МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
НОВОТАРМАНСКАЯ СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА
Тюменского муниципального района

РАССМОТРЕНО
На заседании методического совета
Протокол от «29» 08 2026 г.
№ 1

УТВЕРЖДАЮ
Директор МАОУ Новотарманской СОШ
_____ И.А.Белькович
Приказ от 09 2026г.
№

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ «ОСНОВЫ ПРОГРАММИРОВАНИЕ (PYTHON)»

Возраст обучающихся: 13-14 лет
Срок реализации: 1 год

Автор-составитель: Сивкова Олеся Сергеевна,
педагог дополнительного образования

п. Новотарманский, 2026 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Нормативно-правовые основы разработки программы

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» (далее – программа) разработана в соответствии со следующими нормативными документами:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Распоряжение Правительства Российской Федерации от 31.03.2022 № 678-р «Об утверждении Концепции развития дополнительного образования детей до 2030 года»;
- Приказ Министерства просвещения Российской Федерации от 03.09.2019 № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей» (в редакции от 02.02.2021 № 38);
- Приказ Министерства просвещения Российской Федерации от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодёжи»;
- 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 (или) безвредности для человека факторов среды обитания»;
- Методические рекомендации по проектированию дополнительных общеобразовательных общеразвивающих программ / сост. О. Н. Бугаева. – Тюмень, 2024. – 56 с.;
- Устав МАОУ Новотарманская СОШ.

Направленность программы

Направленность техническая. Предмет освоения — программирование как способ создания работающих программных продуктов. Ведущий вид деятельности — техническое творчество: проектирование, написание, тестирование и отладка программ. Программа развивает алгоритмическое мышление и инженерный подход к решению задач и знакомит с профессиональной сферой информационных технологий.

Актуальность программы

Государственный заказ. Указ Президента РФ № 309 относит цифровую трансформацию и реализацию потенциала каждого человека к национальным целям развития. Концепция развития дополнительного образования детей до 2030 года ставит задачу обновлять содержание программ технической направленности и вовлекать детей в техническое творчество.

Педагогическая проблема. Между блочной средой и текстовым языком лежит разрыв: к новой логике добавляются синтаксис, типы данных и сообщения об ошибках на английском языке. В школьном курсе информатики обучающиеся группы с языком Python практически не работают, поэтому переход остаётся без сопровождения.

Запрос детей и родителей. Программа продолжает программу по Python «Основы программирование (Python)», реализуемую в организации. Её выпускники хотят развиваться дальше, участвовать в конкурсах и создавать собственные творческие проекты.

Потенциал организации. Компьютерный класс с индивидуальным рабочим местом для каждого обучающегося, интерактивная панель и доступ к образовательным платформам позволяют вести практико-ориентированное обучение.

Программа решает названную проблему: выстраивает управляемый переход от блоков к тексту.

Отличительные особенности и новизна

Программа обеспечивает преемственность со Python: конструкции Python сопоставляются со знакомыми блоками, а исполнитель turtle сохраняет наглядность результата. Сквозная проектная линия включает мини-проект в конце каждого раздела и итоговый проект года с планированием, доской задач и распределением ролей. Задания трёх уровней сложности по выбору обучающегося позволяют при одноуровневой программе подобрать нагрузку под возможности каждого.

Педагогическая целесообразность

Методы и формы работы подобраны с учётом возрастных особенностей подростков 13–14 лет и закономерностей усвоения нового материала. Потребность в общении со сверстниками и самостоятельности реализуется через парное программирование со сменой ролей, групповые мини-проекты, взаимное ревью программ и выбор уровня задания, темы и формата проекта. Становление формально-логического мышления и ограниченный объём рабочей памяти учтены в заданиях на трассировку и предсказание результата, в разборе готовых примеров и введении одной новой конструкции за занятие.

