

КОМИТЕТ ПО ДЕЛАМ ОБРАЗОВАНИЯ ГОРОДА ЧЕЛЯБИНСКА
Муниципальное автономное общеобразовательное учреждение
«Средняя общеобразовательная школа № 43 г. Челябинска»

СОГЛАСОВАНО
Протокол методического
совета *№ 4*

« *15* » *мая* *2023* г.



УТВЕРЖДЕНО

Директор

МАОУ «СОШ № 43 г. Челябинска»

Г.В. Валова

Приказ №

286

« *15* » *мая* *2023* г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ**

«ЧАС КОДА PYTHON»

Срок реализации: 1 год

Возраст обучающихся: 12-13 лет

Петухов Николай Александрович,
учитель физики

Челябинск, 2023г

Оглавление

№	Название раздела	№ страниц
Раздел 1. Комплекс основных характеристик программы		
1.1	Пояснительная записка	3
1.2	Цель и задачи программы	6
1.3	Учебный план	7
1.4	Содержание программы	8
1.5	Планируемые результаты	9
Раздел 2. Комплекс организационно-педагогических условий		
2.1	Календарный учебный график (со сроками промежуточной аттестации)	11
2.2	Условия реализации программы	11
2.3	Формы промежуточной и итоговой аттестации	12
2.4	Оценочные материалы	12
2.5	Методическое обеспечение дополнительной образовательной программы	13
2.6	Воспитательная деятельность	13
2.7	Список литературы	16
Приложения		17
	Приложение 1 Календарно-тематическое планирование	
	Приложение 2 Список примеров дидактического материала	
	Приложение 3 Система диагностики результативности образовательной программы	

Раздел 1. «Комплекс основных характеристик программы»

1.1 Пояснительная записка

Данная программа является дополнительной общеобразовательной общеразвивающей технической направленности, очной формы обучения, сроком реализации 1 год для детей 12-13 лет.

Программа позволяет обучающимся приобщиться к инженерно-техническим знаниям в области инновационных технологий, сформировать техническое мышление.

Персональные компьютеры широко внедряются в науку и технику, управленческую деятельность, технологические процессы. Программирование применяется в различных областях науки и техники является одним из основных способов реализации технологических решений. Отечественная промышленность испытывает огромную потребность в высококвалифицированных специалистах и, прежде всего, в специалистах занятых в наукоёмких областях, где применяются инновационные технологии. Профессия программист не только увлекательна, но и одна из наиболее востребованных на рынке труда. Особенно востребованы специалисты по языку программирования Python.

Python – это простой в освоении и мощный язык программирования. Он предоставляет эффективные высокоуровневые структуры данных, а также простой, но эффективный подход к объектно-ориентированному программированию. Его синтаксис и динамическая типизация наряду с тем, что он является интерпретируемым, делают его идеальным языком для написания сценариев и быстрой разработки приложений в различных областях и на большинстве платформ. Для успешной работы программист должен знать определенный минимальный набор алгоритмов, а также владеть способами их программной реализации. Программа обучения предусматривает изучение основных возможностей среды разработки Python и методов работы в ней, изучение фундаментальных понятий программирования и базовых структур данных. Для освоения программы предварительного знания языков программирования и практики программирования на каком-либо другом языке не требуется.

Проект программы составлен в соответствии с государственными требованиями к образовательным программам системы дополнительного образования детей на основе следующих нормативных документов

1. Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»;
2. Приказ Министерства просвещения Российской Федерации от 09 ноября 2018 года № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
3. Письмо Минобрнауки РФ от 18.11.2015 № 09-3242 «О направлении рекомендаций» (вместе «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»);
4. Распоряжение Правительства Российской Федерации от 4 сентября 2014 года № 1726-р «Концепция развития дополнительного образования детей»;
5. Распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года»;
6. Постановление Правительства Российской Федерации от 30 декабря 2015 года № 1493 «О государственной программе «Патриотическое воспитание граждан Российской Федерации на 2016-2020 годы»;
7. Постановление Главного государственного санитарного врача РФ от 04.07.2014 № 41 «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций

дополнительного образования детей»;

8. Федеральная целевая программа развития образования на 2016-2020 годы, утвержденная Постановлением Правительства Российской Федерации от 23 мая 2015 года № 1499;

9. Устав учреждения. Локальные нормативные акты учреждения.

Программа **актуальна** в связи с широким распространением цифровой техники в обществе, в связи с возрастающей потребностью обучающихся в освоении цифровых технологий и повышением их интереса к современным языкам программирования. В век информационных технологий практически каждый школьник к 5-6 классу уже владеет навыками работы на персональном компьютере и работы в сети Интернет. Содержание программы предусматривает ознакомление воспитанников с основами работы в интегрированной среде программирования и основными возможностями языка программирования Python. Программа позволяет освоить практически все операторные конструкции и познакомиться с основными функционального и объектного программирования. Воспитанники, прошедшие обучение по данной программе, получают знания, умения и навыки, необходимые для сознательного выбора в будущем профессии связанной с программированием. Программа позволяет школьникам познакомиться и понять тонкости профессии программист и профессий связанных с разработкой IT-приложений, оценить себя в этих профессиях, выработать профессионально важные качества.

Новизна данной дополнительной общеобразовательной общеразвивающей программы заключается в построении индивидуальной образовательной траектории обучающегося, в приобретении им знаний, востребованных на рынке труда, в повышении самооценки и осознании перспектив будущей жизни, дальнейшей социализации.

Программа курса:

- обеспечивает знакомство с фундаментальными понятиями алгоритмизации и программирования на доступном уровне;
- имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика;
- охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний;
- ориентирована на существующий парк вычислительной техники и дополнительные ограничения;
- допускает возможность варьирования в зависимости от уровня подготовки и интеллектуального уровня учащихся (как группового, так и индивидуального);
- предусматривает возможность индивидуальной работы с учащимися.

Отличительная особенность данной программы заключается в том, что на занятиях учащиеся познакомятся с теоретическими аспектами и синтаксисом языка, а также обучатся практическим навыкам программирования в среде Python. Занятия начинаются с практического знакомства со средой программирования Python, далее идет непосредственное изучение синтаксических конструкций языка и отработка навыков применения элементов программирования при решении задач и создании игр. Каждая новая тема завершается практическими задачами, способствующими овладению методики программирования и изучению языка Python.

Программа модифицированная: за основу взят и переработан ряд программ других педагогических работников школ и дополнительного образования, а также пособия Бриггса Джейсона «Python для детей. Самоучитель по программированию».

Особенности возрастной группы

Обучение рассчитано на детей 12-13 лет

Уровень освоения программы – стартовый

Объём программы – 36 часов

Срок освоения программы – 1 год

Режим занятий: 1 раз в неделю по 1 академическому часу (*45 минут*)

Условия реализации программы:

Количество объединений (групп): 1

Наполняемость группы: 11 человек - по количеству рабочих мест (компьютеров)

Группа набирается из учащихся по желанию участвовать в технической деятельности по направлению.

Формы занятий определяются количеством детей и особенностями материала. Занятия проводятся в компьютерном классе. Занятий для самостоятельного изучения материала вне аудитории не предусмотрено.

Форма занятий — по количеству детей, фронтальная, при которой, работа преподавателя ведется сразу со всей группой в едином темпе и с общими задачами. При этом выделяется время на проверку работы каждого обучающегося через определенные промежутки времени, для фиксации правильности выполнения полученного материала

1.2 Цель и задачи

Цель программы:

формирование интереса к программированию и развитие творческих способностей обучающихся средствами компьютерных технологий.

