

Муниципальное казённое общеобразовательное учреждение
Куйбышевского муниципального района
Новосибирской области
«Средняя общеобразовательная школа № 2»

Принята на заседании
педагогического совета
Протокол № 1
от «28» августа 2026 г.

УТВЕРЖДЕНО:
Директор  О. А. Дицель



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**
подвид: разноуровневая (стартовый уровень)
технической направленности
«Программирование на Python»
(наименование)

Возраст учащихся: 12 - 14 лет

Срок реализации: 1 год

Автор-составитель:
Абрамовских Виктория Вячеславовна,
педагог дополнительного образования

г. Куйбышев
2026г.

Внутренняя экспертиза проведена. Программа рекомендована к рассмотрению на педагогическом совете учреждения.

Заместитель директора по ВР

Афанасьева О. В. / Афанасьева
ФИО Подпись
«28» августа 2026г.

РАЗДЕЛ 1 «КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ»

1.1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» имеет техническую направленность.

Актуальность программы обусловлена государственной политикой в сфере цифрового развития и технического образования, подтверждается Концепцией развития дополнительного образования детей до 2030 года (утв. Распоряжением Правительства РФ от 31.03.2022 № 678-р), а также растущей потребностью общества в кадрах, владеющих навыками алгоритмического мышления и программирования.

Информационные технологии прочно вошли в повседневную жизнь подростков: они используют цифровые сервисы не только для общения и развлечения, но и как инструмент познания. Данная программа переводит интерес обучающихся к компьютерам в осознанную созидательную деятельность — от потребления цифрового контента к его созданию. Язык Python выбран как один из самых востребованных и доступных для стартового обучения: он обладает понятным синтаксисом, широкой сферой профессионального применения и активно используется в олимпиадном и конкурсном движении. Освоение программы способствует развитию логического и системного мышления, усидчивости и способности доводить задачу до результата.

Отличительной особенностью и новизной программы является её выраженная практико-ориентированность и проектный характер построения содержания. Обучение программированию не сводится к заучиванию синтаксических конструкций: каждая изученная тема сразу закрепляется созданием работающего программного продукта — от графических рисунков в среде «Черепашка» до собственных мини-игр и прикладных утилит. Новизна проявляется в поэтапном усложнении проектных заданий в рамках каждого раздела, что позволяет обучающемуся уже на стартовом уровне ощутить себя автором реального продукта, а не исполнителем типовых упражнений.

Программа рассчитана на обучающихся 12–15 лет (5–9 классы). В этом возрасте у подростков активно формируются способности к абстрактному и формальному мышлению, появляется устойчивый интерес к техническому творчеству и цифровым технологиям, что делает данный период оптимальным для старта в программировании.

Принципы формирования групп: по возрасту, на стартовом (ознакомительном) уровне подготовки, со смешанным составом с учетом уровня базовой компьютерной грамотности.

Условия приема: свободный набор, предварительная специальная подготовка не требуется, возможно проведение вводного собеседования и входной диагностики для определения стартового уровня и уровня владения компьютером.

Объем программы составляет 72 учебных часа. Срок реализации программы – 1 год.

Уровень программы: стартовый (ознакомительный).

Форма обучения: очная.

Язык обучения: русский.

Форма реализации образовательной программы: традиционная.

Занятия проводятся 1 раз в неделю по 2 академических часа с 5-минутным перерывом. Режим организации соответствует возрастным особенностям обучающихся и санитарно-гигиеническим требованиям.

1.2. Цель и задачи программы

Цель: Формирование у обучающихся основ алгоритмического мышления и устойчивых навыков программирования на языке Python через разработку собственных программных продуктов (игр, утилит, графических проектов).

Задачи программы:

а) Предметные задачи:

- освоение базового синтаксиса и структуры языка Python;
- работа с переменными, типами данных и базовыми операторами;
- изучение условных конструкций и циклов для управления ходом программы;
- освоение работы со структурами данных (списки, кортежи, словари);
- создание простых графических приложений с использованием модуля Turtle;
- приобретение навыков написания, чтения и отладки (поиска ошибок) кода.

