

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 338  
НЕВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

ПРИНЯТА  
педагогическим советом  
ГБОУ школы № 338 Невского района  
Санкт-Петербурга  
Протокол №8 от 18.06.2025

УТВЕРЖДЕНА  
Директор ГБОУ школы № 338  
Невского района Санкт-Петербурга  
М.Н. Свирко  
приказом №4871-д от 19.06.2025



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ  
ПРОГРАММА  
«ТЕХНОЛОГИКА»**

Возраст учащихся 13–17 лет  
Срок реализации: 1 год

Дата и время 30.07.2025 14:00  
ФИО: Свирко Мария Николаевна  
Должность: Директор  
КЭП: 0088A382C1F179DFEA80D361858F7E2D1B  
Действителен с 06.12.2023 по 28.02.2025

Автор – составитель:  
Канышев Артемий Денисович  
педагог дополнительного образования

## Пояснительная записка

История развития информационных технологий уходит корнями в глубокую древность. Первым вычислительным устройством считается абак, появившийся более 5000 лет назад в Древнем Вавилоне. В XVII веке Блез Паскаль создал первую механическую счётную машину, а в XIX веке Чарльз Бэббидж разработал проект аналитической машины — прообраза современного компьютера, заложив основы программирования.

Революционным этапом стало появление в середине XX века первой электронно-вычислительной машины ENIAC. С тех пор развитие цифровых технологий идёт стремительными темпами: от персональных компьютеров 1980-х годов до массового распространения интернета в 1990-х, а затем — к развитию мобильных технологий, облачных платформ, больших данных и искусственного интеллекта в XXI веке. Сегодня мы стоим на пороге новой технологической эры, формируемой в рамках Национальной технологической инициативы (НТИ), где ключевыми направлениями становятся автоматизация процессов, роботизация, машинное обучение и интеллектуальные системы.

Актуальность программы кружка «Технологика» обусловлена:

- массовой цифровизацией всех сфер жизни общества, включая образование, промышленность, медицину и управление;
- растущим спросом на специалистов в области цифровых технологий, особенно в таких перспективных направлениях, как автоматизация бизнес-процессов и искусственный интеллект (по прогнозам экспертов, к 2030 году дефицит IT-кадров достигнет сотен тысяч человек);
- необходимостью ранней профориентации школьников в условиях стремительно меняющегося рынка труда;
- высоким интересом детей к современным технологиям, программированию и созданию «умных» систем;
- важностью развития у учащихся алгоритмического, логического и системного мышления, навыков работы с данными и решения реальных задач.

Кружок «Технологика» ориентирован на подготовку учащихся к участию в Национальной технологической олимпиаде (НТО) по профилям «Автоматизация бизнес-процессов» и «Искусственный интеллект», что делает его не только образовательной, но и олимпиадно-ориентированной площадкой. Программа учитывает требования НТИ и современные тренды в технологическом образовании.

Отличительной особенностью программы является практико-ориентированный подход: учащиеся не просто изучают теорию, а решают прикладные задачи, разрабатывают собственные проекты по автоматизации процессов, создают элементы систем искусственного интеллекта на языке Python, работают с API, базами данных, простыми нейросетями и инструментами RPA (Robotic Process Automation). Такой подход способствует глубокому пониманию материала, развитию исследовательских навыков и формированию технологического мышления.

В результате занятий участники кружка не только осваивают актуальные компетенции, но и применяют их в учебной и проектной деятельности, что повышает мотивацию к обучению, раскрывает их потенциал и формирует устойчивый интерес к STEM-дисциплинам и инженерным профессиям будущего.

Дополнительная общеобразовательная общеразвивающая программа «Технологика» технической направленности.

Уровень освоения – профильный.

Программа разработана в соответствии со следующими документами:

- Федеральный Закон Российской Федерации от 29.12.2012 г. № 273 «Об образовании в Российской Федерации» (далее – ФЗ № 273).

- • Федеральный закон от 31.07.2020 № 304-ФЗ «О внесении изменений в Федеральный закон Российской Федерации «Об образовании в Российской Федерации по вопросам воспитания».
- • Приказ Министерства просвещения РФ от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
- • Концепция развития дополнительного образования детей до 2030 года (утв. распоряжением Правительства РФ от 31.03.2022г. № 678-р).
- • Распоряжение Комитета по образованию Санкт-Петербурга от 25.08.2022 №1672-р «Об утверждении критериев оценки качества дополнительных общеразвивающих программ, реализуемых организациями, осуществляющими образовательную деятельность, и индивидуальными предпринимателями Санкт-Петербурга».
- • Постановление Главного государственного санитарного врача РФ 28.09.2020 санитарные правила СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания, обучения, отдыха и оздоровления детей и молодежи», зарегистрированными Министерством юстиции Российской Федерации РГ№ 61753 от 18.12.2020.

**Актуальность** данной программы заключается в формировании у школьников ключевых навыков цифровой эпохи: логического, алгоритмического и системного мышления, необходимых для успешной реализации в условиях стремительно развивающейся технологической среды. В условиях роста автоматизации и внедрения решений на основе искусственного интеллекта особенно важным становится понимание принципов работы алгоритмов, обработки данных и создания «умных» систем. Программа направлена на развитие этих компетенций через практико-ориентированное обучение.

