

ТАМБОВСКОЕ ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«ШКОЛА № 3 — ЦЕНТР ПРОФНАВИГАЦИИ И РАЗВИТИЯ КАРЬЕРЫ»

Рассмотрена на заседании
педагогического совета
Протокол № 1
от 27.08.2025

УТВЕРЖДАЮ
Директор (ТОГАОУ «Школа №3 –
Центр профнавигации и развития
карьеры»)
В.Б. Яковлева
Приказ № 596 от 28.08.2025

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**
технической направленности
«Программирование Minecraft на языке Python»

Возраст обучающихся: *10-13 лет*

Срок реализации: *1 год*

Уровень программы: *базовый*

Разработчик программы:
*Педагог дополнительного образования
Воронов-Олемской Георгий Витальевич*

г. Тамбов, 2025

ИНФОРМАЦИОННАЯ КАРТА ПРОГРАММЫ

1. Учреждение	Тамбовское областное государственное автономное образовательное учреждение «Школа №3 – Центр профнавигации и развития карьеры»
2. Полное название программы	Дополнительная общеобразовательная общеразвивающая программа технической направленности «Программирование Minecraft на языке Python»
3. Сведения об авторах	
3.1. Ф.И.О., должность автора	Воронов-Олемской Георгий Витальевич, педагог дополнительного образования
4. Сведения о программе:	
4.1 Дата разработки	
4.2. Нормативная база:	<ul style="list-style-type: none"> ✓ Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями от 01.04.2024); ✓ Приказ Министерства просвещения Российской Федерации от 27 сентября 2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»; ✓ Приказ Министерства просвещения Российской Федерации от 3 сентября 2019 г. № 467 «Об утверждении Целевой модели развития региональной системы дополнительного образования детей» (ред. от 21.04.2023) ✓ Распоряжение Правительства Российской Федерации от 31 марта 2022 года № 678-р «Концепция развития дополнительного образования детей до 2030 года»; ✓ Письмо Министерства образования и науки Российской Федерации от 18 ноября 2015г. № 09-3242 «О направлении информации». Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы); ✓ Постановление Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 г. № 28 «Об утверждении санитарных правил 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»; ✓ Устав ТОГАОУ «Школа №3 – Центр профнавигации и развития карьеры»; ✓ Локальные акты, регламентирующие образовательную деятельность ТОГАОУ «Школа №3 – Центр профнавигации и развития карьеры».
4.3. Область применения	дополнительное образование
4.4. Направленность	техническая
4.6. Вид программы	дополнительная общеобразовательная общеразвивающая программа
4.7. Образовательная область	познавательное развитие
4.8. Уровень освоения	базовый
4.9. Возраст обучающихся	10-13 лет
4.10. Продолжительность обучения	1 год

1 КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

1.1 Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Творческое программирование: Minecraft и графика с Turtle на языке python» (далее – Программа) имеет техническую направленность, уровень освоения программы – базовый. Реализация программы позволит обучающимся получить необходимый объем знаний о возможностях использования Python и создать положительную мотивацию, способствующую развитию личностных, метапредметных и предметных умений, а также позволит им раскрыть свой творческий потенциал в сфере современных it-технологий.

Актуальность и практическая значимость программы

При этом Python является современным, востребованным и перспективным языком разработки, позволяющим не только заложить фундамент знаний будущих программистов и инженеров, но получить востребованные и актуальные навыки, необходимые в учебной, проектной и повседневной деятельности.

Язык программирования Python принято считать одним из самых простых в освоении. Обучаться языку возможно имея лишь базовые навыки использования компьютера, установка необходимого ПО не сложнее установки любой другой программы, а синтаксис языка не перегружен и интуитивно понятен. Данные факторы раскрывают Python максимально эффективным инструментом при обучении детей программированию.

Создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления у детей. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления. По итогам освоения программы обучающиеся получают возможность для социального, культурного и профессионального самоопределения, творческой самореализации личности по средствам мероприятий технической направленности, а также овладеют soft и hard компетенциями.

Отличительная особенность программы заключается в возможности получения обучающимися универсальных компетенций, необходимых при дальнейшем изучении не только информационных технологий, но и предметов гуманитарного и естественно-научного цикла. Также в результате изучения парадигмы объектно-ориентированного подхода к программированию происходит формирование базовых знаний и умений для работы с большинством популярных языков и необходимых при освоении других IT-направлений.

Педагогическая целесообразность программы заключается в возможности сформировать у обучающихся представление о базовых понятиях структурного программирования (данных, переменных, ветвлениях, циклах и функциях). В то же время Python является востребованным языком, он отлично подходит для знакомства с концепцией объектно-ориентированного программирования и активно применяется в различных областях от разработки веб-приложений до машинного обучения. Научившись программировать на языке Python, обучающиеся получают мощный и удобный в использовании инструмент для решения учебных задач и для создания собственных проектов. Вместе с тем

чистота и ясность его конструкций позволит в дальнейшем с лёгкостью выучить любой другой язык программирования.

Адресат программы: программа предназначена для детей в возрасте от 10 до 13 лет.

Возрастные особенности обучающихся

Подростковый возраст – остро протекающий переход от детства к взрослости. С одной стороны, для этого сложного периода показательные негативные проявления, дисгармоничность в строении личности, вызывающий характер его поведения по отношению к взрослым. С другой стороны, подростковый возраст отличается и множеством положительных факторов: возрастает самостоятельность ребенка, более разнообразными и содержательными становятся отношения с другими детьми и взрослыми, значительно расширяется сфера его деятельности. Главное, данный период отличается выходом ребенка на качественно новую социальную позицию, в которой формируется его сознательное отношение к себе как члену общества.