б) Метапредметные задачи:

- развитие логического и алгоритмического мышления;
- умение разбивать сложную задачу на подзадачи (декомпозиция) и планировать последовательность действий;
- развитие навыков самостоятельного поиска, анализа информации и работы с документацией;
- умение аргументировать выбор того или иного алгоритма решения задачи.

в) Личностные задачи:

- воспитание усидчивости, внимательности к деталям и аккуратности при написании кода;
- формирование мотивации к освоению IT-профессий и пониманию роли программирования в современном мире;
- развитие самостоятельности, уверенности в своих силах и способности доводить начатое дело до логического завершения.

1.3. Планируемые результаты

а) Предметные результаты:

- *Теоретические знания:* понимание принципов работы ЭВМ; знание базового синтаксиса языка Python; понимание типов данных, условных операторов, циклов и базовых структур данных (списки, словари, кортежи); знание правил написания читаемого кода и использования функций.
- *Практические умения:* способность применять полученные знания для решения типовых и творческих задач (написание простых скриптов, создание калькуляторов, текстовых квестов, графических рисунков с помощью модуля Turtle, отладка программ и поиск синтаксических ошибок).

б) Метапредметные результаты:

- *Универсальные способы деятельности:* способность к алгоритмизации и декомпозиции сложных задач; умение моделировать процессы; навык самостоятельного поиска информации и работы с документацией для решения возникающих проблем.
- *Коммуникативные навыки:* умение четко формулировать свои мысли при описании алгоритма; способность объяснять логику работы своего кода; навыки сотрудничества при работе над парными или групповыми проектами.

в) Личностные результаты:

- *Ценностно-смысловые установки и мотивация:* формирование устойчивого интереса к информационным технологиям и программированию; понимание этики работы с данными и авторским кодом; развитие потребности в саморазвитии в цифровой среде.
- *Личностные качества:* воспитание внимательности, дисциплинированности мышления, терпения при преодолении ошибок; развитие ответственности за результат своей программной разработки.

1.4. Содержание программы

Учебный план

№ п/п	Название раздела и тем	Всего часов	Теория	Практика	Формы контроля
1	Введение в Python и среду разработки	4	2	2	
1.1	Знакомство с Python. Установка среды	2	1	1	Устный опрос, проверка установки ПО
1.2	Первый скрипт. Синтаксис, отступы, комментарии, print()	2	1	1	Запуск первой программы
2	Переменные, типы данных, ввод/вывод	8	3	5	
2.1	Переменные и типы данных (int, float, str, bool)	4	2	2	Блиц-тест по типам данных
2.2	Операторы ввода (input) и вывода. Преобразование типов	4	1	3	Мини-проект: программа-приветствие и калькулятор
3	Сравнения и логические выражения	6	2	4	
3.1	Операторы сравнения (==, !=, >, <, >=, <=)	2	1	1	Логические задачи
3.2	Логические операторы (and, or, not). Порядок действий	4	1	3	Решение логических задач в коде
4	Условные конструкции	8	2	6	

4.1	Ветвление: if, else	4	1	3	Программа-проверка «пароля»
4.2	Множественный выбор: elif. Вложенные условия	4	1	3	Текстовый квест с выбором сюжета
5	Списки	10	3	7	
5.1	Создание списков. Индексация и срезы	4	1	3	Задания на работу с элементами списка
5.2	Методы списков (append, remove, sort). Перебор циклом for	6	2	4	Мини-проект: «Список покупок» или «Журнал оценок»
6	Словари и кортежи	6	2	4	
6.1	Кортежи: особенности и применение	2	1	1	Задание на различие списков и кортежей
6.2	Словари: ключи и значения, перебор	4	1	3	Мини-проект: «Телефонная книга»
7	Циклы	8	2	6	
7.1	Цикл while. Операторы break и continue	4	1	3	Задачи на итерации
7.2	Цикл for и функция range(). Перебор коллекций	4	1	3	Генерация числовых последовательностей и узоров
8	Черепашья графика (Turtle)	10	2	8	
8.1	Знакомство с Turtle. Базовые команды, цвет, перо	4	1	3	Рисование геометрических фигур
8.2	Циклы в графике. Создание орнаментов и узоров	6	1	5	Творческий проект: авторский орнамент
9	Знакомство с функциями	8	3	5	
9.1	Объявление и вызов функций. Аргументы	4	2	2	Написание собственных функций
9.2	Оператор return. Использование функций в проектах	4	1	3	Создание программы с использованием функций
10	Итоговый проект	4	1	3	

