

Количество уроков: 68.

**Направленность программы:** современные языки программирования.

Категория обучающихся (возраст) по программе: учащиеся 8-11 классов общеобразовательных организаций.

### **Актуальность**

Психологически обоснованный и компетентный выбор будущей профессии для школьников 8-11 классов определяется возможностью реализовать учебно-профессиональную деятельность в период школьного обучения. В современных условиях, ускоренного внедрения

цифровых технологий в экономике и социальной сфере востребованными являются IT-специалисты. Уровень решаемых ими задач настолько высок, что выбор IT-профессии не может быть ситуативно-вынужденным. Успешность в ней будет определяться тем, насколько хорошо будущий специалист осведомлен о специфике деятельности и имеет ли некоторый практический опыт.

Предлагаемая дополнительная общеобразовательная программа «Нейро.PY - Искусственный интеллект с Python» отвечает цели осознанного компетентного выбора профессии. Это возможно за счет того, что содержание программы позволяет уяснить азы профессии будущего уже сейчас. Профессиональные компетенции, как результат освоения настоящей программы, позволят создавать и обучать нейронные сети.

Разработчик нейронных сетей (программист-разработчик нейронных сетей, конструктор нейронных сетей) - специалист, создающий искусственные нейронные сети под прикладные задачи в разных отраслях экономики, например, сфера финансов и экономической безопасности, логистики и грузовых перевозок, здравоохранение, сельское хозяйство и др. В рамках программы обучающиеся смогут освоить умения и навыки, которые в дальнейшем лягут в основу цифровых профессиональных компетенций в любой из указанных сфер.

Создание нейронных сетей является одним из интересных и востребованных направлений в IT-сфере. Возможность создавать искусственный интеллект волнует молодые умы и тем самым мотивирует их на качественное обучение и решение практических задач, заложенных в дополнительной общеобразовательной программе. Язык программирования Python является универсальным и подходит для создания IT-продуктов на разнообразных платформах, что существенно расширяет возможности обучающихся в создании собственных игр и приложений.

На примере создания собственной нейронной сети на Python обучающиеся познакомятся с базовыми принципами и алгоритмами их разработки и обучения, а также переобучения и оптимизации. Это ключевые аспекты реальных профессиональных действий разработчика нейронных сетей, которые составят базовый портфель компетенций, что позволит сделать осознанный выбор IT-профессии и создавать продукт нейронных сетей уже по окончании программы.

### **Описание программы**

Программа формирует первичные представления и универсальные умения и навыки создания нейронных сетей на Python, что соответствует начальному уровню освоения специальных знаний, являющимися основой для развития будущих профессиональных умений и навыков.

### **Формы реализации**

очная с применением дистанционных образовательных технологий, очная без применения дистанционных образовательных технологий (в том числе, с применением средств электронного обучения).

Общий объем программы 68 академических часа. Длительность и количество занятий - 1 раза в неделю по 2 академических часа.

Программа включает 4 модуля, каждый из которых завершается промежуточной аттестацией - выполнением теста. В каждом модуле в рамках самостоятельной работы предусмотрено решение практических задач. По мере освоения программы, обучающиеся

выполняют разнообразные упражнения по созданию игр, приложений с использованием нейронных сетей.

В результате обучающиеся смогут создать собственную нейронную сеть на Python.

Содержание модуля 1 программы включает общие вопросы программирования на Python и знакомство с нейронными сетями: классификация нейронных сетей; переменные, типы данных и функции; импорт библиотеки; отрисовка графических примитивов; метод списка и метод строк; нейросети OpenCV. Основная часть программы (модули 2-4) раскрывает принципы объектно-ориентированного программирования на Python (модуль 2); алгоритмы создания графических приложений и решение прикладных задач (модуль 3); особенности создания и обучения нейронных сетей (модуль 4). В рамках указанных модулей обучающиеся также познакомятся с использованием программных инструментов для создания нейронных сетей на Python.

Программа обеспечена учебно-методическими ресурсами для качественного освоения содержания учебного материала, что также обеспечивает результативность освоения модулей. Обучающимся обеспечен доступ к современному практикоориентированному содержанию электронных образовательных и информационных ресурсов. Предусмотрены материально-технические условия - программное обеспечение и оборудование.

#### **Аннотация, адресат программы**

Программа рекомендована для учащихся 8-11 классов с целью получения ими начальных навыков создания нейронных сетей на Python. Обучение не требует специальных умений и навыков в предметной области «Математика и информатика». Результаты обучения: знания об особенностях создания и функционирования нейронных сетей, о программировании на языке Python; универсальные умения и навыки, позволяющие: создавать собственные базы данных, работать с различными библиотеками (Tensorflow, SQLite3, Turtle, Math), создавать и группировать виджеты под интерфейс, отрисовывать и настраивать картинки, разрабатывать игры и простые оконные приложения минимальной сложности на Python, создавать, обучать и переобучать, а также оптимизировать собственную нейронную сеть. Результаты обучения востребованы при создании продуктов нейронных сетей, простых игр и оконных приложений, а также служат основой для развития предпрофессиональных умений и навыков.

Практикоориентированный характер обучения обеспечивается решением задач, связанных: с переменными и типами данных, с использованием циклов For и While, с использованием графических библиотек. Предусмотрено решение практических задач по созданию файлов с помощью Python и записи данных с помощью графических библиотек; по использованию объектно-ориентированного программирования; по созданию собственных модулей и библиотек; по созданию и подключению баз данных; по написанию интерфейса игры и подключению виджетов; по созданию поколений нейронной сети; по созданию сверточных нейронных сетей и решение задач на правдоподобие. В рамках практики обучающиеся смогут научиться работать с программными инструментами (интерпретатор Python, среда разработки PyCharm, библиотеки Tensorflow, SQLite3, Turtle, Math), с сайтом PyPI и нейронной сетью OpenCV. Итог - создание собственной нейронной сети на Python.

***Цель программы:***

сформировать у обучающихся 8-11 классов начальные умения по созданию нейронных сетей на Python, тем самым мотивировать их к развитию предпрофессиональных умений в области программирования, создания, обучения и переобучения, оптимизации нейронных сетей.

***Задачи обучения, развития, воспитания:******Задачи обучения:***

сформировать отношение к компьютерной технике, смартфону, планшету и другим современным техническим устройствам как инструменту для решения творческих задач и реализации полезных (общественно полезных) творческих IT-замыслов, реализации потребности в созидании;

сформировать мотивацию к изучению программных IT-инструментов для создания и реализации различных IT-проектов;

сформировать первичное целостное представление о программных разработках с помощью языка программирования Python;

сформировать первичное целостное представление о нейронных сетях и их функционировании включая обучение и переобучение, оптимизацию нейронных сетей;

сформировать первичное представление о целесообразности и значимости нейронных сетей для реализации прикладных задач в разных отраслях экономики; изучить основы разработки игр и простых оконных приложений;

сформировать основу для развития навыков работы с основными программными продуктами по созданию нейронных сетей.

***Задачи воспитания:***

сформировать нормы культурного проявления социальной активности и гражданской позиции в отношении тенденций развития отечественных IT-технологий в области создания нейронных сетей;

сформировать морально-этическую основу для развития профессиональной позиции в отношении использования технологий искусственного интеллекта и нейронных сетей в различных отраслях экономики;

сформировать основу для развития культуры профессионального общения в IT-сфере.

***Задачи развития:***

способствовать развитию образного мышления; развивать творческий подход к решению задач;

развивать профессиональный тезаурус и понимание принципов создания нейронных сетей.

