Министерство общего и профессионального образования Ростовской области
Государственное бюджетное учреждение дополнительного образования Ростовской области «Областной центр технического творчества учащихся»
Детский технопарк «Кванториум»

Утверждаю
Директор ГБУ ДО РО ОЦТТУ
_____ Арутюнова Р.Г.

Принята
на заседании педагогического совета
Детского технопарка «Кванториум»
Ростовской области
от «31» августа 2026 г.
Протокол от 31.08.2026 № 147/2

Согласовано директором
(руководителем структурного подразделения)
Детского технопарка «Кванториум»
Ростовской области

_____ Кобизь Л.Н.

Дополнительная общеобразовательная (общеразвивающая) программа технической направленности вводного модуля «IT-квантум» по теме «Практическое программирование на Python: решение алгоритмических задач»

Авторы-составители: Колегаева Анастасия Ивановна, п.д.о. СП ДТ «Кванториум» ГБУ ДО РО ОЦТТУ

Возрастной состав учащихся: 10-17 лет

Объем учебной нагрузки: 72 часа

г. Ростов-на-Дону
2026 г.

Оглавление

I. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА 3
II. СОДЕРЖАНИЕ ПРОГРАММЫ 6
III. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ 9
IV. ФОРМЫ АТТЕСТАЦИИ 10
V. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ 15
VI. ВОСПИТАТЕЛЬНЫЙ КОМПОНЕНТ 20
VII. УЧЕБНЫЙ ПЛАН 23
VIII. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК 24
IX. МАТЕРИАЛЬНО-ТЕХНИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ 26
ПРОГРАММЫ 26
Список литературы 27

I. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Предметом IT как учебной дисциплины является создание и применение цифровых продуктов. Программирование даёт школьнику возможность отработать навыки алгоритмизации, работы с данными и разработки приложений. В рамках проектной деятельности в IT-квантуме ученики проводят исследования, анализируют задачи и создают работающие программы. Кроме того, IT - это сфера, где требуется командная работа, навыки коммуникации и умение отстаивать свою точку зрения. Работа над проектами учит планировать время и распределять задачи. Итог — презентация и защита групповых проектов, что создаёт ситуацию успеха и развивает навыки публичных выступлений.

Ведущая идея программы - создание практико-ориентированной IT-экосистемы, где обучение строится через решение реальных задач и разработку проектов. Деятельность выбирается ребёнком свободно, а теория сразу проверяется практикой.

Ключевые понятия:

Программирование — процесс создания программ для решения задач с помощью написания кода. Python — один из самых востребованных языков благодаря простоте и широкому спектру применения.

Алгоритм — точная последовательность шагов, описывающая решение задачи.

Структуры данных — способы организации данных (списки, словари, множества).

IT-проект — ограниченное по времени мероприятие по созданию уникального цифрового продукта.

Интерфейс — часть программы, с которой взаимодействует пользователь.

Данные — информация, пригодная для обработки компьютером.

Искусственный интеллект — область информатики, посвящённая созданию обучаемых алгоритмов.

Дополнительная общеобразовательная общеразвивающая программа «Практическое программирование на Python: решение алгоритмических задач» имеет ***техническую направленность.***

Уровень освоения программы – базовый.

Программа разработана в соответствии с:

— Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

— Приказом Минпросвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

— Концепцией развития дополнительного образования детей до 2030 года (распоряжение Правительства РФ от 31.03.2022 № 678-р);

— Указом Президента РФ от 09.11.2022 № 809 «Об утверждении Основ государственной политики по сохранению и укреплению традиционных

российских духовно-нравственных ценностей»;

— Указом Президента РФ от 07.05.2024 № 309 «О национальных целях развития Российской Федерации на период до 2030 года и на перспективу до 2036 года»;

— Федеральным проектом «Педагоги и наставники» национального проекта «Молодежь и дети».

Актуальность программы обусловлена высоким спросом на специалистов в сфере IT. Python используется в веб-разработке, Data Science, искусственном интеллекте и автоматизации.

Педагогическая целесообразность программы заключается в том, что после её освоения обучающиеся получат прочную базу для продолжения изучения программирования. Сочетание индивидуальной и командной работы над проектами, а также развитие soft-skills через освоение hard-skills позволит сформировать целостную систему знаний, умений и навыков.

Практическая значимость программы — формирование навыков работы с современными IT-технологиями и языком Python. Учащиеся освоят написание кода, основы проектной деятельности, что обеспечит осознанное применение знаний в собственных проектах.

Принципы отбора содержания:

- Интеграция обучения и практики;
- Постепенное усложнение;
- Доступность и наглядность;
- Проектная коллаборация;
- Междисциплинарность;
- Индивидуальная траектория.

Отличительная особенность программы - проектно-ориентированный подход с акцентом на решение алгоритмических задач и создание работающих приложений.

Цель программы - сформировать у учащихся базовые навыки программирования на Python через решение широкого круга алгоритмических задач и создание проектов.

