

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ ЦЕНТР ДЕТСКОГО (ЮНОШЕСКОГО) ТЕХНИЧЕСКОГО
ТВОРЧЕСТВА «СТАРТ+» НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

ПРИНЯТА

Решением Педагогического совета
ГБУ ДОЦД (Ю)ТТ «Старт+»
Невского района Санкт-Петербурга
Протокол от _____ № _____

УТВЕРЖДЕНА

Приказом от _____ № _____
Директор ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт-Петербурга
_____ Подобаева О.Г.

**ДОПОЛНИТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«ПРОГРАММИРОВАНИЕ 2 СТУПЕНЬ»**

Срок освоения программы: 2 года
Возраст обучающихся: 13-17 лет

Разработчик: Тупицын Пётр Викторович,
педагог дополнительного образования
высшей квалификационной категории

2026 г.

Пояснительная записка

Направленность

Дополнительная общеобразовательная общеразвивающая программа **«Программирование 2 ступень»** (далее – Программа) является программой **технической направленности**.

Занятия по программе ориентированы на формирование у обучающихся навыков эффективного целеполагания, командной работы, практической деятельности по достижению целей и получению результата.

Актуальность Программы

Актуальность программы обусловлена перечнем приоритетов и перспектив научно-технологического развития Российской Федерации, перечисленных в Стратегии научно-технологического развития Российской Федерации, где создание систем обработки больших объемов данных, машинного обучения и искусственного интеллекта названо в числе приоритетных направлений на ближайшие 10–15 лет. Кроме того, в течение последних лет возрастает роль программных систем и продуктов во всех сферах жизни современного общества: от промышленности и медицины до транспорта, финансов и образования. Это приводит к необходимости подготовки специалистов, способных проектировать и реализовывать качественные и эффективные алгоритмы, что особенно важно в условиях динамичного развития цифровой экономики и высокой конкуренции в ИТ-отрасли.

Современные промышленные предприятия активно внедряют решения на основе искусственного интеллекта и нейросетевых технологий для оптимизации производственных процессов, прогнозирования и анализа, повышения качества продукции и снижения затрат. Таким образом, понимание принципов работы промышленных программных систем и нейросетей становится не только научно-познавательным, но и прикладным навыком, напрямую востребованным на рынке труда.

Для учащихся 13-17 лет обучение промышленному программированию и основам нейронных сетей играет важную роль не только в освоении современных технологий, но и в формировании критического и алгоритмического мышления, умений решать нестандартные задачи и обрабатывать большие массивы информации. Овладение навыками составления алгоритмов для обработки данных, представленных в различной форме, расширяет кругозор учащихся, способствует развитию цифровой грамотности и формирует представления о профессии разработчика программного обеспечения, выполняя задачу профессиональной ориентации и обеспечивая раннюю подготовку к инженерным и ИТ-специальностям.

Программа составлена в соответствии с нормативными документами в сфере образования:

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
2. Федеральный закон от 31.07.2020 № 304-ФЗ «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации» по вопросам воспитания обучающихся».
3. Стратегия развития воспитания в Российской Федерации на период до 2030 года (Стратегия разработана и реализуется в преемственности со Стратегией развития воспитания

в Российской Федерации на период до 2025 года, утвержденной Распоряжением Правительства Российской Федерации от 29 мая 2015 г. № 996-р)

4. Концепция развития дополнительного образования детей до 2030 года, утвержденная распоряжением Правительства Российской Федерации от 31.03.2022 № 678-р.

5. Распоряжение Правительства Российской Федерации от 1 июля 2025 г. № 1745-р «О внесении изменений в Концепцию развития дополнительного образования детей до 2030 года», утв. распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р, и утверждении плана мероприятий по реализации Концепции

6. Приказ Министерства просвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

7. Постановление Правительства РФ от 11 октября 2023 г. № 1678 «Об утверждении правил применения электронного обучения и дистанционных образовательных технологий при реализации образовательных программ с 1 сентября 2024 г. по 1 сентября 2029 г.»

8. Приказ Министерства просвещения Российской Федерации от 13.03.2019 № 114 «Об утверждении показателей, характеризующих общие критерии оценки качества условий осуществления образовательной деятельности организациями, осуществляющими образовательную деятельность по основным общеобразовательным программам, образовательным программам среднего профессионального образования, основным программам профессионального обучения, дополнительным общеобразовательным программам».

9. Приказ Министерства просвещения Российской Федерации от 03.09.2019 № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей».

10. Приказ Министерства труда и социальной защиты Российской Федерации от 22.09.2021 № 652н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых».

11. Приказ Министерства образования и науки Российской Федерации и Министерства просвещения Российской Федерации от 05.08.2020 № 882/391 «Об организации и осуществлении образовательной деятельности при сетевой форме реализации образовательных программ».

12. Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 N. 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».

13. Постановление 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 безвредности для человека факторов среды обитания».

14. Паспорт федерального проекта «Успех каждого ребенка» национального проекта «Образование» (утвержден президиумом Совета при Президенте Российской Федерации по стратегическому развитию и национальным проектам, протокол от 24.12.2018 № 16).

15. Письмо Минобрнауки России от 29.03.2016 № ВК-641/09 «О направлении методических рекомендаций».

16. Паспорт национального проекта «Молодежь и дети» (утвержден Министерством науки и высшего образования Российской Федерации в 2025 году).

Адресат Программы

Программа рассчитана на детей в возрасте 13-17 лет, имеющих хорошие навыки работы с персональным компьютером, интересующихся сферой информационных технологий, программирования и искусственного интеллекта.

Уровень освоения Программы – **общекультурный**

Объём и срок реализации Программы

Обучение по Программе запланировано на 288 учебных часов.

Срок реализации Программы: 2 года.

1-й год обучения – 144 часа;

2-й год обучения – 144 часа.

Режим занятий: 2 раза в неделю по 2 академических часа.

Отличительные особенности Программы

Программа спроектирована в модульном формате и включает в себя два модуля: «Промышленное программирование» и «Нейронные сети». За счёт выбора одного из модулей для обучения у учащегося есть возможность выбора индивидуальной образовательной траектории. Модуль «Нейронные сети» создан в рамках соглашения между ГБУ ДО ЦД(Ю)ТТ «Старт+» Невского района Санкт-Петербурга и Благотворительным фондом «Вклад в будущее».

Отличительной особенностью Программы является сочетание фундаментальной подготовки и практико-ориентированного обучения, обеспечивающее постепенный переход от освоения базовых понятий к разработке сложных приложений и интеллектуальных систем. Программа выстроена по принципу последовательного углубления знаний: сначала формируются основы алгоритмического и математического мышления, навыки работы с языком Python и обработки данных, затем — умения проектировать архитектуру приложений, применять современные библиотеки и инструменты разработки, организовывать безопасный и эффективный код.

Значимым элементом образовательного процесса является проектная деятельность: учащиеся регулярно создают прикладные приложения и модели искусственного интеллекта, отрабатывая навыки самостоятельного проектирования, командного взаимодействия, тестирования и презентации результатов. Особое внимание уделяется развитию критического и системного мышления, способности анализировать задачу, выбирать оптимальные методы её решения и оценивать эффективность полученного результата.

Таким образом, программа формирует у школьников комплекс компетенций — от владения промышленным программированием и основами работы с большими данными до проектирования нейросетевых решений, готовя их к профессиональной и исследовательской деятельности в сфере современных цифровых технологий.

Цель Программы

Формирование начальных знаний и навыков в области проектирования и разработки программных продуктов, машинного обучения и нейронных сетей, а также интереса к профессии разработчика программного обеспечения.

Задачи Программы

Обучающие

- Познакомить учащихся с основами программирования и синтаксисом языка Python.
- Развить навыки разработки алгоритмов обработки данных, включая использование базовых структур данных и пользовательских функций.
- Сформировать понимание парадигм программирования: процедурной, событийно-ориентированной и объектно-ориентированной; освоить создание собственных классов.
- Научить использовать библиотеки Python для численных расчетов, статистики, анализа и визуализации данных.
- Развить умение проектировать архитектуру программных решений, разрабатывать, тестировать и отлаживать приложения под конкретные задачи.
- Освоить технологии промышленного программирования: многопоточность, асинхронность, сетевые протоколы, API, клиент-серверные архитектуры и облачные сервисы.
- Научить проектировать и развертывать веб- и серверные приложения с учетом принципов безопасности и управления сетевыми взаимодействиями.
- Развить навыки построения простых моделей машинного обучения (регрессия, классификация) и понимания принципов работы нейронных сетей.
- Сформировать умение подготавливать и обрабатывать данные для обучения моделей искусственного интеллекта.
- Развить навыки проектирования и анализа нейросетей различной сложности, выбора архитектур и настройки гиперпараметров.
- Научить оценивать качество работы моделей, оптимизировать их и интегрировать в прикладные решения и сервисы.
- Сформировать системное и критическое мышление при проектировании программных продуктов и нейросетевых решений, а также готовность к командной и исследовательской деятельности.

Развивающие

- Расширить представления о сферах применения Python в промышленной и прикладной разработке.
- Развить навыки системного мышления при проектировании архитектуры программ.
- Сформировать способность к целеполаганию и логическому подходу к решению комплексных задач.
- Развить умение работать с реальными инструментами разработки, контролировать качество кода.
- Развить навыки командной работы при реализации проектов, распределении ролей и задач.

Воспитательные

- Воспитать целеустремленность, усидчивость и дисциплину при реализации сложных проектов.
- Воспитать уважение к процессу разработки и результатам труда.

- Сформировать уверенность в собственных силах при решении нестандартных и комплексных задач.
- Сформировать профессиональное отношение к труду и понимание ценности качественной работы.

Планируемые результаты освоения Программы

У учащихся:

Предметные:

- Сформированы базовые знания по программированию и владение синтаксисом языка Python.
- Развиты навыки разработки алгоритмов обработки данных с использованием структур данных и пользовательских функций.
- Сформировано понимание процедурной, событийно-ориентированной и объектно-ориентированной парадигм программирования, освоено создание собственных классов.
- Развиты умения работать с библиотеками Python для численных расчетов, статистической обработки, анализа и визуализации данных.
- Сформированы навыки проектирования архитектуры программных решений, разработки, тестирования и отладки приложений.
- Освоены технологии промышленного программирования: многопоточность, асинхронность, сетевые протоколы, API, клиент-серверные архитектуры и облачные сервисы.
- Сформированы умения разрабатывать и развертывать веб- и серверные приложения с учетом принципов безопасности и управления сетевыми взаимодействиями.
- Развиты навыки построения и анализа простых моделей машинного обучения, понимание принципов работы нейронных сетей.
- Сформированы умения подготовки и обработки данных для обучения моделей искусственного интеллекта.
- Развиты навыки проектирования и анализа нейросетей различной сложности, выбора архитектур и настройки гиперпараметров.
- Сформированы умения оценки качества моделей, их оптимизации и интеграции в прикладные решения и сервисы.
- Сформировано системное и критическое мышление при проектировании программных продуктов и нейросетевых решений, готовность к командной и исследовательской деятельности.

Метапредметные:

- Расширены представления о сферах применения Python в промышленной и прикладной разработке.
- Развиты навыки системного мышления при проектировании архитектуры программ.
- Сформированы умения целеполагания и логического подхода к решению комплексных задач.
- Развиты навыки работы с инструментами разработки и контроля качества кода.
- Развиты навыки командной работы при реализации проектов и распределении ролей и задач.

Личностные:

- Воспитана целеустремленность, усидчивость и дисциплина при реализации сложных проектов.
- Воспитано уважение к процессу разработки и результатам труда.
- Сформирована уверенность в собственных силах при решении нестандартных и комплексных задач.
- Сформировано профессиональное отношение к труду и понимание ценности качественной работы.

Организационно-педагогические условия реализации Программы

Язык реализации

Образовательная деятельность осуществляется на государственном языке Российской Федерации – русском языке.

Форма обучения

Очная, с возможностью использования дистанционных образовательных технологий.

Особенности организации образовательного процесса

В рамках модуля «Промышленное программирование» обучение направлено на освоение языка Python как инструмента разработки пользовательских приложений. Учащиеся последовательно переходят от изучения базовых конструкций и структур данных к созданию полноценных программных продуктов, включая клиент-серверные приложения и работу с базами данных. Особое внимание уделяется качеству и безопасности кода, применению современных библиотек, инструментов тестирования и деплоя. Важной частью является проектная деятельность и командная работа, позволяющая отрабатывать навыки проектирования архитектуры, взаимодействия компонентов и презентации готовых решений.

В рамках модуля «Нейронные сети» акцент сделан на формировании фундаментальной базы в области обработки данных, математических методов и алгоритмического мышления с последующим углублением в построение и анализ сложных нейросетевых моделей. Учащиеся знакомятся с библиотеками Python для анализа и визуализации данных, получают опыт работы с задачами компьютерного зрения, обработки текста и генерации информации. Проектная деятельность позволяет закрепить навыки подготовки и обработки данных, проектирования архитектуры сетей, оценки эффективности моделей и их применения в прикладных задачах, что формирует готовность к исследовательской и профессиональной деятельности в области искусственного интеллекта.

Программа направлена на развитие навыков, нужных для успешной реализации собственных идей. Программа способствует многостороннему личностному развитию ребенка и побуждает получать новые знания, создавать новое. Программа учитывает психологические, возрастные и индивидуальные особенности детей, нуждающихся в развитии эмоционально-волевых качеств.

С целью повышения устойчивости образовательной учебной траектории, вовлеченности учащихся, развития самостоятельности и формирование культуры работы над ошибками в рамках освоения дополнительной общеразвивающей программы «Программирование 2 ступень» в структуру обучения внедрена Виртуальная лаборатория

программирования. Виртуальная лаборатория программирования является основным практическим инструментом освоения ДОП «Программирование 2 ступень».

Условия набора и формирования групп

В объединение на добровольной основе принимаются учащиеся в возрасте 13-17 лет, которые уже получили знания и навыки работы с ПК, не имеющие медицинских противопоказаний. Зачисления в группу производится по заявлению родителей учащихся.

Собеседование и анализ первых выполненных работ позволяет выявить исходный уровень подготовки и индивидуальные особенности каждого учащегося. На основе полученных данных определяются направление и формы индивидуальной работы с учащимися.

Первый год обучения - группа в количестве не менее 15 человек.

Второй год обучения - группа в количестве не менее 12 человек.

Формы проведения занятий

Для реализации Программы используются несколько форм занятий:

- беседа;
- объяснение;
- обсуждение;
- обобщение;
- работа за компьютером;
- практические работы;
- выполнение самостоятельных заданий;
- работа над проектом;
- творческие отчеты;
- мини-игры;
- мини-конкурсы.

Выбор представленных форм обуславливается, в первую очередь, ориентированностью Программы на развитие гибких навыков и взаимодействием учащихся между собой. Применяемые в рамках данной Программы формы занятий носят развивающий характер и направлены на формирование опыта учащихся. Все формы направлены на выполнение заявленных задач и обоснованы спецификой Программы.

Формы организации деятельности учащихся на занятиях

В соответствии с темами Программы используются преимущественно следующие формы:

- групповая;
- индивидуальная;
- индивидуально-групповая;
- фронтальная.

Фронтальная – взаимодействие педагога и всех детей объединения осуществляется одновременно, применяется преимущественно при изучении учащимися новых тем, обсуждении сюжета, алгоритма действий на занятии.

При групповой работе дети распределяются по подгруппам (или парам) в зависимости от уровня подготовки, возраста. Особое внимание оказывается детям, участвующим в различные соревнования за команду.

Индивидуально-групповая – используется при акценте на теоретические занятия в совокупности с практическими.

При индивидуальной работе дети работают по одиночке над своими проектами.

Условия реализации программы в условиях вынужденного временного перехода в дистанционный режим

Согласно Положению ГБУ ДО ЦД(Ю)ТТ «Старт+» Невского района Санкт-Петербурга «Об использовании дистанционных образовательных технологий и электронных образовательных ресурсов при реализации дополнительных общеобразовательных общеразвивающих программ» по решению внепланового педагогического совета учреждения может быть принято решение о внеплановом временном переходе на дистанционный режим в связи с особыми обстоятельствами, например с эпидемиологической обстановкой.

В период подготовки к переходу на дистанционное обучение проводится мониторинг материально-технического и программного обеспечения учащихся и уровня их информационно-коммуникационной грамотности. Затем учащиеся (их родители или законные представители) извещаются о переходе на дистанционный режим обучения.

Если темы из календарно-тематического планирования адаптировать под дистанционный режим затруднительно, то составляется корректировка программы (в соответствии с Приложением 3 к Положению «Об использовании дистанционных образовательных технологий и электронных образовательных ресурсов...»), в которой при необходимости:

- указываются темы, которые добавляются в учебный план, или происходит перераспределение часов между разделами или темами,
- производится изменение содержания,
- корректируется календарно-тематическое планирование (например, на период дистанционного обучения переносятся темы, ориентированные на освоение теории),
- прописывается режим оказания педагогом консультационной помощи учащимся, при выполнении заданий,
- описывается характер дистанционного взаимодействия и конкретизируется необходимое материально-техническое и программное обеспечение, а также информационно коммуникационные умения, необходимые для дистанционного взаимодействия.

Корректировка утверждается директором ГБУ ДО ЦД(Ю)ТТ «Старт+» Невского района Санкт-Петербурга и предлагается для ознакомления учащимся и их родителям (законным представителям), которые подтверждают свое согласие на занятие по скорректированной на время дистанционного режима программе.

Если темы, предусмотренные на этот период возможно реализовать дистанционно, лишь изменив форму предоставления заданий и формат взаимодействия, то корректировка не составляется, а темы Программы реализуются в дистанционном режиме с даты его введения.