Адресат программы

Программа адресована обучающимся 13–14 лет (7–8 классы). Требования к уровню подготовки не предъявляются. Опыт визуального программирования желателен, но не обязателен: вводное занятие и первый раздел выстроены так, чтобы включиться мог и новичок. Медицинских противопоказаний, кроме общих ограничений для работы за компьютером, нет.

В этом возрасте подросток стремится к признанию сверстников, остро переживает неуспех на глазах у группы и охотнее включается в дело, результат которого можно показать. Способность к отвлечённым рассуждениям ещё формируется, поэтому абстрактные понятия вводятся через наглядный результат и собственное действие.

Условия набора и формирования групп:

- набор свободный, по заявлению родителей (законных представителей);
- группа разновозрастная, постоянного состава, до 10 человек;
- добор в течение года возможен при наличии свободных мест и готовности обучающегося;

Уровень, объём и срок освоения

- Уровень программы: стартовый.
- Объём: 72 академических часа.
- Срок освоения: 1 год, 36 учебных недель.

Форма обучения и особенности реализации

Форма обучения очная. Язык реализации — русский. Модель реализации традиционная: разделы осваиваются последовательно. Электронные образовательные ресурсы используются на очных занятиях как средство обучения. Дистанционные образовательные технологии не применяются (исключение при ухудшении эпидемиологический обстановки).

Режим занятий

- Периодичность: 1 занятие в неделю.
- Продолжительность занятия: 2 академических часа по 45 минут.

Работа за компьютером чередуется с безэкранной: трассировкой на бумаге, обсуждением, планированием у доски. Работа за компьютером занимает не более 30 минут каждого академического часа (СанПиН 1.2.3685-21, таблица 6.8). В каждой части занятия проводится гимнастика для глаз.

Формы организации и проведения занятий

Форма организации деятельности — групповая, с парной и индивидуальной работой внутри занятия. Парная работа отвечает возрастной потребности в общении и формирует умение сотрудничать. Индивидуальная работа по уровням обеспечивает собственный темп. Формы проведения занятий: эвристические беседы, практическое занятие, практикум, парное программирование, творческая мастерская, взаимное ревью программ.

ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

Цель: развитие алгоритмического мышления и формирование начальных умений программирования на языке Python у обучающихся 13–14 лет средствами технического творчества — через анализ готовых программ, создание собственных и проектную работу в команде.

Задачи.

Обучающие :

- Знакомить с базовыми понятиями и конструкциями языка Python: типы данных, переменные, ввод и вывод, ветвление, циклы, строки, списки, функции.
- Обучать чтению программ: трассировке и предсказанию результата без запуска.
- Формировать умение записывать на Python алгоритмы линейной, разветвляющейся и циклической структуры.
- Обучать приёмам тестирования и отладки программ.
- Знакомить с этапами создания программного продукта от замысла до презентации и правилами безопасной работы за компьютером.

Развивающие:

- Развивать алгоритмическое и логическое мышление: анализ, декомпозицию задачи, сравнение способов решения.
- Формировать умение планировать работу: ставить цель, делить её на шаги, сверять результат с планом.
- Развивать умение сотрудничать в паре и группе: распределять роли, договариваться, давать и принимать обратную связь.
- Развивать умение представлять результат: объяснять логику программы и выступать перед аудиторией.

Воспитательные:

- Воспитывать настойчивость и конструктивное отношение к ошибке как к рабочему этапу.

- Формировать ответственность за личный вклад в общий результат и уважение к чужому труду и авторству.
- Поддерживать устойчивый интерес к техническому творчеству и профессиям в сфере информационных технологий, уважение к достижениям отечественной науки.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

Каждой задаче программы соответствует один результат и способ его проверки. Результаты сформулированы для стартового уровня: это первичный опыт деятельности, часть которого обучающийся выполняет по образцу, часть — самостоятельно.