***Формы и режим занятий***

Формы занятий - лекция, практическое занятие, самостоятельная работа. Режим занятий - 1 раз в неделю по 2 академических часа. Объем часов в неделю - 2 ак. часов.

***Планируемые результаты обучения:******Предметные результаты***

Сформированные у обучающихся начальные умения и навыки создания нейронных сетей на Python.

Обучающийся узнает:

характеристики существующих нейронных сетей и особенности работы в среде PyCharm;

принципы создания переменных и базовое понятие функции;

типы данных в Python, их отличия и особенности использования в разных ситуациях;  
базовые математические функции; предназначение ассоциативного массива в Python;  
особенности работы с ООП в Python;

базовые понятия обучения нейронной сети с помощью фреймворка Tensorflow.

особенности реализации на Python нейронной сети;

особенности создания базы данных на Python;

базовые принципы построения интерфейса приложений;

особенности создания приложений.

Обучающийся научится: устанавливать среду разработки PyCharm;

применять переменные и базовые простые функции в решении задач, использовать типы данных;

импортировать, скачивать и устанавливать библиотеки;

рисовать с помощью исполнителя, осуществлять построение графических фигур, отрисовывать фракталы с помощью графических примитивов; применять правила написания кода на Python; использовать библиотеки Turtle, Tkinter, PyQt, Pillow, SQLite3; скачивать и работать в нейронной сети OpenCV;

создавать собственные функции, просторно использовать однажды написанный код; реализовать хранение данных и представлять их графическом примитиве; работать с разными файлами и картинками (html, xml и т. д.); отличать процедурный стиль написания кода от ООП; создавать и обучать нейронные сети на начальном уровне; создавать собственную небольшую игру.

Обучающийся овладеет навыками:

установки интерпретатора Python и среды разработки PyCharm; решения задач с переменными и типами данных; создания собственного файла с помощью Python; создания базы данных с помощью библиотеки SQLite3 в Python; создания простого приложения с использованием Tkinter; создания игры в жанре гонки на Turtle; создания приложения рандомайзер; связывания приложения с базой данных;

создания приложений с использованием PyQt для реализации интерфейса OpenCV;

создания нескольких поколений нейронной сети;

поиска информации для обучения нейронной сети;

использования сверточных нейронных сетей;

использования метода переобучения нейронной сети;

построения правдоподобия нейронных сетей.

### ***Личностные результаты:***

ответственное отношение к учению, способность к саморазвитию и самообразованию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в профессиональных предпочтениях, с учетом устойчивых познавательных интересов и уважительного отношения к труду;

готовность и способность к самообразованию на протяжении всей жизни как условию успешной профессиональной и общественной деятельности.

### **Метапредметные результаты:**

умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в познавательной деятельности, развивать её мотивы и интересы;

умение самостоятельно планировать альтернативные пути достижения целей, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;

умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий и требований, корректировать их в соответствии с изменяющейся ситуацией;

умение самостоятельно определять цели деятельности и составлять её планы; контролировать и корректировать деятельность; использовать все возможные ресурсы для реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;

умение продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников, эффективно разрешать конфликты.

Предметные результаты: представление о нейронных сетях;

представление о возможностях языка программирования Python для создания нейронных сетей, простых игр и оконных приложений;

представление о программных инструментах, используемых для создания нейронных сетей;

первичные навыки и умения написания программного кода на языке программирования Python в соответствии с техническим заданием по созданию нейронной сети;

первичные навыки и умения создания баз данных, нейронных сетей, простых игр и оконных приложений.

### **2. Содержание учебного плана по компетенциям**

Наименование компетенции	Тип компетенции	Знания, соответствующие компетенции	Умения, соответствующие компетенции	Владение инструментами, соответствующие компетенции
ПК-1 Способен ориентироваться в переменных, типах данных и функциях	Профессиональная Модуль 1, 2	Знает, что такое переменная, данные, функция, тип данных. Знает принципы создания переменных и базовое понятие функции. Знает типы данных в Python, их отличия и особенности использования в разных ситуациях.	Умеет применять переменные и базовые простые функции в решении задач, использовать типы данных. Умеет создавать собственные функции, использовать однажды написанный код. Умеет реализовать хранение данных и представлять их графическом примитиве.	Владеет навыком решения задач с переменными и типами данных. Владеет навыком решения задач с помощью функции и графической библиотеки. Владеет навыком создания собственного файла с помощью Python. Владеет навыком записи данных с помощью графической библиотеки Turtle. Владеет навыком установки



				интерпретатора Python и среды разработки PyCharm.
ПК-2 Способе н ориентир оваться в возможн остях библиоте к для нейронн ых сетей.	Профессиона льная Модуль 1 Модуль 2 Модуль 3	Знает библиотеки и инструменты для реализации в Python. Знает особенности скачивания и установки библиотек (Turtle, Math) и модулей для проектов. Знает предназначение сайта PyPI. Знает особенности работы с библиотекой Turtle. Знает особенности библиотеки для создания интерфейсов на Python PyQt.	Умеет импортировать, скачивать и устанавливать библиотеки для построения таблиц в консоли, для отрисовки графических объектов. Умеет рисовать с помощью исполнителя, осуществлять построение графических фигур, отрисовывать фракталы с помощью графических примитивов. Умеет применять правила написания кода на Python. Умеет использовать циклы и библиотеки для отрисовки графики Turtle. Умеет использовать операторы для циклов. Умеет использовать метод списков. Умеет создавать множества исполнителей в библиотеке Turtle Умеет использовать библиотеку Tkinter, PyQt.	Владеет навыком импортирования библиотеки и использования сайта PyPI. Владеет начальным навыком использования графических библиотек при решении задач на списки. Владеет первичным навыком реализации игры на библиотеке Turtle. Владеет навыком использования библиотеки SQLite3 в Python. Владеет навыком использования Tkinter. PyQt
ПК-3 Способе н применя ть принцип ы объектно ориентир ованно го програм мирован ия (ООП) в Python.	Профессиона льная Модуль 2	Знает, что такое объектно-ориентированное программирование. Знает особенности работы с ООП в Python. Знает особенности создания конструктора и базовые принципы ООП наследования. Знает особенности реализации (использования)	Умеет создавать множества исполнителей в библиотеке Turtle. Умеет создавать собственные классы. Умеет отличать процедурный стиль написания кода от ООП. Умеет применять наследование и создавать иерархию классов. Умеет осуществлять разграничение кода и отображать с помощью библиотеки Turtle. Умеет использовать ООП.	Владеет навыком создания игры с помощью классов на библиотеке Turtle.

		принципов полиморфизма и инкапсуляции ООП в Python.		
ПК-4 Способен создавать модули, ориентироваться в базах данных	Профессиональная Модуль 2 Модуль 3	Знает, что такое модуль, база данных, приложение, виджет, кнопка и др. термины. Знает особенности создания модулей. Знает особенности	Умеет использовать готовые алгоритмы для библиотеки Turtle. Умеет использовать ранее написанные	Владеет навыком создания собственного модуля для хранения нескольких алгоритмов. Владеет навыком создания базы данных с помощью
Наименование компетенции	Тип компетенции	Знания, соответствующие компетенции	Умения, соответствующие компетенции	Владение инструментами, соответствующими компетенции
		работы в GitHub. Знает особенности создания базы данных на Python. Знает менеджмент файлов.	классы и функции в новых проектах. Умеет создавать базу данных с помощью библиотеки SQLite3 и размещать данные. Умеет подключать базу данных.	библиотеки SQLite3 в Python.
ПК-5 Способен ориентироваться в типах и видах нейронных сетей, работать ними	Профессиональная Модуль 1 Модуль 2 Модуль 4	Знает, что такое нейронная сеть и каких видов она бывает. Знает характеристики существующих нейронных сетей и особенности работы в среде PyCharm. Знает базовые понятия обучения нейронной сети с помощью фреймворка Tensorflow. Знает особенности функционирования нейронных сетей и умеет проводить аналогию с работой человеческого мозга. Знает особенности создания и обучения нейронной сети с помощью	Умеет устанавливать среду разработки PyCharm. Умеет создавать и обучать нейронные сети на начальном уровне. Умеет осуществлять поиск задач для нейронных сетей. Умеет на начальном уровне осуществлять обучение нейронных сетей. Умеет на начальном уровне использовать методы оптимизации нейронных сетей. Умеет создавать сверточные нейронные сети на начальном уровне. Умеет определять необходимость использования сверточных нейронных сетей. Умеет определять	Владеет навыком установки среды разработки PyCharm. Владеет навыком использования библиотеки Tensorflow. Владеет первичным навыком создания нескольких поколений нейронной сети. Владеет навыком поиска информации для обучения нейронной сети. Владеет методами оптимизации нейронной сети. Владеет первичным навыком использования сверточных нейронных сетей. Владеет первичным