Задачи:

Обучающие:

- сформировать представление об основах программирования в среде Python;
- сформировать навыки грамотной работы в системе программирования Python;
- ознакомить с базовыми понятиями теории алгоритмов при решении математических задач;
- обучить методам решения задач, реализуемым на языке Python;
- сформировать практические навыки решения прикладных задач;
- сформировать навыки поиска информации, работы с технической литературой.

Развивающие:

- развить навыки самостоятельного и творческого подхода к решению задач с помощью средств современной вычислительной техники;
- развить алгоритмическое и логическое мышление учащихся;
- развить творческие способности обучающихся, их потребность в самореализации;
- развить интеллектуальные и практические умения, самостоятельно приобретать и применять на практике полученные знания.

Воспитательные:

- содействовать воспитанию устойчивого интереса к изучению программирования;
- содействовать воспитанию информационной культуры;
- формировать потребность в творческой деятельности, стремление к самовыражению через техническое творчество;
- содействовать воспитанию интереса профессиям, связанным с программированием.

1.3. Учебный план

№ п/п	Наименование темы	Всего	Теория	Практика	Формы аттестации/ контроля
1	Вводное занятие. Инструктаж по технике безопасности. О языке программирования Python	1	1	-	Дискуссия, наблюдение
2	Вычисления в Python	1	0,5	0,5	Наблюдение, практическая работа
3	Переменные как ярлыки данных. Использование переменных	1	0,5	0,5	Наблюдение, практическая работа
4	Строки	1	0,5	0,5	Наблюдение, практическая работа
5	Списки и кортежи	1	0,5	0,5	Наблюдение, практическая работа
6	Словари в Python	1	0,5	0,5	Наблюдение, практическая работа
7	Конструкция <i>if, if-then-else</i> . Команды <i>if</i> и <i>elif</i>	2	1	1	Наблюдение, практическая работа
8	Объединение условий. Переменные без значения – <i>None</i>	1	-	1	Наблюдение, практическая работа
9	Использование цикла <i>for</i>	2	1	1	Наблюдение, практическая работа
10	Цикл <i>while</i>	1	0,5	0,5	Наблюдение, практическая работа
11	Применение функций	1	-	1	Практическая работа
12	Применение модулей	1	-	1	Практическая работа
13	Разделение объектов на классы	2	1	1	Наблюдение, практическая работа
14	Другие полезные свойства объектов и классов	1	-	1	Практическая работа
15	Использование встроенных функций	4	2	2	Наблюдение, практическая работа
16	Создание копий. Ключевые слова	1	-	1	Практическая работа
17	Генерация случайных чисел. Управление оболочкой	2	-	2	Практическая работа
18	Управление временем	1	-	1	Практическая работа
19	Использование модуля черепашки	1	-	1	Практическая работа
20	Создание анимации	3	1	2	Наблюдение, практическая работа
21	Первая игра «Человек спешит к выходу»	6	-	6	Творческая работа
22	Итоговое занятие	1	-	1	Защита итоговой работы

1.4 Содержание программы

Тема 1. «Вводное занятие. Инструктаж по технике безопасности. О языке программирования Python»

Теория: инструктаж по технике безопасности, знакомство со средой программирования Python.

Тема 2. «Вычисления в Python»

Практика: применение простых математических расчетов, операторов в Python; порядок выполнения операций.

Тема 3. «Переменные как ярлыки данных. Использование переменных»

Практика: создание переменных, печать значения переменной, изменения значения переменной, использование переменных в расчетах.

Тема 4. «Строки»

Практика: создание строк (введение текста в кавычках, синтаксические ошибки, проблемы со строками, переменные внутри строк, умножение строк).

Тема 5. «Списки и кортежи»

Практика: создание списков, отображение элементов списка, добавление элементов в список, списковая арифметика, кортежи (списки в круглых скобках).

Тема 6. «Словари в Python»

Практика: создание элементов и соответствующих им значений («ключи» и связанные «значения»)

Тема 7. «Конструкция if, if-then-else. Команды if и elif»

Теория: конструкция *if*, группа команд (блок), условие и сравнение значений, конструкция *if-then-else*, усиление конструкции *if* с помощью *elif*

Практика: применение конструкций *if*, *if-then-else* и *elif*.

Тема 8. «Объединение условий. Переменные без значения – None»

Практика: объединение условий (использование *and* (*и*) и *or* (*или*)), назначение переменной пустого значения (*None*), разница между строками и числами.

Тема 9. «Использование цикла for»

Теория: цикл *for* (определенное число повторов), функции *range* и *list*

Практика: применение цикла *for* в различных ситуациях.

Тема 10. «Цикл while»

Практика: применение цикла *while* (неизвестное число повторов).

Тема 11. «Применение функций»

Практика: строение функции (имя, аргумент тело), переменные и область видимости.

Тема 12. «Применение модулей»

Практика: применение модулей для группировки функций, переменных и других фрагментов Python-кода (команда *import*, функция *asctime*, модуль *sys*, функция *readline*).

Тема 13. «Разделение объектов на классы»

Теория: разделение объектов по смысловым группам – создание классов (функция *pass*), определение функций класса и их добавление (функция *def*), вызов других функций определенного класса (аргумент *self*), использование функций для задания характеристик класса.

Практика: применение классов и объектов, объекты и классы в картинках.

Тема 14. «Другие полезные свойства объектов и классов»

Практика: унаследованные функции, функции, вызывающие другие функции, инициализация объектов (задание свойств объектов – функции *__init__*, *def__init__* (два аргумента)).

Тема 15. «Использование встроенных функций»

Теория: функции *abs* (возврат абсолютного значения (модуля) числа), *bool* (булево значение – одно или два значения), *dir* (информация и любом переданном значении), *eval* (принятие в качестве аргумента строки и выполнение ее в качестве кода), *exec* (аналогично функции *eval* для выполнения более сложного кода), *float* (преобразование строки или числа в

вещественное число), *int* (преобразование строки или числа в целое число), *len* (возвращение количества элементов объекта), *max* и *min* (возврат наибольшего и наименьшего элементов списка или кортежа), *range* (возврат итератора – повтор некоего действия), *sum* (сложение элементов списка, возвращая их сумму).

Практика: использование встроенных функций *abs*, *bool*, *dir*, *eval*, *exec*, *float*, *int*, *len*, *max* и *min*, *range*, *sum*; работа с файлами (создание, открытие, запись файлов).

Тема 16. «Создание копий. Ключевые слова»

Практика: создание копий с помощью модуль *copy*, ключевые слова и модуль *keyword*.

Тема 17. «Генерация случайных чисел. Управление оболочкой»

Практика: генерация случайных чисел с помощью модуля *random* (использование *randint*, получение случайного элемента списка с помощью *choice*, перетасовка списка с помощью *shuffle*); управление оболочкой с помощью модуля *sys* (выход из оболочки с помощью функции *exit*, ввод данных и объект *stdin*, вывод данных и объект *stdout*).

Тема 18. «Управление временем»

Практика: работа со временем и модуль *time* (преобразование дат с помощью *asctime*, получение даты и времени с помощью *localtime*, замедление выполнения программы – функция *sleep*); сохранение информации.

Тема 19. «Использование модуля черепашки»

Практика: создание холста, перемещение черепашки.

Тема 20. «Создание анимации»

Теория: знакомство с модулем *tkinter*.

Практика: создание кнопки, именованные аргументы, рисование линий, прямоугольников и множества прямоугольников, рисование в цвете, рисование дуг, рисование многоугольников, отображение текста, вывод изображений, создание простых анимаций).

Тема 21. «Первая игра «Человечек спешит к выходу»»

Практика: создание собственной игры с использованием элементов пройденного курса по Python-программированию.

Тема 22. «Итоговое занятие»

Практика: защита зачетной работы.