Предметные результаты

К концу обучения обучающийся:

– Объясняет назначение изученных конструкций Python и верно использует термины: переменная, тип данных, условие, цикл, список, функция, параметр
– Выполняет трассировку программы из 10–15 строк с ветвлением и циклом и предсказывает её вывод без запуска
– Самостоятельно пишет по текстовому условию программу из 20–30 строк с вводом данных, ветвлением, циклом, списком и собственной функцией
– Находит и исправляет синтаксические и логические ошибки в своей и чужой программе, используя сообщения интерпретатора и пошаговое выполнение
– Создаёт программный продукт, проходя этапы: замысел, план, разработка, тестирование, презентация; соблюдает правила безопасной работы за компьютером

Метапредметные результаты

К концу обучения обучающийся:

– Делит задачу на подзадачи, составляет алгоритм словесно или блок-схемой, предлагает не менее двух способов решения и сравнивает их
– Составляет план работы над проектом с задачами и сроками, ведёт доску задач, корректирует план по ходу работы
– Работает в паре в ролях «штурман» и «пилот», выполняет принятую роль в группе, даёт обратную связь по заданным критериям
– Представляет проект: объясняет назначение и логику программы, отвечает на вопросы

Личностные результаты

К концу обучения у обучающегося:

– Проявляется настойчивость: доводит задачу до работающего результата, воспринимает ошибку как рабочий этап, обращается за помощью и помогает другим

– Проявляется ответственность: соблюдает договорённости группы, отвечает за свою часть общего проекта, указывает источник заимствованного кода
– Сформирован устойчивый интерес к программированию: участвует в событиях объединения, выполняет добровольные задания, называет профессии в сфере информационных технологий и достижения отечественных учёных

УЧЕБНЫЙ ПЛАН

Объём программы — 72 часа: 17,5 часа теории (24 %) и 54,5 часа практики (76 %). Промежуточная аттестация проводится в теме 4.3, итоговая — на итоговом занятии.

№	Раздел, тема	Всего, ч	Теория, ч	Практика, ч
1	**Введение в программирование. От блоков к тексту**	**2**	**1**	**1**
2	**Линейные программы**	**16**	**4**	**12**
2.1	Вывод, типы данных и переменные. Ввод данных	4	1	3
2.2	Арифметика строк	4	1	3
2.3	Арифметика чисел	4	1	3
2.4	Практикум: повторение, задачи по выбору, мини-проект «Полезный калькулятор»	2	0,5	1,5
2.5	Среда разработки	2	0,5	1,5
3	**Ветвление**	**8**	**2**	**6**
3.1	Условный оператор и множественный выбор	4	1	3
3.2	Логические операции и вложенные условия	2	0,5	1,5
3.3	Мини-проект «Текстовый квест»	2	0,5	1,5
4	**Циклы и графика turtle**	**12**	**3**	**9**
4.1	Исполнитель turtle и цикл for	4	1	3
4.2	Цикл while и вложенные циклы	4	1	3
4.3	Мини-проект. Промежуточная аттестация	2	0,5	1,5

№	Раздел, тема	Всего, ч	Теория, ч	Практика, ч
4.4	Практикум: повторение, отладка, задачи по выбору	2	0,5	1,5
5	**Строки и списки**	**10**	**2,5**	**7,5**
5.1	Строки: индексы, срезы, методы	2	0,5	1,5
5.2	Списки	4	1	3
5.3	Мини-проект «Викторина» или «Шифровальщик»	2	0,5	1,5
5.4	Практикум: повторение, отладка, задачи по выбору	2	0,5	1,5
6	**Функции и декомпозиция**	**8**	**2**	**6**
6.1	Функции: параметры и возврат значения	4	1	3
6.2	Декомпозиция задачи	2	0,5	1,5
6.3	Мини-проект «Вторая версия»	2	0,5	1,5
7	**Проектная мастерская**	**14**	**2,5**	**11,5**
7.1	Замысел, планирование и проектирование	4	1,5	2,5
7.2	Разработка проекта	4	0	4
7.3	Тестирование, взаимное ревью и доработка	4	0,5	3,5
7.4	Подготовка презентации и предзащита	2	0,5	1,5
8	**Итоговое занятие.**	**2**	**0,5**	**1,5**
	Итого	**72**	**17,5**	**54,5**

СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА

1. Введение в программирование. От блоков к тексту (2 ч)

Теория (1 ч). Цели и содержание программы. Правила работы объединения. Инструктаж по технике безопасности и правилам поведения в компьютерном классе. Блочные и текстовые языки программирования. Язык Python: области применения, примеры созданных на нём продуктов. Соответствие блоков Python и команд Python.