Центральным инструментом обучения выступает язык программирования Python — один из самых востребованных и доступных языков в современной IT-индустрии. Его простота делает его идеальным выбором для начинающих, а широкая экосистема библиотек (таких как Pandas, NumPy, Scikit-learn, TensorFlow/Keras, RPA-инструменты) позволяет решать задачи как в области автоматизации рутинных процессов, так и в создании элементов искусственного интеллекта — от анализа данных до построения моделей машинного обучения. Python является основным языком в большинстве профилей Национальной технологической олимпиады (НТО), что делает его освоение стратегически важным для участников кружка.

**Отличительная особенность** образовательной программы — интеграция программирования как части современной цифровой грамотности, приравненной по значимости к традиционным видам грамотности. Обучение строится не на заучивании кода, а на развитии технологического мышления: учащиеся учатся анализировать задачи, проектировать алгоритмы, тестировать решения и работать в команде над реальными проектами.

Ребята применяют полученные знания на практике: создают скрипты для автоматизации учебных и бытовых задач, разрабатывают прототипы чат-ботов, системы распознавания образов, модели классификации данных, а также учатся строить цифровые потоки обработки информации — всё это соответствует задачам профилей НТО «Автоматизация бизнес-процессов» и «Искусственный интеллект». Такой подход развивает не только технические навыки, но и креативность, критическое мышление, способность находить нестандартные решения и воплощать их в виде работающих программных продуктов.

Таким образом, программа кружка «Технологика» формирует у учащихся не просто навыки программирования, а комплексную технологическую компетентность, необходимую для участия в олимпиадах, дальнейшего обучения в IT-сфере и карьеры в индустрии будущего.

#### **Адресат программы**

Данная программа составлена для учащихся 13–17 лет, занимающихся в системе дополнительного образования, и имеющих желание попробовать себя в техническом направлении.

### **Объём и срок реализации программы**

Курс включает 144 занятия. 1 год – 144 часа.

Срок реализации: 1 год.

### **Цель программы**

Формирование у обучающихся системных компетенций в области цифровых технологий через освоение языка программирования Python, основ алгоритмизации, автоматизации процессов и элементов искусственного интеллекта, с целью подготовки школьников к участию в Национальной технологической олимпиаде (НТО) и дальнейшему развитию в перспективных направлениях Национальной технологической инициативы (НТИ).

### **Задачи программы**

#### *Обучающие:*

- Познакомить учащихся с базовыми концепциями информационных технологий, автоматизации и искусственного интеллекта;
- Освоить синтаксис и основные конструкции языка Python как ключевого инструмента для решения задач НТО;
- Научить разрабатывать алгоритмы для автоматизации типовых задач (включая обработку данных, работу с файлами, парсинг, RPA-подобные скрипты);
- Познакомить с основами машинного обучения: классификацией, кластеризацией, обучением моделей на простых примерах;
- Обучить работе с API, библиотеками Python (Pandas, NumPy, Scikit-learn, Tkinter и др.) и простыми нейросетевыми архитектурами;
- Сформировать навыки проектирования и реализации IT-решений по профилям «Автоматизация бизнес-процессов» и «Искусственный интеллект»;
- Подготовить учащихся к выполнению заданий различного уровня сложности, включая олимпиадные задачи НТО.

#### *Развивающие:*

- Развивать логическое, алгоритмическое и системное мышление;
- Формировать аналитические способности и умение работать с данными;
- Развивать креативность и способность находить нестандартные технические решения;
- Способствовать развитию проектного мышления — от идеи до работающего прототипа;
- Развивать навыки презентации и защиты собственных проектов;
- Формировать умение самостоятельно изучать новые технологии и адаптироваться к изменяющейся цифровой среде;
- Стимулировать техническое и математическое мышление в контексте реальных прикладных задач.

#### *Воспитательные:*

- Воспитывать культуру работы с информацией, этическое отношение к данным и ИИ;
- Формировать ответственность, дисциплину и настойчивость в решении сложных задач;
- Развивать самостоятельность, инициативность и стремление к самообразованию;
- Воспитывать навыки командной работы, распределения ролей и эффективного взаимодействия в проектной деятельности;
- Формировать профессиональную направленность на карьеру в IT, особенно в областях автоматизации, анализа данных и искусственного интеллекта;
- Способствовать формированию интереса к инновациям, научно-техническому творчеству и участию в олимпиадах и конкурсах.

### **Условия реализации программы**

**Условия набора детей в коллектив:** в группу первого года обучения принимаются все желающие заниматься в данном объединении, на основании письменного заявления

родителей.

Комплектование групп 1 года обучения проводится до 10 сентября.