10.1	Проектирование и написание кода	2	1	1	Защита черновика проекта
10.2	Отладка, тестирование и презентация	2	0	2	Защита итогового проекта
	ИТОГО	72	22	50	

Содержание учебного плана

Раздел 1. Введение в Python и среду разработки (4 часа)

Тема 1.1. Знакомство с Python. Установка среды (2 ч. Теория: 1, Практика: 1)

Теория: История создания языка Python. Сферы применения (веб, наука, игры, автоматизация). Установка интерпретатора и среды разработки (IDLE или PyCharm).

Практика: Установка ПО, знакомство с интерфейсом среды, запуск первой команды.

Форма контроля: Устный опрос, проверка установки ПО.

Тема 1.2. Первый скрипт. Синтаксис, отступы, комментарии, print() (2 ч. Теория: 1, Практика: 1)

Теория: Структура программы на Python. Понятие синтаксиса. Роль отступов (индентации).

Комментарии в коде. Функция print() для вывода данных.

Практика: Написание, запуск и модификация первой программы «Hello, World!».

Форма контроля: Запуск первой программы.

Раздел 2. Переменные, типы данных, ввод/вывод (8 часов)

Тема 2.1. Переменные и типы данных (4 ч. Теория: 2, Практика: 2)

Теория: Понятие переменной как «ярлыка» для данных. Целые числа (int), дробные (float), строки (str), логический тип (bool). Динамическая типизация. Функция type().

Практика: Создание переменных, присваивание значений, проверка типов данных.

Форма контроля: Блиц-тест по типам данных.

Тема 2.2. Операторы ввода (input) и вывода. Преобразование типов (4 ч. Теория: 1, Практика: 3)

Теория: Математические операторы (+, -, *, /, //, %, **). Приоритет операций. Функция input() для получения данных от пользователя. Преобразование типов (int(), float(), str()).

Практика: Создание интерактивного калькулятора, программы-конвертера величин, программы-приветствия.

Форма контроля: Мини-проект: калькулятор.

Раздел 3. Сравнения и логические выражения (6 часов)

Тема 3.1. Операторы сравнения (2 ч. Теория: 1, Практика: 1)

Теория: Операторы сравнения: == (равно), != (не равно), >, <, >=, <= . Возвращаемое значение True/False.

Практика: Решение простых логических задач на бумаге и в коде.

Форма контроля: Логические задачи.

Тема 3.2. Логические операторы (and, or, not). Порядок действий (4 ч. Теория: 1, Практика: 3)

Теория: Логические операторы: and (и), or (или), not (не). Таблицы истинности. Приоритет логических операций.

Практика: Решение составных логических задач в коде (например, проверка попадания в диапазон).

Форма контроля: Решение логических задач в коде.

Раздел 4. Условные конструкции (8 часов)

Тема 4.1. Ветвление: if, else (4 ч. Теория: 1, Практика: 3)

Теория: Конструкция if. Синтаксис. Блок else для альтернативного действия. Вложенные условия.

Практика: Создание программы-проверки «пароля», программы определения високосного года.

Форма контроля: Программа-проверка «пароля».

Тема 4.2. Множественный выбор: elif. Вложенные условия (4 ч. Теория: 1, Практика: 3)

Теория: Конструкция elif для множественного выбора. Комбинация if-elif-else.

Практика: Создание текстового квеста с несколькими вариантами развития сюжета (например, «Выбери свой путь»).

Форма контроля: Текстовый квест с выбором сюжета.

Раздел 5. Списки (10 часов)

Тема 5.1. Создание списков. Индексация и срезы (4 ч. Теория: 1, Практика: 3)

Теория: Понятие списка (list). Создание списков. Индексация (начало с 0). Срезы списков. Отрицательные индексы.