Практика (1 ч). Выработка правил группы «Кодекс команды». Задание «Мост»: сопоставление скрипта Python и программы на Python, предсказание результата. Запуск и изменение готовой программы в онлайн-редакторе.

2. Линейные программы (16 ч)

2.1. Вывод, типы данных и переменные. Ввод данных (4 ч)

Теория (1 ч). Функция print(), строки и числа, параметры sep и end. Комментарии. Переменная, оператор присваивания, правила именования. Типы данных int, float, str. Функция input(), преобразование типов. Синтаксическая ошибка, чтение сообщения интерпретатора. Таблица трассировки.

Практика (3 ч). Предсказание вывода готовой программы, запуск, изменение. Программа-визитка, анкета с вводом данных. Трассировка программ с присваиванием, обмен значений переменных. Задачи с автопроверкой.

2.2. Арифметика строк (4 ч)

Теория (1 ч). Сцепление и повторение строк. Длина строки. Форматированный вывод (f-строки). Ошибка при сложении строки и числа.

Практика (3 ч). Рамки и узоры из символов, генератор приветствий и подписей. Сборка программы из перемешанных строк. Задания «Найди ошибку» на несоответствие типов.

2.3. Арифметика чисел (4 ч)

Теория (1 ч). Арифметические операции, целочисленное деление и остаток, возведение в степень, приоритет операций. Целые и дробные числа, округление.

Практика (3 ч). Задачи: расчёты времени и стоимости, перевод единиц, цифры числа. Трассировка вычислений. Задачи с автопроверкой.

2.4. Практикум: повторение, задачи по выбору (2 ч)

Теория (0,5 ч). Разбор типичных ошибок раздела: несоответствие типов, пропущенное преобразование ввода, опечатки в именах переменных.

Практика (1,5 ч). Решение задач трёх уровней сложности по выбору. Мини-проект «Полезный калькулятор» на выбор: возраст питомца, конвертер валют, время в пути.

2.5. Среда разработки (2 ч)

Теория (0,5 ч). Зачем нужна среда разработки на своём компьютере. Установка среды разработки. Редактор и оболочка, сохранение и запуск программы, пошаговое выполнение с просмотром значений переменных.

Практика (1,5 ч). Совместная установка среды разработки. Перенос своей программы, запуск и пошаговое выполнение. Первое знакомство с графическим исполнителем: запуск готовой программы «Радужная спираль», предсказание и проверка результата после изменения двух чисел.

3. Ветвление (8 ч)

3.1. Условный оператор и множественный выбор (4 ч)

Теория (1 ч). Логический тип bool. Операции сравнения. Условный оператор if, полная форма if–else. Отступ как часть синтаксиса. Множественный выбор elif. Блок-схема ветвления. Сравнение с блоками «если» в Python.

Практика (3 ч). Трассировка программ с ветвлением для разных входных данных. Запись программы по блок-схеме и составление блок-схемы по программе. Задачи: чётность числа, проверка пароля, оценка по баллам, светофор.

3.2. Логические операции и вложенные условия (2 ч)

Теория (0,5 ч). Операции and, or, not, таблицы истинности. Вложенные условия. Логическая ошибка и её отличие от синтаксической.

Практика (1,5 ч). Задания «Найди ошибку» с логическими ошибками. Задачи: високосный год, принадлежность числа промежутку, вид треугольника. Два способа решения одной задачи и их сравнение в парах.