Задачи:

Образовательные:

- сформировать устойчивые знания синтаксиса и базовых конструкций Python;
- обучить основным структурам данных и алгоритмам работы с ними;
- дать понимание объектно-ориентированного программирования;
- научить создавать консольные и графические приложения, простые веб-серверы;
- обучить работе с внешними библиотеками.

Развивающие:

- развить алгоритмическое и логическое мышление;
- развить способность к самообучению и поиску информации;

- сформировать техническое мышление и творческий подход.

Воспитательные:

- сформировать коммуникативную культуру, умение аргументировать свою точку зрения;
- воспитать трудолюбие, целеустремлённость, умение доводить дело до конца;
- сформировать способности к продуктивному сотрудничеству.

Программа предназначена для детей в возрасте ***10–17 лет***.

Особенности организации образовательного процесса: индивидуальные, групповые, фронтальные формы работы. Состав групп 10–15 человек.

Форма обучения – очная.

Общее количество часов — 72 часа. Продолжительность занятий — 45 минут (академический час), между занятиями — 10-минутные перемены. Недельная нагрузка — 6 часов (2 раза в неделю). Срок освоения — 1 год.

Организация учебного процесса. Каждое занятие включает три этапа: краткое теоретическое введение (объяснение необходимого минимума для решения задач), основная практическая часть (решение задач на компьютере) и рефлексия. Основное время отводится на решение задач по программированию.

Методы обучения: метод проектов, игровые технологии, групповые дискуссии, практическое моделирование. Программа предусматривает участие в конкурсах и хакатонах.

Программа разработана в соответствии с приоритетами государственной политики в сфере образования, установленными указами Президента и национальными проектами.

Приоритетные направления обновления содержания и технологий:

— проектная и исследовательская деятельность;

— цифровые технологии и практико-ориентированное обучение;

— ранняя профориентация в IT-сфере;

— наставничество и командная работа;

— воспитание ответственного поведения в цифровой среде.

Воспитательный компонент реализуется через:

— формирование интереса к IT-сфере и профессиональное самоопределение;

— воспитание трудолюбия, целеустремлённости, умения доводить дело до конца;

— развитие коммуникативной культуры и навыков продуктивного сотрудничества;

— социально значимые проекты и участие в конкурсах, хакатонах, проектных защитах.

II. СОДЕРЖАНИЕ ПРОГРАММЫ

Тема 1. Введение в программирование (3 ч.)
История языка Python
Сильные и слабые стороны Python
Python 2 VS Python 3
Установка Python на компьютер
Установка VS Code и (или) Wing IDE на компьютер

Тема 2. Ввод-вывод данных, переменные (3 ч.)
Вывод данных, команда print()
Ввод данных, команда input()
Переменные
Частые ошибки

Тема 3. Sep, end, PEP 8 (3 ч.)
Необязательные параметры команды print(): sep, end
Множественное присваивание
Стандарт PEP 8
Комментарии

Тема 4. Работа с целыми числами (3 ч.)
Целочисленный тип данных
Преобразование строки к целому числу
Операции над целыми числами
Частые ошибки

Тема 5. Работа с целыми числами (3 ч.)
Операция возведения в степень (**)
Операция нахождения остатка (%)
Операция целочисленного деления (//)
Обработка цифр числа
Частые ошибки
Решение задач

Тема 6. Итоговая работа на ввод-вывод данных. Часть 1 (3 ч.)

Тема 7. Итоговая работа на ввод-вывод данных. Часть 2 (3 ч.)

Тема 8. Условный оператор (3 ч.)
Условный оператор
Отступы
Операторы сравнения
Цепочки сравнений

Решение задач
Частые ошибки

Тема 9. Логические операторы (3 ч.)
Логическое умножение and
Логическое сложение or
Логическое отрицание not
Решение задач
Частые ошибки

Тема 10. Условный оператор (3 ч.)
Вложенные условия
Каскадные условия
Решение задач

Тема 11. Итоговая работа на условный оператор. Часть 1 (3 ч.)

Тема 12. Числовые типы данных (3 ч.)
Целочисленный тип данных int
Числа с плавающей точкой float
Встроенные функции max(), min(), abs()
Решение задач

Тема 13. Строковый тип данных (3 ч.)
Строковый тип данных str
Функции len() и str()
Конкатенация строк
Умножение строки на число
Оператор in

Тема 14. Модуль math (3 ч.)
Модуль math
Решение задач

Тема 15. Цикл for (3 ч.)
Цикл for
Переменная цикла
Решение задач

Тема 16. Цикл for (3 ч.)
Цикл for
Функции range() с одним параметром
Функции range() с двумя параметрами
Функции range() с тремя параметрами

Решение задач

Тема 17. Частые сценарии (3 ч.)
Подсчёт количества
Вычисление суммы и произведения
Определение максимума и минимума
Обмен значений переменных
Расширенные операторы присваивания (+=, -=, //= и т.д.)
Сигнальные метки

Тема 18. Количество чисел (3 ч.)
Решение задач

Тема 19. Цикл while (3 ч.)
Цикл while
Считывание данных до стоп значения
Бесконечный цикл

Тема 20. Обработка цифр числа (3 ч.)
Обработка цифр числа с помощью цикла while
Обработка цифр числа с помощью цикла for

Тема 21. Break, continue, else (3 ч.)
Оператор break
Оператор continue
Бесконечные циклы
Ключевое слово else в циклах
Решение задач

Тема 22. Ревью кода (3 ч.)
Ревью кода
Поиск ошибок
Производительность кода

Тема 23. Вложенные циклы (3 ч.)
Вложенные циклы
Операторы break и continue во вложенных циклах
Решение задач

Тема 24. Делители-1 (3 ч.)
Решение задач

III. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

Личностные:

- формирование интереса к IT-сфере;
- развитие патриотического сознания через социально значимые проекты;
- воспитание ответственности.