При этом задания для выполнения учащимися предоставляются средствами официальной группы ВКонтакте не позднее даты занятия по расписанию. Срок выполнения по умолчанию (если иное не оговорено в задании) устанавливается до времени и даты следующего ближайшего занятия. Консультативная поддержка учащимся (их родителей и законных представителей) оказывается по телефону, через электронную почту или группу

ВКонтакте в день занятия по расписанию в течение 2 часов со времени начала занятия по расписанию.

Для выполнения заданий учащимся потребуется компьютер (ноутбук), имеющий выход в Интернет, платформа Сферум а также бумага и пишущие средства. Они должны иметь адрес электронной почты, аккаунт ВКонтакте, аккаунт на платформе Сферум и уметь ими пользоваться. Наличие у учащихся должного материально-технического и программного обеспечения и их умение этим пользоваться определяется в период подготовки к переходу на дистанционное обучение. Выполненные задания учащиеся высылают в виде текстовых файлов, картинок/скриншотов, файлов программы и иных файлов (в соответствии с характером задания), направляемых через группу ВКонтакте. Если некоторые учащиеся не имеют должного обеспечения и не владеют информационно-коммуникационными технологиями, то для них возможна выдача индивидуальных заданий иного характера.

Трудоемкость дистанционного задания в часах в этом случае приравнивается к количеству часов, отведенных на эту тему в календарно-тематическом планировании. Если на период временного перехода на дистанционный режим приходятся контрольные или итоговые занятия, то они проводятся также в дистанционном режиме

Материально-техническое оснащение

Материалы и инструменты, необходимые для работы:

1. Бумага А4, тетради, блокноты.
2. Персональный компьютер (15 единиц);
3. Проектор;
4. Маркерная доска;
5. Маркеры разных цветов;

Кадровое обеспечение: педагог с соответствующим профилем объединения образованием и опытом работы, имеющий компетенции данного направления деятельности.

.

Учебный план

Модуль «Промышленное программирование»

1 год обучения

№	Название раздела	Количество часов			Форма/способ контроля
		Теория	Практика	Всего	
1.	Введение в программирование	6	12	18	Фронтальная/опрос
2.	Структура пользовательского приложения. Часть 1	6	8	14	Индивидуальная/наблюдение
3.	Контейнеры данных	8	16	24	Индивидуальная/практикум
4.	Структура пользовательского приложения. Часть 2	8	14	22	Комбинированная/практикум
5.	Таблицы и СУБД	10	18	28	Индивидуальная/практикум
6.	Объектно-ориентированное программирование	9	25	34	Комбинированная/практикум
7.	Итоговое занятие	1	3	4	Индивидуальная/анализ
ИТОГО		48	96	144	

Модуль «Промышленное программирование»

2 год обучения

№	Название раздела	Количество часов			Форма/способ контроля
		Теория	Практика	Всего	
1.	Углубленные технологии программирования. Облачные вычисления	14	28	42	Фронтальная/опрос
2.	Сетевое администрирование для программистов	5	9	14	Индивидуальная/наблюдение
3.	Основы фронтенд-разработки	14	28	42	Индивидуальная/практикум
4.	Основы бэкенд-разработки	14	28	42	Комбинированная/практикум
5.	Итоговое занятие	1	3	4	Индивидуальная/анализ
ИТОГО		48	96	144	

**Модуль «Нейронные сети»
1 год обучения**

№	Название раздела	Количество часов			Форма/способ контроля
		Теория	Практика	Всего	
1.	Введение в технологию искусственного интеллекта и основы программирования на python	4	10	14	Фронтальная/опрос
2.	Создание алгоритмов искусственного интеллекта на python	12	48	60	Индивидуальная/наблюдение
3.	Машинное обучение как методология ИИ	5	17	22	Индивидуальная/практикум
4.	Нейронные сети и компьютерное зрение	2	10	12	Комбинированная/практикум
5.	Продвинутые алгоритмы машинного обучения	7	25	32	Индивидуальная/практикум
6.	Итоговое занятие (защита проекта)	2	2	4	Индивидуальная/анализ
ИТОГО		32	112	144	

**Модуль «Нейронные сети»
2 год обучения**

№	Название раздела	Количество часов			Форма/способ контроля
		Теория	Практика	Всего	
1.	Библиотека Keras	6	14	20	Фронтальная/опрос
2.	Инструменты и библиотеки глубокого обучения	7	15	22	Индивидуальная/наблюдение
3.	Сетевое администрирование для программистов	4	10	14	Индивидуальная/практикум
4.	Инфраструктура и развертывание нейронных сетей	8	14	22	Комбинированная/практикум
5.	Основы проектирования нейронных сетей	6	10	16	Индивидуальная/практикум
6.	Рекуррентные сети и последовательности	2	6	8	Комбинированная/практикум

7.	Архитектуры генеративных сетей	8	16	24	Индивидуальная/практикум
8.	Введение в архитектуру трансформеров	4	10	14	Комбинированная/практикум
9.	Итоговое занятие	1	3	4	Индивидуальная /анализ
ИТОГО		46	98	144	

Методические материалы

В процессе занятий чаще используются методы самостоятельной работы. Самостоятельная работа не только закрепляет практические умения и навыки, но и развивает творческую активность. Одним из видов самостоятельной работы является поиск материалов и исследование по заданным темам, проектные работы.

Объяснительно-иллюстративный метод (показ презентаций, показ видеоматериалов, демонстрация образцов);

- Наглядный и частично-поисковый метод обучения (внедрение улучшений в проектах, выбор оптимального варианта алгоритма);
- Исследовательский метод, метод проектов (усовершенствовать модель-прототип, предложить свою модификацию или новую конструкцию)

Перечень дидактических материалов, используемых в процессе обучения

- Демонстрационные схемы;
- Шаблоны;
- Дидактические материалы с поясняющими рисунками, планом выполнения заданий
- и т.п.;
- Инструкции, описания;
- Таблицы с рекомендациями по выполнению заданий.

Итоги реализации Программы и достижения учащихся, осваивающих Программу представляются в следующих формах:

- презентации;
- показы индивидуальных разработанных учащимися проектов, моделей, полученных объектов и пр.;
- участие в конкурсах и мероприятиях Центра технического творчества «Старт+».

Оценочные материалы

Контроль степени освоения учащимися Программы осуществляется посредством организации следующих видов контроля:

Вид контроля	Формы контроля	Срок контроля
Входной	Опрос	Сентябрь 1 года обучения
Промежуточный	Изучение результатов работы учащихся	• Декабрь-январь 1 года обучения • Май 1 года обучения

		Декабрь-январь 2 года обучения
Текущий	Наблюдение	В течение учебного года
Итоговый	Изучение результатов работы учащихся	Май 2 года обучения

Результативность освоения Программы демонстрируется кодом программ, написанных участниками при решении задач и разработке проектов.

Входной контроль проводится в сентябре с целью выявления у учащихся уровня подготовки в области разработки алгоритмов, работы с компьютером, а так же с целью выявления уровня развития гибких навыков у учащихся.

Входная диагностическая работа выполняется всеми учащимися в устной форме – опрос, педагог заполняет **Бланк входной диагностики**.

Критерии входного контроля:

- Высокий уровень – 10 баллов
- Средний уровень – 5 баллов
- Допустимый уровень – 1 балл

Текущий контроль осуществляется наблюдением на занятиях в течение всего учебного года.

Промежуточный контроль проводится в середине каждого учебного года (в течение декабря-января) согласно календарно-тематическому плану, а также в конце 1 года обучения (май).

Контроль осуществляется изучением готовых работ учащихся и заносится в **ведомость контроля освоения материала**, в последующем в **Бланк промежуточной диагностики**.

Итоговый контроль проводится в конце 2 года обучения (май) согласно календарно-тематическому плану.

Контроль осуществляется изучением готовых работ учащихся и заносится в **ведомость контроля освоения материала**, в последующем в **Бланк итоговой диагностики**.

Модуль «Промышленное программирование»

Для педагогов

- Кнут, Д. Искусство программирования. Том 1–4. — 3-е изд. — М. : Вильямс, 2019. (Базовый фундамент по алгоритмам и структурам данных, необходимый для методической глубины преподавания, классическая монография по программированию)
- Лутц, М. Изучаем Python. — 5-е изд. — СПб. : Питер, 2021. — 1648 с. (Для системного объяснения языка на разных уровнях сложности)
- Марк Лутц, Д. Аслани. Программирование на Python. — М. : ДМК Пресс, 2020. — 752 с. (Хорошее пособие по современным практикам Python для реальных приложений)
- Гамма, Э., Хелм, Р., Джонсон, Р., Влиссидес, Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб. : Питер, 2021. — 416 с. (Для методического обеспечения тем по архитектуре и шаблонам проектирования)
- Тененбаум, Э. Системное программирование и операционные системы. — 4-е изд. — М. : Вильямс, 2018. — 1056 с. (Для углубления тем многопоточности, взаимодействия с ОС, сетевого программирования)

Для учащихся

- Б. Хардт, Т. Хелмес. Совершенный Python — СПб. : Питер, 2020. — 512 с. (Понятное и современное введение для углубленного освоения Python)
- Шоу, З. Изучаем Python. — 4-е изд. — СПб. : Питер, 2020. — 352 с. (Для начальной подготовки — практический курс с заданиями)
- Маккинни, У. Python и анализ данных. — 3-е изд. — СПб. : Питер, 2023. — 672 с. (Практическое освоение библиотек NumPy, Pandas)
- Дауни, А. Как мыслит программист. — СПб. : Питер, 2019. — 304 с. (Развитие алгоритмического мышления и грамотного стиля программирования)
- Стерлинг, Т. Основы веб-программирования на Python и Django. — М. : ДМК Пресс, 2021. — 368 с. (Для знакомства с практическим веб-разработкой)

Модуль «Нейронные сети»

Для педагогов

- Кнут, Д. Искусство программирования. Том 1–4. — 3-е изд. — М. : Вильямс, 2019. (Фундамент для понимания алгоритмов, применимых в ИИ)
- Гудфеллоу, И., Бенджио, Й., Курвиль, А. Глубокое обучение. — М. : ДМК Пресс, 2018. — 796 с. (Классический курс по глубокому обучению — теория и практика)
- Бишоп, К. Распознавание образов и машинное обучение. — М. : Техносфера, 2019. — 784 с. (Для глубокой теоретической базы по машинному обучению)
- Хейкин, С. Нейронные сети: полный курс. — 2-е изд. — М. : Вильямс, 2020. — 1104 с. (Системное пособие по нейросетевым моделям и их применению)
- Рассел, С., Норвиг, П. Искусственный интеллект: современный подход. — 4-е изд. — М. : Вильямс, 2022. — 1136 с. (Методологическая база для понимания общей теории искусственного интеллекта)

Для учащихся

- Мирджалили, В. Python и машинное обучение. — 3-е изд. — СПб. : Питер, 2022. — 848 с. (Пошаговое освоение машинного обучения на Python)
- Чоллет, Ф. Глубокое обучение на Python. — 2-е изд. — М. : ДМК Пресс, 2020. — 480 с. (Практическое руководство по Keras и TensorFlow)
- Грин, Дж. Нейронные сети и глубокое обучение. — СПб. : Питер, 2021. — 416 с. (Понятное введение в глубокое обучение для старшеклассников)
- Прован, Д. Нейросети для начинающих. — СПб. : Питер, 2020. — 352 с. (Хорошее популярное объяснение основных идей и примеров)
- Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. Алгоритмы. Построение и анализ — 4-е изд. — М. : Вильямс, 2021. — 1328 с. (Для развития фундаментального понимания алгоритмов, необходимых в ИИ)

Контрольно-измерительные материалы

Оценка результатов проводится по бальной системе, за каждый правильный ответ обучающийся получает один балл.

Суммируя результаты выполнения всех заданий, определяется общая сумма результатов.

Низкий уровень: 1-3 баллов

Средний уровень: 4-6 баллов

Высокий уровень: 7-10 баллов

Учащиеся, набравшие более 4 баллов считаются прошедшими аттестацию.

Вывод:

Требуют особого педагогического внимания:

- учащиеся с результатом менее 4 баллов
- учащиеся с результатом более 8 баллов

Итоговый контроль

Главным критерием успешного освоения Программы является реализация проекта в сфере информационных технологий. Диагностика результативности освоения учащимися Программы происходит по окончании учебного года в форме открытого аттестационного занятия – защиты индивидуального или командного проекта.

Индивидуальная карточка входного контроля

Объединение «Программирование 2 ступень» модуль «_____»

группа № _____, 1 год обучения _____.

Ф.И.О. учащегося _____

«___» _____ 20__ г.

№	критерии / показатели	Оценки		
		Баллы по показателю	Баллы по критерию	Примечания
1	Техника безопасности			
	понимание инструкций по технике безопасности			
	навыки безопасного поведения			
2	Мотивированность			
	проявление интереса к созданию, разработке, анализу			
	желание работать над проектами, решать практические задачи			
	готовность к самостоятельной работе			
	умение задавать уточняющие и исследовательские вопросы			
3	Информированность и готовность			
	представление о том, что такое программирование			
	понимание слов: «цикл», «блок-схема», «язык программирования», «среда разработки»			
	общий кругозор в области современных технологий			
4	Умелость			
	базовые навыки работы с компьютером: запуск компьютера и программ, ввод информации, сохранение файлов в редакторе			
	специальные навыки: слепая печать, умение печатать в английской и русской раскладках клавиатуры			
5	Практичность			
	умение выделять главное в проблеме / задаче			
	способность объяснить свой выбор или решение			
6	Коммуникативность			
	готовность работать в паре или группе			
	умение слушать, отвечать по существу			
7	Средний балл всего			

Промежуточный контроль

Тест №1

Объединение «Программирование 2 ступень» модуль « _____ », группа № _____, 1 год обучения _____.
Ф.И.О. учащегося _____

« ____ » _____ 20 ____ г.

Решите задачу и загрузите код решения в тестирующую систему.

Задача А. Дедушка Paint

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 с
Ограничение по памяти	64 МБ

Псевдографический интерфейс, предшествовавший появлению полноценных графических операционных систем, стал важным этапом в развитии взаимодействия компьютера с пользователем. Этот метод визуализации информации на экране имитировал графику, используя символы и управляющие коды для создания иллюзии графического интерфейса. Псевдографика активно использовалась во всех операционных системах с интерфейсом командной строки (CLI). Однако благодаря созданию компьютерной мыши ещё в 1980-х именно для таких операционных систем были разработаны первые графические редакторы.

Представьте, что вы оказались на заре эпохи персональных компьютеров. Разработайте псевдографический редактор с возможностью рисования прямоугольников на изначально пустом холсте размером $N \times M$. При инициализации программа создаёт холст шириной N и высотой M , заполняя его символами ' ' (пробел). Затем программа принимает от пользователя K команд для рисования прямоугольников. После окончания ввода команд программа выводит получившийся холст на экран.

Пользователь вводит команды для рисования в виде строки формата «флаг $X_1 Y_1 X_2 Y_2$ », где флаг – режим рисования, X_1 и Y_1 – координаты верхнего левого угла, $X_2 Y_2$ – координаты нижнего правого угла прямоугольника. Начало координат – точка $(0, 0)$ – верхний левый угол холста. Координатой X считается расстояние по ширине, а координатой Y считается расстояние по высоте. Прямоугольники могут выходить за пределы холста, в этом случае на холсте отображается только видимая часть фигур.

У программы есть ровно два режима рисования прямоугольника: с белой заливкой (символами пробела) – флаг '+' (плюс) и без заливки – флаг '-' (минус). В обоих режимах у фигуры рисуется контур из символов '*' (звёздочка) толщиной 1. Толщина контура является частью общего размера прямоугольника.

Прямоугольники могут накладываться друг на друга. Каждая следующая команда пользователя создаёт прямоугольник поверх созданного предыдущей командой.

Формат ввода:

На первых двух строках вводятся натуральные числа N – ширина холста и M – высота холста ($3 \leq N, M \leq 1000$). На третьей строке вводится натуральное число K – количество команд, вводимых пользователем ($1 \leq K \leq 100$). Далее на K строках вводятся

команды для рисования, состоящие из разделенных пробелами флага (символ ‘+’ или ‘-’) и целых чисел X_1, Y_1, X_2, Y_2 ($-500 \leq X_1 \leq X_2 \leq 1500, -500 \leq Y_1 \leq Y_2 \leq 1500$).

Формат вывода:

Необходимо вывести М строк длиной N, состоящих из символов ‘*’ (звёздочка) и ‘ ’ (пробел) – холст псевдографического редактора с нарисованными заданным набором команд прямоугольниками.

Примеры:

Ввод	Вывод
10 10 2 - 0 0 3 3 + 1 1 8 6	***** ***** ** * ** * * * * * *****
5 6 3 - 1 2 4 2 + -100 1 100 5 - 3 3 4 4	***** ** ** *****

Критерии автоматической проверки решения

Количество верных тестов	Итоговый балл
2	1
3	5
6	10

Промежуточный контроль
Тест №2

Объединение «Программирование 2 ступень» модуль « _____ »,
группа № _____, 1 год обучения _____.
Ф.И.О. учащегося _____

« ____ » _____ 20 ____ г.

Решите задачу и загрузите код решения в тестирующую систему.

Задача В. Дедушка Принтер

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 с
Ограничение по памяти	64 МБ

Старый принтер в офисе принимает задания на печать по проводной сети. Однако именно сегодня сразу N сотрудников офиса отправили сразу по X файлов на печать.