Условия набора обучающихся: для обучения в объединении принимаются все желающие, независимо от уровня первоначальных знаний. Данная программа является базой для перехода на более сложные программы обучения.

Объем и срок освоения программы: программа реализуется в течение 1 учебного года (144 академических часа).

Формы обучения: программа предполагает использование очной и дистанционной формы обучения. При применении формата дистанционного обучения используется информационно-коммуникационная платформа «Сферум».

Режим занятий, периодичность и продолжительность занятий: в программе предусматривается выделение фиксированного количества часов на освоение учащимися определенного раздела, при этом обучающиеся могут самостоятельно распределить учебную нагрузку, предоставив выполненное задание к установленной контрольной дате.

1.2 Цель и задачи программы

Цель программы: формирование у обучающихся базовых навыков прикладной разработки на языке программирования Python для решения практических задач и разработки продуктов.

Образовательные задачи:

сформировать навыки алгоритмического и логического мышления, грамотной разработки программ;

познакомить с принципами и методами объектно-ориентированного программирования;

обучить навыкам работы в системе программирования Python;

изучить конструкцию языка программирования Python;

обучить навыкам разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;

Развивающие задачи:

развить у обучающихся интереса к программированию;

развить логическое мышление, творческий и познавательный потенциал обучающегося, его коммуникативных способностей;

развить навыки инженерного мышления, умения работать как по предложенным инструкциям, так и находить свои собственные пути решения поставленных задач;

развить навыки эффективной деятельности в проекте;

развить стрессоустойчивость;

развить способности к самоанализу, самопознанию.

Воспитательные задачи:

воспитать мотивацию обучающихся к изобретательству, созданию собственных инженерных и программных реализаций;

привить стремление к получению качественного законченного результата;

привить информационную культуру: ответственное отношение к информации;

сформировать потребность в самостоятельном приобретении и применении знаний, потребность к постоянному саморазвитию;

воспитать социально-значимые качества личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность.

1.3 Учебный план

№ п/п	Название раздела, темы	Количество часов			Форма аттестации/контроля
		всего	теория	практика	
	Вводное занятие	2	1	1	Стартовая диагностика
1	Раздел 1. Введение в Python и модуль turtle	6	3	3	

1.1.	Установка Python и настройка среды разработки	2	1	1	Беседа, практическое задание
1.2.	Основы работы с модулем Turtle: импорт, создание объектов черепахи	2	1	1	Лекция, практическое задание
1.3.	Примеры простых команд: forward(), left(), right(), pendown()	2	1	1	Лекция, практическое задание
2.	Раздел 2. Типы данных и структуры	12	4	8	
2.1.	Обзор базовых типов данных: int, float, string, bool.	2	1	1	Лекция, практическое задание
2.2.	Работа со структурами данных: списки, кортежи, множества, словари.	2	1	1	Опрос, практическое задание
2.3.	Применение структур данных для хранения координат и параметров фигур в Turtle	4	2	2	Лекция, практическое задание
2.4.	Практические задания по созданию и манипуляции данными.	4	0	4	Контрольное занятие
3.	Раздел 3. Условные операторы и циклы	12	4	8	
3.1.	Использование операторов ветвления if, elif, else	4	2	2	Лекция, практическое задание
3.2.	Введение в циклы for и while.	2	1	1	Лекция, практическое задание
3.3.	Применение циклов для рисования фигур с использованием Turtle	2	1	1	Беседа, самостоятельная работа
3.4.	Создание графиков и узоров с помощью условий и циклов	4	0	4	Взаимооценки обучающимися работ друг друга
4	Раздел 4. Функции и анонимные функции	6	3	3	
4.1.	Определение и использование функций. Введение в анонимные функции (lambda)	2	1	1	Беседа, творческая работа
4.2.	Создание функций для рисования фигур с параметрами	2	1	1	Опрос, самостоятельная работа

4.3.	Практическое задание на написание функций для различных задач	2	1	1	Тестирование, практическое задание
5.	Раздел 5. Функции высшего порядка	6	3	3	
5.1.	Понятие и использование функций высшего порядка	2	1	1	Лекция, практическое задание
5.2.	Применение в контексте создания сложных рисунков в Turtle	2	1	1	Опрос, практическое задание
5.3.	Задание на использование функций высшего порядка в проектах	2	1	1	Беседа. Самостоятельная работа
6.	Раздел 6. Объектно-ориентированное программирование (ООП)	12	6	6	
6.1.	Основы ООП: классы и объекты	4	2	2	Лекция, практическое задание
6.2.	Инкапсуляция, наследование и полиморфизм	4	2	2	Лекция, практическое задание
6.3.	Создание собственных классов для расширения функциональности Turtle	4	2	2	Беседа, практическое задание
7.	Раздел 7. Проектная работа	4	2	2	
7.1.	Разработка индивидуальных проектов с использованием всех пройденных тем	2	0	2	Подготовка к созданию творческих проектов
7.2.	Примеры проектов: создание интерактивных игр или анимации с использованием Turtle	2	0	2	Создание творческих проектов
8	Раздел 8. Введение в Minecraft и библиотеку MCPi	6	3	3	
8.1.	Установка PyCharm, скачивание необходимых файлов, знакомство со средой и её командами	2	1	1	Беседа, практическое задание
8.2.	Установка Minecraft и библиотеки MCPi	2	1	1	Лекция, практическое задание
8.3.	Настройка сервера. Основы работы с MCPi: подключение к игре, создание объекта игрока	2	1	1	Лекция, практическое задание