Метапредметные:

- умение работать в команде;
- развитие алгоритмического и критического мышления;
- навыки презентации и защиты своих идей.

Образовательные (предметные):

- знание синтаксиса и базовых конструкций Python;
- умение работать со структурами данных (списки, словари, множества, строки);
- умение создавать функции, классы, работать с файлами;
- способность разрабатывать консольные, GUI и веб-приложения на Python;
- владение навыками отладки и тестирования кода.

Механизм оценивания:

- текущий контроль – решение задач на каждом занятии, тесты, практические работы;
- промежуточная аттестация — темы 6, 7, 11, 18, 24: защита мини-проектов, практические работы, тестирование;
- итоговая аттестация — итоговый проект и его защита.

Критерии оценки:

- техническое исполнение;
- сложность и оптимальность решения;
- читаемость и качество кода;
- качество презентации;
- умение отвечать на вопросы;
- самостоятельность выполнения.

IV. ФОРМЫ АТТЕСТАЦИИ

1. Общие положения

Аттестация обучающихся по дополнительной общеобразовательной общеразвивающей программе «Практическое программирование на Python: решение алгоритмических задач» (вводный модуль, 72 часа) проводится в соответствии с:

- Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Минпросвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- локальными актами ГБУ ДО РО ОЦТТ / Детского технопарка «Кванториум».

Цель аттестации — установить соответствие результатов освоения программы планируемым результатам, определить уровень сформированности предметных, метапредметных и личностных компетенций обучающихся.

Задачи аттестации:

- оценить качество освоения теоретического материала и практических навыков программирования на Python;
- выявить уровень алгоритмического и логического мышления;
- определить динамику индивидуального развития каждого обучающегося;
- создать ситуацию успеха и мотивацию к дальнейшему изучению IT-технологий;
- получить объективные данные для корректировки образовательного процесса.

Виды аттестации:

- текущий контроль;
- промежуточная аттестация;
- итоговая аттестация.

Формы аттестации:

- тестирование;
- решение задач;
- практическая работа;
- защита мини-проекта;
- итоговый проект и его защита;
- педагогическое наблюдение;
- анализ продуктов деятельности.

Принципы аттестации:

- систематичность и регулярность;
- объективность и прозрачность;
- дифференцированный подход;
- практико-ориентированность;

- сочетание индивидуальных и групповых форм;
- учёт возрастных особенностей (10–17 лет).

2. Текущий контроль

Цель: проверка усвоения текущего материала, своевременное выявление затруднений и корректировка образовательного маршрута.

Периодичность: на каждом занятии.

Формы текущего контроля:

- устный опрос;
- решение задач по теме занятия;
- выполнение практических заданий на компьютере;
- тестирование по отдельным темам;
- педагогическое наблюдение;
- самопроверка и взаимопроверка кода;
- рефлексия по итогам занятия.

Содержание текущего контроля:

- знание синтаксиса и базовых конструкций Python;
- умение применять команды print(), input(), работать с переменными;
- владение операциями над целыми числами, строками, списками;
- умение использовать условный оператор, логические операторы, циклы for и while;
- навыки отладки и поиска ошибок;
- способность решать алгоритмические задачи.

Результат: текущий контроль не предполагает выставления отметок в балльной системе, но фиксируется в листе наблюдений педагога. Обучающийся получает обратную связь и рекомендации.

3. Промежуточная аттестация

Цель: оценка результатов освоения блоков программы и уровня сформированности ключевых компетенций.

Периодичность: по итогам изучения блоков и тем, указанных в учебном плане.

Формы промежуточной аттестации:

№ темы	Название темы	Форма промежуточной аттестации
1	Введение в программирование	Тестирование
6	Итоговая работа на ввод-вывод данных. Часть 1	Тестирование
7	Итоговая работа на ввод-вывод данных. Часть 2	Решение задач
11	Итоговая работа на условный оператор	Решение задач
17	Частые сценарии	Тестирование
18	Количество чисел	Практическая работа
24	Делители-1	Практическая работа

Содержание промежуточной аттестации:

- *Тестирование* включает 10–15 вопросов закрытого и открытого типа по пройденному материалу. Время выполнения — 20–25 минут. Критерии: 80–100% — высокий уровень, 50–79% — средний, менее 50% — низкий.
- *Решение задач* предполагает самостоятельное написание программ по условию. Оценивается корректность, оптимальность, читаемость кода,

самостоятельность.