Каждому файлу требуется некоторое время Y_i для загрузки в память принтера. Поэтому за годы тяжелой работы принтер выработал несколько правил работы с торопливыми сотрудниками офиса:

- Для каждого сотрудника формируется отдельная очередь его файлов, при этом файлы очереди загружаются в память принтера последовательно: следующий файл одного сотрудника начинает загрузку только после печати предыдущего его файла. Одновременно в принтер загружается только один файл одного сотрудника.
- После печати очередного файла принтер считает, чей следующий файл загрузится быстрее – именно его он печатает. Принтер печатает файлы мгновенно, поэтому если одновременно загрузятся файлы сразу нескольких сотрудников, они будут напечатаны все. А затем принтер снова переключится на ближайший загружаемый файл среди очередей всех сотрудников.

Ваша задача проста: определите, чьи файлы принтер напечатает быстрее. Напишите программу, которая принимает на ввод время печати файлов каждого из сотрудников, а затем выводит номер сотрудника (нумерация с 1 в порядке ввода), все файлы которого будут напечатаны быстрее остальных. Гарантируется, что такой сотрудник всегда один.

Формат ввода:

На первой строке вводится натуральное число N – количество сотрудников. На второй строке вводится натуральное число X ($1 \leq N, X \leq 1000$) – количество файлов, отправленных на печать каждым сотрудником. Далее следует N строк, на каждой из которых перечислены X разделённых пробелом натуральных чисел $Y_0 \dots Y_X$ – время, необходимое для загрузки каждого файла сотрудника ($1 \leq Y_i \leq 1000$).

Формат вывода:

Необходимо вывести одно число – номер сотрудника, чьи файлы принтер напечатает быстрее.

Пример:

Ввод	Вывод
2 2 1 99 50 50	2

Пояснение к примеру:

- 1) Сначала принтер напечатает файл сотрудника №1.
 - 2) Теперь в очереди файл сотрудника №1 (через 99 секунд) и файл сотрудника №2 (50 секунд). Принтер выбирает сотрудника №2.
 - 3) Через 50 секунд принтер напечатает файл сотрудника №2.
 - 4) Теперь в очереди файл сотрудника №1 (через 99 секунд) и файл сотрудника №2 (50 секунд). Принтер выбирает сотрудника №2.
 - 5) Через 50 секунд принтер напечатает файл сотрудника №2.
- Сотрудник №2 свободен. Теперь принтер напечатает файл сотрудника №1.

Критерии автоматической проверки решения

Количество верных тестов	Итоговый балл
1	1
2	5
5	10

Промежуточный контроль

Тест №3

Объединение «Программирование 2 степень» модуль «_____»,
группа № _____, 2 год обучения _____.
Ф.И.О. учащегося _____

«__» _____ 20__ г.

Решите задачу и загрузите код решения в тестирующую систему.

Задача С. Дедушка USB

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	1 с
Ограничение по памяти	32 МБ

Каждый человек на нашей планете знаком с интерфейсом USB. Для кого-то это удобный разъём, в который можно подключить и мышь, и флешку. Но для нас, программистов, это в первую очередь чудо инженерной мысли. Задумайтесь, ещё в 1995 году был разработан последовательный интерфейс, способный автоматически работать с множеством устройств через один разъём, подбирать драйвера для каждого из этих устройств и даже мгновенно включать их в работу компьютера. Так уже устаревший USB стандарта 2.0 способен через несколько разветвителей (USB-хабов) работать одновременно с 90-120 периферийными устройствами!

Предположим, что у Вас есть некоторое количество одинаковых разветвителей и ровно один свободный разъём USB, способный работать одновременно с любым количеством периферии. Рассчитайте, сколько же устройств Вы сможете подключить через эти разветвители к компьютеру и определите, сможет ли компьютер работать с ними одновременно через один изначальный USB-разъём. Не забывайте, что некоторые разветвители придётся подключать в другие разветвители.

Формат ввода:

На первой строке вводится натуральное число U – количество разветвителей USB ($1 \leq U \leq 1000$). На второй строке вводится натуральное число S – количество разъёмов в каждом из разветвителей ($1 \leq S \leq 1000$). На третьей строке вводится натуральное число B – максимальное количество периферийных устройств, с которым может одновременно работать компьютер через один USB-интерфейс ($1 \leq B \leq 1000000$).

Формат вывода:

На первой строке натуральное число – количество устройств, которые можно подключить через разветвители. На второй строке фраза “YES”, если интерфейс сможет работать с этими устройствами одновременно, или “NO” в ином случае.

Примеры:

Ввод	Вывод
5	46
10	NO
20	
3	7

3 40	YES
---------	-----

Критерии автоматической проверки решения

Количество верных тестов	Итоговый балл
2	1
3	5
6	10

**Мониторинг результатов обучающихся по дополнительной общеразвивающей программе
«Программирование 2 ступень» модуль «_____»
Критерии оценки итогового контроля (защиты проектов)**

Показатели	Критерии	Степень выраженности оцениваемого качества	Возможное количество баллов	Методы диагностики
Предметные				
I. Теоретическая подготовка ребенка: 1.1. Понимание структуры проектной деятельности (этапы: цель, задачи, результат, анализ) 1.2. Знание ключевых понятий: проект, прототип, критерий, задача, этап 1.3. Умение сформулировать цель и задачи проекта 1.4. Представление о рисках, ресурсах, ограничениях	- Соответствие теоретических знаний ребенка программным требованиям - Осмысленность и правильность использования специальной терминологии	- минимальный уровень (ребенок овладел менее чем 50% объема знаний, предусмотренных Программой, ребенок, как правило, избегает употреблять специальные термины);	1	Наблюдение, тестирование, контрольный опрос и др.
		- средний уровень (объем усвоенных знаний составляет более 50%, ребенок сочетает специальную терминологию с бытовой);	5	
		- максимальный уровень (ребенок освоил практически весь объем знаний, предусмотренных Программой за конкретный период, специальные термины употребляет осознанно и в полном соответствии с их содержанием);	10	
II. Практическая подготовка: 2.1. Навыки планирования проекта: а) структурирование идеи б) составление последовательного плана 2.2. Визуализация проекта: а) создание схем, набросков, макетов б) оформление блоков в презентационной или табличной форме 2.3. Практическое описание проекта: а) фиксация промежуточных и финальных результатов б) определение функций участников	- Соответствие практических умений и навыков программным требованиям - Отсутствие затруднений в использовании специального оборудования и оснащения	- минимальный уровень (ребенок овладел менее чем 50% предусмотренных умений и навыков, ребенок испытывает серьезные затруднения при работе с оборудованием, ребенок в состоянии выполнять лишь простейшие практические задания педагога);	1	Контрольное задание
		- средний уровень (объем усвоенных умений и навыков составляет более 50%, работает с оборудованием с помощью педагога, выполняет в основном задания на основе образца);	5	
		- максимальный уровень (ребенок овладел практически всеми умениями и навыками, предусмотренными Программой за конкретный период, работает с оборудованием самостоятельно, не испытывает особых трудностей, выполняет практические задания с элементами творчества);	10	
III. Выполнение	- Креативность в	- минимальный уровень (ребенок в	1	

творческих заданий: 3.1. Проявление нестандартного подхода: а) оригинальность идеи проекта б) гибкость в выборе решений 3.2. Работа без жёстких шаблонов: а) попытка отойти от предложенного образца б) доработка проекта по собственной инициативе 3.3. Вовлечённость в проект: а) стремление дополнить проект деталями б) инициатива в представлении результата	выполнении заданий	состоянии выполнять лишь простейшие практические задания педагога); - средний уровень (выполняет в основном задания на основе образца); - максимальный уровень (выполняет практические задания с элементами творчества);	5	Контрольное задание
			10	
Итого				
Метаредметные (общеучебные умения и навыки ребенка)				
IV. Интеллектуально-познавательные умения: 4.1. Умение выделять главное: а) анализ условия б) абстрагирование от второстепенного 4.2. Построение логики проекта: а) последовательность шагов б) связь задачи и результата 4.3. Аргументация решений: а) логика выбора идеи и способа реализации б) защита своей позиции в команде	- Способность анализировать информацию из кейса, выделять главное и второстепенное - Умение строить логическую последовательность действий (от проблемы к решению, от цели к результату) - Аргументация своих проектных решений — как в письменной части, так и в устной защите.	- минимальный уровень умений (обучающийся затрудняется выделить суть задания, не может объяснить, почему предложил ту или иную идею, ответы неполные, поверхностные, не связаны логически); - средний уровень (обучающийся способен определить ключевую цель и предложить логичный план, использует шаблонные формулировки без глубокой аргументации, допускает незначительные логические неточности); - максимальный уровень (обучающийся уверенно анализирует ситуацию, предлагает осмысленные шаги, аргументирует выбор решений, учитывает риски, обосновывает действия, способен сопоставлять альтернативные варианты и делать логические выводы)	1	Наблюдение, анализ
			5	
			10	
V. Коммуникативные умения: 5.1. Восприятие чужой точки зрения: а) умение слушать и не перебивать б) корректные уточнения 5.2. Устная	- Активность в общении с командой, умение слушать и ясно выражать мысли, участие в обсуждении.	- минимальный уровень умений (обучающийся молчалив, не участвует в обсуждении, либо перебивает и не слышит других, затрудняется формулировать свою позицию, демонстрирует признаки конфликтности или избегания общения);	1	Наблюдение, анализ
			5	
			10	

аргументация: а) ясное выражение мыслей б) ответы на вопросы по теме проекта 5.3. Командное взаимодействие: а) участие в обсуждении б) поддержка и дополнение идей других		- средний уровень (обучающийся слушает, но редко сам инициирует реплики, формулирует мысли с неуверенностью или требует поддержки, участвует в обсуждении, но не всегда конструктивно); - максимальный уровень (обучающийся активно и уважительно взаимодействует с партнёрами, свободно и логично выражает свои мысли, аргументирует и задаёт уточняющие вопросы, выступает в роли коммуникативного посредника или лидера в команде)		
VI. Организационно-регулятивные умения и навыки: 6.1. Самоорганизация: а) подготовка к работе б) соблюдение этапов задания 6.2. Ответственность за этап: а) выполнение без напоминаний б) своевременная сдача 6.3. Корректировка действий: а) принятие замечаний б) доработка после обсуждения	- Способность организовать свою работу, соблюдать сроки, адекватно реагировать на обратную связь	- минимальный уровень умений (обучающийся работает медленно, с постоянными напоминаниями, не может распределить действия во времени, отказывается от исправлений, реагирует на замечания негативно); - средний уровень (обучающийся выполняет работу, но иногда теряет последовательность, может отреагировать на замечания, но не всегда их внедряет, требует умеренного сопровождения от педагога); - максимальный уровень (обучающийся чётко планирует и реализует свою часть задания, вносит корректировки по замечаниям без сопротивления, анализирует ошибки, завершает работу вовремя и соблюдает договорённости команды)	1 5 10	Наблюдение, анализ
Итого				
Личностные				
VII. Организационно-волевые качества: 7.1. Выдержка: а) концентрация в течение всей работы б) отсутствие раздражения при трудностях 7.2. Настойчивость: а) продолжение работы при ошибках б) стремление довести до результата	- Способность переносить (выдерживать) известные нагрузки в течение определенного времени, преодолевать трудности. - Способность активно побуждать себя к практическим действиям.	- минимальный уровень умений (терпения хватает меньше чем на 50% занятия, волевые усилия ребенка побуждаются извне, ребенок постоянно находится под воздействием контроля извне); - средний уровень (терпения хватает больше чем на 50% занятия, волевые усилия ребенка побуждаются иногда – самим ребенком, периодически контролирует себя сам); - максимальный уровень (терпения хватает на все занятие, волевые усилия ребенка	1 5 10	Наблюдение

	- Умение контролировать свои поступки (приводить к должному свои действия).	побуждаются всегда – самим ребенком, постоянно контролирует себя сам)		
VIII. Ориентационные качества: 8.1. Самооценка: а) осознанность сильных и слабых сторон б) соответствие самооценки результату 8.2. Мотивация: а) интерес к содержанию работы б) желание выполнять задание, а не просто закрыть тему	- Способность оценивать себя адекватно реальным достижениям. - Осознанное участие ребенка в освоении образовательной программы	- минимальный уровень умений (завышенная/заниженная самооценка, интерес к занятиям продиктован ребенку извне); - средний уровень (адекватная самооценка, интерес периодически поддерживается самим ребенком); - максимальный уровень (адекватная самооценка, интерес постоянно поддерживается ребенком самостоятельно)	1 5 10	Анкетирование, тестирование
IX. Поведенческие качества: 9.1. Поведение в конфликтной ситуации: а) адекватная реакция б) умение разрешать спор 9.2. Взаимодействие в группе: а) принятие совместного решения б) соблюдение договорённостей	- Способность занять определенную позицию в конфликтной ситуации - Умение воспринимать общие дела, как свои собственные	- минимальный уровень умений (обучающийся отстраняется от работы, не проявляет интереса к результату команды, не выполняет взятые на себя обязательства, нарушает договорённости); - средний уровень (обучающийся выполняет свою часть, но не проявляет инициативы, участвует в обсуждении, но редко влияет на общую стратегию); - максимальный уровень (обучающийся активно вовлечён в командную работу, предлагает идеи, берёт на себя ответственность, поддерживает партнёров, соблюдает договорённости, вносит вклад в достижение общего результата)	1 5 10	Наблюдение
Итого				

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ ЦЕНТР ДЕТСКОГО (ЮНОШЕСКОГО) ТЕХНИЧЕСКОГО
ТВОРЧЕСТВА «СТАРТ+» НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

ПРИНЯТА

Решением Педагогического совета
ГБУ ДО ЦД Ю) ТТ «Старт+»
Невского района Санкт-Петербурга
Протокол от _____ № _____

УТВЕРЖДЕНА

Приказом от _____ № _____
Директор ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт-Петербурга
_____ Подобаева О.Г

**Рабочая программа
к дополнительной общеразвивающей программе
«ПРОГРАММИРОВАНИЕ 2 СТУПЕНЬ»
МОДУЛЬ «ПРОМЫШЛЕННОЕ ПРОГРАММИРОВАНИЕ»**

1 год обучения
Возраст обучающихся 13-17 лет

Разработчик: Тупицын Пётр Викторович,
педагог дополнительного образования
высшей квалификационной категории

ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА 1 ГОДА ОБУЧЕНИЯ

Группу составляют учащиеся в возрасте 13-17 лет, имеющие хорошие навыки работы с персональным компьютером, интересующиеся сферой информационных технологий и разработкой комплексных пользовательских приложений, в количестве не менее 15 человек.

Занятия объединения по Программе «Программирование 2 ступень» модуль «Промышленное программирование» 1 года обучения проводятся 2 раза в неделю в очном формате, длительность каждого занятия – 2 учебных часа с 10-минутным перерывом. Длительность освоения Программы - 144 часа за 1 учебный год.

Особенности организации образовательного процесса по Программе заключаются в модульном построении содержания и последовательном переходе от освоения базовых элементов языка Python к разработке прикладных проектов. Каждый раздел объединяет несколько занятий, включающих изучение теоретического материала, выполнение практических заданий и завершение в форме проектной работы.

Теоретическая часть каждого занятия строится в форме краткой беседы с демонстрацией программных конструкций, алгоритмов и приёмов. Важное внимание уделяется логике их применения. Практическая часть занятия является основной. Она включает решение задач различного уровня сложности под руководством педагога, что позволяет отрабатывать новый материал и сразу применять его в работе.

Особым элементом образовательного процесса является проектная деятельность. После каждого раздела, посвященного синтаксису или структурам данных, учащиеся разрабатывают тематическое приложение с использованием знаний и навыков, полученных в ходе изучения модуля. Так учащиеся осваивают проектное мышление: учатся формулировать задачу, разрабатывать архитектуру приложения, распределять роли и доводить проект до презентационного вида. Формат хакатонов позволяет смоделировать реальные условия командной разработки и стимулирует креативность, самостоятельность и инициативу.

Программа ориентирована на формирование комплексных компетенций. Учащиеся осваивают язык Python как инструмент промышленного программирования, знакомятся с библиотеками, файловыми системами, базами данных и объектно-ориентированными принципами. При этом внимание уделяется не только результату, но и процессу: качеству кода, читаемости, соблюдению стандартов (PEP-8, PEP-20), безопасности и устойчивости программ.

Таким образом, обучение по Программе сочетает последовательное изучение теории, систематическую практику и проектную деятельность, что обеспечивает комплексное развитие как профессиональных, так и универсальных навыков, необходимых будущим специалистам в IT-сфере.

ЗАДАЧИ 1 ГОДА ОБУЧЕНИЯ

Обучающие

- Познакомить с основными понятиями программирования;
- Познакомить с синтаксисом языка Python;
- Научить разработке алгоритмов для обработки данных, представленных в различном виде;
- Научить проектировать архитектуру крупных программных продуктов;
- Научить разрабатывать и отлаживать работу программы под конкретную задачу;
- Познакомить с процедурной, событийно-ориентированной и объектно-ориентированной парадигмами программирования.

Развивающие

- Расширить представления о сферах применения информационных технологий;
- Развить навыки работы с информацией разного вида на компьютере;
- Развить навыки целеполагания и логического подхода к решению поставленной задачи;
- Сформировать способность к саморазвитию, самообразованию;
- Развить умение работать в команде, грамотно распределять роли и задачи между участниками команды.

Воспитательные

- Воспитать целеустремленность, усидчивость и терпение в достижении творческих результатов;
- Воспитать уважение к труду и его результатам;
- Сформировать уверенность в своих силах.
- Сформировать позитивное отношение к труду.

СОДЕРЖАНИЕ ПРОГРАММЫ 1 ГОДА ОБУЧЕНИЯ

1. Введение в программирование

Раздел является вводным и направлен на формирование базового понимания принципов работы программы и логики её построения. Учащиеся знакомятся с **основными элементами языка Python**: переменными, типами данных, вводом и выводом информации, операторами ветвления и циклами. Раскрывается алгоритмическая логика: как программа принимает решения, каким образом выполняются повторяющиеся действия, чем отличаются различные виды циклов и в каких случаях они применяются.