3.3. Мини-проект «Текстовый квест» (2 ч)

Теория (0,5 ч). Парное программирование: роли «штурман» и «пилот», правила смены ролей. Схема сюжета как дерево решений. Критерии оценки мини-проекта.

Практика (1,5 ч). Разработка квеста в парах: схема сюжета, программа, проверка всех веток. Обмен квестами между парами, взаимная оценка по критериям.

4. Циклы и графика turtle (12 ч)

4.1. Исполнитель turtle и цикл for (4 ч)

Теория (1 ч). Модуль и его подключение. Исполнитель Черепаха: команды движения, поворота, управления пером и цветом. Цикл for, функция range() с одним, двумя и тремя аргументами, тело цикла. Сравнение с блоком «повторить» в Python.

Практика (3 ч). Предсказание рисунка по программе. Многоугольники, звёзды, орнаменты. Изменение готовой программы для получения заданного рисунка. Выставка работ «Галерея узоров».

4.2. Цикл while и вложенные циклы (4 ч)

Теория (1 ч). Цикл while, условие продолжения. Счётчик и накопитель. Бесконечный цикл, оператор break. Модуль random, случайное число. Вложенные циклы.

Практика (3 ч). Заполнение таблицы трассировки цикла. Задачи: сумма чисел, обратный отсчёт, проверка корректности ввода, таблица умножения. Каркас игры «Угадай число».

4.3. Мини-проект. Промежуточная аттестация (2 ч)

Теория (0,5 ч). Требования к мини-проекту и критерии оценки. Порядок аттестации.

Практика (1,5 ч). Индивидуальное контрольное задание на трассировку. Доработка и защита мини-проекта на выбор: игра «Угадай число» с подсказками и счётчиком попыток или генератор узоров.

4.4. Практикум: повторение, отладка, задачи по выбору (2 ч)

Теория (0,5 ч). Разбор результатов промежуточной аттестации и типичных ошибок: бесконечный цикл, ошибка на единицу в границах range().

Практика (1,5 ч). Решение задач трёх уровней сложности по выбору. Задания «Найди ошибку». Индивидуальная работа над затруднениями.

5. Строки и списки (10 ч)

5.1. Строки: индексы, срезы, методы (2 ч)

Теория (0,5 ч). Строка как последовательность символов. Индексы и срезы, проверка вхождения. Методы строк: изменение регистра, замена, поиск, подсчёт. Перебор символов в цикле. Сцепление, повторение и длина строки изучены в теме 2.2.

Практика (1,5 ч). Трассировка программ со срезами. Задачи: палиндром, подсчёт гласных, шифр Цезаря.

5.2. Списки (4 ч)

Теория (1 ч). Список, индексы, изменение элементов. Добавление и удаление элементов, сортировка. Функции len(), sum(), min(), max(). Перебор списка в цикле. Преобразование строки в список и обратно. Случайный выбор элемента.

Практика (3 ч). Задачи: список покупок, поиск наибольшего элемента без встроенной функции, отбор элементов по условию, статистика оценок. Задания «Найди ошибку»: выход за границы списка.

5.3. Мини-проект «Викторина» или «Шифровальщик» (2 ч)

Теория (0,5 ч). Роли в группе из 2–3 человек. Короткий план работы: что делаем, кто отвечает, как проверяем.

Практика (1,5 ч). Разработка мини-проекта в группах: викторина с вопросами в списке и подсчётом баллов или программа шифрования и расшифровки сообщений. Взаимное тестирование.

5.4. Практикум: повторение, отладка, задачи по выбору (2 ч)

Теория (0,5 ч). Разбор типичных ошибок раздела: индексация с нуля, неизменяемость строк, изменение списка во время перебора.

Практика (1,5 ч). Решение задач трёх уровней сложности по выбору. Сборка программ из перемешанных строк. Доработка мини-проекта.