- *Практическая работа* включает разработку мини-программы или решение комплексной задачи с применением изученных конструкций. Оценивается техническое исполнение, сложность, качество презентации результата.

Результат: промежуточная аттестация фиксируется в протоколе и оценочном листе обучающегося. Уровень освоения определяется как высокий, средний или низкий.

4. Итоговая аттестация

***Цель*:** определение уровня достижения планируемых результатов по завершении вводного модуля.

***Форма*:** итоговый проект и его защита.

***Срок проведения*:** последнее занятие модуля (тема 24 или отдельное занятие по расписанию).

***Требования к итоговому проекту*:**

- проект должен быть выполнен на языке Python;
- демонстрировать владение базовыми конструкциями: ввод-вывод, переменные, условия, циклы, строки, списки;
- иметь практическую значимость или игровую направленность;
- сопровождаться краткой презентацией (3–5 слайдов);
- защита проекта — 5–7 минут, включая ответы на вопросы.

Примерные темы итоговых проектов:

- «Калькулятор для решения задач по математике»;
- «Программа-профориентатор “Твой путь в IT”»;
- «Игра “Угадай число”»;
- «Помощник для подсчёта оценок»;
- «Генератор случайных задач по Python»;
- «Программа для обработки результатов соревнований».

Критерии оценивания итогового проекта:

№	Критерий	Баллы
1	Техническое исполнение (работоспособность, отсутствие ошибок)	0–5
2	Сложность и оригинальность решения	0–5
3	Читаемость и структурированность кода	0–5
4	Качество презентации	0–5
5	Умение отвечать на вопросы	0–5
6	Самостоятельность выполнения	0–5
Итого		**0–30**

Уровни:

- 25–30 баллов — высокий;
- 15–24 балла — средний;
- менее 15 баллов — низкий.

Результат: итоговая аттестация фиксируется в протоколе итоговой аттестации. Обучающимся, успешно прошедшим итоговую аттестацию, выдаётся сертификат или диплом об окончании вводного модуля (в

соответствии с локальными актами организации).

5. Критерии оценивания

Критерии оценивания предметных результатов:

- знание синтаксиса Python;
- умение работать с переменными, типами данных;
- владение условными операторами и логическими выражениями;
- умение использовать циклы for и while;
- навыки работы со строками, списками, словарями;
- способность решать алгоритмические задачи;
- умение отлаживать и тестировать код.

Критерии оценивания метапредметных результатов:

- умение планировать свою деятельность;
- способность работать в команде;
- навыки поиска информации и самообучения;
- умение презентовать результаты своей работы;
- критическое мышление.

Критерии оценивания личностных результатов:

- интерес к IT-сфере;
- ответственность и трудолюбие;
- целеустремлённость;
- коммуникативная культура;
- умение доводить дело до конца.

Шкала оценивания:

- *высокий уровень* — обучающийся свободно владеет материалом, применяет знания в новых ситуациях, самостоятельно решает задачи повышенной сложности, активно участвует в проектной деятельности;
- *средний уровень* — обучающийся владеет основным материалом, справляется с типовыми задачами, допускает незначительные ошибки, нуждается в поддержке педагога;
- *низкий уровень* — обучающийся испытывает затруднения при решении задач, требует постоянного сопровождения, не в полной мере освоил базовые конструкции.

6. Фонды оценочных средств

Фонд оценочных средств включает:

- *Тесты* по темам: «Введение в программирование», «Итоговая работа на ввод-вывод данных. Часть 1», «Частые сценарии».
- *Практические задания* по темам: «Количество чисел», «Делители-1».
- *Комплект задач* для текущего контроля по всем темам.
- *Темы мини-проектов* для промежуточной аттестации (темы 6, 7, 11).
- *Темы итоговых проектов* для итоговой аттестации.
- *Критерии оценивания* по каждому виду аттестации.

- *Оценочные листы и протоколы* для фиксации результатов.

Пример теста по теме «Введение в программирование»:

- Какой язык программирования мы изучаем?
- Что такое алгоритм?
- Какая команда используется для вывода данных?
- Как объявить переменную в Python?
- Что такое PEP 8?

Пример практической задачи по теме «Количество чисел»:

Напишите программу, которая запрашивает у пользователя число N и выводит количество цифр в этом числе.

Пример задания для мини-проекта:

Создайте программу-калькулятор, которая выполняет четыре арифметических действия и выводит результат.

7. Система фиксации результатов

Результаты аттестации фиксируются в:

- *журнале учёта работы педагога* (текущий контроль);
- *оценочных листах* обучающихся (промежуточная аттестация);
- *протоколах промежуточной и итоговой аттестации;*
- *портфолио обучающихся* (проекты, задачи, сертификаты);
- *итоговом отчёте педагога* по завершении модуля.

Формы отчётности для педагога:

- *аналитическая справка по результатам аттестации;*
- *динамика индивидуальных достижений;*
- *рекомендации по дальнейшему обучению.*

Информирование родителей (законных представителей):

- результаты аттестации доводятся до сведения родителей на родительских собраниях, через индивидуальные консультации или электронные дневники (по согласованию).