Особое внимание уделяется **понятию структуры алгоритма**: линейного, разветвляющегося и циклического. На основе этих элементов учащиеся учатся строить простые прикладные программы, выполнять пошаговую отладку, анализировать результаты и исправлять ошибки. Раздел служит фундаментом для последующего изучения более сложных структур данных и проектирования приложений.

Теория:

Понятие программы и алгоритма. Основы работы в среде Python. Ввод и вывод данных. Переменные и их типы. Условный оператор и сравнение данных. Ветвление и вложенные условия. Отступы и структура кода. Сложные условия и оператор `elif`. Цикл

while и работа с условием. Цикл с постусловием. Бесконечный цикл. Операторы break и continue. Цикл for и работа с параметрами.

Практика:

Написание первых программ с вводом и выводом данных. Работа с различными типами переменных. Решение задач на использование условных операторов. Отладка и исправление ошибок во вложенных условиях. Применение сложных логических выражений при ветвлениях. Решение задач с циклами while и for. Использование операторов break и continue для управления циклом. Практическое сравнение алгоритмов с разными циклами.

2. Структура пользовательского приложения. Часть 1

Раздел посвящён изучению принципов логического построения классических пользовательских приложений. Учащиеся знакомятся с идеей работы программы в **формате бесконечного цикла**, когда приложение ожидает действий пользователя и реагирует на них. Осваиваются приёмы использования **системы флагов** для управления логикой работы и сохраняемыми состояниями. Важной частью раздела является работа с готовыми библиотеками и модулями, что вводит учащихся в основы API и показывает возможности расширения языка средствами стандартного репозитория.

В практической части большое внимание уделяется созданию прикладных программ, которые **постоянно взаимодействуют с пользователем**. Итогом раздела становится разработка собственного приложения «Игра в города» с использованием бесконечного цикла и системы флагов. Работа завершается мини-хакатоном, где учащиеся дорабатывают и презентуют свои решения, что формирует первый опыт проектной деятельности и демонстрирует, как базовые алгоритмические конструкции могут складываться в полноценное пользовательское приложение.

Теория:

Логический тип данных. Логика условного оператора и её применение в построении приложений. Система флагов и управление состоянием программы. Бесконечный цикл и его назначение. Использование встроенных функций Python. Модули и библиотеки стандартного репозитория. Рациональное использование переменных и множественное присваивание. Приёмы рационального сокращения кода.

Практика:

Создание простых программ на основе бесконечного цикла. Управление логикой работы через систему флагов. Решение задач с использованием встроенных функций Python. Подключение и применение стандартных модулей. Тренировка в рациональном использовании переменных и приёмах множественного присваивания. Отработка компактных решений в одну строку. Разработка приложения «Игра в города» с использованием бесконечного цикла. Отладка и улучшение пользовательского взаимодействия.

3. Контейнеры данных

Раздел посвящён **изучению контейнеров** данных в Python и формированию понимания того, как информация может храниться и обрабатываться в программах. Учащиеся осваивают работу со списками и строками, знакомятся с их методами и возможностями. Особое внимание уделяется тому, как списки и строки упрощают решение прикладных задач за счёт удобных встроенных инструментов. Учащиеся осваивают методы работы со словарями, учатся выбирать подходящий тип контейнера

в зависимости от задачи. Важной частью раздела становится использование списочных выражений и приёмов записи программ «в одну строку», что развивает алгоритмическое мышление и показывает возможности сокращённой записи кода.

Завершением раздела является хакатон «Генераторы текста», на котором учащиеся применяют все изученные приёмы для создания прикладных программ по обработке текстовых данных и их преобразованию. В основе применяемого в данном разделе генератора текста частотный анализ слов и анализ последовательностей слов без применения нейронных сетей. Это формирует **умение комбинировать разные контейнеры и методы**, проектировать алгоритмы обработки информации и представлять результаты в удобной форме.

Теория:

Знакомство с объектно-ориентированным стилем языка. Объекты и типы данных. Списки и новые возможности цикла `for`. Функции `min` и `max` для обработки данных в списке. Строка как контейнер символов. Посимвольная работа со строками. Методы списков и строк. Решение задач с использованием методов. Списочные выражения. Рациональное использование списочных выражений в классических алгоритмах. Словарь как контейнер для эффективного доступа к данным.

Практика:

Создание и обработка списков с использованием цикла `for`. Решение задач с функциями `min` и `max`. Посимвольная обработка строк. Применение методов списков и строк для анализа и преобразования данных. Отработка задач на использование методов. Использование списочных выражений для компактных решений. Написание программ в одну строку. Решение прикладных задач с использованием словарей. Хакатон «Генераторы текста» и презентация решений.

4. Структура пользовательского приложения. Часть 2

Раздел направлен на формирование у учащихся понимания принципов **процедурного подхода в программировании**. Учащиеся осознают, как функции помогают структурировать код, упрощают повторное использование алгоритмов и делают программы более гибкими и устойчивыми к изменениям. Раздел развивает навыки проектирования программ с учётом логики работы и взаимодействия различных компонентов, учит оценивать, как выбор структуры кода влияет на читаемость, поддержку и расширяемость приложения.

Ключевой акцент раздела – **применение функций для реализации сложных проектов**. Учащиеся учатся комбинировать разные элементы языка, работать с внешними ресурсами и системными инструментами, проверять корректность работы программы и отлаживать её. Раздел формирует умение мыслить проектно: видеть программу как набор взаимодействующих компонентов, планировать архитектуру и подходы к реализации.

Теория:

Знакомство с функциями. Аргументы функции. Функции, возвращающие значение. Области видимости переменных, локальные и глобальные переменные. Аргументы по умолчанию. Именованные аргументы. Конструкции `*args` и `**kwargs`. Проектирование программ на основе функции. Функция как объект. Работа с файлами. Библиотека `os` и взаимодействие с операционной системой. Утилита `pyinstaller` и создание исполняемого файла программы. PEP-8 и PEP-20.

Практика:

Создание функций с различными аргументами и возвращаемыми значениями. Практическое применение локальных и глобальных переменных. Использование аргументов по умолчанию и именованных аргументов. Решение задач с применением *args и **kwargs. Проектирование и реализация программных модулей на основе функций. Работа с файлами для чтения и записи данных. Применение библиотеки os для взаимодействия с файловой системой. Создание исполняемых файлов с помощью pyinstaller.

5. Таблицы и СУБД

Раздел представляет собой логическое продолжение изучения контейнеров данных, расширяя понимание способов хранения и обработки информации на примере **сложных и многомерных структур**. Учащиеся учатся работать с матрицами и многомерными списками, осваивают алгоритмы сортировки, транспонирования и обработки данных в различных форматах. Важной частью раздела становится освоение принципов работы с таблицами и базами данных, что формирует понимание того, как структурированная информация может эффективно храниться, извлекаться и модифицироваться в пользовательских приложениях.

Особое внимание уделяется **связям между данными**, индексам, способам оптимизации доступа и обеспечению корректности информации. Раздел также знакомит с использованием современных инструментов для работы с базами данных и форматами файлов, включая csv и SQL, а также с применением библиотек Python для управления данными и их безопасной обработки. Итоговая работа раздела в формате хакатона позволяет учащимся интегрировать полученные знания в полноценное приложение с базой данных, демонстрируя системное мышление и способность решать комплексные задачи обработки информации.

Теория:

Матрицы и многомерные списки. Алгоритмы работы с матрицами. Алгоритмы сортировки и транспонирования. Lambda-функции. Функции высшего порядка. Регулярные выражения. Работа с csv-таблицами. Базы данных и СУБД. Связанные таблицы, сортировка и индексы. SQLAlchemy. Пользовательская оболочка для программ с базой данных. Безопасность баз данных и защита от инъекций кода.

Практика:

Создание и обработка многомерных списков. Решение задач с применением алгоритмов сортировки и транспонирования. Применение lambda-функций и функций высшего порядка для обработки данных. Работа с регулярными выражениями. Чтение и запись csv-таблиц. Организация работы с базой данных. Создание связанных таблиц, настройка сортировки и индексов. Использование SQLAlchemy для управления данными. Разработка пользовательской оболочки для работы с базой данных. Обеспечение безопасности базы данных и предотвращение инъекций.

6. Объектно-ориентированное программирование

Раздел посвящён изучению **объектно-ориентированного программирования** – ключевую парадигму программирования на python, использующуюся в профессиональном программировании. ООП позволяет организовать код более структурированно и обеспечивать масштабируемость приложений. Учащиеся учатся создавать классы и объекты, определять атрибуты и методы, формировать иерархии классов с

наследованием, а также использовать механизмы инкапсуляции для управления доступом к данным.

Важным этапом изучения раздела является понимание логики проектирования приложений в рамках ООП с достаточным уровнем абстракции.

Особое внимание уделяется **принципам расширяемости и повторного использования кода, пониманию полиморфизма и интерфейсов**, а также композиции и агрегирования объектов для построения сложных систем. Раздел включает разработку итогового проекта первого года обучения, в ходе которой учащиеся выбирают тему, разрабатывают архитектуру программы, создают классы и интегрируют работу с базами данных. Работа с многопоточностью и интерфейсами позволяет применять изученные принципы в условиях, близких к промышленной разработке.

Теория:

ООП. Классы и объекты. Конструктор `init` и методы класса. Атрибуты и методы: объектные и классовые. Инкапсуляция. Скрытие данных и свойства. Наследование, базовые и дочерние классы. Функция `super()`, расширение и изменение поведения. Полиморфизм и интерфейсы. Множественное наследование и миксины. Магические методы классов. Композиция и агрегирование объектов.

Практика:

Создание классов и объектов с различными атрибутами и методами. Определение и использование конструктора `init`. Работа с объектными и классовыми атрибутами и методами. Применение инкапсуляции и свойств для управления доступом к данным. Разработка иерархий классов с наследованием. Использование `super()` для расширения и изменения поведения. Применение полиморфизма и интерфейсов. Реализация множественного наследования и миксинов. Работа с магическими методами классов. Создание сложных систем с использованием композиции и агрегирования объектов. Разработка итогового проекта.

7. Итоговое занятие

Теория: Подведение итогов обучения. Анализ достижений.

Практика: Защита проектов. Коллективная рефлексия.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ 1 ГОДА ОБУЧЕНИЯ

После обучения по Программе учащиеся получают следующие результаты:

Предметные:

- Учащиеся знают основные понятия программирования;
- Учащиеся умеют программировать с использованием языка Python;
- Учащиеся умеют работать с контейнерами данных;
- Учащиеся умеют работать с файлами и базами данных;
- Учащиеся умеют разрабатывать программы, передающие данные по локальной сети;
- Учащиеся умеют проектировать архитектуру программного обеспечения;
- Учащиеся умеют разрабатывать интерфейс программы;
- Учащиеся знают принципы процедурного программирования и разработки функций;

- Учащиеся знают принципы событийно-ориентированного программирования и обработки событий;
- Учащиеся знают принципы объектно-ориентированного программирования;
- Учащиеся умеют писать читаемый код программы, разрабатывать грамотный API для использования модулей другими разработчиками.

Метапредметные:

- Развиты представления о сферах применения информационных технологий;
- Развиты навыки работы с информацией разного вида на компьютере;
- Развиты навыки целеполагания и логического подхода к решению поставленной задачи;
- Сформированы способность к саморазвитию, самообразованию;
- Развиты умение работать в команде разработчиков, грамотно распределять роли и задачи между участниками команды.

Личностные:

- Сформировано ответственное отношение к труду, понимание важности соблюдения сроков и качества выполнения.
- Развита настойчивость в достижении цели, готовность преодолевать трудности в процессе работы.
- Учащиеся демонстрируют способность воспринимать конструктивную критику и высказывать мнение корректно и аргументированно.
- Учащиеся проявляют интерес к сфере технического творчества.
- Учащиеся осознают значимость участия в общественно полезной и профессионально ориентированной проектной деятельности.

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ ЦЕНТР ДЕТСКОГО (ЮНОШЕСКОГО) ТЕХНИЧЕСКОГО
ТВОРЧЕСТВА «СТАРТ+» НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

ПРИНЯТА

Решением Педагогического совета
ГБУ ДО ЦД Ю) ТТ «Старт+»
Невского района Санкт-Петербурга
Протокол от _____ № _____

УТВЕРЖДЕНА

Приказом от _____ № _____
Директор ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт-Петербурга
_____ Подобаева О.Г

**Рабочая программа
к дополнительной общеразвивающей программе
«ПРОГРАММИРОВАНИЕ 2 СТУПЕНЬ»
МОДУЛЬ «ПРОМЫШЛЕННОЕ ПРОГРАММИРОВАНИЕ»**

2 год обучения
Возраст обучающихся 13-17 лет

Разработчик: Тупицын Пётр Викторович,
педагог дополнительного образования
высшей квалификационной категории

ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА 2 ГОДА ОБУЧЕНИЯ

Группу составляют учащиеся в возрасте 13-17 лет, успешно завершившие обучение по программе 1 года обучения, имеющие хорошие навыки работы с персональным компьютером, интересующиеся сферой информационных технологий и разработкой комплексных пользовательских приложений, в количестве не менее 12 человек.

Занятия объединения по Программе «Программирование 2 ступень» модуль «Промышленное программирование» 2 года обучения проводятся 2 раза в неделю в очном формате, длительность каждого занятия – 2 учебных часа с 10-минутным перерывом. Длительность освоения Программы - 144 часа за один учебный год.

Особенности организации образовательного процесса по Программе заключаются в углубленном освоении языка Python через изучение его применения в реальных сферах разработки, прежде всего облачных и веб-приложений. Программа направлена на формирование у учащихся профессиональных навыков проектирования, разработки и тестирования программных решений, которые могут использоваться в реальных проектах.

Занятия построены так, чтобы теоретическая часть была компактной и целенаправленной, а практическая позволяла сразу применять изученные технологии на реальных примерах. Учащиеся осваивают основы многопоточности, сетевого взаимодействия, работы с API, построения серверных и клиентских приложений, оттачивают навыки тестирования и организации корректного взаимодействия компонентов программы.

Для развития навыков самостоятельной развёртки приложений на сервере учащиеся изучают раздел «Сетевое администрирование для программистов», который является общим для второго года обучения модулей «Промышленное программирование» и «Нейронные сети» ДОП «Программирование 2 ступень».

Программа формирует профессиональное мышление: учащиеся учатся планировать архитектуру приложения, оценивать влияние выбранных технологий на производительность и безопасность, применять инструменты разработки и контроля качества кода. В процессе работы над проектами учащиеся создают полноценные приложения, отрабатывают практики взаимодействия клиента и сервера, управления потоками данных, деплоя и отладки программ. Итоговые проекты включают проектирование архитектуры, реализацию компонентов, тестирование и презентацию результатов, что формирует системное видение процесса разработки и подготовку к профессиональной деятельности в сфере программирования.

ЗАДАЧИ 2 ГОДА ОБУЧЕНИЯ

Обучающие

- Познакомить с углубленными технологиями программирования на Python, применяемыми в реальных проектах;
- Освоить принципы многопоточности, асинхронного и параллельного программирования;
- Научить работать с сетевыми протоколами, API, клиент-серверной архитектурой и облачными технологиями;
- Развить навыки проектирования и разработки полноценных веб- и серверных приложений;

- Научить организовывать тестирование кода, управлять зависимостями и деплоем приложений;
- Познакомить с основами безопасности данных и управления сетевыми взаимодействиями.

Развивающие

- Расширить представления о сферах применения Python в промышленной и прикладной разработке;
- Развить навыки системного мышления при проектировании архитектуры программ;
- Сформировать способность к целеполаганию и логическому подходу к решению комплексных задач;
- Развить умение работать с реальными инструментами разработки, контролировать качество кода;
- Развить навыки командной работы при реализации проектов, распределении ролей и задач.

Воспитательные

- Воспитать целеустремленность, усидчивость и дисциплину при реализации сложных проектов;
- Воспитать уважение к процессу разработки и результатам труда;
- Сформировать уверенность в собственных силах при решении нестандартных и комплексных задач;
- Сформировать профессиональное отношение к труду и понимание ценности качественной работы.

СОДЕРЖАНИЕ ПРОГРАММЫ 2 ГОДА ОБУЧЕНИЯ

1. Углубленные технологии программирования. Облачные вычисления

Раздел объединяет изучение **углубленных технологий программирования** с практическими навыками **построения приложений для обмена данными** и работы в распределённой среде. С одной стороны рассматриваются продвинутое темы, которые расширяют профессиональное понимание Python и его возможностей в реальных проектах: многопоточность, асинхронное и параллельное программирование, логирование и тестирование кода, управление зависимостями и виртуальные окружения. С другой стороны акцент сделан на практическом освоении основ передачи данных между компьютерами и приложениями: работа с сокетами, протоколом HTTP, создание простейшего клиента и сервера, построение облачного сервера на Flask.

В процессе раздела формируется понимание того, как локальная логика программы связана с сетевым взаимодействием, какие инструменты обеспечивают корректность и безопасность передачи данных, и как проектировать приложения для работы в условиях параллельных и распределённых вычислений.

Теория:

Логирование программ. Тестирование кода: unittest и pytest. Виртуальное окружение. Основы многопоточности и потоков в threading. Многопроцессорность и запуск программ на нескольких ядрах процессора. Асинхронное программирование и параллельные вычисления. Сокеты и низкоуровневое сетевое взаимодействие.