		библиотеки Tensorflow. Знает в каких случаях следует использовать простые алгоритмы. Знает особенности оптимизации нейронных сетей.		навыком использования метода переобучения нейронной сети. Владеет первичным навыком построения правдоподобия нейронных сетей.
Наименование компетенции	Тип компетенции	Знания, соответствующие компетенции	Умения, соответствующие компетенции	Владение инструментами, соответствующие компетенции
		Знает особенности сверточных нейронных сетей и необходимость их использования. Знает особенности регуляризации и нормализации нейронных сетей. Знает методы максимального правдоподобия в нейронных сетях.	условия необходимости переобучения или создания новой нейронной сети. Умеет определять необходимость построения правдоподобия нейронной сети.	
ПК-6 Способен ориентироваться в особенностях создания интерфейса.	Профессиональная Модуль 3	Знает особенности построения графических приложений. Знает библиотеки и инструменты для реализации в Python. Знает особенности работы с библиотекой Turtle. Имеет представление о событии в графических примитивах. Знает особенности библиотеки для создания интерфейсов на Python PyQt. Знает базовые принципы построения приложений.	Умеет использовать библиотеку Tkinter. Умеет создавать события нажатия и их виды. Умеет создавать оконное приложение с помощью PyQt.	Владеет навыком создания простого приложения с использованием Tkinter. Владеет навыком создания игры в жанре гонки на Turtle. Владеет навыком создания приложения рандомайзер.

ПК-7 Способе н ориентир оваться в особенно стях создания приложе ний и использо вании виджето в.	Профессиона льная Модуль 3	Знает особенности создания приложений. Знает особенности добавления виджетов для взаимодействия пользователя с программой.	Умеет создавать приложения с виджетами в том числе для просмотра событий, сделанных в PyQt. Умеет использовать	Владеет первичным навыком создания игр. Владеет первичным навыком связывания приложения с базой данных. Владеет первичным навыком создания
Наимено вание компетен ции	Тип компетенции	Знания, соответствующие компетенции	Умения, соответствующие компетенции	Владение инструментами, соответствующи е компетенции
		Знает особенности визуальной настройки кнопок и группировки виджетов в нужном порядке. Знает возможные события в PyQt. Знает особенности работы с картинками и принципы настройки виджета под интерфейс. Знает особенности создания интерфейса для нейронной сети OpenCV.	виджеты в PyQt. Умеет настраивать картинки с помощью библиотеки Pellow. Умеет считывать информацию с базы данных и представлять её в выпадающем списке в приложении. Умеет создавать макет приложения для библиотеки OpenCV.	приложений с использованием PyQT для реализации интерфейса OpenCV.

### 3. Учебный план

	Всего	Теория	Практика	СР
<b>Модуль 1. «Знакомство с Python и нейросетями»</b>	<b>16</b>	<b>8</b>	<b>5</b>	<b>3</b>
<b>Раздел 1. Основы Python</b>				
Тема 1.1. Введение в нейросети. Классификация нейросетей	1	1	0	0
Тема 1.2. Переменные, типы данных, функции	2	1	1	0
Тема 1.3. Импортированием библиотек	2	1	1	0
<b>Раздел 2. Условные операторы и циклы</b>				
Тема 2.1. Отрисовка графических примитивов, условные операторы	2	1	0	1
Тема 2.2. Цикл FOR и цикл while	2	1	0	1
Тема 2.3. Списки, кортежи. Методы списков и методы строк	2	1	1	0
Тема 2.4. Словари, множества	2	1	1	-

Тема 2.5. Использование нейросети OpenCV	2	1	1	0
Промежуточная аттестация по модулю 1.	1			1
<b>Модуль 2. «Объектноориентированное программирование на Python»</b>	<b>22</b>	<b>9</b>	<b>9</b>	<b>4</b>
<b>Раздел 3. Функции и классы</b>				
Тема 3.1. Работа с функциями	2	1	1	-
Тема 3.2. Работа с файлами: создание, хранение	2	1	1	-
Тема 3.3. Знакомство с классами. Атрибуты класса	3	1	1	1
Тема 3.4. Конструктор. Что такое self? Наследование	3	1	1	1
<b>Раздел 4. Дочерние классы</b>				
Полиморфизм в Python	2	1	1	0
Тема 4.2. Инкапсуляция в Python	2	1	1	-
Тема 4.3. Создание собственных модулей	3	1	1	1
Тема 4.4. Работа с нейросетью Tensorflow	2	1	1	-
Тема 4.5. Работа с базами данных и использование SQLite3	2	1	1	-
Промежуточная аттестация по модулю 2.	1			1
<b>Модуль 3. «Создание графических приложений и решение прикладных задач»</b>	<b>18</b>	<b>9</b>	<b>2</b>	<b>7</b>
<b>Раздел 5. Основы графического интерфейса</b>				
Тема 5.1. Знакомство с созданием графического интерфейса	1	1	0	0
Тема 5.2. Работа с библиотекой Turtle. Обработка событий	2	1	0	1
Тема 5.3. Начало работы с PyQt	2	1	1	-
Тема 5.4. Создание собственного приложения и добавление виджетов на экран	2	1	0	1
Тема 5.5. Виджет, кнопка, метка и группировка виджетов.	2	1	1	0
<b>Раздел 6. Разработка графического интерфейса</b>				
Тема 6.1. Обработка событий в PyQt	2	1	0	1
Тема 6.2. Работа с картинками и настраивание каждого виджета	2	1	0	1
Тема 6.3. Виджет ввода текста, радиокнопки и выпадающий список	2	1	0	1
Тема 6.4. Создание интерфейса для нейросети OpenCV	2	1	0	1
Промежуточная аттестация по модулю 3	1			1
<b>Модуль 4. «Создание и обучение нейронных сетей»</b>	<b>12</b>	<b>7</b>	<b>4</b>	<b>1</b>
<b>Раздел 7. Основы нейронных сетей</b>				
Тема 7.1. Введение в нейронные сети и нейроны	2	1	1	0
Тема 7.2 Первая нейронная сеть	2	1	1	0
Тема 7.3. Как и где использовать нейронные сети	1	1	0	0
<b>Раздел 8. Методы работы с нейросетями</b>				
Тема 8.1. Методы оптимизации	2	1	1	0
Тема 8.2. Сверточные нейронные сети	2	1	1	0
Тема 8.3. Регуляризация и нормализация нейронных сетей	1	1	0	0
Тема 8.4. Методы нейронных сетей, максимальное подобие	1	1	0	0
Промежуточная аттестация по модулю 4	1			1
<b>Итого</b>	<b>68</b>	<b>33</b>	<b>20</b>	<b>15</b>