1.5 Планируемые результаты

Предметные результаты:

- знание основных предметных понятий программирования, компьютерных наук и их свойств;
- знание базового синтаксиса и инструментария языка программирования Python, умение применять язык программирования Python на практике;
- умение применять объектно-ориентированную модель в программировании;
- навык разработки эффективных алгоритмов и программ на основе изучения языка программирования Python.

Личностные результаты:

- способность организовывать учебное сотрудничество и совместную деятельность со сверстниками в процессе проектной и учебно-исследовательской деятельности;
- понимание необходимости уважительного, организованного и ответственного отношения к учению, труду, другому человеку, его мнению и деятельности;
- умение алгоритмически и логически мыслить;
- знание правил поведения, социальных норм, ролей и форм социального взаимодействия в группах.

Метапредметные результаты:

- умение работать с различными источниками информации, извлекать и анализировать необходимую информацию из открытых источников;
- способность составлять и изменять план действий, необходимый для достижения цели, предвидеть результат и достигать его;
- умение выполнять проекты в соответствии с техническим заданием;

– знание правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой и оборудованием.

Раздел 2. «Комплекс организационно-педагогических условий»

2.1 Календарно - тематический план

1	Продолжительность освоения программы	36 недель, 9 месяцев
2	Начало реализации программы (начало учебного года)	05.09.2023
3	Окончание реализации программы (окончание учебного года)	31.05.2024
4	Входной (вводный) мониторинг (входное обследование уровня подготовленности обучающихся)	11.09.2023
5	Регламентирование образовательного процесса (режим работы объединения)	Занятия проводятся 1 раз в неделю, продолжительностью 45 минут (1 академический час)
6	Выходные и праздничные дни	Праздничные дни, установленные Правительством РФ: 4 ноября - День народного единства; 23 февраля - День защитника Отечества; 8 марта - Международный женский день; 1 мая - Праздник Весны и Труда; 9 мая - День Победы.
7	Каникулы (сроки, продолжительность)	31.12.23 - 10.01.24 01.06.24 – 31.08.24
9	Сроки промежуточной аттестации	18.12.2023-23.12.2023
	Сроки проведения воспитательных мероприятий	24.12.23-29.04.23
10	Дата итоговой аттестации	22.05.2024-31.05.2024

2.2 Условия реализации программы

Материально-техническое обеспечение

Успешной реализации учебного процесса способствует соответствующая материально-техническая база.

Наличие:

- 1) учебного кабинета для занятий с детьми;
- 2) освещения (естественного и искусственного в соответствии с гигиеническими требованиями);
- 3) мебели (учебных стульев и столов – стандартные, комплектные).

Наглядное обеспечение

Коллекция видеоматериалов, электронных презентаций, Интернет-ресурсов по тематическим разделам программы.

Дидактическое обеспечение

Дидактический материал включает в себя примеры оценочных материалов по темам программы и задание к итоговому продукту учащихся (Приложение 2).



2.3 Формы промежуточной и итоговой аттестации

В группе проводится текущая, промежуточная и итоговая аттестация, которая отслеживает личностный рост ребёнка по следующим параметрам:

- усвоение знаний по базовым темам программы;
- овладение умениями и навыками, предусмотренными программой;
- овладение знаниями использования базового синтаксиса и инструментария языка программирования Python;
- формирование коммуникативных качеств, трудолюбия и работоспособности.

Время проведения	Методы проверки	Формы проверки
Текущий контроль		
По окончании изучения раздела	Анализ практических работ	Практическая работа (контрольное задание)
Промежуточный контроль		
В конце первого полугодия	Анализ контрольной работы	Контрольная работа
Итоговый контроль		
В конце учебного года	Анализ зачетной работы	Итоговая (зачетная) работа

Используются следующие формы проверки: защита практических работ.

Методы проверки: наблюдения, беседа с учащимися, педагогический анализ проводимых отчетных мероприятий.

2.4 Контрольно-оценочные материалы

На занятиях применяется поурочный, тематический и итоговый контроль. Уровень освоения материала выявляется в беседах, в выполнении практических и творческого задания. В течение года ведется индивидуальное педагогическое наблюдение за развитием каждого обучающегося (Приложение 3).

Результаты освоения программного материала определяются по трём уровням: высокий, средний, низкий (Приложение 3).

Основная форма подведения итогов – выполнение практических работ и итогового задания. Критериями оценки результативности обучения являются уровень теоретической и практической подготовки учащихся (приложение 3).

Оценка личностного развития определяются через изучение качеств личности.

Результаты развития качеств личности заносятся в диагностическую карту мониторинга (приложение 3).

Оцениваются следующие качества личности:

- Самооценка.
- Активность, организаторские способности.
- Коммуникативные навыки, коллективизм.
- Ответственность, самостоятельность, дисциплинированность.
- Нравственность, гуманность.
- Креативность, склонность к исследовательско-проектировочной деятельности.

Используется 3-бальная система оценки результатов:

3 балла ярко - проявляются

2 балла - проявляются в соответствии с возрастом

1 балл - слабо проявляются

0 баллов - не проявляются

Оценка развития метапредметных результатов учащихся определяются по трём уровням: высокий, средний, низкий (приложение 3).

2.5 Методическое обеспечение

Организация образовательного процесса по данной программе предполагает создание для обучающихся творческой, свободной, комфортной среды. Этому способствует использование педагогом **методов обучения**, позволяющих достичь максимального результата. К ним относятся беседа, практическое занятие, мастер-класс, защита проектов. Применяются **активные методы обучения**: выполнение практических работ. Педагогом активно используются **современные образовательные технологии**: проектные, информационно-коммуникационные, личностно ориентированного обучения. Занятия по программе строятся на следующих принципах: усвоения материала от простого к сложному, единства воспитания и обучения, последовательности, доступности, индивидуальности, самореализации. **Характер деятельности обучающихся**: поисковый, исследовательский.

Познавательный интерес детей усиливается за счет систематического проведения мероприятий **практической деятельности**. Данный вид деятельности включает в себя решение задач и защиту практических работ.

Занятия по программе строятся на следующих принципах: усвоения материала от простого к сложному, единства воспитания и обучения, последовательности, доступности, индивидуальности, самореализации.

Характер деятельности обучающихся.

По внешним признакам деятельности педагога и учащихся:

Словесный – беседа, лекция, обсуждение, рассказ, анализ.

Наглядный – показ педагогом решения примера задачи.

Практический – создание мини проектов, участие в различных конкурсах и олимпиадах по программированию.

По степени активности познавательной деятельности учащихся:

Объяснительно-иллюстративные - учащиеся воспринимают и усваивают готовую информацию;

Репродуктивный – учащиеся воспроизводят полученные знания и освоенные способы деятельности, это практические работы, а также участие учащихся в олимпиадах, конкурсах.

Исследовательский – овладение учащимися методами научного познания, самостоятельной творческой работы это - самостоятельный поиск и создание алгоритма для решения задач.

По логичности подхода:

Аналитический – анализ созданного алгоритма.

По критерию степени самостоятельности и творчества в деятельности обучаемых:

Частично-поисковый – учащиеся участвуют в коллективном поиске, в процессе решения задач на программирование, разборе уже созданных проектов.

2.6 Воспитательная деятельность

Целью воспитания является развитие личности, самоопределение и социализация детей на основе социокультурных, духовно-нравственных ценностей и принятых в российском обществе правил и норм поведения в интересах человека, семьи, общества и государства, формирование чувства патриотизма, гражданственности, уважения к памяти защитников Отечества и подвигам Героев Отечества, закону и правопорядку, человеку труда и старшему поколению.