**Наполняемость учебной группы:** 1-й год обучения – 15 чел.

**Форма организации деятельности учащихся на занятии:**

Обучение в кружке «Технологика» строится на принципах дифференцированного и деятельностного подхода, что позволяет эффективно сочетать теоретическую подготовку с практической отработкой навыков, необходимых для участия в Национальной технологической олимпиаде (НТО). Используются следующие формы организации учебной и проектной деятельности:

- **Фронтальная** форма: применяется при изучении теоретических основ программирования, алгоритмизации, принципов работы систем искусственного интеллекта и автоматизации. Включает объяснение ключевых концепций, демонстрацию примеров кода, разбор типовых задач НТО и обсуждение актуальных технологических кейсов.
- **Коллективная** форма: используется в ходе командных соревнований, таких как мини-хакатоны, IT-квесты, тематические недели (например, «Неделя ИИ» или «День автоматизации»), а также при подготовке к олимпиадам и участию в онлайн-конкурсах. Такие мероприятия способствуют формированию духа командного соперничества, развитию скоростного мышления и умения работать в условиях ограниченного времени — важных навыков для олимпиадников.
- **Групповая** форма: лежит в основе проектной деятельности. Учащиеся работают в командах по 3–5 человек над созданием собственных прототипов решений по профилям «Автоматизация бизнес-процессов» и «Искусственный интеллект». Каждый участник команды выполняет определённую роль: программист, аналитик данных, тестировщик, координатор проекта, презентатор. Это имитирует реальную IT-команду и формирует навыки распределённой работы, управления задачами и ответственности за результат.
- **Индивидуальная** форма: направлена на отработку базовых навыков программирования, решение алгоритмических задач различного уровня сложности, самостоятельное освоение новых библиотек и подготовку к индивидуальным этапам олимпиад. Ученики получают персонализированные задания в зависимости от уровня подготовки.

**Интерактивные методы и технологии обучения:**

На занятиях активно используются современные интерактивные формы работы, способствующие вовлечённости и глубокому усвоению материала:

- Практические задания в средах разработки Python (Jupyter Notebook, PyCharm, Google Colab);
- Программирование скриптов для автоматизации рутинных задач (работа с Excel, парсинг сайтов, отправка писем);
- Разработка прототипов систем искусственного интеллекта: чат-ботов, моделей классификации изображений, анализ текста;
- Создание простых приложений с графическим интерфейсом (на базе Tkinter или Flask);
- Решение логических и олимпиадных задач, аналогичных заданиям НТО;
- Работа с онлайн-платформами и тренажёрами по программированию (например, Stepik, Codeforces, Timus, Яндекс.Контест);
- Использование систем автоматической проверки кода (CI/CD-подобные сценарии в учебном формате) для развития культуры тестирования и отладки.

Такое разнообразие форм и методов обеспечивает высокую мотивацию учащихся, развивает как технические, так и метапредметные компетенции, необходимые для успешного участия в технологических конкурсах и дальнейшего профессионального роста в сфере цифровых технологий.

**Материально-техническое оснащение**

Занятия проходят в компьютерном классе, оснащённом:

- Современными компьютерами с установленным ПО
- Проектором и интерактивной доской
- Доступом к интернету

- Сервером для хранения проектов
- Средствами видеоконференцсвязи

Программное обеспечение:

- Python и необходимые библиотеки
- Среда разработки (например, PyCharm)
- Система контроля версий Git
- Платформы для онлайн-обучения

### ***Дидактические материалы***

Наглядность:

1. Презентации по темам курса
2. Образцы программного кода
3. Алгоритмы решения типовых задач
4. Схемы основных конструкций языка

Раздаточный материал:

1. Задачи для практики
2. Контрольные работы
3. Методические указания
4. Шаблоны проектной документации

### ***Печатные пособия***

1. Лутц, М. Изучаем Python : практическое руководство для начинающих программистов / М. Лутц. — Москва : ООО "Издательство 'Диалектика'", 2020. — 624 с.
2. Доусон, М. Программируем на Python : создание игр и приложений / М. Доусон. — Санкт-Петербург : Питер, 2019. — 400 с.
3. Свейгарт, Э. Автоматизация рутинных задач с помощью Python : практическое пособие для начинающих / Э. Свейгарт. — Москва : ДМК Пресс, 2021. — 500 с.
4. Барри, П. Head First Python : учебное пособие для изучения языка программирования Python / П. Барри. — Москва : Вильямс, 2020. — 624 с.
5. Шрайбер, Э. Изучаем Python через проекты : практические задания для освоения навыков программирования / Э. Шрайбер. — Москва : Питер, 2022. — 350 с.
6. Массарон, Л. Python за 24 часа : самоучитель для быстрого старта / Л. Массарон. — Москва : Эксмо, 2021. — 384 с.

### ***Необходимое кадровое и материально-техническое обеспечение программы***

Кадровое обеспечение: педагог с соответствующим профилем объединения образованием и опытом работы.

### **Планируемые результаты освоения программы**

Основной показатель качества освоения программы - развитие цифровых компетенций обучающегося, его способность создавать IT-продукты и готовность к дальнейшему профессиональному развитию в сфере информационных технологий.

Программа кружка «Технологика» направлена на формирование у учащихся комплексных компетенций, необходимых для успешного участия в Национальной технологической олимпиаде (НТО), а также для дальнейшего развития в перспективных направлениях цифровой экономики. Результаты освоения программы представлены по трём ключевым компонентам: образовательному, развивающему и воспитательному, а также в виде личностных и метапредметных достижений.