#### 4. Календарно-тематическое планирование

№	Тема и № модуля	Тема занятия	Кол-во занятий*	Кол-во часов	Дата
1	1 Модуль. Знакомство с Python и нейросетями	Раздел 1. Основы Python			
2		Тема 1.1. Введение в нейросети. Классификация нейросетей	1	1	
3		Тема 1.2. Переменные, типы данных, функции	1	2	
4		Тема 1.3. Импортирование библиотек	2	2	
5		Раздел 2. Условные операторы и циклы			
6		Тема 2.1. Отрисовка графических примитивов, условные операторы	2	2	
7		Тема 2.2. Цикл FOR и цикл while	2	2	
8		Тема 2.3. Списки, кортежи. Методы списков и методы строк	2	2	
9		Тема 2.4. Словари, множества	2	2	
10		Тема 2.5. Использование нейросети OpenCV	2	2	
10	Аттестация		1	1	
11	2 Модуль. Объектно-ориентированное программирование на Python	Раздел 3. Функции и классы			
12		Тема 3.1. Работа с функциями	2	2	
13		Тема 3.2. Работа с файлами: создание, хранение	2	2	
14		Тема 3.3. Знакомство с классами. Атрибуты класса	3	3	
15		Тема 3.4. Конструктор. Что такое self? Наследование	3	3	
16		Раздел 4. Дочерние классы			
17		Тема 4.1. Полиморфизм в Python	2	2	
18		Тема 4.2. Инкапсуляция в Python	2	2	
19		Тема 4.3. Создание собственных модулей	3	3	
20		Тема 4.4. Работа с нейросетью Tensorflow	2	2	
21	Тема 4.5. Работа с базами данных и использование SQLite3	2	2		
22	Аттестация		1	1	
23	3 Модуль. Создание графических приложений и решение прикладных задач	Раздел 5. Основы графического интерфейса			
24		Тема 5.1. Знакомство с созданием графического интерфейса	1	1	
25		Тема 5.2. Работа с библиотекой Turtle. Обработка событий	2	2	
26		Тема 5.3. Начало работы с PyQt	2	2	
27		Тема 5.4. Создание собственного приложения и добавление виджетов на экран	2	2	

28		Тема 5.5. Виджет, кнопка, метка и группировка виджетов.	2	2	
29		Раздел 6. Разработка графического интерфейса			
30		Тема 6.1. Обработка событий в PyQt	2	2	
31		Тема 6.2. Работа с картинками и настраивание каждого виджета	2	2	
32		Тема 6.3. Виджет ввода текста, радиокнопки и выпадающий список	2	2	
33		Тема 6.4. Создание интерфейса для нейросети OpenCV	2	2	
34	Аттестация		1	1	
35	4 Модуль. Создание и обучение нейронных сетей	Раздел 7. Основы нейронных сетей			
36		Тема 7.1. Введение в нейронные сети и нейроны	2	2	
37		Тема 7.2 Первая нейронная сеть	2	2	
38		Тема 7.3. Как и где использовать нейронные сети	1	1	
39		Раздел 8. Методы работы с нейросетями			
40		Тема 8.1. Методы оптимизации	2	2	
41		Тема 8.2. Сверточные нейронные сети	2	2	
42		Тема 8.3. Регуляризация и нормализация нейронных сетей	1	1	
43		Тема 8.4. Методы нейронных сетей, максимальное подобие	1	1	
44	Аттестация		1	1	

### 5. Рабочая программа в таблице

Порядковый номер модуля	Наименование темы	Лекции. Количество академических часов	Содержание лекций	Практические занятия. Количество академических часов	Содержание практических занятий	Самостоятельная работа. Количество академических часов	Содержание самостоятельной работы
1	<b>Раздел 1. Основы Python</b>						
1	Тема 1.1. Введение в нейросети. Классификация нейросетей	1	Краткий обзор существующих нейросетей. Работа в Pycharm.	1	Изучение характеристик существующих нейросетей в контексте их практического применения. Отработка практических навыков по установке среды разработки PyCharm.	1	Установка интерпретатора Python и среду разработки PyCharm.
1	Тема 1.2 Переменные, типы данных, функции	1	Создание переменных и знакомство с базовым понятием функции. Существующие типы данных в Python и их отличия. Использование типов данных в разных ситуациях.	2	Отработка практических навыков по применению переменных и базовых простых функций (print, input и т. д.) в решении задач, использование того или иного типа данных.	1	Решение задач, связанных с переменными и типами данных.
1	Тема 1.3 Импортирование библиотек	1	Скачивание и установка библиотек и модулей для проектов. Рассмотрение сайта PyPI, особенности быстрого поиска любой библиотеки для будущих проектов.	1	Отработка практических навыков по импортированию библиотек. Реализация действий по скачиванию и установке библиотеки	1	Самостоятельный импорт библиотеки и тренировка в использовании сайта PyPI.



			Использование команды <code>print</code> .		для построения таблиц в консоли.		
1	<b>Раздел 2. Условные операторы и циклы</b>						
1	Тема 2.1. Отрисовка графических примитивов, условные операторы	1	Использование библиотеки Turtle для представления данных в визуальном формате. Разбор условных операторов: понятие и особенности работы.	5	Реализация действий по импорту библиотеки для отрисовки графических объектов. Обучение рисованию с помощью исполнителя. Практическая реализация и реализация правил написания кода на Python. Отрисовка собственных объектов. Решение задач на построение графических фигур.	1	Решение задач на основе материалов лекции и практики.
1	Тема 2.2. Цикл FOR и цикл While	1	Особенности подключения библиотеки Math, базовые математические функции. Обучение созданию цикла.	5	Решение задач с использованием циклов и библиотеки для отрисовки графики turtle. Реализация действий по использованию операторов для циклов.	1	Создание собственного рисунка с использованием циклов For и While.
Порядковый номер модуля	Наименование темы	Лекции. Количество академических часов	Содержание лекций	Практические занятия. Количество академических часов	Содержание практических занятий	Самостоятельная работа. Количество академических часов	Содержание самостоятельной работы
			Понятие о циклах For и While. Взаимодействие вложенных конструкций.		Отрисовка фракталов с помощью графических примитивов.		

1	Тема 2.3 Списки, кортежи. Методы списков и методы строк	1	Создание списков и работа с ними. Хранение большого объема информации в одном месте. Рассмотрение методов списков и похожей структуры кортеж. Методы строк.	2	Решение задач на основе материала лекции. Реализация действий по использованию метода списков. Ответ на вопрос «Почему split() упрощает жизнь программиста?». Практическое применение понятия «синтаксический сахар» и его примеры.	1	Решение задач на списки с использованием графических библиотек.
1	Тема 2.4. Словари, множества	1	Понятие «ассоциативный массив»: предназначение и реализация в Python. Сравнение с другими коллекциями и возможности использования структур.	2	Решение практических задач на словари и множества. Ответ на вопрос «Почему множества приближены к математике?» Отработка практических навыков создания множества исполнителей в библиотеке Turtle. и собственной небольшой	1	Реализация игры на библиотеке Turtle.
Порядковый номер модуля	Наименование темы	Лекции. Количество академических часов	Содержание лекций	Практические занятия. Количество академических часов	Содержание практических занятий игры.	Самостоятельная работа. Количество академических часов	Содержание самостоятельной работы
1	Тема 2.5. Использование нейросети OpenCV	1	Нейросеть для компьютерного зрения и особенности её реализации на Python.	1	Реализация действий по скачиванию и нейросети OpenCV. Работа с нейросетью OpenCV. Решение задач.	1	Скачивание и применение OpenCV для решения задач.
2	<b>Раздел 3. Функции и классы</b>						