Задачи воспитания

1. Усвоение знаний норм, духовно-нравственных ценностей, традиций; Формирование и развитие личностных отношений к этим нормам, ценностям, традициям (их освоение, принятие); приобретении соответствующего этим нормам, ценностям, традициям социокультурного опыта поведения, общения, межличностных и социальных отношений,

применения полученных знаний. Разработчик программы конкретизирует задачи воспитания детей по программе с учётом её предметного содержания, направленности.

Целевые ориентиры воспитания: интерес к технической деятельности, истории техники в России и мире, к достижениям российской и мировой технической мысли; понимание значения техники в жизни российского общества; интереса к личностям конструкторов, организаторов производства; ценностей авторства и участия в техническом творчестве; навыков определения достоверности и этики технических идей; отношения к влиянию технических процессов на природу; ценностей технической безопасности и контроля; отношения к угрозам технического прогресса, к проблемам связей технологического развития России и своего региона; уважения к достижениям в технике своих земляков; воли, упорства, дисциплинированности в реализации проектов; опыта участия в технических проектах и их оценки;

Основной формой воспитания и обучения детей является **учебное занятие**. В ходе учебных занятий в соответствии с предметным и метапредметным содержанием программы обучающиеся: усваивают информацию, имеющую воспитательное значение; получают опыт деятельности, в которой формируются, проявляются и утверждаются ценностные, нравственные ориентации; осознают себя, способными к нравственному выбору; участвуют в освоении и формировании среды своего личностного развития, творческой самореализации

Практические занятия детей по программе «Кодим в среде scratch» способствуют усвоению и применению правил поведения и коммуникации, формированию позитивного и конструктивного отношения к событиям, в которых они участвуют, к членам своего коллектива

В **коллективных играх** проявляются и развиваются личностные качества: эмоциональность, активность, нацеленность на успех, готовность к командной деятельности и взаимопомощи. **Итоговые мероприятия:** шахматные турниры, соревнования— способствуют закреплению ситуации успеха, развивают рефлексивные и коммуникативные умения, ответственность, благоприятно воздействуют на эмоциональную сферу детей.

К методам оценки результативности реализации программы воспитания можно отнести: — педагогическое наблюдение, в процессе которого внимание педагогов сосредотачивается на проявлении в деятельности детей и в её результатах определённых в данной программе целевых ориентиров воспитания, а также на проблемах и трудностях достижения воспитательных задач программы

Самоанализ и самооценка обучающихся по итогам деятельности, отзывы родителей (законных представителей) и других участников образовательных событий и мероприятий также дают возможность для выявления и анализа наиболее значимых результатов воспитания детей.

План воспитательной работы			
Мероприятие	Сроки	Форма	Практический результат
«Техника безопасности», «Пожарная безопасность».	Сентябрь	Беседа	Фотоматериалы
«Взаимопомощь-это важно!»	Сентябрь	Беседа	Фотоматериалы
Участие в творческом конкурсе «Алло, мы ищем таланты»	Октябрь	Творческий конкурс	Фото- и видеоматериалы с выступлением детей
«Мои интересы и склонности»	Октябрь	Беседа	Фотоматериалы
Экскурсия в Кванториум	Ноябрь	Экскурсия	Фотоматериалы
Урок мужества	Декабрь	Беседа с презентацией	Фотоматериалы
«ЗОЖ-наше будущее»	Январь	Беседа	Фотоматериалы
«День памяти»	Февраль	Час общения на тему вывода войск из Афганистана	Фотоматериалы
«Ложь и правда»	Март	Час обсуждения	Фотоматериалы
«Удивительный мир космоса»	Апрель	Игра-викторина	Фотоматериалы
«Умей сказать: «Нет!»»	Май	Беседа-практикум	Фотоматериалы

2.7 Список литературы

Список литературы для педагога

1. Долинский М.С. Решение сложных и олимпиадных задач по программированию : Учебное пособие / М.С. Долинский. – М. : – 2006. – 366 с.
2. Краля Н. А. Метод учебных проектов как средство активизации учебной деятельности учащихся: Учебно-методическое пособие / Под ред. Ю. П. Дубенского. – Омск : Изд-во ОмГУ, 2005. – 9 с.
3. Майк МакГрат «Программирование на Python для начинающих» / МакГрат Майк. – М. : Эксмо, 2015. – 192 с.
4. Федоров Д. Ю. Основы программирования на примере языка Python : учебное пособие / Д.Ю. Федоров. – Санкт-Петербург: 2016.
5. Щерба А.В. Изучение языка программирования Python на основе задач УМК авторов И.А. Калинин и Н.Н. Самылкина – учебное пособие / А.В. Щерба. – М. : МПГУ, 2015. – 30 с.

Список литературы для обучающегося

1. Бриггс Джейсон Python для детей. Самоучитель по программированию / Джейсон Бриггс; пер. с англ. Станислава Ломакина. – 3-е изд. – М. : Манн, Иванов и Фербер, 2019. – 320 с.

2. Майк МакГрат «Программирование на Python для начинающих» / МакГрат Майк. – М. : Эксмо, 2015. – 192 с.
3. Федоров Д. Ю. Основы программирования на примере языка Python : учебное пособие / Д.Ю. Федоров. – Санкт-Петербург: 2016.

ПРИЛОЖЕНИЕ 1

Календарно-тематическое планирование
 программы «Час кода PYTHON», 12-13 лет.
 учитель физики – Петухов Николай Александрович

№ занятия	№ темы	Тема учебного занятия	Дата	Часы	Содержание деятельности			
					Теоретическая часть занятия		Практическая часть занятия	
					Количество часов	Форма организации деятельности	Количество часов	Форма организации деятельности
1	1	Вводное занятие. Инструктаж по технике безопасности. О языке программирования Python		1	1	групповая		
2	2	Вычисления в Python		1	0,5	групповая	0,5	групповая
3	3	Переменные как ярлыки данных. Использование переменных		1	0,5	групповая	0,5	групповая
4	4	Строки		1	0,5	групповая	0,5	групповая
5	5	Списки и кортежи		1	0,5	групповая	0,5	групповая
6	6	Словари в Python		1	0,5	групповая	0,5	групповая
7	7	Конструкция <i>if, if-then-else</i> . Команды <i>if</i> и <i>elif</i>		1	1	групповая		
8	7	Конструкция <i>if, if-then-else</i> . Команды <i>if</i> и <i>elif</i>		1			1	групповая
9	8	Объединение условий. Переменные без значения – <i>None</i>		1			1	групповая
10	9	Использование цикла <i>for</i>		1	1	групповая		

11	9	Использование цикла <i>for</i>		1			1	<i>групповая</i>
12	10	Цикл <i>while</i>		1	0,5	<i>групповая</i>	0,5	<i>групповая</i>
13	11	Применение функций		1			1	<i>групповая</i>
14	12	Применение модулей		1			1	<i>групповая</i>
15	13	Разделение объектов на классы		1	1	<i>групповая</i>		
16	13	Разделение объектов на классы		1			1	<i>групповая</i>
17	14	Другие полезные свойства объектов и классов		1			1	<i>групповая</i>
18	15	Использование встроенных функций		1	1	<i>групповая</i>		
19	15	Использование встроенных функций		1	1	<i>групповая</i>		
20	15	Использование встроенных функций		1			1	<i>групповая</i>
21	15	Использование встроенных функций		1			1	<i>групповая</i>
22	16	Создание копий. Ключевые слова		1			1	<i>групповая</i>
23	17	Генерация случайных чисел. Управление оболочкой		1			1	<i>групповая</i>
24	17	Генерация случайных чисел. Управление оболочкой		1			1	<i>групповая</i>
25	18	Управление временем		1			1	<i>групповая</i>
26	19	Использование модуля черепашки		1			1	<i>групповая</i>
27	20	Создание анимации		1	1	<i>групповая</i>		
28	20	Создание анимации		1			1	<i>групповая</i>
29	20	Создание анимации		1			1	<i>групповая</i>
30	21	Первая игра «Человечек спешит к выходу»		1			1	<i>групповая</i>
31	21	Первая игра «Человечек спешит к выходу»		1			1	<i>групповая</i>
32	21	Первая игра «Человечек спешит к выходу»		1			1	<i>групповая</i>
33	21	Первая игра «Человечек спешит к выходу»		1			1	<i>групповая</i>