9	Раздел 9. Основные команды MCPi	12	3	9	
9.1	Использование команд setBlock(), getBlock(), getTilePos()	4	1	3	Лекция, практическое задание
9.2	Создание блоков различных типов и манипуляция с ними	4	1	3	Беседа, практическое задание
9.3	Примеры использования команд для создания простых объектов	4	1	3	Опрос, практическое задание
10	Раздел 10. Работы с координатами и позициями	12	4	8	
10.1	Использование getTilePos() для получения позиции игрока	4	1	3	Лекция, практическое задание
10.2	Перемещение игрока с помощью команд player.setTilePos(x, y, z) и player.setPos(x, y, z)	4	1	3	Лекция, практическое задание
10.3	Создание системы навигации по миру Minecraft	4	2	2	Беседа, практическое задание
11	Раздел 11. Создание простых построек с помощью циклов	12	4	8	
11.1	Использование циклов для автоматизации строительства	4	1	3	Беседа, практическое задание
11.2	Создание стен, полов и других конструкций с помощью вложенных циклов	4	1	3	Лекция, практическое задание
11.3	Примеры: автоматизированное строительство	4	2	2	Лекция, практическое задание
12	Раздел 12. Работа с блоками и их свойствами	12	4	8	
12.1	Использование команд getBlock() и setBlock() для изменение состояние блоков	4	2	3	Лекция, практическое задание
12.2	Изучение различных типов блоков и их взаимодействие	4	1	3	Беседа, практическое задание
12.3	Создание интерактивных объектов на основе свойств блоков	4	1	3	Опрос, практическое задание

13	Раздел 13. Создание игровых механик	12	4	8	
13.1	Разработка простых игр и квестов внутри Minecraft	4	1	3	Лекция, практическое задание
13.2	Использование логики условий для создания интерактивных элементов	4	1	3	Опрос, практическое задание
13.3	Примеры проектов: автоматизированные фермы, системы управления, умные дома	4	2	2	Беседа, практическое задание
14	Раздел 14. Проектная работа в Minecraft	16	4	12	
14.1	Разработка индивидуальных или групповых проектов с использованием всех изученных тем	4	2	2	Беседа, практическое задание
14.2	Работа с файлами на Python для сохранения кода в txt документов, а также использование для модификаций	4	1	3	Лекция, практическое задание
14.3	Знакомство с модулями csv файлами для хранения координат игрока и блоков	4	1	3	Лекция, практическое задание
14.4	Защита проектов	4	0	4	Защита проектов
	Итоговое занятие	2	0	2	Презентация и защита творческих проектов
	ИТОГО:	144	52	92	

СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА

Вводное занятие.

Теория. Введение в историю языка программирования Python. Основные преимущества данного языка над другими. Техника безопасности.

Практика. Стартовая диагностика. Знакомство с конструктором.

Раздел 1. Введение в Python и модуль Turtle

Тема 1.1. Установка Python и настройка среды разработки

Теория. Откуда скачивать установочный файл. Процесс установки Python на компьютер и графических редакторов, знакомство и выбор редакторов кода.

Практика. Загрузка и установка инсталлятора python. Выбор и установка редактора кода и его настройка.

Тема 1.2. Основы работы с модулем Turtle: импорт, создание объектов черепахи.

Теория. Понятие модуля, использование методов turtle для рисования объектов черепахи.

Практика. Применение изученных методов для построения объектов.

Тема 1.3. *Примеры простых команд: forward(), left(), right(), pendown()*

Теория. Знакомство с методами построения линий, декартова система координат.

Практика. Использование простых команд для построения геометрических фигур.

Раздел 2. Типы данных и структуры

Тема 2.1. *Обзор базовых типов данных: int, float, string, bool.*

Теория. Знакомство с понятием типы данных. Какие арифметические операции существуют на языке Python.

Практика. Применение на практике для построения таблиц и расчетов.

Тема 2.2. *Работа со структурами данных: списки, кортежи, множества, словари.*

Теория. Понятие коллекций данных, знакомство с их методами.

Практика. Решение практических задач применяя полученные знания.

Тема 2.3. *Применение структур данных для хранения координат и параметров фигур в Turtle.*

Теория. Использование переменных и структур данных для хранения координат и взаимодействия с ними.

Практика. Обращение к переменным для получения координат из структур данных и их использования при построении графиков.

Тема 2.4. *Практические задания по созданию и манипуляции данными.*

Практика. Решение практических задач.

Раздел 3. Условные операторы и циклы

Тема 3.1. *Использование операторов ветвления if, elif, else*

Теория. Введение в понятие логики и использование операторов ветвления.

Практика. Использование условий для динамического построения фигур.

Тема 3.2. *Введение в циклы for и while*

Теория. Понятие итераций и применение циклов для повторения условий.

Практика. Использование циклов для рутинных построений

Тема 3.3. *Применение циклов для рисования фигур с использованием Turtle*

Теория. Объединение полученных знаний для построения более сложных рисунков.

Практика. Упрощение написания кода через использование циклов

Тема 3.4. *Создание графиков и узоров с помощью условий и циклов*

Практика. Написание интерактивного помощника для рисования

Раздел 4. Функции и анонимные функции

Тема 4.1. *Определение и использование функций. Введение в анонимные функции (lambda)*

Теория. Дефиниция функции и её использование в коде. Разница между функцией и lambda функцией.

Практика. Упрощение и написание более красивого и читабельного кода с применением функций.

Тема 4.2. *Создание функций для рисования фигур с параметрами*

Теория. Пример написания часто повторяющейся функции с её последующим вызовом

Практика. Написание собственных функций и их применения для рисования.