#### ***Образовательный компонент:***

- Овладеют основами алгоритмизации и программирования на языке Python;
- Освоят базовые и продвинутое конструкции Python, включая работу с модулями, файлами, API и библиотеками (Pandas, NumPy, Scikit-learn и др.);
- Научатся разрабатывать алгоритмы для автоматизации типовых процессов (обработка данных, генерация отчётов, парсинг информации);
- Получат первичные навыки работы с элементами искусственного интеллекта: создание простых моделей машинного обучения, классификация данных, работа с текстом и изображениями;

- Смогут реализовывать собственные проекты по профилям «Автоматизация бизнес-процессов» и «Искусственный интеллект», включая прототипы чат-ботов, систем анализа данных и RPA-скриптов;
- Научатся решать задачи различного уровня сложности, аналогичные заданиям НТО, в том числе с использованием логического анализа, оптимизации и обработки больших объёмов информации.

*Развивающий компонент:*

- Логическое, алгоритмическое и системное мышление;
- Аналитические способности и умение работать с данными;
- Креативность и способность находить нестандартные решения технических задач;
- Навыки проектирования и реализации IT-решений «от идеи до прототипа»;
- Способность к самостоятельному изучению новых технологий и инструментов;
- Техническое и математическое мышление, необходимое для понимания принципов работы ИИ и автоматизированных систем;
- Проектные компетенции: управление сроками, тестирование решений, документирование кода.

Воспитательный компонент:

- Культуры обращения с информацией, этического подхода к данным и использованию ИИ;
- Ответственного отношения к выполнению учебных и проектных задач;
- Самостоятельности, инициативности и настойчивости в достижении целей;
- Навыков командной работы, эффективного взаимодействия и распределения ролей в IT-команде;
- Профессиональной ориентации на карьеру в сфере цифровых технологий, особенно в областях автоматизации, анализа данных и искусственного интеллекта.

Личностные результаты:

- Устойчивый интерес к современным технологиям и инновациям;
- Готовность к непрерывному самообразованию и профессиональному росту;
- Ответственность за результат своей работы и вклад в общий успех команды;
- Умение работать в группе, распределять обязанности и координировать действия;
- Критическое мышление при анализе задач и выборе решений.

Метапредметные результаты (формирование универсальных учебных действий):

Регулятивные УУД:

- Умение ставить цели при решении технических задач;
- Планирование этапов реализации проекта или алгоритма;
- Анализ ошибок в коде и поиск путей их устранения;
- Оценка эффективности различных подходов к решению одной и той же задачи;
- Контроль и корректировка процесса выполнения задания с использованием обратной связи.

Познавательные УУД:

- Поиск и анализ информации в официальной документации, технических мануалах и справочниках;

- Интерпретация и разбор технических заданий, включая олимпиадные условия;
- Систематизация знаний по темам программирования, автоматизации и ИИ;
- Применение алгоритмического подхода к решению задач из разных предметных областей;
- Использование различных источников информации для расширения кругозора и углубления понимания технологий.

Коммуникативные УУД:

- Эффективная коммуникация в команде при реализации проектов;
- Представление и защита собственных решений и идей;
- Участие в обсуждении технических решений, предложение альтернатив;
- Распределение ролей и координация действий в рамках командной разработки;
- Навыки презентации проектов перед аудиторией, включая использование визуальных материалов и демонстрацию работающих прототипов.

Основным показателем качества освоения программы является сформированность цифровых компетенций, проявляющаяся в способности учащихся:

- самостоятельно разрабатывать и реализовывать IT-проекты;
- применять полученные знания для участия в олимпиадах и конкурсах (в первую очередь — в НТО);
- демонстрировать готовность к обучению в STEM-направлениях и строительству карьеры в технологической сфере.

Успешное освоение программы создаёт прочный фундамент для дальнейшего профессионального роста и активного участия в цифровой трансформации общества.

### Учебный план 1 года обучения

№ п/п	Наименование тем	Количество часов			Формы контроля
		Всего	Теория	Практика	
1.	Вводное занятие. Введение в программирование. Техника безопасности	2	1	1	Фронтальный Тестирование
2.	Основы Python: синтаксис, типы данных, переменные	8	4	4	Текущий Фронтальный Индивидуальный
3.	Условные операторы и ветвления	6	2	4	Текущий Фронтальный Индивидуальный
4.	Циклы и их применение	6	2	4	Текущий Фронтальный Индивидуальный
5.	Работа с коллекциями данных (списки, кортежи, словари, множества)	10	4	6	Текущий Фронтальный Индивидуальный
6.	Функции и модули	12	4	8	Текущий Фронтальный Индивидуальный



7.	Объектно-ориентированное программирование	8	4	4	Текущий Фронтальный Индивидуальный
8.	Работа с файлами и данными	16	8	8	Текущий Фронтальный Индивидуальный
9.	Графические интерфейсы и визуализация	8	4	4	Текущий Фронтальный Индивидуальный
10.	Базы данных и веб-программирование	16	8	8	Текущий Фронтальный Индивидуальный
11.	Анализ данных и машинное обучение	12	6	6	Текущий Фронтальный Индивидуальный
12.	Итоговая проектная работа	8		8	Индивидуальный
13.	Дополнительные темы и резерв	32	16	16	Текущий Фронтальный
	<b>Итого</b>	<b>144</b>	<b>63</b>	<b>81</b>	

### Календарный учебный график

Год обучения	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год	08.09.2025	28.05.2026	36	72	144	2 раза в неделю по 2 часа

### Рабочая программа

Рабочая программа составляется в соответствии с локальным актом образовательной организации на каждый текущий год обучения (прилагается к дополнительной общеобразовательной общеразвивающей программе в виде отдельных документов)

Календарно-тематическое планирование по годам обучения прилагается к рабочей программе.