34	21	Первая игра «Человечек спешит к выходу»		<i>1</i>			<i>1</i>	<i>групповая</i>
35	21	Первая игра «Человечек спешит к выходу»		<i>1</i>			<i>1</i>	<i>групповая</i>
36	22	Итоговое занятие		<i>1</i>			<i>1</i>	<i>групповая</i>
ИТОГО:				<i>36</i>	<i>10</i>		<i>26</i>	

Список примеров дидактического материала Примеры оценочных заданий темам программы

Задание 1

Создайте список своих любимых развлечений и сохраните его в переменной `games`. Теперь создайте список любимых лакомств, сохранив его в переменной `foods`. Объедините два этих списка, сохранив результат в переменной `favorites`, и напечатайте значение этой переменной.

Задание 2

Есть 3 дома, на крыше каждого из которых прячутся по 25 ниндзя, и есть 2 туннеля, в каждом из которых скрывается по 40 самураев. Сколько всего воинов решили устроить заварушку? (Ответ можно найти, введя в оболочке Python арифметическое выражение.)

Задание 3

Создайте две переменные: пусть одна хранит ваше имя, а другая фамилию. Теперь с помощью строки с метками `%s` напечатайте приветствие вроде такого: «Привет, Брандо Икетт!».

Задание 4

Как думаете, что делает этот код? Сначала попробуйте в этом разобраться, не вводя код в оболочку Python, а затем проверьте свой ответ.

```
>>> money = 2000
>>> if money > 1000:
    print("Я богат!")
else:
    print("Я не богат!")
    print("Может, когда-нибудь потом...")
```

Задание 5

Создайте конструкцию `if`, которая проверяет, действительно ли количество бисквитов (которое задано в переменной `twinkies`) меньше 100 или больше 500. Если это условие выполняется, пусть ваша программа напечатает сообщение «Слишком мало или слишком много».

Задание 6

Создайте конструкцию `if`, которая проверяет, соответствует ли заданная в переменной `money` сумма денег диапазону значений от 100 до 500 или диапазону значений от 1000 до 5000.

Задание 7

Создайте конструкцию `if`, которая печатает строку «Их слишком много», если количество ниндзя (заданное в переменной `ninjas`) меньше 50, печатает «Будет непросто, но я с ними разделаюсь», если это количество меньше 30, и печатает «Я одолею этих ниндзя!», если количество меньше 10. Проверьте, как ваш код работает с таким значением:

```
>>> ninjas = 5
```

Задание 8

Как вы считаете, что делает эта программа? Сперва придумайте вариант ответа, а потом запустите код и проверьте, угадали ли вы.

```
>>> for x in range(0, 20):
      print('привет %s' % x)
      if x < 9:
          break
```

Задание 9

Создайте цикл, который печатает четные числа до тех пор, пока не выведет ваш возраст. Если ваш возраст — нечетное число, создайте цикл, который печатает нечетные числа до совпадения с возрастом. Программа должна выводить на экран нечто подобное:

```
2
4
6
8
10
12
14
```

Задание 10

Если бы вы сейчас были на Луне, ваш вес составил бы 16,5 процентов от земного. Чтобы узнать, сколько это, умножьте свой земной вес на 0,165.

Если бы каждый год в течение следующих 15 лет вы прибавляли по одному килограмму веса, каким бы оказался ваш лунный вес в каждый из ежегодных визитов на Луну вплоть до 15-го года? Напишите программу, которая с помощью цикла `for` печатает на экране ваш лунный вес в каждом году.

Задание 11

Одним из заданий к главе 6 было создание цикла `for` для расчета вашего веса на Луне в течение 15 лет. Этот цикл можно оформить в виде функции. Создайте функцию, которая принимает начальный вес и величину, на которую вес увеличивается каждый год. Вызывать эту новую функцию нужно будет примерно так:

```
>>> moon_weight(30, 0.25)
```

Задание 12

Измените функцию из предыдущего задания так, чтобы с ее помощью можно было рассчитывать вес для разного количества лет, например 5 или 20 лет. Пусть эта функция принимает три аргумента: начальный вес, прибавку веса в год и количество лет:

```
>>> moon_weight(90, 0.25, 5)
```

Задание 13

Вместо простой функции, принимающей значения в виде аргументов, можно написать мини-программу, которая будет запрашивать эти значения с помощью `sys.stdin.readline()`. Тогда этой функции вообще не нужны аргументы:

```
>>> moon_weight()
```

Функция должна запросить начальный вес, потом прибавку веса в год и количество лет. Тогда работа с программой будет происходить примерно так:

```
Введите ваш нынешний земной вес
45
Введите ежегодный прирост вашего веса
0.4

Введите количество лет
12
```

Не забудьте импортировать модуль `sys`, прежде чем вводить код функции:

```
>>> import sys
```

Задание 14

Добавьте в класс `Giraffes` функции, при вызове которых жираф переставлял бы правую или левую ногу вперед либо назад. Назвать их можно так: `left_foot_forward`, `left_foot_back`, `right_foot_forward` и `right_foot_back`. Функция, которая ставит левую ногу жирафа вперед, может выглядеть примерно так:

```
>>> def left_foot_forward(self):
        print('левая нога впереди')
```

Теперь создайте функцию `dance`, которая научит Реджинальда танцевать (вызывая четыре только что созданные функции для передвижения ног). В результате должен получиться несложный танец:

```
>>> reginald = Giraffes()
>>> reginald.dance()
левая нога впереди
левая нога сзади
правая нога впереди
правая нога сзади
левая нога сзади
правая нога сзади
правая нога впереди
левая нога впереди
```

Задание 15

Что выведет на экран следующий код? Попробуйте угадать, а затем запустите код и проверьте вашу догадку.

```
>>> a = abs(10) + abs(-10)
>>> print(a)
>>> b = abs(-10) + -10
>>> print(b)
```

Задание 16

Постарайтесь с помощью функций `dir` и `help` узнать, как можно разбить строку на отдельные слова, а затем напишите небольшую программу, которая печатает слова из следующей строки через одно, начиная с первого слова («этот»):

```
"этот если способ вы плохо это подходит читаете для что-то ←
шифрования пошло важных не сообщений так"
```

Задание 17

Напишите программу для копирования файла. (Подсказка: нужно открыть файл, который вы собираетесь скопировать, считать из него данные, создать новый файл-копию и записать туда считанные данные.) Проверьте результат работы программы, напечатав содержимое файла-копии на экране.

Задание 18

Что выведет на экран этот код?

```
>>> import copy
>>> class Car:
    pass

>>> car1 = Car()
>>> car1.wheels = 4
>>> car2 = car1
>>> car2.wheels = 3
>>> print(car1.wheels)

>>> car3 = copy.copy(car1)
>>> car3.wheels = 6
>>> print(car1.wheels)
```

← Что появится на экране?

← Что появится на экране?

Задание 19

Создайте список ваших любимых вещей, а затем с помощью `pickle` запишите его в файл `favorites.dat`. Закройте оболочку Python, запустите ее снова и напечатайте содержимое вашего списка, загрузив его из файла.