Тема 4.3. *Практическое задание на написание функций для различных задач*

Теория. Использование циклов и операторов ветвления внутри функции

Практика. Написание функции с участием типов данных, коллекций, операторов ветвления и циклов.

Промежуточная аттестация.

Раздел 5. Функции высшего порядка

Тема 5.1. *Понятие и использование функций высшего порядка*

Теория. Ключевое отличие от обычных функций и создание собственных

Практика. Применение функций высшего порядка для фильтраций графиков и написание собственных.

Тема 5.2. *Применение в контексте создания сложных рисунков*

Теория. Использование функций высшего порядка.

Практика. Построение сложных рисунков применяя функции высшего порядка.

Тема 5.3. *Задание на использование функций высшего порядка в проектах*

Теория. Постановка проблемы. Инженерные решения. Проектирование модели.

Практика. Самостоятельное проектирование рисунков.

Раздел 6. Объектно-ориентированное программирование

Тема 6.1. *Основы ООП: классы и объекты*

Теория. Знакомство с синтаксисом класса и обращение к его объектам

Практика. Написание собственного класса и вызов объектов.

Тема 6.2. *Инкапсуляция, наследование и полиморфизм.*

Теория. Знакомство с парадигмами программирования и их применением.

Практика. Применение на практике разных парадигм ООП для решение конкретных задач

Тема 6.3. *Создание собственных классов для расширения функциональности Turtle.*

Теория. Использование классов.

Практика. Конструирование и программирование робота. Внутреннее соревнование.

Раздел 7. Проектная работа

Тема 7.1. *Разработка индивидуальных проектов с использованием всех пройденных тем.*

Практика. Разработка и демонстрация индивидуальных проектов

Тема 7.2. *Примеры проектов: создание интерактивных игр или анимации с использованием Turtle*

Практика. Создание анимированных открыток с использованием модуля Turtle

Раздел 8. Введение в Minecraft и библиотеку MCPi

Тема 8.1. *Установка PyCharm, скачивание необходимых файлов, знакомство со средой и её командами.*

Теория. Знакомство с редактором кода PyCharm, использование горячих

клавиш и дополнительные файлы для работы

Практика. Настройка PyCharm.

Тема 8.2. Установка Minecraft и библиотеки MCPi

Теория. Установка необходимых компонентов и модулей.

Практика. Запуск minecraft в PyCharm.

Тема 8.3. Настройка сервера. Основы работы MCPi: подключение к игре, создание объекта игрока.

Теория. Запуск локального сервера

Практика. Подключение своего сервера minecraft

Раздел 9. Основные команды MCPi

Тема 9.1. Использование команд `setBlock()`, `getBlock()`, `getTilePos()`

Теория. Построение блоков в Minecraft при помощи языка Python и получение позиции игрока на карте.

Практика. Использование команд для создания построек.

Тема 9.2. Создание блоков различных типов и манипуляция с ними

Теория. Обзор основных типов блоков в Minecraft и как с их помощью размещать блоки в нужных местах

Практика. Написание скрипта, который строит простую конструкцию из выбранных блоков

Тема 9.3. Примеры использования команд для создания простых объектов

Теория. Объяснение, как с помощью циклов и координат можно создавать линии, квадраты, прямоугольники и другие фигуры.

Практика. Реализовать автоматическую постройку прямоугольника и столба заданной высоты.

Раздел 10. Работы с координатами и позициями

Тема 10.1. Использование `getTilePos()` для получения позиции игрока

Теория. Местонахождение игрока, и использование этих координат для построения объектов или проверки окружения.

Практика. Написать скрипт, который строит блок прямо под игроком, используя его координаты и вывести его координаты в чат.

Тема 10.2. Перемещение игрока с помощью команд `player.setTilePos(x, y, z)` и `player.setPos(x, y, z)`

Теория. Применение: телепортация, автоматизация перемещения, создание мини-игр и испытаний

Практика. Задание: реализовать функцию «домой» — телепортировать игрока в заранее сохранённые координаты

Тема 10.3. Создание системы навигации по миру Minecraft

Теория. Применение: генерация построек, создание защитных стен, платформ, лестниц, автоматизация рутины в игре

Практика. Практическая работа: реализовать построение стены или башни вокруг игрока, используя циклы и вычисления координат

Раздел 11. Создание простых построек с помощью циклов

Тема 11.1. Использование циклов для автоматизации строительства

Теория. Применение циклов `for` и `while` для экономии времени и уменьшения количества ошибок при создании больших или однотипных объектов

Практика. Написать скрипт, который строит линию из 10 блоков вперёд от

позиции игрока, используя цикл `for`, реализовать построение стены высотой 5 и длиной 10 блоков с помощью вложенных циклов.

Тема 11.2. *Создание стен, полов и других конструкций с помощью вложенных циклов.*

Теория. Вложенные циклы позволяют создавать двумерные и трёхмерные конструкции в Minecraft, такие как, платформы, стены, кубы и другие объекты

Практика. Реализовать создание куба размером 4×4×4 блоков из выбранного материала, используя три вложенных цикла

Тема 11.3 *Примеры: автоматизированное строительство.*

Теория. Использование циклов для автоматизированного построения.

Практика. Изменять значения переменных и наблюдать, как меняется результат постройки.

Раздел 12. Работа с блоками и их свойствами

Тема 12.1. *Использование команд `getBlock()` и `setBlock()` для изменения состояния блоков*

Теория. Команда `getBlock(x, y, z)` позволяет узнать, какой блок находится по заданным координатам в мире Minecraft. Команда `setBlock(x, y, z, block_id)` изменяет тип блока по определённым координатам.