Практика: Задания на получение элементов списка, изменение значений, работу со срезами.

Форма контроля: Задания на работу с элементами списка.

Тема 5.2. Методы списков (append, remove, sort). Перебор циклом for (6 ч. Теория: 2, Практика: 4)

Теория: Методы списков: append() (добавление), remove() (удаление), sort() (сортировка), len() (длина). Перебор элементов списка с помощью цикла for.

Практика: Создание программ для сортировки, фильтрации и анализа наборов данных. Мини-проект: «Список покупок» или «Журнал оценок».

Форма контроля: Мини-проект: «Список покупок».

Раздел 6. Словари и кортежи (6 часов)

Тема 6.1. Кортежи: особенности и применение (2 ч. Теория: 1, Практика: 1)

Теория: Понятие кортежа (tuple). Отличия от списка. неизменяемость данных. Когда использовать кортежи.

Практика: Задания на использование кортежей для хранения неизменяемых данных (координаты, даты).

Форма контроля: Задание на различие списков и кортежей.

Тема 6.2. Словари: ключи и значения, перебор (4 ч. Теория: 1, Практика: 3)

Теория: Понятие словаря (dict). Ключи и значения. Создание словарей. Добавление, удаление и изменение элементов. Перебор словаря (цикл for, методы keys(), values(), items()).

Практика: Создание программы «Телефонная книга» или «Словарь английских слов».

Форма контроля: Мини-проект: «Телефонная книга».

Раздел 7. Циклы (8 часов)

Тема 7.1. Цикл while. Операторы break и continue (4 ч. Теория: 1, Практика: 3)

Теория: Синтаксис цикла while. Условие выхода из цикла. Бесконечные циклы. Операторы break (выход) и continue (пропуск итерации).

Практика: Задачи на итерации, поиск числа, выход из цикла по условию.

Форма контроля: Задачи на итерации.

Тема 7.2. Цикл `for` и функция `range()`. Перебор коллекций (4 ч. Теория: 1, Практика: 3)

Теория: Синтаксис цикла `for`. Функция `range()` для генерации последовательностей. Перебор элементов списков и строк с помощью `for`.

Практика: Генерация числовых последовательностей, подсчет элементов, создание паттернов.

Форма контроля: Генерация числовых последовательностей.

Раздел 8. Черепашня графика (Turtle) (10 часов)

Тема 8.1. Знакомство с Turtle. Базовые команды, цвет, перо (4 ч. Теория: 1, Практика: 3)

Теория: Импорт модуля `turtle`. Объект «Черепашка». Базовые команды: `forward()`, `backward()`, `left()`, `right()`. Изменение цвета (`color()`), толщины пера (`pensize()`), скорости (`speed()`).

Практика: Рисование базовых геометрических фигур (квадрат, треугольник, звезда).

Форма контроля: Рисование геометрических фигур.

Тема 8.2. Циклы в графике. Создание орнаментов и узоров (6 ч. Теория: 1, Практика: 5)

Теория: Использование циклов `for` для рисования повторяющихся элементов. Вложенные циклы в графике. Создание функций для рисования фигур.

Практика: Создание сложных орнаментов, спиралей, узоров. Творческое задание: авторский орнамент.

Форма контроля: Творческий проект: авторский орнамент.

Раздел 9. Знакомство с функциями (8 часов)

Тема 9.1. Объявление и вызов функций. Аргументы (4 ч. Теория: 2, Практика: 2)

Теория: Ключевое слово `def`. Объявление функции. Аргументы и параметры. Вызов функции.

Практика: Написание собственных функций для вычислений и обработки данных (например, функция расчета площади).

Форма контроля: Написание собственных функций.

Тема 9.2. Оператор `return`. Использование функций в проектах (4 ч. Теория: 1, Практика: 3)

Теория: Оператор `return` для возврата значения. Отличие `print()` от `return()`. Использование функций в более сложных программах.

Практика: Создание программы с использованием функций (например, калькулятор с функциями для каждой операции).

Форма контроля: Создание программы с использованием функций.