8. Связь аттестации с планируемыми результатами

Планируемые результаты	Формы аттестации
Знание синтаксиса Python	Тестирование, решение задач
Умение работать с переменными и типами данных	Решение задач, практическая работа
Владение условными операторами	Тестирование, мини-проект
Умение использовать циклы	Решение задач, практическая работа
Навыки работы со строками и списками	Решение задач, итоговый проект
Алгоритмическое мышление	Решение задач, итоговый проект
Командная работа	Групповые проекты, защита
Навыки презентации	Защита мини-проектов, итогового проекта

V. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

1. Общие положения

Оценочные материалы разработаны для дополнительной общеобразовательной общеразвивающей программы технической направленности «Практическое программирование на Python: решение алгоритмических задач» (вводный модуль, 72 часа, возраст 10–17 лет).

Цель оценочных материалов — обеспечить объективную и системную оценку достижения обучающимися планируемых результатов: предметных, метапредметных и личностных.

***Задачи*:**

- определить уровень освоения теоретических знаний и практических навыков программирования на Python;
- оценить динамику индивидуального развития каждого обучающегося;
- выявить затруднения и своевременно скорректировать образовательный процесс;
- создать ситуацию успеха и поддержать мотивацию к дальнейшему изучению IT.

Принципы: систематичность, объективность, прозрачность, практико-ориентированность, дифференцированный подход, сочетание индивидуальных и групповых форм, учёт возрастных особенностей.

2. Состав оценочных материалов

В комплект оценочных материалов входят:

1. *Тесты* (закрытые и открытые вопросы) по темам: «Введение в программирование», «Итоговая работа на ввод-вывод данных. Часть 1», «Частые сценарии».
2. *Практические задания* по темам: «Количество чисел», «Делители-1».
3. *Комплект задач* для текущего контроля по всем темам.
4. *Темы мини-проектов* для промежуточной аттестации (темы 6, 7, 11).
5. *Темы итоговых проектов* для итоговой аттестации.
6. *Критерии оценивания* по каждому виду аттестации.
7. *Оценочные листы **и** протоколы* для фиксации результатов.
8. *Портфолио обучающегося* (рекомендуемая форма).

3. Оценочные материалы для текущего контроля

Форма: решение задач на каждом занятии, устный опрос, педагогическое наблюдение, самопроверка и взаимопроверка кода.

Примеры заданий по темам:

Тема	Задание
Тема 2. Ввод-вывод данных	Напишите программу, которая запрашивает имя пользователя и выводит приветствие.
Тема 4. Работа с целыми числами	Напишите программу, которая запрашивает два числа и выводит их сумму, разность, произведение и частное.

Тема 8. Условный оператор	Напишите программу, которая определяет, является ли введённое число чётным.
Тема 15. Цикл for	Выведите все числа от 1 до 10 с помощью цикла for. Найдите сумму чисел от 1 до 100.
Тема 19. Цикл while	Напишите программу, которая запрашивает числа до тех пор, пока не будет введён 0, и выводит их сумму.
Форма фиксации: лист наблюдений педагога.	

4. Оценочные материалы для промежуточной аттестации

4.1. Тестирование (темы 1, 6, 17)

Пример теста по теме «Введение в программирование»:

1. Какой язык программирования изучается в модуле?
а) Java
б) Python
в) C++
г) JavaScript
2. Что такое алгоритм?
а) последовательность команд
б) язык программирования
в) тип данных
г) программа
3. Какая команда используется для вывода данных?
а) input()
б) print()
в) output()
г) write()
4. Как объявить переменную в Python?
а) int x = 5
б) x = 5
в) var x = 5
г) let x = 5
5. Что такое PEP 8?
а) стандарт оформления кода
б) библиотека Python
в) среда разработки
г) тип данных

Критерии: 80–100% — высокий уровень; 50–79% — средний; менее 50% — низкий.

4.2. Практическая работа (темы 18, 24)

Тема 18. Количество чисел

Задание: Напишите программу, которая запрашивает у пользователя число N и выводит количество цифр в этом числе.

Пример решения:

```
n = int(input("Введите число: "))
count = 0
while n > 0:
    n = n // 10
```

```
    count += 1
print("Количество цифр:", count)
```

Критерии: программа работает корректно — 5 баллов; программа работает с недочётами — 3–4 балла; программа не работает — 0–2 балла.

Тема 24. Делители-1

Задание: Напишите программу, которая выводит все делители введённого числа.

Пример решения:

```
n = int(input("Введите число: "))
for i in range(1, n + 1):
    if n % i == 0:
        print(i)
```

Критерии: аналогично.

4.3. Мини-проекты (темы 6, 7, 11)

Темы мини-проектов:

- «Калькулятор» (тема 6);
- «Программа-опросник» (тема 7);
- «Игра “Угадай число”» (тема 11).

Критерии оценивания мини-проекта:

№	Критерий	Баллы
1	Техническое исполнение	0–5
2	Сложность решения	0–5
3	Качество презентации	0–5
4	Умение отвечать на вопросы	0–5
Итого		**0–20**

Уровни: 16–20 баллов — высокий; 10–15 баллов — средний; менее 10 баллов — низкий.