Практика. Написать скрипт, который определяет тип блока под игроком с помощью `getBlock()` и выводит его в чат. Использование `setBlock()`, чтобы построить стену, дорожку или заменить один тип блока на другой в выбранной области.

Тема 12.2. *Изучение различных типов блоков и их взаимодействие*

Теория. Рассматриваются основные типы блоков в Minecraft: земля, камень, дерево, вода, лава, стекло и другие и их взаимодействие между собой.

Практика. Задания на определение типа блока по его свойствам (например, отличить твёрдый блок от прозрачного). Эксперименты с размещением различных блоков рядом друг с другом для изучения их взаимодействия.

Тема 12.3. *Создание интерактивных объектов на основе свойств блоков.*

Теория. Примеры программ: замена всех блоков определённого типа на другой в заданной области, построение лабиринтов, автоматическое создание дорог и туннелей.

Практика. Создать автоматическую систему, которая реагирует на тип блока под игроком — например, если под игроком вода, строить мост из дерева.

Раздел 13. Создание игровых механик

Тема 13.1. *Разработка простых игр и квестов внутри Minecraft*

Теория. Рассматривается процесс создания собственных игровых проектов и сценариев в среде Minecraft. Обсуждаются основные этапы проектирования: выбор идеи, постановка целей, определение механики игры или сценария, планирование этапов реализации.

Практика. Задание: придумать и описать идею собственной мини-игры или сценария для Minecraft, определить основные правила и цели. Практическая работа: реализовать базовую механику выбранной игры с помощью Python.

Тема 13.2. *Использование логики условий для создания интерактивных элементов.*

Теория. Создание событий в Minecraft, которые реагируют на действия

игрока или изменения в игровом мире.

Практика. Реализация интерактивной двери, которая открывается только если у игрока есть определенный предмет.

Тема 13.3. *Примеры проектов: автоматизированные фермы, системы управления, умные дома.*

Теория. Обсуждение разных видов проектов, которые можно реализовать в Minecraft: паркур-карты, квесты, мини-игры, элементы «умного дома»

Практика. Выбрать одну из тем (паркур, квест или умный дом) и описать идею своего проекта. Реализация: написать базовый скрипт для ключевого элемента проекта

Раздел 14. Проектная работы в Minecraft

Тема 14.1. *Разработка индивидуальных или групповых проектов с использованием всех изученных тем.*

Теория. Обсуждение выбранных тем, планирование этапов, распределение задач и интеграция разных элементов: автоматизация построек, интерактивные механики, мини-игры и умный дом.

Практика. Придумать и реализовать проект в Minecraft с использованием кода Python - от идеи до итогового решения.

Тема 14.2. *Работа с файлами на Python для сохранения кода в txt документов, а также использование для модификаций.*

Теория. Использование функции open() для работы с файлами и её режимами: "r", "w", "a", "r+", "w+". Знакомство с методами функции: write(), read().

Практика. Создание файла для хранения постоянных данных

Тема 14.3. *Знакомство с модулями и csv-файлами для хранения координат игрока и блоков.*

Теория. Понятие comma separated values – csv, для чего используются эти файлы и в чем их преимущество над txt.

Практика. Создание csv файла и хранения всех координат игрока и блоков.

Тема 14.4. *Защита проектов.*

Практика. Подготовить презентацию своего проекта, выделить ключевые моменты и сложности, с которыми столкнулись в процессе работы.

Демонстрация: показать работу проекта в Minecraft, объяснить, как реализованы основные функции и механики. Обратная связь: получить рекомендации и замечания, обсудить возможные улучшения и доработки.

Итоговое занятие.

Практика. Презентация и защита творческого проекта (Обучающиеся работают над проектами в Minecraft, индивидуально или в составе команды. Тематику выбирают самостоятельно или с помощью наставника. Защита проходит в виде презентации проектов на открытом занятии, конференции, родительском собрании и др. мероприятиях).

1.4 Планируемые результаты

Программа направлена на достижение учащимися следующих **личностных** результатов:

формирование ответственного отношения к учению, готовности и способности учащегося к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

формирование целостного мировоззрения, соответствующего современному уровню общества;

развитие осознанного и ответственного отношения к собственным поступкам в сфере использования информации;

формирование коммуникативной компетентности в различных сферах деятельности.

Метапредметные результаты:

умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;

умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;

смысловое чтение;

владение устной и письменной речью;

умение осознанно использовать речевые средства в соответствии с задачей коммуникации;

формирование и развитие компетентности в области использования информационно-коммуникационных технологий (далее ИКТ-компетенции).

Предметные результаты:

в результате реализации программы обучающиеся будут

знать:

место языка Python среди языков программирования высокого уровня;

что такое операция, операнд и их характеристики;

принципиальные отличия величин, структурированных и неструктурированных;

основные операторы языка Python, их синтаксис;

правила описания функций в Python и построение вызова;

принципиальные отличия между формальными, локальными и глобальными переменными;

область действия описаний в функциях;

свойства данных типа «массив», «матрица»;

уметь:

записывать примеры арифметических и логических выражений всех

атрибутов, которые могут в них входить;

разрабатывать программы обработки числовой и символьной информации;

разрабатывать программы (линейные, разветвляющиеся и с циклами);

воспроизводить алгоритмы сортировки массивов и двумерных

массивов, поиска в упорядоченном массиве, распространять эти алгоритмы на сортировку и поиск в нечисловых массивах;

читать и записывать текстовые файлы в заданном формате.