#### Особенности организации образовательного процесса 1 года обучения

Рабочая программа рассчитана на 1 год для детей 13–17 лет, в группе 15 человек.

Набор осуществляется на основе свободного выбора. Занятия проходят 2 раза в неделю по 2 часа.

За учебный год учащиеся осваивают программу в 144 часа.

В дни осенних, зимних и весенних каникул проводится индивидуальная работа, индивидуальные консультации.

### Содержание программы 1 года обучения.

#### 1. Вводное занятие. Введение в программирование. Техника безопасности

- Теория: Знакомство с кружком «Технологика». Цели и перспективы изучения программирования. Роль IT в современном мире. Основные направления НТИ и НТО: автоматизация, искусственный интеллект, цифровые решения. Правила безопасной

работы за компьютером, этика использования данных, основы цифровой гигиены. Обзор сред программирования: Replit, Jupyter Notebook, установка Python.

- Практика: Диагностическое задание — первые шаги в Python. Запуск среды, выполнение простых команд (print, input), создание анкеты пользователя. Знакомство с интерфейсом и базовыми операциями.

## **2. Основы Python и базовые конструкции**

- Теория: Синтаксис языка Python. Переменные и типы данных (int, float, str, bool). Операции: арифметические, логические, сравнения. Форматирование строк (f-строки). Ввод и вывод данных.
- Практика: Разработка простых программ — конвертер единиц измерения, калькулятор площади фигур, генератор приветствий. Отработка навыков написания читаемого кода.

## **3. Условные операторы и ветвления**

- Теория: Условные конструкции if, elif, else. Логические выражения (and, or, not). Вложенные условия. Обработка ошибок: try-except.
- Практика: Создание интерактивных программ — тест на возраст, система оценки успеваемости, "умный помощник", предлагающий действия в зависимости от условий.

## **4. Циклы и их применение**

- Теория: Циклы for и while. Использование range(), break, continue. Вложенные циклы. Применение циклов для обработки данных и повторяющихся задач.
- Практика: Реализация игр ("Угадай число"), построение графических узоров в консоли, обработка списков (подсчёт, фильтрация), автоматизация простых рутинных действий.

## **5. Работа с коллекциями данных**

- Теория: Списки, кортежи, множества, словари. Методы работы с коллекциями: добавление, удаление, поиск, изменение. Перебор элементов (for, enumerate, .items()). Вложенные структуры данных.
- Практика: Создание базы данных студентов или книг. Реализация системы хранения и поиска информации. Анализ успеваемости класса, фильтрация данных по критериям.

## **6. Функции и модули**

- Теория: Функции: определение, параметры, возврат значений (return). Локальные и глобальные переменные. Lambda-функции. Создание и использование модулей. Стандартные библиотеки: math, random, datetime.
- Практика: Разработка библиотеки полезных функций. Создание калькулятора с модульной структурой. Генератор паролей, случайных вопросов, расписаний.

## **7. Объектно-ориентированное программирование (ООП)**

- Теория: Классы и объекты. Конструктор \_\_init\_\_. Атрибуты и методы. Наследование, инкапсуляция (на уровне понимания). Полиморфизм.
- Практика: Моделирование реальных объектов — классы "Ученик", "Книга", "Банковский счёт". Проект "Транспорт": родительский класс и его наследники (машина, велосипед, самолёт).

## **8. Работа с файлами и данными**

- Теория: Чтение и запись текстовых файлов. Формат CSV и JSON. Работа с файлами через контекстный менеджер with. Введение в библиотеку Pandas: Series, DataFrame, чтение/запись таблиц.
- Практика: Создание дневника с сохранением записей. Импорт и анализ таблицы с оценками. Автоматизация формирования отчётов. Парсинг данных с сайта (упрощённо, через requests и BeautifulSoup).

## **9. Графические интерфейсы и визуализация**

- Теория: Библиотека Tkinter: виджеты (Label, Entry, Button), обработка событий. Введение в визуализацию данных: Matplotlib, построение графиков и диаграмм.
- Практика: Разработка оконного калькулятора, формы обратной связи. Построение графиков изменения температуры, диаграмм успеваемости класса.

## **10. Базы данных и веб-программирование**

- Теория: Основы SQL и SQLite: таблицы, запросы (SELECT, INSERT, UPDATE,

DELETE), фильтрация. Работа с базой данных через Python (sqlite3). Основы веба: HTTP, HTML, клиент-серверная архитектура. Введение в Flask.

- Практика: Создание базы данных "Школьная библиотека" или "Заметки". Разработка простого веб-приложения на Flask — сайт-визитка кружка, страница с прогнозом погоды.