Раздел 10. Итоговый проект (4 часа)

Тема 10.1. Проектирование и написание кода (2 ч. Теория: 1, Практика: 1)

Теория: Этапы разработки ПО. Проектирование интерфейса текстовой программы или графического проекта.

Практика: Написание чернового кода итогового проекта (текстовый квест, графический орнамент или утилита).

Форма контроля: Защита черновика проекта.

Тема 10.2. Отладка, тестирование и презентация (2 ч. Теория: 0, Практика: 2)

Теория: (не предусмотрена).

Практика: Тестирование программы, исправление ошибок, публичная защита проекта.

Форма контроля: Защита итогового проекта.

1.5. Система контроля

Формы текущего контроля и аттестации служат инструментом для подтверждения обучающимися планируемых результатов и определяются учебным планом.

Текущий контроль: блиц-опросы по синтаксису, проверка написанного кода на занятиях, мини-задачи на поиск и исправление ошибок («найди баг»), выполнение практических шагов (например, рисование фигуры в Turtle).

Промежуточная аттестация: выполнение самостоятельных практических работ по ключевым темам (работа со списками, написание текстового квеста с условиями, создание графического орнамента).

Итоговая аттестация: защита итогового творческого проекта (текстовая игра, утилита или графический проект).

Комплект диагностических методик, заданий и критерии оценки представлены в Приложении 1.

РАЗДЕЛ 2. РАБОЧАЯ ПРОГРАММА ВОСПИТАНИЯ

Цель рабочей программы воспитания – формирование у обучающихся цифровой культуры, ответственности за результаты своего труда, развитие навыков алгоритмического мышления и воспитание уважительного отношения к интеллектуальной собственности в среде IT.

Задачи программы

- Формировать понимание основ цифровой этики (недопустимость плагиата кода, уважение к данным пользователей).
- Способствовать осознанному выбору будущей траектории развития в сфере информационных технологий.
- Воспитывать усидчивость, терпение и умение доводить начатое дело до конца (преодоление ошибок в коде).
- Развивать навыки сотрудничества и умения работать в команде при создании совместных проектов.

Целевые ориентиры воспитания

- Понимание важности чистоты и читаемости кода как уважения к тем, кто будет его читать.
- Умение конструктивно реагировать на ошибки и воспринимать их как часть процесса обучения.
- Активный интерес к IT-сфере и понимание роли программирования в современном мире.

Виды, формы и содержание деятельности

Для достижения цели используются: мастер-классы, диспуты по цифровой этике, мини-хакатоны, разбор реальных кейсов из мира IT, создание контента и программных продуктов.

Условия организации воспитания

- Создание атмосферы «IT-стартапа» в классе, где поощряется инициатива и нестандартные решения.
- Поддержка соревновательного духа (кто быстрее найдет ошибку, кто сделает красивее орнамент).
- Индивидуальный подход: поддержка детей, которым программирование дается сложнее, через pair-programming (парное программирование).

Календарный план воспитательной работы

№ п/п	Название мероприятия, события	Цель проведения	Сроки проведения
1	Мастер-класс «Код, который меняет мир: от игр до космоса»	Мотивация к обучению, расширение кругозора о сферах применения Python	Сентябрь
2	«Осенний узор в Turtle»	Развитие творческого подхода, сплочение коллектива через совместное решение задач	Октябрь

3	Диспут «Этика программиста: что такое плагиат и читерство?»	Формирование понимания авторского права и цифровой этики	Февраль
4	Лекция «Профессии в IT: кем я могу стать?»	Профориентация	Март
5	«Защита моих первых программ»	Развитие навыков публичных выступлений, рефлексия, празднование успеха	Май

РАЗДЕЛ 3. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

3.1. Календарный учебный график

Год обучения (уровень)	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год обучения	07 сентября 2026 г.	26 мая 2027 г.	36	36	72 ч., 2 часа в неделю	1 раз в нед. по 2 часа

3.2. Условия реализации программы

Кадровое обеспечение:

Программу реализует педагог дополнительного образования, имеющий соответствующую квалификацию и компетенции в сфере информатики и программирования.

Материально-техническое обеспечение:

Занятия проводятся в компьютерном классе, оснащённом столами, стульями и соответствующим программным обеспечением.