Результаты фиксируются в ***протоколе промежуточной аттестации.***

5. Оценочные материалы для итоговой аттестации

Форма: итоговый проект и его защита.

Примерные темы итоговых проектов:

- «Калькулятор для решения задач по математике»;
- «Программа-профориентатор “Твой путь в IT”»;
- «Игра “Угадай число”»;
- «Тест на знание правил дорожного движения»;
- «Помощник для подсчёта оценок»;
- «Генератор случайных задач по Python»;
- «Программа для обработки результатов соревнований».

Требования к проекту:

- проект выполнен на языке Python;
- демонстрирует владение базовыми конструкциями: ввод-вывод, переменные, условия, циклы, строки, списки;
- имеет практическую значимость или игровую направленность;
- сопровождается краткой презентацией (3–5 слайдов);
- защита — 5–7 минут, включая ответы на вопросы.

Критерии оценивания итогового проекта:

№	Критерий	Баллы
1	Техническое исполнение (работоспособность, отсутствие ошибок)	0–5
2	Сложность и оригинальность решения	0–5
3	Читаемость и структурированность кода	0–5
4	Качество презентации	0–5
5	Умение отвечать на вопросы	0–5
6	Самостоятельность выполнения	0–5
Итого		**0–30**

Уровни: 25–30 баллов — высокий; 15–24 балла — средний; менее 15 баллов — низкий.

Результаты фиксируются в ***протоколе итоговой аттестации***. Обучающимся, успешно прошедшим итоговую аттестацию, выдаётся сертификат или диплом (в соответствии с локальными актами организации).

6. Критерии оценивания

Предметные результаты:

- знание синтаксиса Python;
- умение работать с переменными, типами данных;
- владение условными операторами и логическими выражениями;
- умение использовать циклы for и while;
- навыки работы со строками, списками, словарями;
- способность решать алгоритмические задачи;
- умение отлаживать и тестировать код.

Метапредметные результаты:

- умение планировать свою деятельность;
- способность работать в команде;
- навыки поиска информации и самообучения;
- умение презентовать результаты своей работы;
- критическое мышление.

Личностные результаты:

- интерес к IT-сфере;
- ответственность и трудолюбие;
- целеустремлённость;
- коммуникативная культура;
- умение доводить дело до конца.

Шкала оценивания:

Уровень	Характеристика
Высокий	Обучающийся свободно владеет материалом, применяет знания в новых ситуациях, самостоятельно решает задачи повышенной сложности, активно участвует в проектной деятельности.
Средний	Обучающийся владеет основным материалом, справляется с типовыми задачами, допускает незначительные ошибки, нуждается в поддержке педагога.
Низкий	Обучающийся испытывает затруднения при решении задач, требует постоянного сопровождения, не в полной мере освоил базовые конструкции.

7. Формы фиксации результатов

Результаты аттестации фиксируются в:

- *журнале учёта работы педагога* (текущий контроль);

- *оценочных листах* обучающихся (промежуточная аттестация);
- *протоколах промежуточной и итоговой аттестации*;
- *портфолио обучающихся* (проекты, задачи, сертификаты);
- *итоговом отчёте педагога* по завершении модуля.

Формы отчётности для педагога:

- аналитическая справка по результатам аттестации;
- динамика индивидуальных достижений;
- рекомендации по дальнейшему обучению.

Информирование родителей (законных представителей):

- результаты аттестации доводятся до сведения родителей на родительских собраниях, через индивидуальные консультации или электронные дневники (по согласованию).

8. Связь оценочных материалов с планируемыми результатами

Планируемые результаты	Формы и материалы оценивания
Знание синтаксиса Python	Тестирование, решение задач
Умение работать с переменными и типами данных	Решение задач, практическая работа
Владение условными операторами	Тестирование, мини-проект
Умение использовать циклы	Решение задач, практическая работа
Навыки работы со строками и списками	Решение задач, итоговый проект
Алгоритмическое мышление	Решение задач, итоговый проект
Командная работа	Групповые проекты, защита
Навыки презентации	Защита мини-проектов, итогового проекта

9. Перечень оценочных материалов

1. Тесты по темам 1, 6, 17 (по 10–15 вопросов).
2. Практические задания по темам 18, 24.
3. Комплект задач для текущего контроля (по всем темам).
4. Темы мини-проектов (темы 6, 7, 11).
5. Темы итоговых проектов.
6. Критерии оценивания (текущий, промежуточный, итоговый).
7. Оценочные листы.
8. Протоколы промежуточной и итоговой аттестации.
9. Портфолио обучающегося (рекомендуемая форма).

VI. ВОСПИТАТЕЛЬНЫЙ КОМПОНЕНТ

1. Цель и задачи воспитательной работы

Цель воспитательной работы — создание условий для формирования социально ответственной, интеллектуально развитой личности обучающегося через освоение IT-технологий, развитие технического творчества, коллективного взаимодействия и ценностного отношения к знаниям и труду.