2	Тема 3.1 Работа с функциями	1	Особенности работы над функциями. Создание собственных функций с учетом правил синтаксиса.	3	Реализация действий по созданию собственных функций, повторного использования однажды написанного кода. Практическое применение видов функций, анонимных функций. Решение задач с помощью функции и графической библиотеки.		
2	Тема 3.2. Работа с файлами: создание, хранение	1	Особенности работы с файлами. Создание, обработка, хранение текстовой информации с помощью Python в файле формата txt.	3	Решение задач на хранение данных и представление их на графическом примитиве. Реализация действий по записи данных в .txt файл. Работа с разными файлами и картинками (html, xml и т.д.).		
Порядк овый номер модуля	Наименование темы	Лекции. Количество во академических часов	Содержание лекций	Практические занятия. Количество академических часов	Содержание практических занятий	Самостоятельная работа. Количество академических часов	Содержание самостоятельной работы
					Создание собственного файла с помощью Python, запись данных с помощью графической библиотеки Turtle.		

2	Тема 3.3. Знакомство с классами. Атрибуты класса	1	Начало работы с ООП. Понятие о классе и объекте. Ответ на вопрос «Как ООП построено на Python?». Базовые методы классов.	5	Практическое знакомство с особенностями создания множества исполнителей в библиотеке Turtle. Реализация действий по созданию собственных классов. Практическое задание на определение отличий процедурного стиля написания кода от ООП.	1	Создание собственной игры с помощью классов на библиотеке Turtle.
2	Тема 3.4. Конструктор. Что такое self? Наследование	1	Понятие о конструкторе и особенностях его создания. Ответ на вопрос «Почему конструктор - это важная часть класса?». Базовые принципы ООП наследования.	2	Решение задач с помощью ООП. Практическое применение наследования. Реализация действий по созданию иерархии классов. Ответ на вопросы «Что такое обращение на объект? Для чего это	1	Создание класса наследник от класса Turtle.
Порядковый номер модуля	Наименование темы	Лекции. Количество академических часов	Содержание лекций	Практические занятия. Количество академических часов	Содержание практических занятий	Самостоятельная работа. Количество академических часов	Содержание самостоятельной работы
					необходимо?».		
2	<b>Раздел 4. Дочерние классы</b>						

2	Тема 4.1. Полиморфизм в Python	1	Особенности смены алгоритма работы одного метода, когда он наследуется в другом классе. Ответы на вопросы «Для чего это необходимо, если можно создать новый класс? В каких ситуациях можно обойтись без этого?».	1	Реализация действий по использованию «магических методов» и ответ на вопрос «Почему там нет магии?». Применение на практике понятия об утиной типизации. Решение задач.	1	Решение задач с использованием «магических методов».
2	Тема 4.2. Инкапсуляция в Python	1	Использование инкапсуляции для разграничения прав и предоставления информации. Принцип инкапсуляции. Ответ на вопрос «Почему реализация принципа инкапсуляции в 90% случаев используется не так, как задумывалось?».	1	Отработка практических навыков по разграничению кода и отображению с помощью библиотеки Turtle. Решение задач с использованием ООП.	1	Создание собственной квест-игры с 3 основными фантастическими персонажами, отображение в Turtle.
2	Тема 4.3.	1	Создание	1	Реализация действий по	1	Создание собственного
	Создание собственных модулей		собственных модулей и библиотеки. GitHub. Менеджмент файлов.		созданию собственного модуля, в котором будут лежать готовые алгоритмы для библиотеки Turtle. Практическое обучение использованию ранее написанных классов и функций в новых проектах.		модуля, в котором будет храниться несколько алгоритмов.

2	Тема 4.4. Работа с нейросетью Tensorflow	1	Базовые понятия обучения собственной нейросети с помощью фреймворка Tensorflow.	1	Практическое изучение нейросети и применение знаний в практике её написания.	1	Скачивание и использование библиотеки Tensorflow
2	Тема 4.5 Работа с базами данных и использование SQLite3	1	Создание собственной базы данных на Python. Обучение написанию простых запросов в БД. Понятие о СУБД. Ответ на вопрос «Почему СУБД не всегда она нужна?».	2	Реализация действий по созданию собственной БД с помощью библиотеки SQLite3. Практическая работа по размещению данных в БД. Реализация действий по скачиванию SQLite3и подключению собственной БД.	1	Создание собственной БД с помощью библиотеки SQLite3 в Python.
3	<b>Раздел 5. Основы графического интерфейса</b>						
Порядк овый номер модуля	Наименование темы	Лекции. Количество во академическ их часов	Содержание лекций	Практическое занятия. Количество академическ их часов	Содержание практических занятий	Самостояте льная работа. Количество академическ их часов	Содержание самостоятельной работы
3	Тема 5.1 Знакомство с созданием графического интерфейса	1	Понятие о том, как строятся графические приложения и какие есть библиотеки и инструменты для реализации в Python.	1	Практическое применение библиотеки Tkinter. Создание простого оконного приложения.	1	Создание простого приложения с использованием Tkinter.

3	Тема 5.2 Работа с библиотекой Turtle. Обработка событий	1	Продолжение изучения библиотеки Turtle. Знакомство с событием в графических примитивах.	2	Создание игры и собственных событий нажатия. Реализация действий по закреплению за персонажем определенной кнопки на клавиатуре.	1	Создание игры в жанре гонки на Turtle.
3	Тема 5.3. Начало работы с PyQt	1	Изучение популярной библиотеки для создания интерфейсов на Python PyQt. Базовые принципы построения интерфейса приложений.	1	Реализация действий по созданию оконного приложения с помощью PyQt. Решение задач с использованием интерфейса.	1	Создание собственного приложения рандомайзер.
3	Тема 5.4. Создание собственного приложения и	1	Создание собственного приложения. Добавление	4	Реализация действий по созданию приложения с виджетами, виджет-кнопки и реализация	1	Создание приложения, которое парсит информацию с файла.
Порядк овый номер модул я	Наименование темы	Лекции. Количество во академичес ких часов	Содержание лекций	Практические занятия. Количество академичес ких часов	Содержание практических занятий	Самостояте льная работа. Количество академическ их часов	Содержание самостоятельной работы
	добавление виджетов на экран		виджетов для взаимодействия пользователя с программой.		события «нажатие». Реализация действий по созданию виджет- надписи для отображения информации. Практическое использование виджетов в PyQt		

3	Тема 5.5. Виджет, кнопка, метка и группировка виджетов	1	Создание кнопок и текста, их визуальная настройка. Группировка виджетов в нужном порядке.	2	Реализация действий по созданию игры-теста. Реализация практического обучения группировке элементов в нужном порядке. Реализация действий по написанию интерфейса для игры «Угадай число».	1	Создание игры-кликер.
3	<b>Раздел 6. Разработка графического интерфейса</b>						
3	Тема 6.1. Обработка событий в PyQt	1	Обзор возможных событий в PyQt (нажатие, удержание, перетаскивание и т. д.).	2	Реализация действий по созданию приложения, в котором можно будет рассматривать события, сделанные PyQt.	1	Реализация отключения интерфейса по нажатию на кнопку.
3	Тема 6.2. Работа с картинками и настраивание	1	Работа с картинками: добавление, изменение размера с	2	Реализация действий по добавлению в создаваемое приложение картинок, их	1	Создание игры «Угадай кино».
Порядк овый номер модуля	Наименование темы	Лекции. Количество во академическ их часов	Содержание лекций	Практическое занятия. Количество академическ их часов	Содержание практических занятий	Самостояте льная работа. Количество академическ их часов	Содержание самостоятельной работы
	каждого виджета		помощью сторонних библиотек, настройка виджета под интерфейс.		настройка с помощью библиотеки Pellow.		
3	Тема 6.3. Виджет ввода текста, радиокнопки и выпадающий список	1	Обзор виджетов для ввода текста и для выбора одного или нескольких указателей.	1	Реализация действий по считыванию информации с БД и её представление в выпадающем списке в создаваемом приложении.	1	Связывание создаваемого приложения с БД.