Безопасность передачи данных по сети. Протокол HTTP. Работа с REST API и модулем requests. HTTP-сервер. Организация обмена данными между клиентом и сервером через Flask-поток. Проектирование серверного приложения. Проблема очереди запросов и её распределение. Redis-сервер для автоматического распределения запросов.

Практика:

Настройка логирования и написание тестов для отдельных модулей кода. Создание виртуальных окружений и управление зависимостями проектов. Разработка многопоточных и многопроцессорных программ. Реализация асинхронных и параллельных вычислений. Практическое взаимодействие через сокеты, создание клиента и сервера. Реализация асимметричной криптосистемы для передачи данных. Отправка и обработка HTTP-запросов через модуль requests. Разработка REST API. Создание и запуск простого Flask-сервера. Организация обмена данными между клиентом и сервером через несколько потоков. Проектирование, отладка и тестирование серверного приложения. Решение задач по распределению запросов вручную и с использованием Redis.

2. Сетевое администрирование для программистов

Раздел посвящён освоению **принципов работы компьютерных сетей** и организации серверной инфраструктуры для приложений. Рассматриваются сетевые модели и протоколы, способы адресации и маршрутизации данных, методы настройки локальных сетей и удалённых серверов, а также принципы обеспечения безопасности серверных приложений.

В процессе раздела формируется понимание того, как взаимодействуют различные уровни сетевой архитектуры, как проектировать серверное приложение с учётом **безопасности и доступности**, и какие технологии позволяют управлять **потоками запросов**. Практическая деятельность направлена на закрепление знаний через настройку сетевого окружения, развертывание серверного приложения и обеспечение его корректной и безопасной работы.

Важность раздела обусловлена необходимостью для программиста понимать работу платформ, обеспечивающих работу созданных серверных программ, а также уметь определить необходимые настройки платформы для работы программы.

Теория:

Модели компьютерных сетей и основные сетевые протоколы. Принципы адресации и маршрутизации данных. Организация локальных и удалённых сетей. Основы размещения серверного приложения на сервере. Методы обеспечения безопасности серверных приложений, включая шифрование и контроль доступа. Управление потоками запросов и организация стабильной работы сервера.

Практика:

Настройка локальной сети и подключение к удалённому серверу. Размещение серверного приложения на сервере и проверка доступности. Проброс портов и настройка реверс-прокси. Настройка шифрования и SSL-сертификатов. Контроль запросов по IP-адресу. Отладка и тестирование работы серверного приложения.

3. Основы фронтенд-разработки

Раздел посвящён освоению технологий, которые обеспечивают создание и функционирование **пользовательских интерфейсов веб-приложений**. Основное внимание уделяется тому, как структурировать веб-страницы, как визуально

и функционально организовать элементы интерфейса, а также как реализовать интерактивное взаимодействие между пользователем и приложением.

Изучение фронтенд-разработки позволяет формировать целостное представление о веб-приложении: как данные, поступающие от пользователя, обрабатываются на клиентской стороне, как они передаются на сервер и как результат возвращается обратно. Особое внимание уделяется адаптивной верстке, чтобы интерфейс корректно отображался на разных устройствах и экранах, а также принципам удобства и доступности.

Практическая часть раздела позволяет сразу закреплять теоретические знания, создавая веб-страницы, добавляя интерактивные элементы и подключая их к серверной части приложения. Это формирует навыки проектирования интерфейса, планирования структуры веб-документа, использования CSS и JavaScript для динамического поведения страницы, а также понимание принципов клиент-серверного взаимодействия.

В результате формируется профессиональное мышление, необходимое для разработки веб-приложений: умение анализировать требования пользователя, проектировать интерфейс, реализовывать интерактивные элементы и обеспечивать корректное взаимодействие с серверной частью, при этом учитывая производительность, удобство и безопасность работы приложения.

Теория:

Основы веб-технологий и структура веб-документа. Семантические элементы HTML, работа с формами, таблицами и мультимедиа. Основы CSS: стилизация текста и элементов, блочная модель, позиционирование. Flexbox и Grid для построения адаптивной верстки. Принципы адаптивного дизайна и использование CSS-фреймворков (Bootstrap). Основы программирования на JavaScript: переменные, типы данных, условия и циклы, функции. Взаимодействие с DOM, обработка событий и формы. Асинхронное программирование на клиентской стороне, работа с API и AJAX. Базовое использование библиотек JavaScript.

Практика:

Создание структурированного веб-документа и добавление форм, таблиц и мультимедиа. Применение CSS для стилизации и адаптивного оформления страниц. Использование Flexbox и Grid для построения макета. Разработка интерактивных элементов с помощью JavaScript. Обработка событий, работа с формами и валидация данных. Реализация асинхронного взаимодействия с сервером через fetch и AJAX. Подключение клиентской части к Flask-серверу и тестирование взаимодействия. Создание интерактивной страницы с динамическим обновлением данных и подключением библиотек JavaScript.

4. Основы бэкенд-разработки

Раздел посвящён освоению **серверной части веб-разработки** и организации работы приложений на стороне сервера. Основное внимание уделяется принципам проектирования серверных приложений, обработке данных, взаимодействию с базами данных и обеспечению безопасности пользовательской информации.

В рамках раздела формируется понимание того, как веб-приложение функционирует за пределами пользовательского интерфейса: как обрабатываются **запросы от клиента**, как хранится и **обрабатывается информация на сервере**, как реализуются бизнес-логика и управление доступом пользователей. Рассматриваются ключевые элементы профессиональной разработки на Python с использованием фреймворка Django, включая структуру проекта, маршрутизацию, шаблоны, работу с формами, модели и ORM, панели администратора и CRUD-операции.

Практическая деятельность направлена на закрепление теории через создание реальных серверных приложений. Это включает проектирование архитектуры веб-приложения, разработку серверных компонентов, настройку базы данных и миграций, реализацию системы регистрации и авторизации пользователей, управление доступом, обработку файлов и данных, отладку и тестирование асинхронных потоков. Работа над проектами позволяет формировать навыки комплексной разработки, планирования архитектуры, взаимодействия между клиентской и серверной частями, а также развивать умение создавать приложения, готовые к размещению на сервере и эксплуатации в реальных условиях.

Раздел посвящен не только веб-разработке. Фреймворк Django важен при разработке приложений для облачных вычислений, поскольку способен обеспечивать устойчивую работу потока передачи данных при удобной разработке, на что не способен фреймворк Flask. Поэтому изучение Django – важный навык начинающего python-разработчика и значительный шаг в профориентации учащихся.

Теория:

Основы серверной разработки и ключевые принципы работы веб-приложений. Структура проекта Django и организация приложений. Маршрутизация запросов и обработка маршрутов. Шаблоны Django, наследование и включения. Работа с формами и обработка данных. Модели и ORM, создание базы данных и миграции. Панель администратора и управление CRUD-операциями. Связи между моделями и управление данными пользователей. Регистрация, авторизация и управление доступом. Загрузка и хранение файлов. Принципы проектирования серверной архитектуры и организация асинхронных потоков. Размещение и отладка серверного приложения на сервере.

Практика:

Создание первого приложения в Django и настройка структуры проекта. Настройка маршрутов и обработка запросов. Создание и использование шаблонов, организация наследования и включений. Разработка форм и обработка данных пользователей. Проектирование моделей и работа с базой данных, выполнение миграций. Настройка панели администратора и реализация CRUD-операций. Организация связей между моделями. Реализация регистрации, авторизации и прав доступа. Работа с файлами, загрузка и хранение данных. Проектирование серверного приложения, реализация асинхронных потоков и их отладка. Размещение приложения на сервере и тестирование работы в реальных условиях.

5. Итоговое занятие

Теория: Подведение итогов обучения. Анализ достижений.

Практика: Защита проектов. Коллективная рефлексия.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ 2 ГОДА ОБУЧЕНИЯ

После обучения по Программе учащиеся получают следующие результаты:

Предметные:

- Освоены углубленные технологии программирования на Python, применяемые в реальных проектах;
- Развиты навыки многопоточности, асинхронного и параллельного программирования;

- Сформированы умения работы с сетевыми протоколами, API, клиент-серверной архитектурой и облачными технологиями;
- Сформированы умения проектирования и разработки полноценного веб-и серверного приложения;
- Развиты навыки организации тестирования кода, управления зависимостями и деплоя приложений;
- Сформированы навыки обеспечения безопасности данных и корректного взаимодействия компонентов программы.

Метапредметные:

- Расширены представления о сферах применения Python в промышленной и прикладной разработке;
- Развиты навыки системного мышления при проектировании архитектуры программ;
- Сформированы умения целеполагания и логического подхода к решению комплексных задач;
- Развиты навыки работы с инструментами разработки и контроля качества кода;
- Развиты навыки командной работы при реализации проектов и распределении ролей и задач.

Личностные:

- Воспитана целеустремленность, усидчивость и дисциплина при реализации сложных проектов;
- Воспитано уважение к процессу разработки и результатам труда;
- Сформирована уверенность в собственных силах при решении нестандартных и комплексных задач;
- Сформировано профессиональное отношение к труду и понимание ценности качественной работы.

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ ЦЕНТР ДЕТСКОГО (ЮНОШЕСКОГО) ТЕХНИЧЕСКОГО
ТВОРЧЕСТВА «СТАРТ+» НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

УТВЕРЖДЕНО

Приказ № _____ от _____
Директор ГБУ ДО ЦД (Ю) ТТ «Старт+»
Невского района Санкт-Петербурга
_____ Подобаева О.Г.
« ____ » _____ 20 ____ г.

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК
реализации дополнительной общеразвивающей программы
«Программирование 2 ступень»
Модуль «Промышленное программирование»
на 2026-2027 учебный год

Педагог: Тупицын Пётр Викторович

Год обучения, группа	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Кол-во учебных дней	Кол-во учебных часов	Режим занятий
1 год Группа №1			36	72	144	2 раза в неделю по 2 часа
2 год Группа №1			36	72	144	2 раза в неделю по 2 часа

ПРИНЯТ

Решением Педагогического совета
ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт – Петербурга
Протокол от _____ № _____

УТВЕРЖДЕН

Приказом от _____ № _____
Директор ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт – Петербурга
_____ О.Г. Подобаева

КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН
рабочей программы «Программирование 2 ступень», модуль «Промышленное программирование»
на 2026/2027 учебный год
Группа №1. 1-й год обучения, 144 часа
Педагог дополнительного образования – Тупицын Пётр Викторович

№ п/п	Дата занятия		Раздел	Тема занятия	Кол- во часов	Форма занятия	Способы контроля
	По плану	По факту					
1			Введение в программирован ие	Вводное занятие. Инструктаж по технике безопасности.	2	Аудиторное	опрос
2				Ввод, вывод, переменные и типы данных.	2	Аудиторное	Входящий контроль
3				Условный оператор. Ветвление. Сравнение данных.	2	Аудиторное	наблюдение
4				Вложенный условный оператор. Отступы функций.	2	Аудиторное	наблюдение
5				Сложные условия. Elif.	2	Аудиторное	наблюдение
6				While – цикл с условием.	2	Аудиторное	опрос
7				Цикл с постусловием, бесконечный цикл. Break, continue.	2	Аудиторное	наблюдение
8				For – цикл с параметром.	2	Аудиторное	наблюдение
9				Выбор цикла для алгоритмов. Решение задач.	2	Аудиторное	наблюдение
10			Структура пользовательског о приложения. Часть 1	Значение True/False для программистов. Вся правда об условном операторе.	2	Аудиторное	наблюдение
11				Система флагов, бесконечный цикл.	2	Аудиторное	наблюдение
12				Использование встроенных функций.	2	Аудиторное	опрос

13				Модули и библиотеки стандартного репозитория.	2	Аудиторное	опрос
14				Использование и неиспользование переменных. Множественное присваивание. Задачи «в одну строку».	2	Аудиторное	наблюдение
15				Разработка приложения «Игра в города». Бесконечный цикл.	2	Аудиторное	опрос
16				Разработка приложения «Игра в города». Отладка.	2	Аудиторное	наблюдение
17			Контейнеры данных	Объекты и типы данных. Истинный смысл переменных.	2	Аудиторное	наблюдение
18				Знакомство со списками. Новые возможности цикла for.	2	Аудиторное	опрос
19				Обработка данных в списке. Функции min и max.	2	Аудиторное	наблюдение
20				Строки как контейнер символов. Посимвольная работа со строками.	2	Аудиторное	опрос
21				Обработка данных в строке. Посимвольная работа со строками.	2	Аудиторное	наблюдение
22				Методы списков и строк.	2	Аудиторное	опрос
23				Решение задач на методы списков и строк.	2	Аудиторное	опрос
24				Списочные выражения. Задачи «в одну строку».	2	Аудиторное	наблюдение, опрос
25				Словарь – контейнер для эффективного доступа к данным.	2	Аудиторное	опрос
26				Работа со словарями.	2	Аудиторное	опрос
27				Хакатон «Генераторы текста».	2	Аудиторное	наблюдение
28				Хакатон «Генераторы текста». Презентация решений.	2	Аудиторное	Промежуточный контроль
29			Структура пользовательского приложения.	Знакомство с функциями. Аргументы функции.	2	Аудиторное	
30				Функции, возвращающие значение.	2	Аудиторное	опрос

31			Часть 2	Области видимости. Локальные и глобальные переменные.	2	Аудиторное	опрос
32				Аргументы по умолчанию. Именованные аргументы.	2	Аудиторное	опрос
33				Конструкции *args или **kwargs.	2	Аудиторное	опрос
34				Проектирование программ на основе функции. Функция как объект.	2	Аудиторное	наблюдение
35				Решение задач на применение функций.	2	Аудиторное	опрос
36				Работа с файлами.	2	Аудиторное	наблюдение
37				Библиотека os. Работа с операционной системой в коде программы.	2	Аудиторное	опрос
38				Утилита pyinstaller, исполняемый файл программы.	2	Аудиторное	наблюдение
39				PEP-8 и PEP-20. Правила хорошего тона среди программистов.	2	Аудиторное	опрос
40			Таблицы и СУБД	Матрицы и многомерные списки.	2	Аудиторное	наблюдение
41				Алгоритмы работы с матрицами. Решение задач.	2	Аудиторное	наблюдение
42				Алгоритмы сортировки и транспонирования.	2	Аудиторное	опрос
43				Lambda-функции.	2	Аудиторное	наблюдение
44				Функции высшего порядка.	2	Аудиторное	опрос
45				Регулярные выражения.	2	Аудиторное	наблюдение
46				Работа с csv-таблицами.	2	Аудиторное	опрос
47				Базы данных. СУБД.	2	Аудиторное	опрос
48				Базы данных. Связанные таблицы, сортировка, индексы.	2	Аудиторное	опрос
49				SQLAlchemy.	2	Аудиторное	наблюдение
50				Пользовательская оболочка для программ с базой данных.	2	Аудиторное	наблюдение

51				Безопасность баз данных. Инъекции кода.	2	Аудиторное	наблюдение
52				Хакатон «Покупка билетов».	2	Аудиторное	опрос
53				Хакатон «Покупка билетов». Презентация решений.	2	Аудиторное	наблюдение
54			Объектно-ориентированное программирование	ООП. Классы и объекты.	2	Аудиторное	опрос
55				Конструктор __init__ и методы класса.	2	Аудиторное	опрос
56				Атрибуты и методы: объектные и классовые.	2	Аудиторное	наблюдение
57				Инкапсуляция. Скрытие данных и свойства.	2	Аудиторное	наблюдение
58				Наследование. Базовые и дочерние классы.	2	Аудиторное	опрос
59				Наследование. Функция super(), расширение и изменение поведения.	2	Аудиторное	наблюдение
60				Наследование. Решение задач.	2	Аудиторное	наблюдение
61				Полиморфизм и интерфейсы.	2	Аудиторное	наблюдение
62				Множественное наследование и миксины.	2	Аудиторное	опрос
63				Магические методы классов.	2	Аудиторное	наблюдение
64				Композиция и агрегирование объектов.	2	Аудиторное	опрос
65				Проектная деятельность. Выбор темы, анализ функциональных возможностей.	2	Аудиторное	наблюдение
66				Проектная деятельность. Проектирование архитектуры.	2	Аудиторное	наблюдение
67				Проектная деятельность. Разработка классов.	2	Аудиторное	опрос
68				Проектная деятельность. Работа с СУБД.	2	Аудиторное	наблюдение
69				Проектная деятельность. Введение в многопоточность.	2	Аудиторное	наблюдение
70				Проектная деятельность. Интерфейсы.	2	Аудиторное	опрос

71			Итоговое занятие	Подготовка к презентации проектов. Отладка.	2	Аудиторное	Промежуточный контроль
72				Презентация проектов. Подведение итогов. Вручение грамот.	2	Аудиторное	Фронтальный/анализ
				Итого	144		

ПРИНЯТ
Решением Педагогического совета
ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт – Петербурга
Протокол от _____ № _____

УТВЕРЖДЕН
Приказом от _____ № _____
Директор ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт – Петербурга
_____ О.Г. Подобаева

КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН
рабочей программы «Программирование 2 ступень», модуль «Промышленное программирование»
на 2025/2026 учебный год
Группа №1. 2-й год обучения, 144 часа
Педагог дополнительного образования – Тупицын Пётр Викторович

№ п/п	Дата занятия		Раздел	Тема занятия	Кол- во часов	Форма занятия	Способы контроля
	По плану	По факту					
1			Углубленные технологии программирован ия. Облачные вычисления	Вводное занятие. Инструктаж по технике безопасности.	2	Аудиторное	опрос
2				Логирование программ. Модуль logging.	2	Аудиторное	Входящий контроль
3				Тестирование кода. Модули unittest и pytest.	2	Аудиторное	наблюдение
4				Виртуальное окружение и управление зависимостями.	2	Аудиторное	наблюдение
5				Потоки. Основы многопоточности в threading.	2	Аудиторное	наблюдение
6				Потоки. Многопроцессорность. Запуск программы на нескольких ядрах процессора.	2	Аудиторное	опрос
7				Потоки. Асинхронное программирование.	2	Аудиторное	наблюдение
8				Потоки. Параллельные вычисления.	2	Аудиторное	наблюдение
9				Сокеты. Низкоуровневое сетевое взаимодействие.	2	Аудиторное	наблюдение