## **11. Анализ данных и машинное обучение**

- Теория: Введение в искусственный интеллект и машинное обучение. Типы задач: классификация, регрессия. Подготовка данных: признаки, метки, обучающая и тестовая выборки. Библиотеки: NumPy, Pandas, Scikit-learn. Модели: KNN, дерево решений.
- Практика: Классификация рукописных цифр (MNIST Lite). Прогнозирование успеваемости. Анализ отзывов (позитив/негатив). Создание простого чат-бота на основе правил и ключевых слов.

## **12. Итоговая проектная работа**

- Теория: Этапы проектной деятельности: постановка задачи, сбор требований, планирование, реализация, тестирование, презентация. Подготовка к защите проекта. Критерии оценки.
- Практика: Самостоятельная разработка и защита индивидуального или командного проекта по одному из направлений:
  - *Автоматизация бизнес-процессов*: скрипт для обработки Excel-отчётов, автозаполнение форм, RPA-подобный бот.
  - *Искусственный интеллект*: модель классификации, анализ текста, распознавание изображений. Учащиеся представляют проект с демонстрацией, презентацией и пояснением кода.

## **13. Дополнительные темы и резерв**

- Теория: Расширение знаний: работа с API, регулярные выражения (re), GitHub, DevOps (ознакомительно), асинхронность. IT-профессии будущего: аналитик данных, RPA-разработчик, ML-инженер. Подготовка к НТО: разбор типовых заданий.
- Практика: Хакатоны по автоматизации и ИИ. Создание чат-бота, загрузка проекта на GitHub. Решение олимпиадных задач НТО. Индивидуальные консультации, повторение сложных тем, командные игры на скорость и логику.

### **Планируемые результаты 1 года обучения**

В результате изучения данного курса **1 года обучения** обучающиеся получат возможность формирования:

#### **Личностные результаты:**

- Умение объяснять свое мнение и договариваться в коллективе.
- Развитие креативных навыков для решения нестандартных задач.
- Формирование интереса к IT-сфере и программированию.

#### **Метапредметные результаты:**

##### **Регулятивные УУД :**

- Умение отличать факты от домыслов.
- Способность ставить цели и планировать выполнение задач.
- Оценка своих действий в соответствии с поставленной задачей.

##### **Познавательные УУД :**

- Владение логическими операциями сравнения, анализа и синтеза.
- Умение находить и формулировать решения задач с помощью моделей.
- Переработка полученной информации для создания новых решений.

##### **Коммуникативные УУД :**

- Выполнение различных ролей в команде (лидер, исполнитель, критик).
- Развитие доброжелательности и способности к сотрудничеству.
- Умение вступать в общение с целью быть понятым.

#### **Предметные результаты:**

- Владение основами Python: синтаксис, типы данных, циклы, функции.
- Умение работать с коллекциями данных и файлами.
- Навыки создания графических интерфейсов и веб-приложений.
- Базовые знания в области анализа данных и машинного обучения.
- Умение реализовывать IT-проекты от идеи до готового продукта.

### **Оценочные и методические материалы**

#### ***Контроль результативности обучения***

Способом проверки является система педагогической диагностики результатов обучения, развития и воспитания, которые отслеживаются педагогом с помощью методик педагогической диагностики (наблюдение, контрольное задание, опрос, анализ, самоанализ, конкурсы) и фиксируются в журнале учета работы педагога 3 раза в год.

Результаты контроля являются основанием для корректировки программы и поощрения обучающихся

*Объектами контроля являются:*

- знания, умения, навыки по программе «ТехноЛогика»;
- уровень и качество реализуемых проектов;
- степень самостоятельности и уровень творческих способностей.

*Основными формами контроля являются:*

- входной контроль – опрос для определения степени подготовленности детей;
- текущий контроль – конкурсы, самоанализ;
- итоговый контроль – итоговый индивидуальный проект.

В первые дни занятий осуществляется входной контроль, который проводится в виде опроса для определения степени подготовленности детей, степени самостоятельности обучающихся и их интереса к занятиям, уровня культуры, творческих способностей.

Текущий контроль осуществляется в течение учебного года путем наблюдения за работой обучающихся. Текущий контроль позволяет определить степень усвоения обучающимися учебного материала и уровень их подготовленности к занятиям, повышает ответственность и заинтересованность детей в обучении. Выявление отстающих и опережающих обучение обучающихся позволяет своевременно подобрать наиболее эффективные методы и средства обучения.

Итоговый контроль проводится с целью определения степени достижения результатов обучения, ориентации обучающихся на дальнейшее самостоятельное обучение и получение сведений для совершенствования программы объединения и методов обучения.

Одним из способов определения результативности могут стать итоги участия кружковцев в школьных, районных, городских выставках.

В условиях дистанционного обучения демонстрация знаний и навыков в онлайн режиме, при невозможности присылать фото и видео материал и отчет педагогу в качестве обратной связи.

#### **Критерии оценки результатов тестов.**

- 80–100% - высокий уровень освоения программы;
- 60–80% - уровень выше среднего;
- 50–60% - средний уровень;
- 30–50% - уровень ниже среднего;
- меньше 30% - низкий уровень.