2 КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

2.1 Календарный учебный график

Учебный год по дополнительной общеобразовательной общеразвивающей программе технической направленности «Программирование в Minecraft на языке Python» начинается 8 сентября 2025 года. Окончание учебного года – 31 мая 2026 года.

Место проведения: ТОГАОУ «Школа №3 – Центр профнавигации и развития карьеры».

Год обучения: 2025 -2026 гг.

Количество учебных недель: 36

Количество учебных часов: 144

2.2 Условия реализации программы

Материально-техническое обеспечение

Для успешной реализации содержания программы необходимо следующее программное и техническое обеспечение:

ноутбук LENOVO IdeaPad S340-15API, 15.6", IPS, AMD Ryzen 5 3500U
2.1ГГц, 12Гб, 512Гб SSD, AMD RadeonVega 8, Windows 10, 81NC009JRU – 12 штук;

интерактивная панель [LMP6501ELRU] Lumien 65" 3840 x 2160 @ 60 Hz,
инфракрасный тачскрин 20 касаний, яркость 450cd/m2, контрастность 1200:1,
матовое покрытие, память 3GB DDR4 + 32GB, Android 8.0, колонки 2x15 Вт, пульт
ДУ, 2 стилуса – 1 штука;

комплект программного обеспечения "Мой Офис ОБРАЗОВАНИЕ
JetBrainsPyCharm (Python)" – 1 штука;

роутер ASUS RT-AC66U rev B1 // роутер 802.11b/g/n/ac, до 450 + 1300Мбит/с,
2,4 + 5 гГц, 3 антенны, USB, GBT LAN ; 90IG0300-BM3100 – 1 штука;

стол – 12 штук;

стул – 12 штук.

Методическое обеспечение

Методы и формы обучения по программе определяются требованиями федерального государственного образовательного стандарта нового поколения, учетом возрастных и индивидуальных способностей обучающихся, дистанционным характером обучения.

Основные приоритеты методики преподавания по данной программе:

междисциплинарная интеграция, содействующая становлению целостного мировоззрения;

интерактивность;

лично-деятельностный подход в обучении;

вариативное образование, предполагающее построение индивидуальных

траекторий обучения и вариативное изменение образовательных моделей, что делает образовательный процесс более гибким и способным удовлетворять разнообразные образовательные потребности личности;

субъект-субъектное педагогическое взаимодействие обучающихся и педагогов по достижению совместных целей.

Основные технологии, формы и методы обучения

Образовательный процесс строится по двум основным видам деятельности:

обучение детей теоретическим знаниям (вербальная информация, излагаемая педагогом на основе современных педагогических технологий);

самостоятельная и практическая работа обучающихся (изучение основ программирования, выполнение практических заданий, создание собственных проектов и т.д.).

В программе реализуются теоретические и практические блоки, что позволяет наиболее полно охватить и реализовать потребности обучающихся, сформировать практические навыки в области программирования. В ходе выполнения самостоятельных работ, учащиеся приобретают навыки работы с различными ресурсами, используемыми для создания собственных проектов, на основе чего происходит выбор оптимальных средств для представления информации в сети Интернет. Таким образом, данная программа позволяет развить у обучающихся творческий склад мышления, способности к самостоятельному поиску, решению поставленных проблем, и создать условия для творческого самовыражения личности, что в полной мере соответствует тем требованиям, которые обозначены во ФГОС нового поколения.

Программа имеет разноуровневый характер и рассчитана на обучающихся с разным уровнем подготовки. Учебный материал распределен по принципу последовательного расширения и углубления теоретических знаний, приобретения практических умений и навыков.

Кадровое обеспечение программы

Педагоги, организующие образовательный процесс по данной программе должны иметь высшее техническое образование или пройти подготовку на курсах повышения квалификации по применению информационно-коммуникационных технологий. Важным условием, необходимым для реализации программы, является умение педагога осуществлять личностно-деятельностный подход к организации обучения, проектировать индивидуальную образовательную траекторию учащегося, разрабатывать и эффективно применять инновационные образовательные технологии.

2.3 Форма аттестации

Система отслеживания, контроля и оценки результатов процесса обучения по данной программе имеет три основных элемента:

Стартовая диагностика. При приеме детей в объединение педагог проводит тестирование уровня развития мотивации ребенка к обучению для дальнейшего определения образовательного маршрута. Результаты тестирования фиксируются в сводных таблицах.

Текущий контроль предусматривает: тестирование, опросы, соревнования, педагогическое наблюдение, взаимооценка обучающимися работ друг друга, практическое задание, творческая работа, контрольное занятие, зачет, олимпиада, презентация творческих работ.

Уровень освоения программы отслеживается также с помощью выполнения заданий по разработке различных элементов. Задания подбираются в соответствии с пройденным материалом.

Промежуточный контроль, итоговая диагностика. Основной формой подведения итогов является подготовка и защита творческих проектов.