3	Тема 6.4. Создание интерфейса для нейросети OpenCV	1	Создание интерфейса (в индивидуальном порядке) для нейросети OpenCV. Создание слушателями собственных интерфейсов основываясь на материалах лекций.	2	Реализация действий по созданию макета приложения для библиотеки OpenCV.	1	Создание приложения с использованием PyQT для реализации интерфейса OpenCV.
4	<b>Раздел 7. Основы нейронных сетей</b>						
4	Тема 7.1. Введение в нейронные сети. Нейроны	1	Подробное изучение нейронной сети, проведение аналогии с работой человеческого мозга.	2	Реализация действий по поиску задач для нейронных сетей. Практическое использование первых нейросетей.	1	Самостоятельный подбор списка задач для нейронной сети.
Порядковый номер модуля	Наименование темы	Лекции. Количество академических часов	Содержание лекций	Практические занятия. Количество академических часов	Содержание практических занятий	Самостоятельная работа. Количество академических часов	Содержание самостоятельной работы
4	Тема 7.2. Первая нейронная сеть	1	Создание и обучение первой нейронной сети с помощью библиотеки Tensorflow и математических зависимостей.	4	Практическая отработка начала обучения первой нейросети, использование математики на начальном уровне.	1	Создание нескольких поколений собственной нейросети.

4	Тема 7.3. Как и где использовать нейронные сети	1	Рассмотрение случаев, в которых для анализа требуется работа сети и возможность использовать простые алгоритмы.	2	Отработка на практике навыка поиска задачи для нейросети, определения источников для поиска информации для её подробного обучения. Выявление и разбор на практике ограничений в обучении нейросети и «подводных камней» этого процесса.	1	Поиск информации для обучения собственной нейросети.
4	<b>Раздел 8. Методы работы с нейросетями</b>						
4	Тема 8.1. Методы оптимизации нейронной сети	2	Оптимизация нейросети для минимизации использования ресурсов.	2	Реализация действий по созданию методов оптимизации нейросети. Разбор практических случаев, в которых этого можно избежать.	1	Использование методов оптимизации.
4	Тема 8.2. Сверточные нейронные сети	2	Сверточные нейросети для экономии	2	Реализация действий по написанию сверточных нейросетей и	1	Повторение пройденного материала в части использования сверточных
Порядк овый номер модуля	Наименование темы	Лекции. Количество во академичес ких часов	Содержание лекций	Практические занятия. Количество академичес ких часов	Содержание практических занятий	Самостояте льная работа. Количество академическ их часов	Содержание самостоятельной работы

			вычислительных возможностей.		обоснование необходимости этого. Практическое знакомство с существующими сверточными нейросетями.		нейросетей.
4	Тема 8.3. Регуляризация и нормализация нейронных сетей	2	Проблемы переобучения нейросетей и слои нормализации.	4	Реализация действий по переобучению нейросетей. Практическое знакомство со случаями в которых лучше заново писать нейросеть. Ответ на вопрос «Сколько поколений нейросети могут переобучаться?».	1	Создание метода переобучения для своей нейросети.
4	Тема 8.4. Методы нейронных сетей, максимальное подобие.	1	Обзор методов максимального правдоподобия в нейронные сети.	2	Практическое знакомство с ценностями построения правдоподобия нейросети.	1	Решение задачи на построение правдоподобия нейросетей.

**Формы аттестации и оценочные материалы, включая примеры контрольных заданий**

Промежуточная аттестация

В таблице ниже представлены по каждому из модулей:

количество часов промежуточной аттестации;

формы контроля (промежуточной аттестации) с подробным описанием процедуры оценивания результатов обучения;

диагностические инструменты с примерами (контрольные задания, материалы, промежуточные тесты и задачи);

показатели и критерии оценивания;

шкала оценивания.

<b>Аттестация по итогам модуля. Количество ак. часов</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
----------------------------------------------------------	----------	----------	----------	----------

Аттестация по итогам модуля. Формы контроля	Ответы на тестовые вопросы.	Ответы на тестовые вопросы	Ответы на тестовые вопросы	Ответы на тестовые вопросы
<p>Аттестация по итогам модуля. Диагностические инструменты</p>	<p>Тестовые вопросы. Пример тестовых вопросов: 1) Определите значение переменной «a» после выполнения алгоритма: a = 4 b = 6 b = 8 + a * b a = b / 4 * a <b>a) 32</b> <b>b) 33</b> <b>c) 30</b> <b>d) 28</b> 2) Какие существуют типы переменных (выберите несколько вариантов): <b>a) Float;</b> <b>b) Str;</b> <b>c) Null;</b> <b>d) Int;</b> <b>e) Bool;</b> <b>f) Real 2.</b> 3) Переменная int – это? a) вещественная переменная; b) символьная строка; c) логическая</p>	<p>Тестовые вопросы. Пример тестовых вопросов: 1) Конструктор класса задается методом с именем: a) __new__; <b>b) __init__;</b> c) __construct__; g) new; e) init; f) имя конструктора совпадает с именем класса. 2) Если в классе определены два метода с одинаковыми именами и разными списками параметров, то: a) при выполнении скрипта будет сгенерирована ошибка; b) будет сгенерировано предупреждение, второе определение заменит первое; <b>c) не будет сгенерировано ни предупреждения, ни ошибки;</b> d) второе определение заменит первое; e) не будет сгенерировано ни предупреждения, ни ошибки; вызов того или иного метода будет зависеть от типа и количества указанных при вызове параметров; g) будет сгенерировано</p>	<p>Тестовые вопросы. Пример тестовых вопросов: 1) Название класса создания окна приложения: a) QApplication(); <b>b) QMainWindow();</b> c) setText(); d) QtWidgets(). 2) Сколько аргументов принимает setGeometry(): a) 5; <b>b) 4;</b> c) 6; d) 1; e) 2. 3) QPushButton – это: <b>a) графическая кнопка, которую пользователь может нажимать и отпускать;</b> <b>b) отрисовка надписей;</b> <b>c) создание окна приложения.</b> 4) button.clicked.connect() – это: <b>a) обработка события нажатия;</b> <b>b) обработка наведение на кнопку;</b> <b>c) подключение кнопки к приложению.</b> 5) Какой класс используется для создания диалоговых окон</p>	<p>Тестовые вопросы. Пример тестовых вопросов: 1) Как происходит обучение нейронной сети? a) эксперты настраивают нейронную сеть; b) сеть запускается на обучающем множестве, и незадействованные нейроны выкидываются; <b>c) сеть запускается на обучающем множестве, и подстраиваются весовые значения;</b> d) сеть запускается на обучающем множестве, и добавляются или убираются соединения между нейронами. 2) «Обучение с учителем» – это: a) использование знаний эксперта; <b>b) использование сравнения с идеальными ответами;</b> c) подстройка входных данных для получения нужных выходов; d) подстройка матрицы</p>