Необходимое оборудование:

- персональные компьютеры с выходом в сеть Интернет;
- мультимедийный проектор и экран (или интерактивная панель);
- установленный интерпретатор языка Python и среда разработки (IDLE, PyCharm или аналоги);
- расходные материалы (флеш-накопители для сохранения проектов обучающихся).

Требования к специальной одежде обучающихся: специальная форма не требуется, рекомендуется наличие сменной обуви.

Методическое обеспечение:

Учебно-методические материалы, используемые в процессе: инструкции по работе в среде разработки, чек-листы по синтаксису Python, карточки с логическими и практическими задачами, шаблоны для проектирования алгоритмов, памятки по поиску и исправлению ошибок (отладке).

Информационное обеспечение:

- доступ к сети Интернет;
- электронные образовательные ресурсы и онлайн-платформы для изучения программирования;
- нормативно-правовые акты и списки литературы (см. ниже).

3.3. Список литературы и информационных ресурсов

Нормативно-правовые акты и документы

- Об образовании в Российской Федерации : Федеральный закон от 29.12.2012 № 273-ФЗ (с изм. и доп.).

- Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам : Приказ Минпросвещения России от 27.07.2022 № 629.
- Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи : СП 2.4.3648-20 (утв. Постановлением Главного государственного санитарного врача РФ от 28.09.2020 № 28).
- Концепция развития дополнительного образования детей до 2030 года : утв. Распоряжением Правительства РФ от 31.03.2022 № 678-р.

Литература для педагога

- Бриггс, Д. Python для детей. Самоучитель по программированию / Д. Бриггс. – Москва : Манн, Иванов и Фербер, 2018. – 320 с.
- Томашевский, П. Привет, Python! Моя первая книга по программированию / П. Томашевский. – Санкт-Петербург : Наука и Техника, 2018. – 256 с.
- Луцкий, А. В. Программирование на языке Python для школьников / А. В. Луцкий. – Москва : БХВ-Петербург, 2021. – 208 с.

Литература и ресурсы для обучающихся и родителей

- Python-tutor.org : визуализатор выполнения кода [Электронный ресурс]. – Режим доступа: <https://pythontutor.org> (дата обращения: 01.09.2026).
- Stepik : платформа онлайн-курсов по программированию [Электронный ресурс]. – Режим доступа: <https://stepik.org> (дата обращения: 01.09.2026).
- Грамота.ру : информационно-справочный портал [Электронный ресурс]. – Режим доступа: <http://www.gramota.ru> (дата обращения: 01.09.2026).

РАЗДЕЛ 4 «ПРИЛОЖЕНИЕ К ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЕ»

Приложение 1. Оценочные (контрольно-измерительные) материалы

Цели диагностики:

1. **Входная диагностика:** Оценить начальный уровень логического мышления, интерес к сфере IT и базовые навыки работы с компьютером.
2. **Текущая диагностика:** Отслеживать прогресс в освоении синтаксиса Python, структур данных и алгоритмов.
3. **Итоговая диагностика:** Оценить умение применять полученные знания для создания комплексного программного продукта и защищать его.

1. Входная (стартовая) диагностика

Форма: Анкетирование и блиц-опрос (логические задачи).

Содержание:

Часть 1: Анкета

- Почему ты выбрал курс «Программирование на языке Python»?
- Как часто ты пользуешься компьютером и для каких целей? (Учеба, игры, общение, творчество).
- Есть ли у тебя опыт написания кода или создания игр (например, в Scratch или на других платформах)?

Часть 2: Блиц-опрос (на проверку логики и алгоритмического мышления)

- Опиши по шагам (алгоритм), как завязать шнурки или приготовить бутерброд. (Цель: проверить умение формализовать задачу и разбивать ее на последовательные действия).
- Логическая задача: У тебя есть стакан с водой и стакан с соком. Как поменять их содержимое местами, если есть только один пустой стакан? (Цель: проверить интуитивное понимание работы «переменной» / временного буфера).
- Знаком ли ты с понятиями «алгоритм», «цикл», «переменная»? Если да, как ты их понимаешь?

2. Текущая диагностика

Проводится после изучения ключевых блоков.