Также показателем эффективности занятий по курсу являются данные, которые педагог на протяжении года занятий заносил в таблицы в начале и конце года, прослеживая динамику развития познавательных способностей детей.

### **Способ проверки результатов по дополнительной образовательной программе**

Параметры	Учебный период	Критерии	Степень выраженности показателей оцениваемого качества	Методики
<b>Обучающие</b> Теоретические знания по лепке, кругозор.	Сентябрь Январь Май	Знания алгоритма решения задач, планов решения задач, планов конструирования задач, практика.	<p>- <b>Низкий уровень</b> – обучающийся владеет менее 1\2 объема знаний, предусмотренных программой.</p> <p>- <b>Средний уровень</b> - обучающийся владеет 1\2 объема знаний, предусмотренных программой.</p> <p>- <b>Высокий уровень</b> - обучающийся владеет всем объемом знаний, предусмотренных программой.</p>	Наблюдение, опрос
Практические умения и навыки	Сентябрь Январь Май	Умение находить общее в подходах к решению разного рода задач. Умение подбирать продуктивные и эффективные методы и приёмы в решении задач.	<p>- <b>Низкий уровень</b> – педагог оказывает большую помощь, грубые ошибки в самостоятельном решении.</p> <p>- <b>Средний уровень</b> – учащийся выполняет задание с помощью педагога, небольшие ошибки в самостоятельном решении.</p> <p>- <b>Высокий уровень</b> – учащийся самостоятельно выполняет задание. Полностью соответствует программным требованиям.</p>	Контрольное задание
<b>Развивающие</b> 1. Интерес	Сентябрь Январь Май	Осознанное участие ребенка в освоении образовательной программы.	<p>- <b>Низкий уровень</b> – продиктован учащемуся извне.</p> <p>- <b>Средний уровень</b> – периодически поддерживается самим учащимся.</p> <p>- <b>Высокий уровень</b> – постоянно подтверждается учащимся самостоятельно.</p>	Наблюдение
2. Интеллектуальные и творческие способности	Сентябрь Январь Май	Процесс создания и решения поставленных задач.	<p>- <b>Низкий уровень</b> – без педагога не способен привести в процесс новое, создать, самостоятельный продукт. Косность мышления.</p> <p>- <b>Средний уровень</b> – с помощью педагога находит новые пути решения поставленных задач.</p> <p>- <b>Высокий уровень</b> – самостоятельно, неординарно решает задачи, способен сам найти свой путь решения.</p>	Контрольное задание

<b>Воспитательные</b> Организация учебной деятельности	Сентябрь Январь Май	Воля, настойчивость, ответственность за выполнение заданий.	<ul style="list-style-type: none"> <li>- <b>Низкий уровень</b> – слабо развиты волевые черты характера, испытывает трудности в организации учебной деятельности.</li> <li>- <b>Средний уровень</b> – иногда затрудняется в завершении начатого дела.</li> <li>- <b>Высокий уровень</b> – волевой, настойчивый, ответственный, способен организовать и завершить процесс работы на занятии.</li> </ul>	Наблюдение
---	---------------------------	---	---	------------

### МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

№ п/п	Раздел программы или тема	Формы занятий	Приёмы и методы организации учебно-воспитательного процесса	Дидактический материал, техническое оснащение занятий	Формы подведения итогов
1.	Вводное занятие. Введение в программирование. Техника безопасности	Традиционное занятие	Методы: словесный, наглядный, фронтальный, объяснительно-иллюстративный. Приёмы: беседа, устное изложение, демонстрация презентации.	Дидактический материал: презентация, инструкции по технике безопасности. Техническое оснащение: компьютеры, проектор, интерактивная доска, IDLE Python.	Опрос, тестирование
2.	Основы Python и базовые конструкции	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: упражнения, работа с примерами кода.	Дидактический материал: примеры программ, документация Python. Техническое оснащение: компьютеры, IDLE Python.	Коллективный анализ работ
3.	Условные операторы и ветвления	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: упражнения, разбор задач.	Дидактический материал: задачи на условные операторы. Техническое оснащение: компьютеры, IDLE Python.	Самоанализ работ
4.	Циклы и их применение	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: упражнения, проектная работа.	Дидактический материал: задачи на циклы. Техническое оснащение: компьютеры, IDLE Python.	Защита проектов

5.	Работа с коллекциями данных	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: упражнения, работа с примерами.	Дидактический материал: задачи на списки, словари, множества. Техническое оснащение: компьютеры, IDLE Python.	Коллективный анализ работ
6.	Функции и модули	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: упражнения, разработка функций.	Дидактический материал: примеры функций. Техническое оснащение: компьютеры, IDLE Python.	Защита проекта
7.	Объектно-ориентированное программирование	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: упражнения, разработка классов.	Дидактический материал: примеры классов. Техническое оснащение: компьютеры, IDLE Python.	Коллективный анализ работ
8.	Работа с файлами и данными	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: упражнения, работа с файлами.	Дидактический материал: задачи на работу с файлами. Техническое оснащение: компьютеры, IDLE Python.	Защита проекта
9.	Графические интерфейсы и визуализация	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: упражнения, создание интерфейсов.	Дидактический материал: библиотеки Tkinter, Pygame. Техническое оснащение: компьютеры, IDLE Python.	Коллективный анализ работ
10.	Базы данных и веб-программирование	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: упражнения, работа с базами данных.	Дидактический материал: примеры SQL-запросов. Техническое оснащение: компьютеры, IDLE Python, Flask.	Защита проекта
11.	Анализ данных и машинное обучение	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: упражнения, работа с библиотеками NumPy, Pandas.	Дидактический материал: задачи на анализ данных. Техническое оснащение: компьютеры, IDLE Python, Scikit-Learn.	Защита проекта