Задание 20

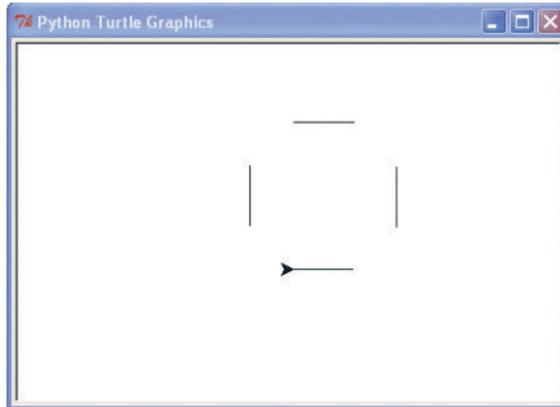
Создайте новый холст с помощью функции `Pen` модуля `turtle` и изобразите на нем прямоугольник.

Задание 21

Создайте новый холст и нарисуйте на нем треугольник. Разворачивая черепашку, сверяйтесь с изображением окружности и градусов поворота

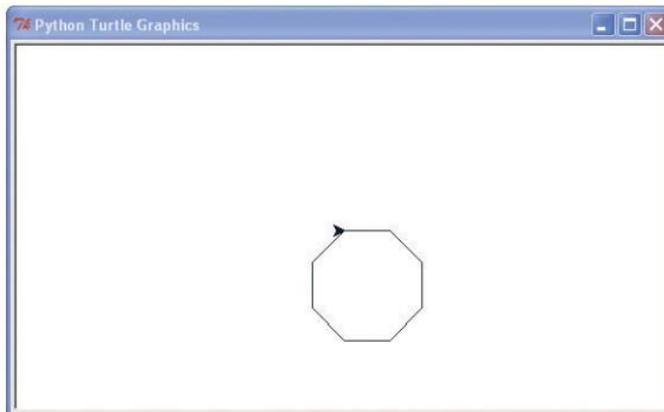
Задание 22

Напишите программу, которая рисует четыре линии, как на этом изображении (размер «квадрата» неважен, только форма):



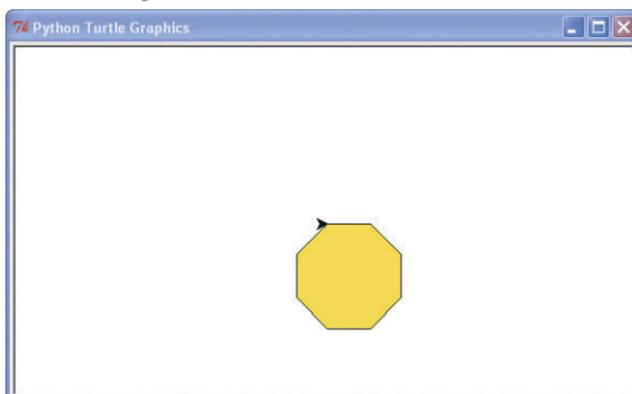
Задание 23

Мы уже умеем рисовать звезды, квадраты и прямоугольники. А теперь создадим функцию для рисования восьмиугольника. (Подсказка: для поворота черепашки используйте угол 45 градусов.)



Задание 24

Измените функцию для рисования восьмиугольника так, чтобы она изображала восьмиугольник заполненным. Попробуйте обвести его контур, как мы это делали раньше со звездой.



Задание 25

Создайте функцию для рисования звезд, которая принимает два аргумента: размер и количество точек, между которыми проведены линии, составляющие фигуру. Определение этой функции будет начинаться примерно так:

```
def draw_star(size, points):
```

Задание 26

Напишите программу, которая с помощью `tkinter` заполняет экран треугольниками. Затем модифицируйте код, чтобы треугольники были раскрашены (заполнены) различными цветами.

Задание 27

Доработайте код движущегося треугольника (см. «Создание простой анимации» на стр. 181), чтобы треугольник двигался вправо, вниз, влево и вверх, вернувшись в итоге в первоначальную позицию.

Задание 28

С помощью `tkinter` отобразите на экране свою фотографию. Не забывайте, что изображение должно быть в формате GIF! А теперь сделайте так, чтобы фотография перемещалась по экрану.

Итоговое занятие

Описание игры

- Человек очутился в плену у злодея, и ваша задача — помочь человеку спастись, добравшись до выхода на верхнем этаже.
- Человек выглядит как фигурка, которая может двигаться вправо, влево, а также прыгать. На этажах расположены платформы, на которые человеку предстоит запрыгивать.
- Цель игры — добраться до двери выхода.

Примерный код

```
from tkinter import *
import random
import time

class Game:
    def __init__(self):
        self.tk = Tk()
        self.tk.title("Человек спешит к выходу")
        self.tk.resizable(0, 0)
        self.tk.wm_attributes("-topmost", 1)
        self.canvas = Canvas(self.tk, width=500, height=500, \
                             highlightthickness=0)
```

```

self.canvas.pack()
self.tk.update()
self.canvas_height = 500
self.canvas_width = 500
self.bg = PhotoImage(file="background.gif")
w = self.bg.width()
h = self.bg.height()
for x in range(0, 5):
    for y in range(0, 5):
        self.canvas.create_image(x * w, y * h, \
            image=self.bg, anchor='nw')
self.sprites = []
self.running = True

def mainloop(self):
    while 1:
        if self.running == True:
            for sprite in self.sprites:
                sprite.move()
            self.tk.update_idletasks()
            self.tk.update()
            time.sleep(0.01)

class Coords:
    def __init__(self, x1=0, y1=0, x2=0, y2=0):
        self.x1 = x1
        self.y1 = y1
        self.x2 = x2
        self.y2 = y2

def within_x(col, co2):
    if (col.x1 > co2.x1 and col.x1 < co2.x2) \
        or (col.x2 > co2.x1 and col.x2 < co2.x2) \
        or (co2.x1 > col.x1 and co2.x1 < col.x2) \
        or (co2.x2 > col.x1 and co2.x2 < col.x2):
        return True

```

```

    else:
        return False

def within_y(col, co2):
    if (col.y1 > co2.y1 and col.y1 < co2.y2) \
        or (col.y2 > co2.y1 and col.y2 < co2.y2) \
        or (co2.y1 > col.y1 and co2.y1 < col.y2) \
        or (co2.y2 > col.y1 and co2.y2 < col.y2):
        return True
    else:
        return False

def collided_left(col, co2):
    if within_y(col, co2):
        if col.x1 <= co2.x2 and col.x1 >= co2.x1:

```

```

        return True
    return False

def collided_right(col1, col2):
    if within_y(col1, col2):
        if col1.x2 >= col2.x1 and col1.x2 <= col2.x2:
            return True
    return False

def collided_top(col1, col2):
    if within_x(col1, col2):
        if col1.y1 <= col2.y2 and col1.y1 >= col2.y1:
            return True
    return False

def collided_bottom(y, col1, col2):
    if within_x(col1, col2):
        y_calc = col1.y2 + y
        if y_calc >= col2.y1 and y_calc <= col2.y2:
            return True
    return False

class Sprite:
    def __init__(self, game):
        self.game = game
        self.endgame = False
        self.coordinates = None
    def move(self):
        pass
    def coords(self):
        return self.coordinates

```

```

class PlatformSprite(Sprite):
    def __init__(self, game, photo_image, x, y, width, height):
        Sprite.__init__(self, game)
        self.photo_image = photo_image
        self.image = game.canvas.create_image(x, y, \
            image=self.photo_image, anchor='nw')
        self.coordinates = Coords(x, y, x + width, y + height)

class StickFigureSprite(Sprite):
    def __init__(self, game):
        Sprite.__init__(self, game)
        self.images_left = [
            PhotoImage(file="figure-L1.gif"),
            PhotoImage(file="figure-L2.gif"),
            PhotoImage(file="figure-L3.gif")
        ]
        self.images_right = [
            PhotoImage(file="figure-R1.gif"),
            PhotoImage(file="figure-R2.gif"),