Блок 1: Переменные, типы данных и условия (после Разделов 2–4)

Задание: Написать программу-калькулятор или программу-квест, которая запрашивает у пользователя данные (имя, возраст, ответ на загадку), использует переменные, преобразование типов и условные операторы (if/elif/else) для выдачи результата.

Критерии: Корректное использование переменных и типов данных, правильная работа условий, отсутствие синтаксических ошибок.

Блок 2: Списки, словари и циклы (после Разделов 5–7)

Задание: Создать программу «Телефонная книга» или «Список покупок». Программа должна позволять добавлять, удалять элементы и выводить список на экран с использованием циклов (for/while).

Критерии: Умение работать со структурами данных, применение циклов, читаемость кода.

Блок 3: Черепашня графика (Turtle) и функции (после Разделов 8–9)

Задание: Написать программу, которая с помощью модуля Turtle рисует сложный геометрический узор или орнамент, используя собственные функции и циклы.

Критерии: Использование функций (def), применение циклов для повторяющихся действий, эстетичность и сложность рисунка.

3. Итоговая диагностика / Проектная работа

Цель: Показать умение комбинировать полученные знания для создания полноценного программного продукта.

Форма: Защита итогового творческого проекта.

Варианты проектов (на выбор ученика):

«Текстовая игра-квест»: Приключенческая игра с разветвленным сюжетом, где игрок делает выбор, влияющий на исход (использует функции, условия, списки/словари).

«Графический орнамент»: Авторский сложный узор или анимация, нарисованная с помощью модуля Turtle (использует функции, циклы, работу с цветом и пером).

«Полезная утилита»: Программа-помощник (калькулятор для хобби, генератор паролей, программа-тест по школьным предметам, конвертер величин).

Критерии оценки итогового проекта:

Критерий	Макс. балл	Описание
Функциональность и работа кода	5	Программа запускается, работает без сбоев и выполняет все заявленные функции.
Техническая реализация (Python)	5	Корректное использование изученных конструкций (переменные, циклы, условия, функции, списки).
Читаемость кода	3	Код хорошо откомментирован, используются понятные имена переменных и функций, соблюдается структура (отступы).
Творческий подход и креатив	3	Оригинальность идеи, нестандартное решение задачи, наличие дополнительных «фишек» или усложнений.

Защита проекта	4	Ученик может объяснить, как работает его программа, ответить на вопросы и уверенно презентовать продукт.
Итого	20	

Приложение 2. Тематический календарный учебный график.

Тематическое планирование

№ п/п	Тема	Количество часов			Тип занятия	Форма контроля
		всего	теория	практика		
Сентябрь	Раздел 1: Введение в Python и среду разработки (4 ч.)					
1, 2	Знакомство с Python. История языка. Установка среды разработки	2	1	1	Лекция-практикум	Устный опрос, проверка установки ПО
3, 4	Первый скрипт. Синтаксис, отступы, комментарии, функция print()	2	1	1	Практикум	Запуск первой программы «Hello, World!»
	Раздел 2: Переменные, типы данных, ввод/вывод (4 ч.)					
5, 6	Переменные и типы данных (int, float, str, bool). Динамическая типизация	2	1	1	Лекция-практикум	Блиц-тест по типам данных
7, 8	Операторы ввода (input) и вывода. Преобразование типов. Калькулятор	2	1	1	Практикум	Мини-проект: программа-калькулятор
Октябрь	Раздел 2: Переменные, типы данных, ввод/вывод (4 ч.)					
9, 10	Арифметические операции. Приоритет операций. Программа-конвертер	2	1	1	Практикум	Практическая работа: конвертер величин
11, 12	Интерактивные программы. Комбинирование ввода, вывода и вычислений	2	0	2	Практикум	Мини-проект: программа-приветствие
	Раздел 3: Сравнения и логические выражения (6 ч.)					