Время проведения	Цель проведения	Формы контроля
<i>Стартовая диагностика</i>		
В начале учебного года	Определение уровня развития детей, их способностей	Тестирование, анкетирование
<i>Текущий контроль</i>		
В течение всего учебного года	<ul style="list-style-type: none"> ➤ Определение степени усвоения обучающимися учебного материала. ➤ Определение готовности детей к восприятию нового материала. ➤ Повышение ответственности и заинтересованности обучающихся в обучении. ➤ Выявление отстающих и опережающих обучение. ➤ Подбор наиболее эффективных методов и средств обучения. 	Тестирование, опрос, соревнование, педагогическое наблюдение, взаимооценки обучающимися работ друг друга, зачет
<i>Промежуточный контроль</i>		
По окончании изучения темы или раздела. В конце месяца, полугодия.	<ul style="list-style-type: none"> ➤ Определение степени усвоения обучающимися учебного материала. ➤ Определение результатов обучения. 	Творческая работа, опрос, контрольное занятие, олимпиада (решение задач повышенной сложности)
<i>Итоговая диагностика</i>		
В конце учебного года или курса обучения	<ul style="list-style-type: none"> ➤ Определение изменения уровня развития детей, их творческих способностей. ➤ Определение результатов обучения. ➤ Ориентирование обучающихся на дальнейшее обучение. ➤ Получение сведений для совершенствования образовательной программы и методов обучения. 	Подготовка к защите проектов, защита проектов.

Формы отслеживания и контроля развивающих и воспитательных результатов:

оценка устойчивости интереса обучающихся к занятиям с помощью наблюдения педагога и самооценки обучающихся;

оценка устойчивости интереса обучающихся к участию в мероприятиях, направленных на формирование и развитие общекультурных компетенций с помощью наблюдения педагога и самооценки обучающихся;

статистический учет сохранности контингента обучающихся;

сравнительный анализ успешности выполнения заданий обучающимися на начальном и последующих этапах освоения программы;

анализ творческих и проектных работ обучающихся;

создание банка индивидуальных достижений обучающихся;

оценка степени участия и активности обучающегося в командных проектах, соревновательной и конкурсной деятельности;

оценка динамики показателей развития познавательных способностей обучающихся (внимания, памяти, изобретательности, логического и пространственного мышления и т.д.) с помощью наблюдения педагога и самооценки обучающихся;

наблюдение и фиксирование изменений в личности и поведении обучающихся с момента поступления в объединение и по мере их участия в деятельности;

индивидуальные и коллективные беседы с обучающимися.

Критериями выполнения программы служат:

знания, умения и навыки обучающихся, позволяющие им комплексно использовать информационные технологии для получения необходимой информации и создания собственных проектов, стабильный интерес к изучению информационно-коммуникационных технологий и их использования в различных сферах деятельности.

2.4 Оценочные материалы

№ п/п	Оцениваемые параметры	Критерии	Методы диагностики
Теоретическая подготовка обучающихся			
1	Теоретические знания по основным разделам учебного плана программы	Соответствие теоретических знаний программным требованиям	Тест-опрос
Практическая работа обучающихся			
3	Практические умения и навыки, знания по основным разделам учебного плана программы	Соответствие практических умений и навыков программным требованиям	Опрос, педагогическое наблюдение

4	Владение специальным программным обеспечением	Отсутствие затруднений при работе с ПО	Анализ информации
5	Творческие навыки	Сформированный интерес к избранному виду деятельности	Индивидуальный проект

Диагностика развития теоретических знаний и практических навыков осуществляется с помощью диагностических контрольных заданий по следующим критериям:

Оценка результатов.

По итогам составляется таблица отслеживания образовательных результатов, в которой обучающиеся по каждой теме выходят на следующие уровни шкалы оценки:

1. Высокий результат – полное освоение содержания;
2. Средний – базовый уровень;
3. Низкий – освоение материала на минимально допустимом уровне

2.5 Методическое обеспечение программы

Педагогические технологии

В процессе обучения по программе, используются разнообразные педагогические технологии:

технологии развивающего обучения, направленные на общее целостное развитие личности, на основе активно-деятельного способа обучения, учитывающие закономерности развития и особенности индивидуума;

технологии личностно-ориентированного обучения, направленные на развитие индивидуальных познавательных способностей каждого обучающегося, максимальное выявление, раскрытие и использование его опыта;

технологии дифференцированного обучения, обеспечивающие обучение каждого обучающегося на уровне его возможностей и способностей;

технологии сотрудничества, реализующие демократизм, равенство, партнерство в отношениях педагога и обучающегося, совместно вырабатывают цели, содержание, дают оценки, находясь в состоянии сотрудничества, сотворчества;

проектные технологии – достижение цели через детальную разработку проблемы, которая должна завершиться реальным, осязаемым практическим результатом, оформленным тем или иным образом;

компьютерные технологии, формирующие умение работать с информацией, исследовательские умения, коммуникативные способности.

В практике выступают различные комбинации этих технологий, их элементов.

Формы организации учебных занятий:

- индивидуальное задание;
- комбинированное занятие;
- лекция;

- беседа;
- олимпиада;
- практическое занятие;
- самостоятельная работа;
- мастер-класс;
- итоговое занятие.

Методы образовательной деятельности

В программе кроме традиционных методов используются:

эвристический метод;

исследовательский метод обучения, дающий обучающимся возможность проявить себя, показать свои возможности, добиться определенных результатов;

метод проблемного изложения материала, когда перед обучающимся ставится некая задача, позволяющая решить определенный этап процесса обучения и перейти на новую ступень обучения;

самостоятельная работа; диалог и дискуссия;

приемы дифференцированного обучения, обеспечивающие обучение каждого обучающегося на уровне его возможностей и способностей.

Еще одним основным методом организации учебной деятельности по программе является метод кейсов.

Кейс – описание проблемной ситуации понятной и близкой обучающимся, решение которой требует всестороннего изучения, поиска дополнительной информации и моделирования ситуации или объекта, с выбором наиболее подходящего.