10				Сокеты. Безопасность передачи данных.	2	Аудиторное	наблюдение
11				Сокеты. Реализация простого клиента-сервера.	2	Аудиторное	наблюдение
12				Сокеты. Реализация асимметричной криптосистемы.	2	Аудиторное	опрос
13				Введение в API. Протокол http. Структура запросов и ответов.	2	Аудиторное	опрос
14				Введение в API. Запросы по протоколу http. Модуль requests.	2	Аудиторное	наблюдение
15				Введение в API. REST API.	2	Аудиторное	опрос
16				Введение в API. Модуль flask. Простейший http-сервер.	2	Аудиторное	наблюдение
17				Облачные вычисления. Обмен данными между клиентом и сервером с помощью двух flask-поток.	2	Аудиторное	наблюдение
18				Облачные вычисления. Проектирование серверного приложения.	2	Аудиторное	опрос
19				Облачные вычисления. Проблема очереди запросов.	2	Аудиторное	наблюдение
20				Облачные вычисления. Распределение запросов вручную.	2	Аудиторное	опрос
21				Облачные вычисления. Redis-сервер для автоматического распределения запросов.	2	Аудиторное	наблюдение
22			Сетевое администрирован ие для программистов	Сетевые модели и протоколы.	2	Аудиторное	опрос
23				Адресация и маршрутизация.	2	Аудиторное	опрос
24				Настройка локальной сети и работа с удаленным сервером.	2	Аудиторное	наблюдение, опрос
25				Размещение серверного приложения на сервере.	2	Аудиторное	опрос

26				Размещение серверного приложения на сервере. Проброс портов и реверс-проксирование.	2	Аудиторное	опрос
27				Безопасность серверного приложения. Шифрование и SSL-сертификат.	2	Аудиторное	наблюдение
28				Безопасность серверного приложения. Контроль запросов по IP адресу.	2	Аудиторное	Промежуточный контроль
29				Введение в веб-технологии.	2	Аудиторное	
30				HTML. Структура документа, теги.	2	Аудиторное	опрос
31				HTML. Списки, таблицы, формы.	2	Аудиторное	опрос
32				HTML. Мультимедиа и семантические теги.	2	Аудиторное	опрос
33				CSS. Основы стилей. Работа с цветом и шрифтами.	2	Аудиторное	опрос
34				CSS. Блочная модель и позиционирование.	2	Аудиторное	наблюдение
35				CSS. Flexbox и Grid.	2	Аудиторное	опрос
36			Основы фронтенд-разработки	CSS. Адаптивная верстка.	2	Аудиторное	наблюдение
37				Bootstrap. Облегченное создание адаптивной верстки.	2	Аудиторное	опрос
38				Введение в JavaScript. Переменные, типы данных, операции.	2	Аудиторное	наблюдение
39				JS. Условия и циклы.	2	Аудиторное	опрос
40				JS. Функции.	2	Аудиторное	наблюдение
41				JS. DOM: доступ к элементам и изменение содержимого.	2	Аудиторное	наблюдение
42				JS. Обработка событий.	2	Аудиторное	опрос
43				JS. Формы и валидация данных.	2	Аудиторное	наблюдение

44				JS. Работа с объектами.	2	Аудиторное	опрос
45				JS. Асинхронное программирование. fetch и setTimeout.	2	Аудиторное	наблюдение
46				JS. AJAX и работа с API.	2	Аудиторное	опрос
47				JS. Библиотека jQuery.	2	Аудиторное	опрос
48				Разработка интерактивной страницы.	2	Аудиторное	опрос
49				Разработка интерактивной страницы. Подключение к flask-серверу.	2	Аудиторное	наблюдение
50			Основы бэкенд-разработки	Django – ключевой фреймворк для серверной разработки на python.	2	Аудиторное	наблюдение
51				Структура проекта Django. Создание первого приложения.	2	Аудиторное	наблюдение
52				Настройка маршрутов.	2	Аудиторное	опрос
53				Шаблоны Django. Введение в шаблоны.	2	Аудиторное	наблюдение
54				Шаблоны Django. Наследование и включения.	2	Аудиторное	опрос
55				Формы и обработка данных.	2	Аудиторное	опрос
56				Модели и ORM Django.	2	Аудиторное	наблюдение
57				Создание базы данных и миграции.	2	Аудиторное	наблюдение
58				Панель администратора Django.	2	Аудиторное	опрос
59				CRUD-операции в Django.	2	Аудиторное	наблюдение
60				Связи между моделями.	2	Аудиторное	наблюдение
61				Работа с пользователями: регистрация, авторизация.	2	Аудиторное	наблюдение
62				Управление доступом и права пользователей.	2	Аудиторное	опрос

63				Загрузка и хранение файлов.	2	Аудиторное	наблюдение
64				Проектная деятельность. Проектирование веб-мессенджера с хранением данных на сервере.	2	Аудиторное	опрос
65				Проектная деятельность. Разработка шаблонов веб-страниц.	2	Аудиторное	наблюдение
66				Проектная деятельность. Разработка серверного приложения.	2	Аудиторное	наблюдение
67				Проектная деятельность. Разработка асинхронных потоков.	2	Аудиторное	опрос
68				Проектная деятельность. Отладка асинхронных потоков.	2	Аудиторное	наблюдение
69				Проектная деятельность. Подготовка сервера.	2	Аудиторное	наблюдение
70				Проектная деятельность. Размещение приложения на сервере.	2	Аудиторное	опрос
71			Итоговое занятие	Подготовка к презентации проектов. Отладка.	2	Аудиторное	Итоговый контроль
72				Презентация проектов. Подведение итогов. Вручение грамот.	2	Аудиторное	Фронтальный/анализ
				Итого	144		

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ ЦЕНТР ДЕТСКОГО (ЮНОШЕСКОГО) ТЕХНИЧЕСКОГО
ТВОРЧЕСТВА «СТАРТ+» НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

ПРИНЯТА

Решением Педагогического совета
ГБУ ДО ЦД(Ю) ТТ «Старт+»
Невского района Санкт-Петербурга
Протокол от _____ № _____

УТВЕРЖДЕНА

Приказом от _____ № _____
Директор ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт-Петербурга
_____ Подобаева О.Г

**Рабочая программа
к дополнительной общеразвивающей программе
«ПРОГРАММИРОВАНИЕ 2 СТУПЕНЬ»
МОДУЛЬ «НЕЙРОННЫЕ СЕТИ»**

1 год обучения
Возраст обучающихся 13-17 лет

Разработчик: Тупицын Пётр Викторович,
педагог дополнительного образования
высшей квалификационной категории

2026 г.

ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА 1 ГОДА ОБУЧЕНИЯ

Группу составляют учащиеся в возрасте 13-17 лет, имеющие хорошие навыки работы с персональным компьютером, интересующиеся сферой информационных технологий, искусственным интеллектом и математикой, в количестве не менее 15 человек.

Занятия объединения по Программе «Программирование 2 ступень» модуль «Нейронные сети» 1 года обучения проводятся 2 раза в неделю в очном формате, длительность каждого занятия – 2 учебных часа с 10-минутным перерывом. Длительность освоения Программы - 144 часа за 1 учебный год. Каждое занятие включает в себя теоретическую часть – краткую беседу по каждой теме с демонстрацией излагаемого материала и практическую – основную, в которой дети выполняют задания под руководством педагога.

Особенности организации образовательного процесса по Программе заключаются в формировании прочной базы для профессиональной работы с методами искусственного интеллекта и нейросетевыми технологиями. Основной упор сделан на развитие математического и алгоритмического мышления, освоение языка программирования Python в контексте задач обработки данных и подготовки моделей машинного обучения.

Важной характеристикой обучения является последовательное построение знаний: учащиеся получают системное представление о структуре данных, алгоритмах их обработки и математических инструментах, необходимых для построения моделей. Акцент сделан на интеграцию теоретических знаний с практическими навыками работы с библиотеками Python для анализа данных, линейной алгебры, статистики и визуализации, что обеспечивает понимание принципов построения моделей машинного обучения.

Особенностью процесса является также раннее знакомство с нейросетевыми технологиями, которое служит мотивирующим элементом: даже в первый год учащиеся получают практический опыт работы с нейронными сетями и компьютерным зрением, что формирует представление о конечной цели обучения и о реальных областях применения полученных знаний. Такой подход позволяет выстраивать образовательную траекторию от фундаментальных концепций к сложным прикладным задачам второго года обучения, обеспечивая прочное понимание основ и готовность к профессиональной практике.

ЗАДАЧИ 1 ГОДА ОБУЧЕНИЯ

Обучающие

- Развить навыки использования базовых конструкций языка Python для разработки алгоритмов обработки данных;
- Развить умение организовывать и структурировать данные с помощью списков, словарей, массивов и пользовательских функций;
- Развить понимание принципов объектно-ориентированного программирования и создание собственных классов для моделирования объектов;
- Развить навыки работы с библиотеками Python для численных расчетов, линейной алгебры и статистической обработки данных;

- Развить умение создавать и анализировать простые модели машинного обучения, включая методы регрессии и классификации;
- Развить навыки визуализации данных для анализа и интерпретации результатов;
- Развить понимание базовых принципов работы нейронных сетей и технологий компьютерного зрения;
- Развить системное мышление при построении алгоритмов искусственного интеллекта и подготовке данных для обучения моделей.

Развивающие

- Расширить представления о сферах применения информационных технологий;
- Развить навыки работы с информацией разного вида на компьютере;
- Развить навыки целеполагания и логического подхода к решению поставленной задачи;
- Сформировать способность к саморазвитию, самообразованию;
- Развить умение работать в команде, грамотно распределять роли и задачи между участниками команды.

Воспитательные

- Воспитать целеустремленность, усидчивость и терпение в достижении творческих результатов;
- Воспитать уважение к труду и его результатам;
- Сформировать уверенность в своих силах.
- Сформировать позитивное отношение к труду.

СОДЕРЖАНИЕ ПРОГРАММЫ 1 ГОДА ОБУЧЕНИЯ

1. Введение в технологию искусственного интеллекта и основы программирования на Python

Раздел посвящён формированию базовой платформы знаний, необходимых для дальнейшего освоения методов искусственного интеллекта и нейросетевых технологий. Основное внимание уделено **пониманию ключевых концепций ИИ**, знакомству с профессиями в сфере искусственного интеллекта, **освоению языка программирования Python** и среды разработки Jupyter Lab, а также основам организации алгоритмического мышления.

В процессе изучения раздела формируется представление о том, как структурировать данные и логику программ, как строить линейные и условные алгоритмы, и как использовать каскадные условные конструкции для построения более сложных решений. Практическая часть направлена на закрепление этих знаний через реализацию простых алгоритмов и программных решений, что создаёт прочную базу для последующего изучения алгоритмов машинного обучения и нейросетевых моделей.

Теория:

Концепции искусственного интеллекта и основные направления развития технологий ИИ. Профессии в сфере ИИ и возможности профессиональной деятельности. Основы языка Python, структура программного кода и базовые синтаксические единицы. Среда разработки Jupyter Lab и дистрибутив Anaconda. Линейные алгоритмы и

последовательность выполнения команд. Условные конструкции и каскадные операторы, принципы ветвления и организации логики программы.

Практика:

Создание и выполнение первых Python-программ в Jupyter Lab. Использование переменных, типов данных и базовых операций. Реализация простых линейных алгоритмов. Применение условных операторов и каскадных конструкций для решения задач. Организация диалога с пользователем через ввод и вывод данных. Практика пошаговой отладки и анализа работы алгоритмов.

2. Создание алгоритмов искусственного интеллекта на Python

Раздел сосредоточен на изучении и отработке **методов организации и обработки данных**, которые станут основой для последующих алгоритмов машинного обучения. Основное внимание уделено работе со **структурами данных** — **строками, списками, словарями, файлами** — и созданию функций и объектов, обеспечивающих системное хранение и обработку информации.

В процессе изучения формируется понимание, как данные могут быть эффективно организованы и подготовлены для решения аналитических и алгоритмических задач. Практическая деятельность направлена на освоение навыков работы с базовыми структурами данных, проектирования функций и классов, а также на умение реализовывать алгоритмы обработки данных, которые впоследствии используются при построении моделей машинного обучения. Такой подход обеспечивает прочную основу для дальнейшего изучения математических и программных методов ИИ.

Теория:

Строки, циклы и методы обработки текстовых данных. Типы данных list и dict, операции над структурами данных. Организация пользовательских функций и возвращение логических констант. Работа с файлами и их обработка в Python. Основы объектно-ориентированного программирования: классы, объекты, конструкторы и методы. Перегрузка операторов и создание собственных объектов для моделирования элементов искусственного интеллекта.

Практика:

Реализация операций со строками и списками, использование циклов с параметром и условием. Создание пользовательских функций для обработки данных. Чтение и запись файлов, обработка их содержимого. Проектирование и создание классов, использование конструкторов и методов. Перегрузка операторов и построение объектов для моделирования элементов нейросетевых систем.

3. Машинное обучение как методология искусственного обучения

Раздел сосредоточен на изучении принципов машинного обучения и методах построения алгоритмов для анализа данных. Основное внимание уделено формированию системного понимания того, как подготовленные структуры данных и алгоритмы **могут быть использованы для решения задач регрессии, классификации и выбора оптимальных моделей**.

Особое место в разделе занимает углублённое изучение **теории вероятностей и статистических методов**, необходимых для корректного анализа данных и оценки качества моделей. Это обеспечивает прочную математическую базу, которая становится фундаментом для построения сложных моделей искусственного интеллекта и нейросетевых систем в последующих разделах программы.

В целом, раздел формирует представление о логике машинного обучения, принципах выбора алгоритмов и методов анализа данных, создавая основу для последующего глубокого изучения нейронных сетей и их применения в реальных прикладных задачах искусственного интеллекта.

Теория:

Основные концепции машинного обучения: регрессия, классификация, пайплайн обучения. Метрики оценки качества моделей. Математические основы машинного обучения: функции анализа, работа с векторами и матрицами, основы теории вероятностей и статистики. Алгоритмы регрессии, логистическая регрессия, решающие деревья и композиции алгоритмов. Построение пайплайна выбора модели и анализ результатов.

Практика:

Построение моделей машинного обучения на основе подготовленных данных. Применение методов ближайших соседей, линейной и логистической регрессии. Реализация алгоритмов решающих деревьев и композиций моделей. Расчет и анализ метрик качества моделей, выполнение кросс-валидации. Применение математических функций и операций с матрицами для реализации алгоритмов. Тестирование и отладка моделей машинного обучения.

4. Нейронные сети и компьютерное зрение

Раздел посвящён ознакомлению с базовыми принципами нейросетевых технологий и их применением в задачах компьютерного зрения. Основное внимание уделено **формированию понимания структуры нейронных сетей**, процесса их обучения и принципов работы с изображениями как с данными для моделей.

Раздел служит мотивационным и подготовительным этапом: учащиеся получают **первые практические навыки работы с библиотекой PyTorch**, строят простые модели и наблюдают процесс их обучения. Это создаёт контекст для применения ранее изученных алгоритмов и структур данных и обеспечивает понимание конечной цели изучения машинного обучения и нейросетей.

Особое внимание уделено знакомству с **архитектурами сверточных нейронных сетей (CNN) и концепцией Transfer Learning**, что позволяет формировать общее представление о подходах к решению задач классификации изображений. Раздел направлен на формирование базового опыта работы с нейросетями и понимание их места в более широкой системе искусственного интеллекта

Теория:

Основные концепции нейронных сетей и принципы их функционирования. Архитектуры сверточных нейронных сетей (CNN). Основы Transfer Learning и

переиспользования моделей. Принципы обучения нейронных сетей и подготовка данных для обучения. Основы компьютерного зрения и обработки изображений.

Практика:

Создание и обучение простых нейронных сетей в библиотеке PyTorch. Работа с изображениями, подготовка данных для моделей. Классификация изображений с использованием простых CNN. Ознакомление с Transfer Learning и использованием предварительно обученных моделей.

5. Продвинутое обучение алгоритмов машинного обучения

Раздел посвящён расширению и углублению знаний о методах машинного обучения, с акцентом на **более сложные алгоритмы классификации, ансамблевые методы и работу с текстовыми и визуальными данными**. Основное внимание уделено формированию понимания того, как комбинировать различные алгоритмы для повышения качества моделей и как внедрять их в прикладные задачи.

В разделе подробно рассматриваются **методы работы с текстовыми данными**, включая классификацию текстов и эмбединги слов, а также задачи обработки изображений с применением нейронных сетей. Особое внимание уделено вопросам **практического внедрения моделей**, включая оценку производительности и ограничений вычислительных ресурсов.

Раздел служит связующим звеном между изучением базовых моделей машинного обучения и последующим углубленным изучением нейронных сетей во втором году программы. Он формирует представление о комплексных подходах к решению задач искусственного интеллекта и подготавливает к самостоятельной работе с реальными данными и проектами.

Теория:

Методы классификации текстов, логистическая регрессия для текстовых данных. Алгоритмы SVM и градиентного бустинга, принципы ансамблевых моделей. Задачи компьютерного зрения и нейронные сети для обработки изображений. Вопросы внедрения моделей машинного обучения в продукт, оценка вычислительных ресурсов. Основы обработки естественного языка, эмбединги слов и языковое моделирование.