13, 14	Операторы сравнения (==, !=, >, <, >=, <=). Возврат True/False	2	1	1	Лекция-практикум	Логические задачи на бумаге
15, 16	Логические операторы (and, or, not). Таблицы истинности	2	1	1	Практикум	Решение логических задач в коде
17, 18	Составные логические выражения. Приоритет операций	2	0	2	Практикум	Практическая работа: проверка попадания в диапазон
Ноябрь	Раздел 4: Условные конструкции (8 ч.)					
19, 20	Ветвление: if, else. Синтаксис. Программа-проверка «пароля»	2	1	1	Лекция-практикум	Программа-проверка пароля
21, 22	Вложенные условия. Программа определения високосного года	2	0	2	Практикум	Практическая работа
23, 24	Множественный выбор: elif. Текстовый квест с выбором сюжета	2	1	1	Проект	Текстовый квест (начало)
25, 26	Множественный выбор: elif. Доработка и защита текстового квеста	2	0	2	Проект	Готовый текстовый квест
Декабрь	Раздел 5: Списки (10 ч.)					
27, 28	Создание списков. Индексация. Срезы списков	2	1	1	Лекция-практикум	Задания на срезы
29, 30	Отрицательные индексы. Изменение элементов списка	2	0	2	Практикум	Практическая работа
31, 32	Методы списков: append(), remove(), sort(), len()	2	1	1	Практикум	Задания на методы списков
33, 34	Перебор списка циклом for. Мини-проект «Список покупок»	2	1	1	Проект	Мини-проект: «Список покупок»
35, 36	Доработка проекта «Список покупок» / «Журнал оценок»	2	0	2	Проект	Защита мини-проекта

Январь	Раздел 6: Словари и кортежи (6 ч.)					
37, 38	Кортежи: особенности, неизменяемость, применение	2	1	1	Лекция-практикум	Задание на различие списков и кортежей
39, 40	Словари: ключи и значения. Создание и доступ	2	1	1	Практикум	Практическая работа
41, 42	Перебор словаря. Мини-проект «Телефонная книга»	2	0	2	Проект	Мини-проект: «Телефонная книга»
	Раздел 7: Циклы (2 ч.)					
43, 44	Цикл while. Операторы break и continue	2	1	1	Лекция-практикум	Задачи на итерации
Февраль	Раздел 7: Циклы (6 ч.)					
45, 46	Цикл while: задачи на поиск числа, выход по условию	2	0	2	Практикум	Практическая работа
47, 48	Цикл for и функция range(). Генерация последовательностей	2	1	1	Лекция-практикум	Генерация числовых последовательностей
49, 50	Перебор коллекций циклом for. Создание паттернов	2	0	2	Практикум	Практическая работа
51, 52	Комбинирование циклов и условий. Практикум	2	1	1	Практикум	Комплексная задача
Март	Раздел 8: Черепаший графика (Turtle) (10 ч.)					
53, 54	Знакомство с Turtle. Импорт модуля. Базовые команды	2	1	1	Лекция-практикум	Рисование базовых фигур
55, 56	Цвет, толщина пера, скорость. Рисование геометрических фигур	2	0	2	Практикум	Рисование квадрата, треугольника, звезды
57, 58	Циклы в графике. Создание орнаментов и узоров	2	1	1	Творческая мастерская	Творческий проект: орнамент (начало)
59, 60	Вложенные циклы в графике. Спираль, узоры	2	0	2	Творческая мастерская	Творческий проект: орнамент (продолжение)

61, 62	Защита творческого проекта «Авторский орнамент»	2	0	2	Творческая мастерская	Защита проекта
Апрель	Раздел 9: Знакомство с функциями (8 ч.)					
63, 64	Объявление и вызов функций. Ключевое слово def. Аргументы	2	1	1	Лекция-практикум	Написание собственных функций
65, 66	Функции с несколькими аргументами. Практикум	2	1	1	Практикум	Функция расчета площади
67, 68	Оператор return. Отличие return от print()	2	1	1	Практикум	Программа с использованием функций
69, 70	Создание программы с функциями (калькулятор с функциями)	2	0	2	Практикум	Программа с функциями
Май	Раздел 10: Итоговый проект (4 ч.)					
71, 72	Проектирование, написание кода, отладка и защита итогового проекта	2	1	1	Проект	Защита итогового проекта
	ИТОГО	72	22	50		