	<p>переменная;  <b>d) целая переменная.</b>  4) Переменная float – это?  <b>a) вещественная переменная;</b>  <b>b)</b> символьная строка;  <b>c)</b> логическая переменная;  <b>d)</b> целая переменная.  5) Переменная str – это?  a) вещественная переменная;  <b>b) символьная строка;</b>  c) логическая переменная;  d) целая переменная.  б) Переменная bool – это?  a) вещественная переменная;  b) символьная строка;  <b>c) логическая переменная;</b>  d) целая переменная.  7) Имена переменных не могут включать (выберите несколько вариантов):  a) русские буквы;  b) латинские буквы;  c) цифры;</p>	<p>предупреждение; вызов того или иного метода будет зависеть от типа и количества указанных при вызове параметров.  3) В языке программирования Python объектами являются:  a) экземпляры классов и переменные;  b) экземпляры классов, переменные и функции;  c) экземпляры классов, классы и переменные;  <b>g) все типы данных.</b>  4) Укажите результат выполнения скрипта:  class Foo:  def foo(self):  print ('1')  def __init__(self):  print ('2')  x = Foo()  a) 1;  <b>b) 2;</b>  c) 2;1;  g) пустая строка;  e) скрипт не будет выполнен, т.к. код содержит ошибки.  5) Укажите результат выполнения скрипта:  class Foo:  def foo(self):  print ('foo')  del self</p>	<p>в PyQt?  <b>a) QDialog</b>  <b>b) QMessageBox</b>  <b>c) QFileDialog</b>  <b>d) QMainWindow</b>  б) Какой метод класса QLabel используется для установки текста на метку?  <b>a) setText()</b>  <b>b) setLabel()</b>  <b>c) setTextLabel()</b>  <b>d) setCaption()</b>  7) Что такое сигнал в PyQt?  <b>a) Событие, которое генерируется объектом и сообщает о том, что произошло какое-то действие.</b>  <b>b)</b> Метод, который вызывается при изменении состояния объекта.  <b>c)</b> Функция, которая обрабатывает нажатие на кнопку.  <b>d)</b> Класс, который отвечает за отображение графических элементов.  4</p>	<p>весов для получения нужных ответов.  3) Синапсами называются:  <b>a) точки соединения нейронов, через которые передаются нейронные сигналы;</b>  <b>b)</b> «усики» нейронов, по которым проходят электрохимические сигналы;  <b>c)</b> тело нейрона, в котором происходит обработка электрохимического сигнала.  4) Какие типы задач можно решать с помощью обработки естественного языка?  a) Распознавание текста на изображениях, определение настроения человека по написанному тексту, распознавание речи.  b) Распознавание звуков, генерация текстовых описаний объектов на основе изображений, анализ синтаксиса текста.  <b>c) Классификация текстов, извлечение</b></p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p><b>d) пробелы;</b>  <b>e) скобки, знаки +</b>  <b>=!/? и др.</b></p>	<pre>def __del__(self): print ('del') obj = Foo() obj.foo() a) foo; b) del; c) del foo; <b>d) foo del;</b> e) скрипт не будет выполнен, т.к. код содержит ошибки. б) Укажите результат выполнения скрипта: class Foo: def __init__(self): print ('construct') del self obj = Foo() if obj: print ('exist')</pre> <p><b>a) construct; b) exist; c) construct exist.</b></p> <p>7) Укажите результат выполнения скрипта:</p> <pre>class Foo (object): obj=0 def __new__(cls,*dt,**mp): obj=1 def __init__(self): self.obj=2 o = Foo() print (o.obj)</pre> <p>a) 0;  b) 1;  c) 2;  <b>d) скрипт не будет выполнен,</b></p>		<p><b>информации, автоматический перевод текста.</b></p> <p>5) Что такое регуляризация в нейронных сетях?  a) Метод уменьшения потерь в нейронных сетях.  b) Метод добавления шума во входные данные нейронной сети.  <b>с) Метод уменьшения переобучения в нейронных сетях.</b></p> <p>б) Что такое нормализация в нейронных сетях?  <b>a) Метод масштабирования данных перед обучением нейронной сети.</b>  <b>b) Метод уменьшения количества параметров в нейронной сети.</b> c) Метод выбора оптимальной функции активации для нейронной сети.</p> <p>7) Какой тип слоев используется в сверточных нейронных сетях для извлечения признаков из изображений?  a) Полносвязные слои.  <b>b) Сверточные слои.</b>  c) Рекуррентные слои.</p>
--	---------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

		<b>т.к. код содержит ошибки.</b>		
Аттестация по итогам модуля. Описание процедуры оценивания Показатели и критерии оценивания	Верный ответ на тестовый вопрос. За каждый правильный ответ на вопрос теста начисляется 1 балл. В случае неправильного ответа баллы не начисляются.	Верный ответ на тестовый вопрос. За каждый правильный ответ на вопрос теста начисляется 1 балл. В случае неправильного ответа баллы не начисляются.	Верный ответ на тестовый вопрос. За каждый правильный ответ на вопрос теста начисляется 1 балл. В случае неправильного ответа баллы не начисляются	Верный ответ на тестовый вопрос. За каждый правильный ответ на вопрос теста начисляется 1 балл. В случае неправильного ответа баллы не начисляются.
Аттестация по итогам модуля. Шкала оценивания, нижнее значение				
Аттестация по итогам модуля. Шкала оценивания, верхнее значение	0	0	0	0
Аттестация по итогам модуля. Шкала оценивания, минимальный проходной балл для успешной сдачи	7	7	7	7

## Организационно-педагогические условия

### Методическое обеспечение: методы, формы и технологии, применяемые при реализации программы

Методы: метод проектов, модульное обучение, проблемное обучение.

Формы: лекции с использованием мультимедиа, практические занятия, самостоятельная работа.

Технологии: традиционное обучение, интерактивное обучение, проектная деятельность, онлайн-обучение.

Материально-техническое обеспечение

Наименование требуемого оборудования и программного обеспечения:

Очная форма с применением дистанционных образовательных технологий:

Персональный компьютер с процессором x86, x64

Минимальная диагональ монитора - не менее 39,6 см.

Организация рабочих мест обучающихся должна обеспечивать зрительную дистанцию до экрана не менее 50 см.

ПРОЦЕССОР: 2 GHz DUAL CORE (Core 2 Duo 2.4 GHz or Athlon X2 2.7 GHz) Вебкамера.

Микрофон, колонка (акустическая система) или наушники.

Клавиатура.

Мышь.

Скорость интернет-соединения: минимальные требования к пропускной способности канала связи - от 128 кбит/сек исходящего потока и от 128 кбит/сек входящего потока. При этом для комфортной работы необходимо хотя бы 1 Мбит/с.

ОС: Windows 7 (SP1+), Windows 10 and Windows 11.

Интерпретатор python <https://www.python.org/downloads/>

Среда разработки Pycharm

<https://www.jetbrains.com/pycharm/download/#section=windows>

Библиотека Pyqt <https://pypi.org/search/?q=Pyqt> Библиотека OpenCV

<https://github.com/opencv/opencv-python/releases>

Очная форма без применения дистанционных образовательных технологий:

Персональный компьютер с процессором x86, x64; архитектура с поддержкой инструкций SSE2; видеокарта: с поддержкой DX10, DX11, DX12.

Минимальная диагональ монитора - не менее 39,6 см.

Организация рабочих мест обучающихся должна обеспечивать зрительную дистанцию до экрана не менее 50 см.

ПРОЦЕССОР: 2 GHz DUAL CORE (Core 2 Duo 2.4 GHz or Athlon X2 2.7 GHz); видеокарта: с поддержкой DX10, DX11, DX12.

Микрофон, колонка (акустическая система) или наушники.

Клавиатура.

Мышь.