```

```

            PhotoImage(file="figure-R3.gif")
        ]
        self.image = game.canvas.create_image(200, 470, \
            image=self.images_left[0], anchor='nw')
        self.x = -2
        self.y = 0
        self.current_image = 0
        self.current_image_add = 1
        self.jump_count = 0
        self.last_time = time.time()
        self.coordinates = Coords()
        game.canvas.bind_all('<KeyPress-Left>', self.turn_left)
        game.canvas.bind_all('<KeyPress-Right>', self.turn_right)
        game.canvas.bind_all('<space>', self.jump)

```

```

def turn_left(self, evt):
    if self.y == 0:
        self.x = -2

def turn_right(self, evt):
    if self.y == 0:
        self.x = 2

def jump(self, evt):
    if self.y == 0:
        self.y = -4
        self.jump_count = 0

def animate(self):
    if self.x != 0 and self.y == 0:
        if time.time() - self.last_time > 0.1:
            self.last_time = time.time()
            self.current_image += self.current_image_add
            if self.current_image >= 2:
                self.current_image_add = -1
            if self.current_image <= 0:
                self.current_image_add = 1
    if self.x < 0:
        if self.y != 0:
            self.game.canvas.itemconfig(self.image, \
                image=self.images_left[2])
        else:
            self.game.canvas.itemconfig(self.image, \
                image=self.images_left[self.current_image])
    elif self.x > 0:
        if self.y != 0:
            self.game.canvas.itemconfig(self.image, \
                image=self.images_right[2])

```

```

        else:
            self.game.canvas.itemconfig(self.image, \
                image=self.images_right[self.current_image])
def coords(self):

```

```

    xy = self.game.canvas.coords(self.image)
    self.coordinates.x1 = xy[0]
    self.coordinates.y1 = xy[1]
    self.coordinates.x2 = xy[0] + 27
    self.coordinates.y2 = xy[1] + 30
    return self.coordinates

```

```

def move(self):
    self.animate()
    if self.y < 0:
        self.jump_count += 1
        if self.jump_count > 20:
            self.y = 4
    if self.y > 0:
        self.jump_count -= 1
    co = self.coords()
    left = True
    right = True
    top = True
    bottom = True
    falling = True
    if self.y > 0 and co.y2 >= self.game.canvas_height:
        self.y = 0
        bottom = False

```

```
elif self.y < 0 and co.y1 <= 0:
    self.y = 0
    top = False
```

```
if self.x > 0 and co.x2 >= self.game.canvas_width:
    self.x = 0
    right = False
elif self.x < 0 and co.x1 <= 0:
    self.x = 0
    left = False
for sprite in self.game.sprites:
    if sprite == self:
        continue
    sprite_co = sprite.coords()
    if top and self.y < 0 and collided_top(co, sprite_co):
        self.y = -self.y
        top = False
    if bottom and self.y > 0 and collided_bottom(self.y, \
        co, sprite_co):
        self.y = sprite_co.y1 - co.y2
        if self.y < 0:
            self.y = 0
        bottom = False
        top = False
    if bottom and falling and self.y == 0 \
        and co.y2 < self.game.canvas_height \
        and collided_bottom(1, co, sprite_co):
        falling = False
```

```
    if left and self.x < 0 and collided_left(co, sprite_co):
        self.x = 0
        left = False
        if sprite.endgame:
            self.game.running = False
    if right and self.x > 0 and collided_right(co, sprite_co):
        self.x = 0
        right = False
        if sprite.endgame:
            self.game.running = False
    if falling and bottom and self.y == 0 \
        and co.y2 < self.game.canvas_height:
        self.y = 4
    self.game.canvas.move(self.image, self.x, self.y)
```

```
class DoorSprite(Sprite):
    def __init__(self, game, photo_image, x, y, width, height):
        Sprite.__init__(self, game)
        self.photo_image = photo_image
        self.image = game.canvas.create_image(x, y, \
            image=self.photo_image, anchor='nw')
        self.coordinates = Coords(x, y, x + (width / 2), y + height)
        self.endgame = True
```

```

g = Game()
platform1 = PlatformSprite(g, PhotoImage(file="platform1.gif"), \
    0, 480, 100, 10)
platform2 = PlatformSprite(g, PhotoImage(file="platform1.gif"), \
    150, 440, 100, 10)
platform3 = PlatformSprite(g, PhotoImage(file="platform1.gif"), \
    300, 400, 100, 10)
platform4 = PlatformSprite(g, PhotoImage(file="platform1.gif"), \
    300, 160, 100, 10)
platform5 = PlatformSprite(g, PhotoImage(file="platform2.gif"), \
    175, 350, 66, 10)
platform6 = PlatformSprite(g, PhotoImage(file="platform2.gif"), \
    50, 300, 66, 10)
platform7 = PlatformSprite(g, PhotoImage(file="platform2.gif"), \
    170, 120, 66, 10)

```

```

platform8 = PlatformSprite(g, PhotoImage(file="platform2.gif"), \
    45, 60, 66, 10)
platform9 = PlatformSprite(g, PhotoImage(file="platform3.gif"), \
    170, 250, 32, 10)
platform10 = PlatformSprite(g, PhotoImage(file="platform3.gif"), \
    230, 200, 32, 10)

```

```

g.sprites.append(platform1)
g.sprites.append(platform2)
g.sprites.append(platform3)
g.sprites.append(platform4)
g.sprites.append(platform5)
g.sprites.append(platform6)
g.sprites.append(platform7)

```

```

g.sprites.append(platform8)
g.sprites.append(platform9)
g.sprites.append(platform10)
door = DoorSprite(g, PhotoImage(file="door1.gif"), 45, 30, 40, 35)
g.sprites.append(door)
sf = StickFigureSprite(g)
g.sprites.append(sf)
g.mainloop()

```

Дополнение к игре

Задание 1

Когда человек доберется до двери, выводите сообщение «Вы победили!». (Вспомните, как делали то же самое, печатая «Конец игры» для созданной в главе 14 игры «Прыг-скок!».)

Задание 2

В главе 15 мы создали два изображения двери — открытую и закрытую. Когда человек в нее войдет, изображение должно поменяться (с закрытой двери на открытую). Затем человек должен исчезнуть, а дверь закрыться. Тогда возникнет иллюзия, что наш герой открывает дверь, заходит и закрывает ее за собой. Измените для этого код классов `DoorSprite` и `StickFigureSprite`.

Задание 3

Создайте новый класс для движущихся платформ — `MovingPlatformSprite`. Они должны непрерывно перемещаться по горизонтали (от левой стороны игрового экрана до правой и обратно), из-за чего человеку будет сложнее добраться до выхода.