Задачи воспитательной работы:

- формировать ценностное отношение к труду, знаниям и созиданию;
- воспитывать информационную и технологическую культуру;
- развивать коммуникативные качества и культуру сотрудничества;
- формировать устойчивый интерес к IT-технологиям и инженерным профессиям;
- воспитывать ответственность за результат своей деятельности и бережное отношение к оборудованию.

2. Направления воспитательной работы

Направление	Содержание в программе
Гражданско-патриотическое	Ознакомление с достижениями российских IT-специалистов и инженеров. Участие в конкурсах и хакатонах как форма защиты чести учреждения. Воспитание гордости за отечественные разработки в области программирования и искусственного интеллекта.
Трудовое	Формирование навыков аккуратной и точной работы с компьютером. Освоение правил организации рабочего места. Развитие умения доводить начатую работу до конца: от идеи и алгоритма до готовой работающей программы. Воспитание уважения к инженерному труду.
Социально-коммуникативное	Обучение навыкам эффективной коммуникации и взаимодействия в команде при разработке проектов. Распределение ролей в группе (разработчик, тестировщик, докладчик). Совместная подготовка к защите проектов и участию в конкурсах.
Цифровая культура	Формирование осознанного использования информационных технологий. Понимание этических норм работы в цифровой среде. Безопасное поведение в интернете.
Профессиональное самоопределение	Ранняя профориентация в IT-сфере. Знакомство с профессиями: программист, аналитик данных, разработчик игр, специалист по искусственному интеллекту

3. Формы и методы воспитательной работы

Формы воспитательной работы:

- беседа, дискуссия, рефлексия;
- деловая и сюжетно-ролевая игра;
- викторина, конкурс;
- проектная и командная работа;
- защита проектов;
- встреча с IT-специалистами;
- экскурсия в IT-компанию или технопарк.

Методы воспитания:

- убеждение, поощрение, поддержка;
- создание ситуации успеха;
- коллективное мнение;
- положительная мотивация;
- личный пример педагога.

Принципы воспитательной работы:

– уважение индивидуальности личности;
– коллективная деятельность;
– возрастной подход;
– педагогическая поддержка

4. Календарный план воспитательной работы

Срок	Мероприятие	Форма	Направление
Сентябрь	«Знакомьтесь — это мы!»	Игра, командообразование	Социально-коммуникативное
Октябрь	«IT-профессии будущего»	Беседа, дискуссия	Профессиональное самоопределение
Ноябрь	«День народного единства»	Беседа, викторина	Гражданско-патриотическое
Декабрь	«Новогодний хакатон»	Командная проектная работа	Социально-коммуникативное, трудовое
Январь	«Безопасный интернет»	Беседа, кейс-игра	Цифровая культура
Февраль	«День защитника Отечества»	Викторина, встреча с ветеранами IT-отрасли	Гражданско-патриотическое
Март	«Женщины в IT»	Беседа, презентации	Профессиональное самоопределение
Апрель	«Космос и технологии»	Викторина, проектная работа	Гражданско-патриотическое, трудовое
Май	«Итоги года: мои достижения»	Защита проектов, рефлексия	Все направления

5. Планируемые результаты воспитательной работы

К концу освоения программы у обучающегося будут сформированы:

– ответственное отношение к результатам своего труда и труда товарищей;
– привычка к самообслуживанию: наведение порядка на рабочем месте, бережное обращение с оборудованием;
– интерес к отечественным научно-техническим разработкам;
– умение работать в паре и малой группе, уважение к мнению другого;
– культура организации своей деятельности;
– адекватность восприятия оценки своей деятельности;
– коллективная ответственность;
– навыки соблюдения нравственно-этических норм;
– уверенность в себе, ответственность, самостоятельность

6. Условия организации воспитания

Воспитательная деятельность в объединении осуществляется преимущественно через:

– вовлечение обучающихся в интересную и полезную деятельность, дающую возможность самореализации;
– формирование детско-взрослой общности, объединяющей детей и педагогов общими позитивными эмоциями;
– создание традиций в детском коллективе;
– поддержку детских инициатив и самоуправления;
– поощрение лидерской позиции

7. Связь воспитательного компонента с содержанием программы

Воспитательный компонент реализуется **интегрированно** через содержание тем программы:

Тема программы	Воспитательный аспект
Введение в программирование	Формирование интереса к IT-сфере, знакомство с историей языка Python
Работа с целыми числами	Воспитание точности, аккуратности, внимания к деталям
Условный оператор	Развитие логического мышления, ответственности за принятые решения
Циклы	Воспитание настойчивости, умения доводить дело до конца
Итоговые работы	Командная работа, взаимопомощь, презентация результатов
Итоговый проект	Профессиональное самоопределение, социальная значимость результатов