Мультимедийный проектор или интерактивная доска.



Скорость интернет-соединения: минимальные требования к пропускной способности канала связи - от 128 кбит/сек исходящего потока и от 128 кбит/сек входящего потока. При этом для комфортной работы необходимо хотя бы 1 Мбит/с.

ОС: Windows 7 (SP1+), Windows 10 and Windows 11.

Интерпретатор Python (<https://www.python.org/downloads>).

Среда разработки Pycharm

(<https://www.jetbrains.com/pycharm/download/#section=window>).

Библиотека Pyqt (<https://pypi.org/search/?q=Pyqt>).

Библиотека OpenCV (<https://github.com/opencv/opencv-python/releases>) Учебно-методические материалы

Перечень методических разработок и материалов для обучения:

*Методические разработки:*

Опорные конспекты лекций.

Презентационные материалы к теме.

Практические задания.

Тестовые вопросы для проверки знаний.

Задачи для самостоятельной работы.

*Материалы для обучения:*

Лекции.

Задачи для разбора на практических занятиях.

Тесты для промежуточной аттестации.

Примеры тестов:

Определите значение переменной «a» после выполнения алгоритма:  $a = 4$

$b = 6$

$b = 8 + a * b$   $a = b / 4 * a$

**32**

33

30

28

Какие существуют типы переменных (выберите несколько вариантов):

**Float;**

**Str;**

№;

**Int;**

**Bool;**

Real 2.

Переменная int - это?

вещественная переменная;

символьная строка;

логическая переменная;

целая переменная.

Переменная float - это?

вещественная переменная;

символьная строка;

логическая переменная;

целая переменная.

Переменная str - это?

вещественная переменная;

символьная строка;

логическая переменная;

целая переменная.

Переменная bool - это?

вещественная переменная;

символьная строка;

логическая переменная;

целая переменная.

Имена переменных не могут включать (выберите несколько вариантов):

русские буквы;

латинские буквы;

цифры;

пробелы;

скобки, знаки + =!/? и др.

Конструктор класса задается методом с именем:

) new\_;

) init ;

\_\_construct\_\_ ; g) new;

init;

имя конструктора совпадает с именем класса.

Если в классе определены два метода с одинаковыми именами и разными списками параметров, то:

при выполнении скрипта будет сгенерирована ошибка;

будет сгенерировано предупреждение, второе определение заменит первое;

не будет сгенерировано ни предупреждения, ни ошибки;

второе определение заменит первое;

не будет сгенерировано ни предупреждения, ни ошибки; вызов того или иного метода будет зависеть от типа и количества указанных при вызове параметров;

будет сгенерировано предупреждение; вызов того или иного метода будет зависеть от типа и количества указанных при вызове параметров.

В языке программирования Python объектами являются:

экземпляры классов и переменные;

экземпляры классов, переменные и функции;  
экземпляры классов, классы и переменные;  
все типы данных.

Укажите результат выполнения скрипта: class Foo:

```
def foo(self): print (T)
def  init  (self):
print ('2') x = Foo()
1;
b) 2;
2;1;
```

пустая строка;

скрипт не будет выполнен, т.к. код содержит ошибки.

Название класса создания окна приложения:

```
QApplication();
QMainWindow();
setText();
QtWidgets().
```

Сколько аргументов принимает setGeometry():

5;  
**4;**  
6;  
1;  
2.

QPushButton - это:

графическая кнопка, которую пользователь может нажимать и отпускать;

отрисовка надписей;

создание окна приложения.

button.clicked.connect() - это:

обработка события нажатия;

обработка наведение на кнопку;

подключение кнопки к приложению.

Как происходит обучение нейронной сети?

эксперты настраивают нейронную сеть;

сеть запускается на обучающем множестве, и неактивные нейроны выкидываются;

сеть запускается на обучающем множестве, и подстраиваются весовые значения;

сеть запускается на обучающем множестве, и добавляются или убираются соединения между нейронами.

«Обучение с учителем» - это:

использование знаний эксперта;

использование сравнения с идеальными ответами;

подстройка входных данных для получения нужных выходов;

подстройка матрицы весов для получения нужных ответов.

Синапсами называются:

точки соединения нейронов, через которые передаются нейронные сигналы;

«усики» нейронов, по которым проходят электрохимические сигналы;

тело нейрона, в котором происходит обработка электрохимического сигнала.

Искусственный нейрон:

является моделью биологического нейрона;

имитирует основные функции биологического нейрона;

по своей функциональности превосходит биологический нейрон.

SetWindowTitle:

устанавливает заголовок окна;

удаляет окно;

скрывает окно;

устанавливает кнопки на окно.

turtle.onclick - это:

нажатие на черепашку;

нажатие на холст экрана;

нажатие на выход.

Укажите результат выполнения скрипта: class Foo (object):

```
obj=0
```

```
def new (cls,*dt,**mp):
```

```
print ('1')
```

```
return object. new (cls,*dt,**mp).obj
```

```
def init (self):
```

```
print ('2') o = Foo() print (type (o))
```

```
1 2 <class ' main .Foo'>;
```

```
2 <class ' main .Foo'>;
```

```
1 <class 'int'>;
```

```
2 1 <class 'int'>.
```

Укажите результат выполнения скрипта: lst=[1]

```
class Foo (object): lst.append(2) print (lst) ob1 = Foo() ob2 = Foo()
```

```
[1, 2];
```

```
[1, 2] [1, 2, 2];
```

```
[1, 2] [1, 2, 2] [1, 2, 2, 2];
```

скрипт не будет выполнен, т. к. код содержит ошибки.

## **6. Перечень источников информационного сопровождения (учебная литература и др.)**

*Учебная литература:*

Бизли Д., Джонс Б. К. Python. Книга рецептов / пер. с англ. Б. В. Уварова. - М.: ДМК Пресс, 2019. - 648 с.

Доусон М. Програмируем на Python, 3-е издание - Пер. с англ. - СПб.: Питер, 2014. - 650 с.

Зингаро Д. Python без проблем: решаем реальные задачи и пишем полезный код. Издательство: Питер, 2023. - 336 с.

Лутц М. Изучаем Python, 3-е издание - Пер. с англ. - СПб.: Символ-Плюс, 2009. - 848 с.,

Николенко С. И., Кадури А. А., Архангельская Е. О. Глубокое обучение: погружение в мир нейронных сетей. Издательство: Питер, 2022. - 480 с.

Персиваль Г. Python. Разработка на основе тестирования / Г. Персиваль. - Москва: Издательство ДМК Пресс, 2018. - 622с.

Рашид, Тарик. Создаем нейронную сеть.: Пер. с англ. — СПб.: ООО «Диалектика», 2019. — 272 с.

Чернышев, С. А. Основы программирования на Python: учебное пособие для среднего профессионального образования / С. А. Чернышев. — Москва: Издательство Юрайт, 2023. — 286 с.

*Электронные информационные ресурсы:*

<https://www.tutorialspoint.com/python/> (сайт на английском языке для изучения Python);

<https://pythonworld.ru/samouchitel-python> ^айт по изучению основ программирования на Python);

<https://github.com/opencv/opencv-python/releases> (репозиторий с информацией по нейросети OpenCV);

<https://pypi.org/search/?q=Pyqt> (сайт с кратким описанием библиотек).

*Электронные образовательные ресурсы:* <https://www.python.org/downloads/> (интерпретатор Python);

<https://www.jetbrains.com/pycharm/download/#section=windows> (среда разработки Pycharm);

<https://pypi.org/search/?q=Pyqt> (библиотека PyQt); <https://github.com/opencv/opencv-python/releases> (нейросеть OpenCV)