ПРИЛОЖЕНИЕ 3

СИСТЕМА ДИАГНОСТИКИ РЕЗУЛЬТАТИВНОСТИ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

	Направление диагностики	Возрастные характеристики обучающихся	Параметры диагностики	Методы диагностики	Контрольные мероприятия, методики
Обучение	I. Теоретические и практические ЗУН	12-13 лет	Предметные достижения обучающихся в процессе усвоения предметной области	Наблюдение, анализ практических работ	Практические задания по темам: вычисления в Python, переменные как ярлыки данных, использование переменных, строки, списки и кортежи, словари в Python, конструкция <i>if, if-then-else</i> . Команды <i>if</i> и <i>elif</i> , объединение условий. Переменные без значения – <i>None</i> , Использование цикла <i>for</i> , Цикл <i>while</i> , Применение функций, Применение модулей, Разделение объектов на классы, Другие полезные свойства объектов и классов, Использование встроенных функций, Создание копий. Ключевые слова, Генерация случайных чисел. Управление оболочкой, Управление временем, Использование модуля черепашки, Создание анимации

	II. Практическая творческая деятельность обучающихся	12-13 лет	Личностные достижения обучающихся в процессе усвоения предметной программы	Анализ продуктов практической деятельности: результат практической работы; метод наблюдения.	Создание групповых практических работ по темам программы
Развитие	I. Особенности личностной сферы	12-13 лет	Творческие способности	Анализ продуктов практической деятельности: результат итоговой работы и практических работ;	Итоговое задание (проект) после освоения программы
	II. Особенности личности в системе социальных отношений	12-13 лет	1. Удовлетворенность отношениями в группе, положение личности в коллективе и его сплоченность	Наблюдение	Практические занятия по темам программы
2. Коммуникативные навыки			Наблюдение	Практические занятия по темам программы	
Воспитание	I. Самоорганизация свободного времени	12-13 лет	Потребность в продуктивном проведении досуга	Наблюдение	Практические занятия по темам программы
	II. Профессиональное самоопределение	12-13 лет	Профессионально важные качества		
			Профессиональные интересы		

Оценивание предметных результатов обучения

Оцениваемые параметры	Критерии	Показатели	Степень выраженности оцениваемого качества
Теоретические знания (по основным разделам учебно-тематического плана программы)	Соответствие теоретических знаний ребенка программным требованиям	Знает базовый синтаксис Python	Высокий уровень: учащийся освоил практически весь объем знаний, предусмотренный программой. Средний уровень: объем усвоенных знаний составляет более 1/2 Низкий уровень: объем усвоенных знаний составляет менее 1/2.
		Знает правила техники безопасности во время работы за ПК.	
Практические умения и навыки, предусмотренные программой	Соответствие практических умений и навыков программным требованиям	Выполняет действия по выполнению заданий, предусмотренных программой	Высокий уровень: учащийся овладел практически всеми умениями и навыками, предусмотренными программой. Средний уровень: объем усвоенных умений и навыков составляет более 1/2. Низкий уровень: учащийся овладел менее 1/2 предусмотренных умений и навыков
		Умеет выполнять самостоятельно задачи практической работы	
		Применяет правила техники безопасности во практической деятельности	

МОНИТОРИНГ
развития качеств личности обучающихся
 Карта оценки личностного развития

Качества личности	Признаки проявления качеств личности			
	ярко проявляются 3 балла	проявляются 2 балла	слабо проявляются 1 балл	не проявляются 0 баллов
1. Активность, организаторские способности	Активен, проявляет стойкий познавательный интерес, целеустремлен, трудолюбив и прилежен, добивается выдающихся результатов, инициативен, организует деятельность других.	Активен, проявляет стойкий познавательный интерес, трудолюбив, добивается хороших результатов.	Мало активен, наблюдает за деятельностью других, забывает выполнить задание. Результативность невысокая.	Пропускает занятия, мешает другим.
2. Коммуникативные навыки, коллективизм	Легко вступает и поддерживает контакты, разрешает конфликты, дружелюбен со всеми, инициативен, по собственному желанию успешно выступает перед аудиторией.	Вступает и поддерживает контакты, не вступает в конфликты, дружелюбен со всеми, по инициативе руководителя или группы выступает перед аудиторией.	Поддерживает контакты избирательно, чаще работает индивидуально, публично не выступает.	Замкнут, общение затруднено, адаптируется в коллективе с трудом, является инициатором конфликтов.
3. Тип сотрудничества (отношение учащегося к общим делам детского объединения)	Умеет воспринимать общие дела, как свои собственные	Инициативен в общих делах	Участвует при побуждении извне	Избегает участия в общих делах
4. Самоконтроль	Умение контролировать свои поступки (приводить к должному свои действия)	Может контролировать себя сам.	Периодически контролирует себя сам	постоянно находится под воздействием контроля извне

5. Ответственность, самостоятельность, дисциплинированность	Выполняет поручения охотно, ответственно, часто по собственному желанию, может привлечь других. Всегда дисциплинирован, везде соблюдает правила поведения, требует того же от других.	Выполняет поручения охотно, ответственно. Хорошо ведет себя независимо от наличия или отсутствия контроля, но не требует этого от других.	Неохотно выполняет поручения. Начинает работу, но часто не доводит ее до конца. Справляется с поручениями и соблюдает правила поведения только при наличии контроля и требовательности преподавателя или товарищей.	Уклоняется от поручений, безответственен. Часто недисциплинирован, нарушает правила поведения, слабо реагирует на воспитательные воздействия.
6. Нравственность, гуманность	Доброжелателен, правдив, верен своему слову, вежлив, заботится об окружающих, пресекает грубость, недобрые отношения к людям,	Доброжелателен, правдив, верен своему слову, вежлив, заботится об окружающих, но не требует этих качеств от других.	Помогает другим по поручению преподавателя, не всегда выполняет обещания, в присутствии старших чаще скромнен, со сверстниками бывает груб.	Недоброжелателен, груб, пренебрежителен, высокомерен с товарищами и старшими, часто обманывает, неискренен.

**ДИАГНОСТИЧЕСКАЯ КАРТА
мониторинга развития качеств личности обучающихся**

№	Фамилия, имя	К а ч е с т в а л и ч н о с т и														
		Активность, организаторские способности			Коммуникативные навыки, коллективизм			Ответственность, самостоятельность, дисциплинированнос ть			Нравственность, гуманность			Креативность, склонность к исследовательско- проектировочной деятельности		
		дата заполнения			дата заполнения			дата заполнения			дата заполнения			дата заполнения		
		сент. 2023		май 2024	сент. 2023		май 2024	сент. 2023		май 2024	сент. 2023		май 2024	сент. 2023		май 2024
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																

Карта оценки развития метапредметных результатов учащихся

Фамилия, имя учащегося	<i>поисковые (исследовательские) умения:</i>	<i>коммуникативные умения:</i>	<i>Умения и навыки работы в сотрудничестве:</i>	<i>презентационные умения и навыки:</i>
<p>— умение самостоятельно генерировать идеи;</p> <p>— умение самостоятельно находить недостающую информацию в информационном поле;</p> <p>— умение находить несколько вариантов решения проблемы;</p> <p>— умение выдвигать гипотезы;</p> <p>— умение устанавливать причинно-следственные связи.</p>	<p>— умение инициировать учебное взаимодействие со взрослыми - вступать в диалог, задавать вопросы;</p> <p>— умение вести дискуссию;</p> <p>— умение отстаивать свою точку зрения;</p> <p>— умение находить компромисс;</p> <p>— навыки интервьюирования, устного опроса .</p>	<p>— умение коллективного планирования;</p> <p>— умение взаимодействовать с любым партнером;</p> <p>— навыки взаимопомощи в группе в решении общих задач;</p> <p>— навыки делового партнерского общения;</p> <p>— умение находить и исправлять ошибки в работе других участников группы.</p>	<p>— навыки монологической речи;</p> <p>— умение уверенно держать себя во время выступления;</p> <p>— умение использовать различные средства наглядности при выступлении;</p> <p>— умение отвечать на незапланированные вопросы.</p>	

Инструкция: опросник является вариантом экспертной оценки, заполняется педагогом на каждого ребенка по четырем видам умений. Необходимо отметить степень выраженности каждого качества, с помощью уровней, где:

Н (низкий уровень) – качество отсутствует у учащегося или выражено слабо и проявляется редко,

С (средний уровень) – качество выражено сильно и проявляется часто,

В (высокий уровень) – выражено сильно и проявляется постоянно.

Нужную букву обвести в каждой графе