6. Функции и декомпозиция (8 ч)

6.1. Функции: параметры и возврат значения (4 ч)

Теория (1 ч). Назначение функций: устранение повторов, понятное имя для действия. Определение и вызов функции. Параметры и аргументы. Возврат значения, отличие return от print(). Локальные переменные. Сравнение с блоками «Другие блоки» в Python.

Практика (3 ч). Трассировка программ с вызовами функций. Выделение повторяющегося кода в функцию. Задачи: функции проверки, расчёта, рисования фигуры с параметрами.

6.2. Декомпозиция задачи (2 ч)

Теория (0,5 ч). Нисходящее проектирование: от задачи к подзадачам и функциям. Главная программа. Понятные имена как документация.

Практика (1,5 ч). Кейс: разбиение игры на функции в группах, сравнение вариантов разбиения. Составление схемы программы до написания кода.

6.3. Мини-проект «Вторая версия» (2 ч)

Теория (0,5 ч). Улучшение программы без изменения её поведения. Критерии читаемой программы.

Практика (1,5 ч). Переработка своего мини-проекта из прошлых разделов: выделение функций, добавление одной новой возможности. Сравнение первой и второй версии.

7. Проектная мастерская (14 ч)

Обязательный результат раздела — работающая программа с текстовым интерфейсом или графикой turtle. Оконный интерфейс на tkinter — вариативный трек по выбору обучающегося: он осваивается по готовому шаблону, и его отсутствие не снижает оценку проекта.

7.1. Замысел, планирование и проектирование (4 ч)

Теория (1,5 ч). Этапы создания программного продукта. Требования к проекту и минимальная работающая версия. Доска задач: нужно сделать, в работе, готово. Роли в команде. Критерии оценки итогового проекта. Вариативный трек «Оконное приложение»: библиотека tkinter, окно, надпись, кнопка, поле ввода, функция-обработчик нажатия.

Практика (2,5 ч). Мозговой штурм идей. Выбор формата работы: индивидуально или в группе из 2–3 человек. Паспорт проекта. План с задачами и сроками. Алгоритм и схема функций. Первый работающий прототип. Для выбравших вариативный трек: запуск и изменение шаблона оконной программы.

7.2. Разработка проекта (4 ч)

Практика (4 ч). Разработка по плану. Короткий отчёт в начале занятия: что сделано, что в работе, что мешает. Ведение доски задач и корректировка плана. Консультации педагога. Вариативный трек: перенос логики проекта в оконный интерфейс на основе шаблона.

7.3. Тестирование, взаимное ревью и доработка (4 ч)

Теория (0,5 ч). Тестовый сценарий, граничные значения. Правила доброжелательной обратной связи.

Практика (3,5 ч). Тестирование своей программы по чек-листу. Обмен проектами, заполнение листа ревью. Исправление найденных ошибок, доработка по замечаниям.

7.4. Подготовка презентации и предзащита (2 ч)

Теория (0,5 ч). Структура выступления: задача, решение, демонстрация, чему научился.

Практика (1,5 ч). Подготовка выступления и демонстрации. Предзащита в малых группах, доработка по замечаниям.

8. Итоговое занятие. Фестиваль проектов (2 ч)

Теория (0,5 ч). Итоги года. Перспективы второго года обучения.

Практика (1,5 ч). Публичная защита проектов перед [приглашённые: родители, младшая группа, педагоги]. Итоговое задание на предсказание результата программы. Итоговая анкета, рефлексия.

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Программа реализуется в течение 36 учебных недель: одно занятие в неделю, 72 часа в год.

Год обучения	Дата начала обучения	Дата окончания обучения	Всего учебных недель	Количество учебных часов	Режим занятий
1	03.09.2026	30.05.2027	36	72	1 раз в неделю по 2 академических часа

ФОРМЫ АТТЕСТАЦИИ

Результаты отслеживаются тремя видами контроля; формы совпадают с указанными в учебном плане.