Практика:

Применение методов классификации текстов с использованием эмбедингов слов. Реализация алгоритмов SVM и градиентного бустинга. Решение задач компьютерного зрения с использованием нейросетевых моделей. Построение и тестирование языковых моделей. Внедрение и проверка моделей машинного обучения в практических условиях с учётом ограничений ресурсов.

6. Итоговое занятие

Теория: Подведение итогов обучения. Анализ достижений.

Практика: Защита проектов. Коллективная рефлексия.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ 1 ГОДА ОБУЧЕНИЯ

После обучения по Программе учащиеся получают следующие результаты:

Предметные:

- Освоены базовые конструкции языка Python в контексте разработки алгоритмов для обработки данных;
- Сформированы навыки организации и структурирования данных с использованием списков, словарей, массивов и пользовательских функций;
- Освоены основы объектно-ориентированного программирования и разработка собственных классов для моделирования объектов;
- Развиты навыки работы с библиотеками Python для численных расчетов, линейной алгебры и статистической обработки данных;
- Сформированы умения создания и анализа простых моделей машинного обучения, включая методы регрессии и классификации;
- Освоены инструменты визуализации данных для анализа и интерпретации результатов;
- Получен базовый опыт работы с нейронными сетями и технологиями компьютерного зрения, обеспечивающий мотивацию и понимание применения изучаемых методов;
- Сформированы навыки системного подхода к построению алгоритмов искусственного интеллекта и подготовки данных для дальнейшего обучения.

Метапредметные:

- Развиты представления о сферах применения информационных технологий;
- Развиты навыки работы с информацией разного вида на компьютере;
- Развиты навыки целеполагания и логического подхода к решению поставленной задачи;
- Сформированы способность к саморазвитию, самообразованию;
- Развиты умение работать в команде разработчиков, грамотно распределять роли и задачи между участниками команды.

Личностные:

- Сформировано ответственное отношение к труду, понимание важности соблюдения сроков и качества выполнения.
- Развита настойчивость в достижении цели, готовность преодолевать трудности в процессе работы.
- Учащиеся демонстрируют способность воспринимать конструктивную критику и высказывать мнение корректно и аргументированно.
- Учащиеся проявляют интерес к сфере технического творчества.
- Учащиеся осознают значимость участия в общественно полезной и профессионально ориентированной проектной деятельности.

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ ЦЕНТР ДЕТСКОГО (ЮНОШЕСКОГО) ТЕХНИЧЕСКОГО
ТВОРЧЕСТВА «СТАРТ+» НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

ПРИНЯТА

Решением Педагогического совета
ГБУ ДО ЦД(Ю) ТТ «Старт+»
Невского района Санкт-Петербурга
Протокол от _____ № _____

УТВЕРЖДЕНА

Приказом от _____ № _____
Директор ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт-Петербурга
_____ Подобаева О.Г

**Рабочая программа
к дополнительной общеразвивающей программе
«ПРОГРАММИРОВАНИЕ 2 СТУПЕНЬ»
МОДУЛЬ «НЕЙРОННЫЕ СЕТИ»**

2 год обучения
Возраст обучающихся 13-17 лет

Разработчик: Тупицын Пётр Викторович,
педагог дополнительного образования
высшей квалификационной категории

2026 г.

ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА 2 ГОДА ОБУЧЕНИЯ

Группу составляют учащиеся в возрасте 13-17 лет, успешно завершившие обучение по программе 1 года обучения, имеющие хорошие навыки работы с персональным компьютером, интересующиеся сферой информационных технологий, искусственным интеллектом и математикой, в количестве не менее 12 человек.

Занятия объединения по Программе «Программирование 2 ступень» модуль «Нейронные сети» 2 года обучения проводятся 2 раза в неделю в очном формате, длительность каждого занятия – 2 учебных часа с 10-минутным перерывом. Длительность освоения Программы - 144 часа за один учебный год. Каждое занятие включает в себя теоретическую часть – краткую беседу по каждой теме с демонстрацией излагаемого материала и практическую – основную, в которой дети выполняют задания под руководством педагога.

Особенности организации образовательного процесса по Программе заключаются в том, что обучение направлено на углубление понимания принципов нейросетевых технологий и их применения в практических задачах. Основное внимание уделено формированию навыков проектирования и анализа сложных моделей, планированию архитектуры нейронных сетей и оценке эффективности алгоритмов в разных сценариях.

Значимым аспектом является развитие способности работать с большими и разнообразными данными, преобразовывать их в подходящие форматы для обучения моделей и строить эффективные решения для задач компьютерного зрения, обработки текста и генерации данных. Обучение способствует развитию логического мышления, системного подхода к решению задач и понимания связи между математическими основами и практическими результатами.

Для развития навыков самостоятельной развертки приложений на сервере учащиеся изучают раздел «Сетевое администрирование для программистов», который является общим для второго года обучения модулей «Промышленное программирование» и «Нейронные сети» ДОП «Программирование 2 ступень».

Программа формирует навыки самостоятельного проектирования и анализа нейросетевых решений, критической оценки качества моделей, принятия архитектурных решений с учётом ограничений ресурсов и требований к результату. Особое внимание уделено подготовке к самостоятельной профессиональной деятельности, развитию исследовательских и аналитических компетенций, а также способности применять знания в комплексных проектах искусственного интеллекта.

ЗАДАЧИ 2 ГОДА ОБУЧЕНИЯ

Обучающие

- Развить навыки проектирования и построения нейронных сетей различной сложности;
- Развить умение подбирать архитектуру сети и настраивать гиперпараметры для решения конкретных задач;
- Развить навыки подготовки и обработки данных для обучения моделей;

- Развить способность анализировать результаты работы моделей и оценивать их качество;
- Развить умение создавать и интегрировать модели в прикладные решения и сервисы;
- Развить понимание взаимодействия различных компонентов нейросетевых систем и принципов их оптимизации;
- Развить способность применять математические и статистические методы для корректного обучения и оценки моделей;
- Развить навыки разработки мультимодальных решений и работы с комплексными архитектурами нейросетей.

Развивающие

- Расширить представления о сферах применения Python в промышленной и прикладной разработке;
- Развить навыки системного мышления при проектировании архитектуры программ;
- Сформировать способность к целеполаганию и логическому подходу к решению комплексных задач;
- Развить умение работать с реальными инструментами разработки, контролировать качество кода;
- Развить навыки командной работы при реализации проектов, распределении ролей и задач.

Воспитательные

- Воспитать целеустремленность, усидчивость и дисциплину при реализации сложных проектов;
- Воспитать уважение к процессу разработки и результатам труда;
- Сформировать уверенность в собственных силах при решении нестандартных и комплексных задач;
- Сформировать профессиональное отношение к труду и понимание ценности качественной работы.

СОДЕРЖАНИЕ ПРОГРАММЫ 2 ГОДА ОБУЧЕНИЯ

1. Библиотека Keras

Раздел является логическим продолжением изучения нейросетевых технологий, начатого в первом году, и направлен на углубление практических навыков **построения и обучения моделей с использованием библиотеки Keras**. Основное внимание уделено закреплению знаний о **структуре нейронных сетей, обработке данных и организации процесса обучения** через Keras, включая функции потерь, метрики и callback-функции.

В процессе изучения раздела формируется умение строить полные рабочие модели в Keras, анализировать их поведение и корректировать архитектуру для достижения лучших результатов. При этом расширяется понимание того, как применять изученные в прошлом году методы машинного обучения и математические основы для решения конкретных прикладных задач с изображениями и текстами.

Теория:

Принципы построения и обучения нейронных сетей с использованием Keras. Структура модели, функции потерь и метрики. Организация процесса обучения и обратной связи через callback-функции. Настройка слоёв и оптимизаторов. Сравнение различных подходов к реализации моделей на Keras и других фреймворках.

Практика:

Создание простой модели нейронной сети в Keras. Компиляция и обучение модели. Сохранение и загрузка моделей. Использование callback-функций и разработка собственных классов обратной связи. Классификация изображений и текстов с помощью Keras. Настройка параметров сети для повышения качества работы модели.

2. Инструменты и библиотеки глубокого обучения

Раздел сосредоточен на освоении современных инструментов и библиотек, которые позволяют эффективно работать с **предобученными моделями, дообучать их под конкретные задачи и форматы данных, а также интегрировать новые компоненты**. Особое внимание уделено пониманию принципов fine-tuning, заморозки слоёв, настройки гиперпараметров и адаптации архитектур под различные типы данных.

В процессе изучения раздела формируются навыки критической оценки существующих решений, выбора подходящей модели для задачи, адаптации её к конкретным данным и расширения функциональности через кастомные слои и функции. Обучение направлено на подготовку к профессиональной работе с современными инструментами глубокого обучения и позволяет применять знания для создания качественных моделей в различных сферах.

Теория:

Принципы работы с предобученными моделями. Fine-tuning и заморозка слоёв. Интеграция кастомных слоёв и функций. Выбор оптимальной модели для задачи. Основы адаптации архитектур под конкретные данные и задачи.

Практика:

Использование предобученных моделей для классификации и генерации данных. Дообучение моделей под собственный датасет. Заморозка и частичная разморозка слоёв. Настройка архитектуры модели и оптимизация гиперпараметров. Сравнение производительности локальных моделей GPT и аналогов.

3. Сетевое администрирование для программистов

Раздел посвящён освоению **принципов работы компьютерных сетей** и организации серверной инфраструктуры для приложений. Рассматриваются сетевые модели и протоколы, способы адресации и маршрутизации данных, методы настройки локальных сетей и удалённых серверов, а также принципы обеспечения безопасности серверных приложений.

В процессе раздела формируется понимание того, как взаимодействуют различные уровни сетевой архитектуры, как проектировать серверное приложение с учётом **безопасности и доступности**, и какие технологии позволяют управлять **потоками запросов**. Практическая деятельность направлена на закрепление знаний через настройку

сетевого окружения, развертывание серверного приложения и обеспечение его корректной и безопасной работы.

Важность раздела обусловлена необходимостью для программиста понимать работу платформ, обеспечивающих работу созданных серверных программ, а также уметь определить необходимые настройки платформы для работы программы.

Теория:

Модели компьютерных сетей и основные сетевые протоколы. Принципы адресации и маршрутизации данных. Организация локальных и удалённых сетей. Основы размещения серверного приложения на сервере. Методы обеспечения безопасности серверных приложений, включая шифрование и контроль доступа. Управление потоками запросов и организация стабильной работы сервера.

Практика:

Настройка локальной сети и подключение к удалённому серверу. Размещение серверного приложения на сервере и проверка доступности. Проброс портов и настройка реверс-прокси. Настройка шифрования и SSL-сертификатов. Контроль запросов по IP-адресу. Отладка и тестирование работы серверного приложения.

4. Инфраструктура и развертывание нейронных сетей

Раздел продолжает изучение сетевого администрирования для программистов и посвящён освоению принципов подготовки и развертывания нейросетевых моделей в реальной инфраструктуре. Учащиеся изучают особенности работы моделей на сервере, взаимодействие с вычислительными ресурсами, такими как CPU и GPU, а также организацию потоков данных и построение интерфейсов для взаимодействия с пользователем. Это позволяет видеть нейросеть не только как инструмент для экспериментов, но и как полноценный компонент прикладного решения, интегрируемый в сервисы и приложения.

В процессе работы развивается понимание того, как обеспечить корректное выполнение модели в разных условиях, оценить требования к вычислительным ресурсам и оптимизировать производительность системы. Акцент делается на построении устойчивого взаимодействия между моделью и внешними приложениями через API, а также на создании удобных веб-интерфейсов, которые позволяют управлять моделями и визуализировать результаты работы.

Особое внимание уделяется проектированию систем, где нейросеть становится частью сервиса, способного обрабатывать запросы пользователей в режиме реального времени. При этом рассматриваются методы контейнеризации, организация REST API, интеграция моделей в веб-интерфейсы и оптимизация работы с серверными вычислительными ресурсами.

Теория:

Принципы развертывания моделей нейронных сетей на сервере. Взаимодействие с вычислительными ресурсами CPU и GPU. Организация потоков данных. Построение REST API и интеграция модели в веб-интерфейсы. Контейнеризация и основы обеспечения масштабируемости.

Практика:

Упаковка модели в Docker-контейнер. Создание API для модели. Развертывание модели как сервиса. Настройка вычислительных ресурсов. Создание веб-интерфейсов с библиотеками для быстрой разработки интерфейсов (Gradio). Тестирование работы модели в реальных условиях.

5. Основы проектирования нейронных сетей

Раздел посвящён углубленному изучению архитектуры базовых типов нейронных сетей (полносвязная, сверточная) и принципов проектирования моделей под конкретные задачи. Основное внимание уделяется пониманию структуры сети, роли гиперпараметров, функций активации и организации слоёв, что позволяет строить модели с учётом требуемого уровня сложности и качества предсказаний.

Важной частью изучения является комбинирование различных подходов: сверточные и полносвязные слои, остаточные связи, разветвляющиеся архитектуры и многовыходные сети. Это даёт возможность экспериментировать с архитектурами, адаптировать их под данные и задачи, а также оценивать влияние изменений на поведение модели, а также формирует базу для изучения архитектуры сложных типов нейронных сетей. При изучении акцентируется внимание на проектировании архитектур, обеспечивающих баланс между сложностью и эффективностью, а также на стратегиях предотвращения переобучения

Теория:

Структура нейронной сети и роль гиперпараметров. Функции активации, нейроны и слои. Сборка архитектуры из модулей. Сверточные сети и остаточные связи. Разветвляющиеся и многовыходные архитектуры. Комбинирование подходов для повышения эффективности модели.

Практика:

Создание и настройка собственной архитектуры сети. Применение функций активации и выбор гиперпараметров. Построение сверточных и полносвязных сетей. Использование остаточных связей и разветвляющихся архитектур. Эксперименты с многовыходными моделями и комбинациями различных подходов для конкретных задач.

6. Рекуррентные сети и последовательности

Раздел посвящён изучению рекуррентных нейронных сетей, позволяющих моделировать последовательные данные и учитывать контекст прошлых элементов при построении предсказаний. В центре внимания находятся простые RNN, блоки LSTM и GRU, а также механизм Attention, который помогает выделять наиболее значимые части последовательностей.

Особое внимание уделяется проектированию архитектур для работы с последовательностями, генерации данных и создания моделей с памятью. Рассматриваются подходы к комбинированию рекуррентных сетей с другими типами архитектур для решения сложных задач, таких как обработка текста и временных рядов. Раздел формирует навыки понимания внутренней логики работы сети и способов управления потоком информации внутри модели.

Теория:

Принципы работы рекуррентных сетей. RNN и их особенности. Блоки LSTM и GRU, механизм памяти. Attention для выбора значимых элементов последовательности. Комбинация рекуррентных сетей с другими архитектурами для генерации данных.

Практика:

Построение простой RNN для предсказания числовых последовательностей. Реализация LSTM и GRU для задач с длинными зависимостями. Использование Attention для выделения ключевых элементов последовательности. Проектирование архитектур для генерации текста и других последовательных данных. Эксперименты с комбинированными сетями для улучшения качества предсказаний.

7. Архитектуры генеративных сетей

Раздел посвящён изучению методов генерации данных с помощью нейронных сетей, включая автоэнкодеры, вариационные автоэнкодеры (VAE), генеративные состязательные сети (GAN) и модели на основе Diffusion. Основное внимание уделяется пониманию принципа работы каждой архитектуры, способов кодирования и восстановления данных, а также построению моделей, способных создавать новые объекты или данные по заданным условиям.

Особое внимание уделяется экспериментальной работе с генеративными архитектурами, оценке качества созданных объектов, поиску баланса между сложностью модели и её способностью к генерализации. Рассматриваются мультимодальные сети, способные работать с различными типами данных одновременно, а также многовыходные сети, что позволяет проектировать гибкие и адаптивные системы генерации.

Теория:

Принципы работы автоэнкодеров и вариационных автоэнкодеров. Состязательные генеративные сети (GAN) и идея соревнования двух моделей. Модели Diffusion и постепенное восстановление данных. Баланс сложности модели и переобучения. Мультимодальные и многовыходные архитектуры.

Практика:

Проектирование и обучение автоэнкодеров для восстановления изображений. Реализация VAE для генерации вариативных данных. Построение GAN для генерации изображений и игровых объектов. Использование Diffusion для постепенного создания данных из шума. Создание мультимодальных и многовыходных сетей для комплексных задач генерации. Эксперименты с балансом сложности моделей и контролем качества генерации.

8. Введение в архитектуру трансформеров

Раздел посвящён изучению архитектуры трансформеров, которая обеспечивает современные возможности обработки текста, изображений и мультимедийных данных. Основное внимание уделяется принципам работы encoder-decoder моделей, decoder-only моделей и особенностям архитектуры GPT.

В рамках раздела рассматривается переход от рекуррентных сетей к трансформерам, объясняется роль внимания (Attention) и механизмы, позволяющие сети эффективно учитывать контекст. Учащиеся исследуют применение трансформеров к различным типам данных и осваивают подходы дообучения предобученных моделей для конкретных задач.

Особое место занимает проектная деятельность: учащиеся создают итоговый проект - мультимодальную GPT-like нейросеть. Проект может выполняться с нуля, либо через дообучение открытой модели под конкретную задачу, либо с использованием заморозки и переписывания слоёв предобученной модели. Это позволяет закрепить знания о трансформерах на практике и применить навыки проектирования сложной нейросетевой системы.

Теория:

Принципы работы трансформеров. Encoder-decoder и decoder-only архитектуры. Механизм внимания (Attention) и управление контекстом. Применение трансформеров к тексту и мультимедиа. Подходы дообучения и модификации предобученных моделей.