VII. УЧЕБНЫЙ ПЛАН

ВВОДНЫЙ МОДУЛЬ

№ п/п	Название раздела, темы	Количество часов				Формы аттестации/контроля
		Всего	Теория	Практика	Самостоятельная подготовка	
Блок 1. Ввод-вывод данных						
1	Введение в программирование	3	2	1	0	Тестирование
2	Ввод-вывод данных, переменные	3	1	1	1	Решение задач
3	Sep, end, PEP	3	1	1	1	Решение задач
4	Работа с целыми числами	3	1	1	1	Решение задач
5	Работа с целыми числами	3	1	1	1	Решение задач
Блок 2. Итоговая работа на ввод-вывод данных						
6	Итоговая работа на ввод-вывод данных. Часть 1	3	0	3	0	Тестирование
7	Итоговая работа на ввод-вывод данных. Часть 2	3	0	0	3	Решение задач
Блок 3. Условный оператор						
8	Условный оператор	3	1	1	1	Решение задач
9	Логические операторы	3	1	1	1	Решение задач
10	Условный оператор	3	1	1	1	Решение задач
Блок 4. Итоговая работа на условный оператор						
11	Итоговая работа на условный оператор.	3	0	0	3	Решение задач
Блок 5. Типы данных						
12	Числовые типы данных	3	1	1	1	Решение задач
13	Строковый тип данных	3	1	1	1	Решение задач
14	Модуль math	3	1	1	1	Решение задач
Блок 5. Циклы for и while						
15	Цикл for	3	1	1	1	Решение задач
16	Цикл for	3	1	1	1	Решение задач
17	Частые сценарии	3	1	2	0	Тестирование
18	Количество чисел	3	0	0	3	Практическая работа
19	Цикл while	3	1	1	1	Решение задач
20	Обработка цифр числа	3	1	1	1	Решение задач
21	Break, continue, else	3	1	1	1	Решение задач
22	Ревью кода	3	1	1	1	Решение задач
23	Вложенные циклы	3	1	1	1	Решение задач
24	Делители-1	3	0	0	3	Практическая работа
Итого		**72**	**20**	**23**	**29**	

VIII. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

ВВОДНЫЙ МОДУЛЬ

№ п/п	Тема занятия	Кол-во часов	Форма занятия	Место проведения	Дата занятия
1	Введение в программирование	3	Теоретическое занятие с практической работой	IT-квантум	
2	Ввод-вывод данных, переменные	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
3	Sep, end, PEP	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
4	Работа с целыми числами	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
5	Работа с целыми числами	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
6	Итоговая работа на ввод-вывод данных. Часть 1	3	Практическое занятие	IT-квантум	
7	Итоговая работа на ввод-вывод данных. Часть 2	3	Самостоятельная работа	IT-квантум	
8	Условный оператор	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
9	Логические операторы	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
10	Условный оператор	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
11	Итоговая работа на условный оператор.	3	Самостоятельная работа	IT-квантум	
12	Числовые типы данных	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
13	Строковый тип данных	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
14	Модуль math	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
15	Цикл for	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
16	Цикл for	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	

17	Частые сценарии	3	Теоретическое занятие с практической работой	IT-квантум	
18	Количество чисел	3	Самостоятельная работа	IT-квантум	
19	Цикл while	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
20	Обработка цифр числа	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
21	Break, continue, else	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
22	Ревью кода	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
23	Вложенные циклы	3	Теоретическое занятие с практической и самостоятельной работой	IT-квантум	
24	Делители-1	3	Самостоятельная работа	IT-квантум	

IX. МАТЕРИАЛЬНО-ТЕХНИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Техническое оснащение:

- компьютерный класс с ПК (не менее 12 рабочих мест);
- ноутбуки для мобильной работы (5–7 шт.);
- интерактивная панель или проектор;
- локальная сеть с выходом в интернет.

Программное обеспечение:

- ОС Windows/Linux;
- среда разработки PyCharm (Community Edition) или VS Code;
- интерпретатор Python 3.8+;
- браузеры последних версий;
- офисный пакет.

Организация пространства: зоны для индивидуальной и групповой работы, зона для презентаций.

Дидактические материалы: сборники задач по Python (в печатном и электронном виде), инструкционные карты, критерии оценки.

Система безопасности: антивирусное ПО, контент-фильтрация, аптечка первой помощи.

Список литературы

Нормативные правовые акты:

Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в РФ».

Указ Президента РФ от 07.05.2024 № 309.

Указ Президента РФ от 09.11.2022 № 809.

Приказ Минпросвещения России от 09.11.2018 № 196.

СанПиН 2.4.4.3172-14.

Для педагога:

Лутц М. «Изучаем Python». — СПб.: Символ-Плюс, 2022.

Доусон М. «Программируем на Python». — СПб.: Питер, 2023.

Свейгарт Э. «Автоматизация рутинных задач с помощью Python». — М.: Диалектика, 2022.

Копец Е.В. «Основы алгоритмизации и программирования на Python для школьников». — М.: Бином, 2023.

Журнал «Информатика в школе» (электронная версия).

Для учащихся:

Бриггс Дж. «Python для детей». — М.: Манн, Иванов и Фербер, 2023.

Свейгарт Э. «Python для детей и родителей». — М.: Эксмо, 2022.

Официальная документация Python (docs.python.org).

Онлайн-курсы «Яндекс.Учебника» по Python.

Stepik.org — курсы по Python.

Электронные ресурсы:

Видеоуроки на «Российской электронной школе».

Репозиторий GitHub.

Методические материалы на «Инфоурок».

Вебинары Академии Минпросвещения России.