Вид контроля	Сроки	Цель	Формы
Входной	Занятие 1	Опыт программирования, мотивация, готовность читать текстовый код	Анкета, задание «Мост»
Промежуточная аттестация	Занятие 18	Предметные результаты по разделам 2–4	Контрольное задание на трассировку, защита мини-проекта
Итоговая аттестация	Занятие 36	Предметные, метапредметные и личностные результаты за год	Защита итогового проекта, итоговое задание на трассировку, анкета мотивации

Уровни освоения программы. Итоговый результат складывается из баллов за защиту проекта (до 20) и итоговое задание на трассировку (до 6).

Уровень	Баллы из 26	Характеристика
Высокий	21–26	Самостоятельно читает и пишет программы с изученными конструкциями, объясняет любую часть своего проекта
Средний	13–20	Читает и пишет программы с опорой на образец или подсказку, объясняет основную логику проекта
Низкий	0–12	Выполняет задания по образцу с помощью педагога, затрудняется объяснить работу программы

РАБОЧАЯ ПРОГРАММА ВОСПИТАНИЯ

Цель: создание условий для развития у подростков ценностного отношения к труду и знаниям, к партнёрам по команде, к результатам своей и чужой интеллектуальной работы и к достижениям отечественной науки.

В подростковом возрасте приоритетом воспитания является развитие социально значимых отношений. Поэтому воспитательная работа встроена в учебную деятельность: отношения формируются в парной и командной работе, а не на отдельных беседах.

Задачи воспитания совпадают с воспитательными задачами программы.

Формы и содержание деятельности

Направление	Формы	Содержание
Трудовое воспитание	Разбор ошибок, мини-проекты, проектная мастерская	Доведение программы до работающего состояния, тестирование, доработка по замечаниям
Духовно-нравственное воспитание	«Кодекс команды», парное программирование, взаимное ревью, беседа «Честный код»	Правила совместной работы, доброжелательная обратная связь, уважение к авторству, честное указание источников
Гражданско-патриотическое воспитание, популяризация научных знаний	Тематические занятия ко Дню информатики и Дню российской науки	Вклад отечественных учёных и инженеров в развитие вычислительной техники и программирования
Профессиональное самоопределение	«Урок цифры», роли в проектной команде, олимпиады и конкурсы	Знакомство с профессиями разработчика, тестировщика, руководителя проекта через собственный опыт
Социальная активность	Мастер-класс для младшей группы, фестиваль проектов с участием родителей	Передача опыта младшим, публичное предъявление результата своего труда

КАЛЕНДАРНЫЙ ПЛАН ВОСПИТАТЕЛЬНОЙ РАБОТЫ

№	Название мероприятия, события	Форма проведения	Сроки проведения
1	«Кодекс команды»	Совместная выработка правил группы	Занятие 1
2	Школьный этап всероссийской олимпиады школьников по информатике	Участие по желанию, разбор задач на практикуме	Сентябрь–октябрь, по графику региона
3	Всероссийский образовательный проект «Урок цифры»	Тематическое занятие с онлайн-тренажёром	по расписанию проекта на 2026/27 учебный год
4	День информатики в России (4 декабря)	Мастер-класс «Первая программа на Python» для младшей группы	03.12.2026
5	День российской науки (8 февраля)	Мини-выступления «Имена российской информатики»	11.02.2027
6	«Честный код»	Беседа об авторстве и заимствовании кода	01.04.2027
7	(совместно выбрать конкурс в котором можно поучаствовать)	Участие с мини-проектами и итоговыми проектами	[сроки конкурса]
8	Фестиваль проектов	Публичная защита	занятие 35

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Материально-техническое обеспечение

Занятия проводятся в компьютерном классе организации, оборудованном в соответствии с санитарно-эпидемиологическими требованиями.