Практика:

Разработка упрощённой GPT-архитектуры. Дообучение предобученной модели под конкретную задачу. Заморозка и частичная модификация слоёв открытой модели. Создание мультимодальной модели для работы с текстом и изображениями. Интеграция итогового проекта в практическое решение и тестирование его работы.

9. Итоговое занятие

Теория: Подведение итогов обучения. Анализ достижений.

Практика: Защита проектов. Коллективная рефлексия.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ 2 ГОДА ОБУЧЕНИЯ

После обучения по Программе учащиеся получают следующие результаты:

Предметные:

- Развита способность проектирования и построения нейронных сетей различной сложности;
- Развита способность подбирать архитектуру сети и настраивать гиперпараметры для решения конкретных задач;
- Развита способность подготовки и обработки данных для обучения моделей;
- Развита способность анализировать результаты работы моделей и оценивать их качество;
- Развита способность создавать и интегрировать модели в прикладные решения и сервисы;
- Развита способность взаимодействия различных компонентов нейросетевых систем и принципов их оптимизации;
- Развита способность применять математические и статистические методы для корректного обучения и оценки моделей;
- Развита способность разработки мультимодальных решений и работы с комплексными архитектурами нейросетей.

Метапредметные:

- Расширены представления о сферах применения Python в промышленной и прикладной разработке;
- Развиты навыки системного мышления при проектировании архитектуры программ;
- Сформированы умения целеполагания и логического подхода к решению комплексных задач;
- Развиты навыки работы с инструментами разработки и контроля качества кода;
- Развиты навыки командной работы при реализации проектов и распределении ролей и задач.

Личностные:

- Воспитана целеустремленность, усидчивость и дисциплина при реализации сложных проектов;
- Воспитано уважение к процессу разработки и результатам труда;
- Сформирована уверенность в собственных силах при решении нестандартных и комплексных задач;
- Сформировано профессиональное отношение к труду и понимание ценности качественной работы.

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ ЦЕНТР ДЕТСКОГО (ЮНОШЕСКОГО) ТЕХНИЧЕСКОГО
ТВОРЧЕСТВА «СТАРТ+» НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

УТВЕРЖДЕНО

Приказ № _____ от _____
Директор ГБУ ДО ЦД (Ю) ТТ «Старт+»
Невского района Санкт-Петербурга
_____ Подобаева О.Г.
« ____ » _____ 20 ____ г.

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК
реализации дополнительной общеразвивающей программы
«Программирование 2 ступень»
Модуль «Нейронные сети»
на 2026-2027 учебный год

Педагог:

Год обучения, группа	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Кол-во учебных дней	Кол-во учебных часов	Режим занятий
1 год Группа №1			36	72	144	2 раза в неделю по 2 часа
2 год Группа №1			36	72	144	2 раза в неделю по 2 часа

ПРИНЯТ

Решением Педагогического совета
ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт – Петербурга
Протокол от _____ № _____

УТВЕРЖДЕН

Приказом от _____ № _____
Директор ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт – Петербурга
_____ О.Г. Подобаева

КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН
рабочей программы «Программирование 2 ступень», модуль «Нейронные сети»
на 2026/2027 учебный год
Группа №1. 1-й год обучения, 144 часа
Педагог дополнительного образования – Тупицын Пётр Викторович

№ п/п	Дата занятия		Раздел	Тема занятия	Кол- во часов	Форма занятия	Способы контроля
	По плану	По факту					
1			Введение в технологии искусственного интеллекта и основы программирования на Python	Вводное занятие. Инструктаж по технике безопасности. Знакомство с технологией ИИ.	2	Аудиторное	опрос
2				Знакомство с профессиями в ИИ.	2	Аудиторное	Входящий контроль
3				Введение в язык программирования Python. Среда разработки Jupyter Lab. Дистрибутив Anaconda.	2	Аудиторное	наблюдение
4				Организация диалога.	2	Аудиторное	наблюдение
5				Основные синтаксические единицы и конструкции.	2	Аудиторное	наблюдение
6				Линейный и условный алгоритмы.	2	Аудиторное	опрос
7				Каскадный условный оператор.	2	Аудиторное	наблюдение
8			Создание алгоритмов искусственного	Строки и циклы. Тип string и операции над строками.	2	Аудиторное	наблюдение
9				Строки и циклы. Цикл с параметром и условием.	2	Аудиторное	наблюдение
10				Списки, функции. Типы данных list и dict.	2	Аудиторное	наблюдение

11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		

**интеллекта на
Python**

Списки, функции. Операции со списками.	2	Аудиторное	наблюдение
Списки, функции, файлы. Пользовательские функции.	2	Аудиторное	опрос
Списки, функции. Возвращение логических констант в зависимости от условия.	2	Аудиторное	опрос
Списки, функции. Обработка файлов.	2	Аудиторное	наблюдение
Основы объектно-ориентированного программирования. Классы и объекты.	2	Аудиторное	опрос
ООП. Конструкторы.	2	Аудиторное	наблюдение
ООП. Методы.	2	Аудиторное	наблюдение
ООП. Перегрузка операторов.	2	Аудиторное	опрос
ООП. Создание класса Нейрон.	2	Аудиторное	наблюдение
Математические основы искусственного интеллекта. Векторы и матрицы.	2	Аудиторное	опрос
Математические основы искусственного интеллекта. Вычисление координат векторов.	2	Аудиторное	наблюдение
Математические основы искусственного интеллекта. Нахождение угла между векторами.	2	Аудиторное	опрос
Математические основы искусственного интеллекта. Нахождение вектора, удовлетворяющего условию.	2	Аудиторное	опрос
Математические основы искусственного интеллекта. Вычисление произведения матриц.	2	Аудиторное	наблюдение, опрос
Математические основы искусственного интеллекта. Вычисление разницы матриц.	2	Аудиторное	опрос
Математические основы искусственного интеллекта. Теория вероятности.	2	Аудиторное	опрос

27			Математические основы искусственного интеллекта. Математическая статистика.	2	Аудиторное	наблюдение
28			Математические основы искусственного интеллекта. Задачи по теории вероятности.	2	Аудиторное	Промежуточный контроль
29			Библиотека Numpy для обработки числовых массивов.	2	Аудиторное	
30			Библиотека Numpy. Обработка одномерных массивов.	2	Аудиторное	опрос
31			Библиотека Numpy. Обработка двумерных массивов.	2	Аудиторное	опрос
32			Библиотека Numpy. Пользовательская функция для обработки одномерных массивов.	2	Аудиторное	опрос
33			Библиотека Numpy и линейная алгебра.	2	Аудиторное	опрос
34			Библиотека Numpy и линейная алгебра. Задача линейной регрессии.	2	Аудиторное	наблюдение
35			Матричное произведение, транспонирование, вычисление обратной матрицы.	2	Аудиторное	опрос
36			Библиотеки Pandas, Matplotlib и Seaborn. Анализ данных о пассажирах Титаника (датасет Titanic).	2	Аудиторное	наблюдение
37			Визуализация данных о пассажирах Титаника (датасет Titanic).	2	Аудиторное	опрос
38		Машинное обучение как методология искусственного обучения	Введение в машинное обучение (МО). Регрессия и классификация.	2	Аудиторное	наблюдение
39			Пайплайн машинного обучения.	2	Аудиторное	опрос
40			Построение модели МО методом ближайших соседей и измерение её качества с помощью кросс-валидации.	2	Аудиторное	наблюдение
41			Метрики машинного обучения.	2	Аудиторное	наблюдение
42			Реализация класса линейной регрессии.	2	Аудиторное	опрос

43				Математические основы машинного обучения. Функции математического анализа.	2	Аудиторное	наблюдение
44				Алгоритмы регрессии в задачах МО.	2	Аудиторное	опрос
45				Логистическая регрессия.	2	Аудиторное	наблюдение
46				Решающие деревья.	2	Аудиторное	опрос
47				Композиции алгоритмов.	2	Аудиторное	опрос
48				Пайплайн выбора модели.	2	Аудиторное	опрос
49			Нейронные сети и компьютерное зрение	Введение в нейросети.	2	Аудиторное	наблюдение
50				Построение нейронных сетей. Введение в библиотеку Pytorch.	2	Аудиторное	наблюдение
51				Обучение нейронных сетей в библиотеке PyTorch.	2	Аудиторное	наблюдение
52				Технология компьютерного зрения. Свёрточные нейронные сети.	2	Аудиторное	опрос
53				Архитектуры CNN и Transfer Learning.	2	Аудиторное	наблюдение
54				Классификация изображений на практике.	2	Аудиторное	опрос
55			Продвинутые алгоритмы машинного обучения	Методы классификации текста.	2	Аудиторное	опрос
56				Логистическая регрессия для задач классификации текста.	2	Аудиторное	наблюдение
57				Алгоритм SVM.	2	Аудиторное	наблюдение
58				Градиентный бустинг.	2	Аудиторное	опрос
59				Градиентный бустинг. Реализация метода.	2	Аудиторное	наблюдение
60				Задачи компьютерного зрения.	2	Аудиторное	наблюдение
61				Решение задач компьютерного зрения с применением нейронных сетей.	2	Аудиторное	наблюдение
62				Внедрение моделей машинного обучения в продукт.	2	Аудиторное	опрос

63				Внедрение моделей машинного обучения в продукт. Проблема мощности устройства.	2	Аудиторное	наблюдение
64				Введение в обработку естественного языка	2	Аудиторное	опрос
65				Эмбединги слов.	2	Аудиторное	наблюдение
66				Классификация текстов с помощью эмбедингов слов.	2	Аудиторное	наблюдение
67				Существующие решения классификации текста.	2	Аудиторное	опрос
68				Языковое моделирование.	2	Аудиторное	наблюдение
69				Языковое моделирование.	2	Аудиторное	наблюдение
70				Языковое моделирование.	2	Аудиторное	опрос
71			Итоговое занятие	Подготовка к презентации проектов. Программы и авторское право. Вопрос коммерциализации.	2	Аудиторное	Итоговый контроль
72				Презентация проектов. Подведение итогов. Вручение грамот.	2	Аудиторное	Фронтальный/ анализ
				Итого	144		

ПРИНЯТ
Решением Педагогического совета
ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт – Петербурга
Протокол от _____ № _____

УТВЕРЖДЕН
Приказом от _____ № _____
Директор ГБУ ДО ЦД(Ю)ТТ «Старт+»
Невского района Санкт – Петербурга
_____ О.Г. Подобаева

КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН
рабочей программы «Программирование 2 степень», модуль «Нейронные сети»
на 2026/2027 учебный год
Группа №1. 2-й год обучения, 144 часа
Педагог дополнительного образования – Тупицын Пётр Викторович

№ п/п	Дата занятия		Раздел	Тема занятия	Кол- во часов	Форма занятия	Способы контроля
	По плану	По факту					
1			Библиотека Keras	Вводное занятие. Инструктаж по технике безопасности.	2	Аудиторное	опрос
2				Введение в keras. Основные классы и функции.	2	Аудиторное	Входящий контроль
3				Построение простой модели в keras. Компиляция и обучение модели.	2	Аудиторное	наблюдение
4				Метрики и функции потерь в keras.	2	Аудиторное	наблюдение
5				Сохранение и загрузка моделей.	2	Аудиторное	наблюдение
6				Callback-функции в keras. Разработка собственного класса обратной связи.	2	Аудиторное	опрос
7				Реализация классификации изображений в keras.	2	Аудиторное	наблюдение
8				Реализация классификации текста в keras.	2	Аудиторное	наблюдение

9				Настройка слоёв и оптимизаторов.	2	Аудиторное	наблюдение
10				Сравнение keras и pytorch на одной задаче.	2	Аудиторное	наблюдение
11			Инструменты и библиотеки глубокого обучения	Обзор современных библиотек для ML и DL.	2	Аудиторное	наблюдение
12				Использование предобученной модели.	2	Аудиторное	опрос
13				Hugging Face – база открытых моделей нейронных сетей.	2	Аудиторное	опрос
14				Дообучение предобученной модели: зачем и когда?	2	Аудиторное	наблюдение
15				Основные идеи дообучения модели.	2	Аудиторное	опрос
16				Заморозка слоёв предобученной модели. Теория.	2	Аудиторное	наблюдение
17				Заморозка начальных слоёв CNN.	2	Аудиторное	наблюдение
18				Частичная разморозка слоёв и fine-tuning.	2	Аудиторное	опрос
19				Интеграция кастомных слоёв и функций в keras.	2	Аудиторное	наблюдение
20				Дообучение модели под конкретный датасет.	2	Аудиторное	опрос
21				Сравнение локальных моделей gpt-oss и yandexgpt.	2	Аудиторное	наблюдение
22			Сетевое администрирование для программистов	Сетевые модели и протоколы.	2	Аудиторное	опрос
23				Адресация и маршрутизация.	2	Аудиторное	опрос
24				Настройка локальной сети и работа с удаленным сервером.	2	Аудиторное	наблюдение, опрос
25				Размещение серверного приложения на сервере.	2	Аудиторное	опрос
26				Размещение серверного приложения на сервере. Проброс портов и реверс-проксирование.	2	Аудиторное	опрос
27				Безопасность серверного приложения. Шифрование и SSL-сертификат.	2	Аудиторное	наблюдение

28				Безопасность серверного приложения. Контроль запросов по IP адресу.	2	Аудиторное	Промежуточн ый контроль
29			Инфраструктура и развертывание нейронных сетей	Docker для ML-проектов.	2	Аудиторное	
30				Упаковывание модели в docker-контейнер.	2	Аудиторное	опрос
31				Основы REST API для машинного обучения.	2	Аудиторное	опрос
32				Создание API для модели на Hugging Face.	2	Аудиторное	опрос
33				Развертывание простой модели как сервиса.	2	Аудиторное	опрос
34				Особенности работы нейронной сети на сервере.	2	Аудиторное	наблюдение
35				CPU и GPU вычисления на сервере.	2	Аудиторное	опрос
36				Веб-интерфейс для нейронной сети на сервере.	2	Аудиторное	наблюдение
37				Библиотека Gradio. Быстрая разработка интерфейса для разных задач.	2	Аудиторное	опрос
38				Библиотека Gradio. Интерфейс чат-бота с искусственным интеллектом.	2	Аудиторное	наблюдение
39				Библиотека Gradio. Интерфейс приложений для обработки изображений.	2	Аудиторное	опрос
40			Основы проектирования нейронных сетей	Что такое архитектура сети и зачем её менять?	2	Аудиторное	наблюдение
41				Гиперпараметры и их роль в архитектуре	2	Аудиторное	наблюдение
42				Нейроны и функции активации. ReLU и sigmoid.	2	Аудиторное	опрос
43				Модули в PyTorch и Keras: как собирать свою архитектуру.	2	Аудиторное	наблюдение
44				CNN. Знакомые сверточные нейросети с незнакомой стороны.	2	Аудиторное	опрос
45				Остаточные связи (skip-connections): как сеть учиться глубже.	2	Аудиторное	наблюдение

46				Разветвляющиеся архитектуры. Нейронные сети с несколькими входами.	2	Аудиторное	опрос
47				Совмещение идей: CNN + MLP.	2	Аудиторное	опрос
48			Рекуррентные сети и последовательност и	Простая RNN как первый шаг к памяти. Предсказание последовательности чисел.	2	Аудиторное	опрос
49				LSTM и GRU как улучшенные блоки рекуррентной сети.	2	Аудиторное	наблюдение
50				Attention как способ выбирать важное.	2	Аудиторное	наблюдение
51				Проектирование собственных архитектур для генерации текста.	2	Аудиторное	наблюдение
52			Архитектуры генеративных сетей	Автоэнкодер как способ кодировать данные. Восстановление сжатого изображения.	2	Аудиторное	опрос
53				Вариации автоэнкодеров (VAE).	2	Аудиторное	наблюдение
54				GAN. Идея соревнования двух нейронных сетей в одном приложении.	2	Аудиторное	опрос
55				GAN. Применение соревнования для создания «ботов» в играх.	2	Аудиторное	опрос
56				GAN. Применение соревнования для генерации изображений.	2	Аудиторное	наблюдение
57				Diffusion. Идея постепенного восстановления из шума.	2	Аудиторное	наблюдение
58				GAN или Diffusion. Тестируем качество генерации.	2	Аудиторное	опрос
59				Баланс сложности и переобучения.	2	Аудиторное	наблюдение
60				Модульность: делим архитектуру на блоки.	2	Аудиторное	наблюдение
61				Мультимодальная сеть. Совмещение разных типов данных.	2	Аудиторное	наблюдение
62				Сеть с несколькими выходами.	2	Аудиторное	опрос
63				Разработка мультимодальной сети.	2	Аудиторное	наблюдение

64			Введение в архитектуру трансформеров	От RNN к трансформерам. Зачем нужен новый подход?	2	Аудиторное	опрос
65				Архитектура трансформера. Encoder + Decoder.	2	Аудиторное	наблюдение
66				Decoder-only модели трансформеров. Как нейронная сеть без энкодера способна понять текст?	2	Аудиторное	наблюдение
67				Архитектура GPT. Предобученный декодер. Проектная деятельность.	2	Аудиторное	опрос
68				Разработка упрощенной GPT архитектуры. Проектная деятельность.	2	Аудиторное	наблюдение
69				Вариации и применение трансформеров. Проектная деятельность.	2	Аудиторное	наблюдение
70				Мультимодальный трансформер. Проектная деятельность.	2	Аудиторное	опрос
71			Итоговое занятие	Подготовка к презентации проектов. Отладка.	2	Аудиторное	Итоговый контроль
72				Презентация проектов. Подведение итогов. Вручение грамот.	2	Аудиторное	Фронтальный/ анализ
				Итого	144		