12.	Итоговая проектная работа	Комбинированное занятие	Методы: словесный, наглядный, индивидуально-фронтальный, практический. Приёмы: защита проектов, демонстрация работ.	Дидактический материал: инструкции для проектов. Техническое оснащение: компьютеры, IDLE Python.	Защита проектов, выставка
13.	Заключительное занятие. Подведение итогов	Комбинированное занятие	Методы: словесный, наглядный, фронтальный. Приёмы: беседа, демонстрация лучших работ, награждение.	Дидактический материал: презентация лучших проектов. Техническое оснащение: компьютеры, проектор, интерактивная доска.	Коллективный анализ работ, выставка

## СПИСОК ЛИТЕРАТУРЫ

### *Литература для педагога:*

1. **Лутц М.** «Изучаем Python». Основное руководство по программированию на Python. Москва, «ДМК Пресс», 2020.
2. **Сузи Р.** «Программирование на Python для детей и начинающих». Практическое пособие для обучения программированию. Москва, «БХВ-Петербург», 2019.
3. **Шоу З.** «Автоматизация рутинных задач с помощью Python». Практическое руководство для начинающих разработчиков. Москва, «Вильямс», 2018.
4. **Гарантаев А.Н.** «Основы программирования в Python». Учебное пособие для внеурочной работы с детьми школьного возраста. Москва, «Просвещение», 2021.
5. **Бизли Д., Джонс Б.К.** «Python. Подробный справочник». Издательство «O'Reilly Media», 2019.
6. **Свейгарт А.** «Программируем с Minecraft. Создание мини-игр с помощью Python». Москва, «Питер», 2017.
7. **Рамальho Л.** «Fluent Python: ясный, пидомный и эффективный код». Москва, «ДМК Пресс», 2020.
8. **Перумов Д.** «Введение в машинное обучение с помощью Python». Москва, «БХВ-Петербург», 2021.
9. **Форрестер К.** «Python для детей. Самоучитель по программированию». Москва, «Эксмо», 2018.
10. **Гуглов А.С.** «Основы анализа данных с использованием Python». Практическое пособие для руководителей IT-кружков. Москва, «Просвещение», 2022.
11. **Гроувс М.** «Основы DevOps с использованием Python». Москва, «ДМК Пресс», 2021.

### *Литература для детей:*

1. **Брайсон Д.** «Программирование на Python для детей». Москва, «Эксмо», 2020.
2. **Свейгарт А.** «Программируем с Minecraft. Создание мини-игр с помощью Python». Москва, «Питер», 2017.
3. **Форрестер К.** «Python для детей. Самоучитель по программированию». Москва, «Эксмо», 2018.
4. **Гуглов А.С.** «Учимся программировать с нуля». Популярное пособие для детей и подростков. Москва, «Академия развития», 2021.
5. **Бизли Д., Джонс Б.К.** «Python. Подробный справочник для начинающих». Москва, «O'Reilly Media», 2019.



## Электронные образовательные ресурсы

Название сайта	Электронный адрес
Федеральный российский общеобразовательный портал	<a href="http://www.school.edu.ru">http://www.school.edu.ru</a>
Министерство образования и науки РФ	<a href="http://mon.gov.ru">http://mon.gov.ru</a>
Федеральный портал «Российское образование»	<a href="http://www.edu.ru">http://www.edu.ru</a>
Образовательный портал «Учеба»	<a href="http://www.uroki.ru">http://www.uroki.ru</a>
Фестиваль педагогических идей «Открытый урок» (Издательский дом «1 сентября»)	<a href="http://festival.1september.ru">http://festival.1september.ru</a>
Python.org — официальный сайт языка Python	<a href="https://www.python.org">https://www.python.org</a>
Real Python — учебные материалы по Python	<a href="https://realpython.com">https://realpython.com</a>
Codecademy — интерактивные курсы по Python	<a href="https://www.codecademy.com">https://www.codecademy.com</a>
Coursera — онлайн-курсы по программированию	<a href="https://www.coursera.org">https://www.coursera.org</a>
Stepik — образовательная платформа	<a href="https://stepik.org">https://stepik.org</a>

### **Интернет-платформы:**

1. [kontur.ru/talk](https://kontur.ru/talk) — для проведения онлайн-занятий и вебинаров.
2. [education.yandex.ru](https://education.yandex.ru) — для просмотра обучающих видео и лекций.
3. [forms.yandex.ru](https://forms.yandex.ru) — для создания тестов и опросов.
4. [gitlab.ru](https://gitlab.ru) — для хранения и управления проектами.
5. [stepik.org](https://stepik.org) — онлайн-редактор кода и платформа для практики программирования.