Позиция	Требования / количество
Помещение	Компьютерный класс
Рабочие места обучающихся	10 рабочих мест: персональный компьютер или ноутбук, мышь, клавиатура, стол и стул, соответствующие росту обучающихся
Рабочее место педагога	1 компьютер с мультимедийным проектором (или интерактивной доской / панелью) для демонстрации материала
Программное обеспечение	Актуальный браузер; доступ к онлайн-платформе Python; операционная система, текстовый и графический редакторы

Позиция	Требования / количество
Доступ к информационным системам	Стабильное подключение к сети Интернет (для всех рабочих мест) для доступа к онлайн-платформе Python и электронным образовательным ресурсам
Дополнительно	Наушники или колонки (по необходимости); раздаточные материалы (технологические карты, карточки заданий)

Программное обеспечение на каждом рабочем месте:

- среда разработки со встроенным интерпретатором Python 3 (библиотеки turtle и tkinter входят в стандартную поставку);
- браузер для работы с образовательной платформой.

Информационное *обеспечение*

- Образовательная платформа: учётная запись педагога и учётные записи всех обучающихся, раздел по программированию на Python как банк задач с автоматической проверкой.
- Официальная документация Python.

Кадровое обеспечение

Программу реализует педагог дополнительного образования, соответствующий требованиям профессионального стандарта «Педагог дополнительного образования детей и взрослых» (приказ Минтруда России от 22.09.2021 № 652н): высшее образование или среднее профессиональное образование по направлению, соответствующему направленности программы (информатика, информационные технологии, педагогическое образование), либо иное образование при условии освоения дополнительной профессиональной программы в области педагогики. Педагог владеет Python на уровне, достаточном для разработки и отладки проектов, методикой проектной деятельности, регулярно повышает квалификацию.

СПИСОК ЛИТЕРАТУРЫ

Для педагога

1. Буйлова, Л. Н. Дополнительная общеразвивающая программа: практическое руководство по проектированию и дизайну : методическое пособие / Л. Н. Буйлова. – Москва : Народное образование, 2023. – 162 с. – Текст : непосредственный.
2. Выготский, Л. С. Педагогическая психология / Л. С. Выготский ; под редакцией В. В. Давыдова. – Москва : Педагогика, 1991. – 480 с. – Текст : непосредственный.

3. Методические рекомендации по проектированию дополнительных общеобразовательных общеразвивающих программ / составитель О. Н. Бугаева. – Тюмень : Дворец творчества и спорта «Пионер», 2023. – 56 с. – (Информационно-методический сборник Регионального модельного центра дополнительного образования детей Тюменской области ; выпуск 7). – Текст : непосредственный.
4. Поляков, К. Ю. Программирование. Python. C++. Часть 1 : учебное пособие / К. Ю. Поляков. – 7-е изд., стер. – Москва : Просвещение, 2025. – 144 с. – Текст : непосредственный.
5. Lister, R. A multi-national study of reading and tracing skills in novice programmers / R. Lister, E. S. Adams, S. Fitzgerald [et al.]. – Текст : непосредственный // ACM SIGCSE Bulletin. – 2004. – Vol. 36, № 4. – P. 119–150.

Для обучающихся

1. Бриггс, Д. Python для детей. Самоучитель по программированию / Джейсон Бриггс ; перевод с английского. – Москва : Манн, Иванов и Фербер, 2018. – 320 с. – Текст : непосредственный.
2. Вордерман, К. Программирование на Python. Иллюстрированное руководство для детей / Кэрол Вордерман ; перевод с английского С. Ломакина. – Москва : Манн, Иванов и Фербер, 2018. – 224 с. – Текст : непосредственный.
3. Свейгарт, Э. Учим Python, делая крутые игры / Эл Свейгарт ; перевод с английского. – Москва : Бомбора, 2021. – 416 с. – Текст : непосредственный.

Для родителей

1. Пэйн, Б. Python для детей и родителей / Брайсон Пэйн ; перевод с английского. – 2-е изд. – Москва : Бомбора, 2021. – 352 с. – Текст : непосредственный.