Преимущества метода кейсов:

практическая направленность. Кейс-метод позволяет применить теоретические знания к решению практических задач;

интерактивный формат. Кейс-метод обеспечивает более эффективное усвоение материала за счет высокой эмоциональной вовлеченности и активного участия обучаемых. Участники погружаются в ситуацию с головой: у кейса есть главный герой, на место которого ставит себя команда и решает проблему от его лица. Акцент при обучении делается не на овладение готовым знанием, а на его выработку;

конкретные навыки. Кейс-метод позволяет совершенствовать универсальные навыки, которые оказываются крайне необходимы в реальном рабочем процессе.

Условно можно выделить следующие **виды кейсов**:

инженерно-практический;

инженерно-социальный;

инженерно-технический;

исследовательский (практический или теоретический).

Построение занятия включает в себя фронтальную, индивидуальную и групповую работу, а также некоторый соревновательный элемент. Программой предусмотрено проведение комбинированных занятий: занятия состоят из теоретической и практической частей, причём большее количество времени занимает именно практическая часть.

Использование дистанционных образовательных технологий при работе с обучающимися

Главным принципом обучения является принцип доступности, который на практике успешно реализуется в ходе использования дистанционных образовательных технологий и электронного обучения. Такой подход позволяет выстроить индивидуальную образовательную траекторию обучающегося, учесть его образовательные потребности, максимально визуализировать процесс обучения.

В режиме дистанционного обучения возможны следующие *виды работы*:

работа через программу Сферум. Такой урок максимально приближен к обычному уроку, поскольку позволяет общаться с учеником в режиме реального времени (выслушать ответ, оценить ученика, построить диалог);

с использованием сервисов, построенных на основе чат-технологий, где дети имеют возможность обмениваться мнениями, вести переписку, участвовать в обсуждении проблемы при выполнении, например, проекта. Чат-технологии полезны для организации групповых форм работы, рассчитанных на длительный период.

Таким образом, применение в практике обучения дистанционных образовательных технологий способствует расширению образовательных возможностей детей, оптимизирует процесс обучения.

Также при организации процесса обучения с использованием дистанционных образовательных технологий необходимо уделять много внимания использованию здоровьесберегающих технологий или их элементов (проведение физкультминуток, гимнастики для глаз и т.д.).

2.6 Воспитательный компонент программы

Реализация программы невозможна без осуществления воспитательной работы с обучающимися. Воспитательная работа ведётся на протяжении всего учебного процесса.

Приоритетные направления в организации воспитательной работы:

воспитание нравственных качеств (трудолюбия, настойчивости, целеустремленности) происходит непосредственно в процессе обучения во время совместной деятельности;

духовно-нравственное воспитание формирует ценностные представления о морали, об основных понятиях этики (добро и зло, истина и ложь, смысл жизни, справедливость, милосердие, проблеме нравственного выбора, достоинство, любовь и др.), о духовных ценностях народов России, об уважительном отношении к традициям, культуре и языку своего народа и др. народов России;

трудовое и профориентационное воспитание формирует знания, представления о трудовой деятельности; выявляет творческие способности и профессиональные направления обучающихся;

воспитание познавательных интересов формирует потребность в приобретении новых знаний, интерес к творческой деятельности;
экологическое воспитание формирует ценностные представления и отношение к окружающему миру.

Основные задачи воспитательной работы:

формирование мировоззрения и системы базовых ценностей личности;
организация инновационной работы в области воспитания и дополнительного образования;
организационно-правовые меры по развитию воспитания и дополнительного образования обучающихся;
приобщение обучающихся к общечеловеческим нормам морали, национальным устоям и традициям образовательного учреждения;
обеспечение развития личности и её социально-психологической поддержки, формирование личностных качеств, необходимых для жизни;
воспитание внутренней потребности личности в здоровом образе жизни, ответственного отношения к природной и социокультурной среде обитания;
развитие воспитательного потенциала семьи;
поддержка социальных инициатив и достижений обучающихся.

Основные воспитательные мероприятия:

просмотр обучающимися тематических материалов и их обсуждение;
тематические диспуты и беседы;
участие в конкурсах, соревнованиях, олимпиадах различного уровня.

Работа с коллективом обучающихся:

формирование практических умений по организации органов самоуправления этике и психологии общения, технологии социального и творческого проектирования;
обучение умениям и навыкам организаторской деятельности, самоорганизации, формированию ответственности за себя и других;
развитие творческого культурного, коммуникативного потенциала ребят в процессе участия в совместной общественно – полезной деятельности;
содействие формированию активной гражданской позиции;
воспитание сознательного отношения к труду, к природе, к своему городу.

Работа с родителями:

организация системы индивидуальной и коллективной работы с родителями (тематические беседы, собрания, индивидуальные консультации);
содействие сплочению родительского коллектива и вовлечение в жизнедеятельность кружкового объединения (организация и проведение открытых занятий для родителей в течение года);
оформление информационных уголков для родителей по вопросам воспитания обучающихся.

Успешная работа детского объединения во многом зависит от степени участия в ней родителей обучающихся. В большинстве родители заинтересованно относятся к занятиям своих детей в объединении, радуются их успехам и достижениям.

Работа с родителями включает в себя следующие формы деятельности:

родительские собрания;
консультации;
беседы;
работа с семьями, находящимися в трудной жизненной ситуации;
совместные праздники обучающихся и их родителей;
привлечение родителей к подготовке и проведению мероприятий;
приглашение родителей на мероприятия объединения и всего учреждения.

Такая работа способствует формированию общности интересов обучающихся и их родителей, служит развитию эмоциональной и духовной близости.

Результат воспитания

В процессе воспитания происходят изменения в личностном развитии обучающихся, в процессе общения со своими сверстниками по достижению общих целей, у ребят формируются такие качества как взаимопомощь, самостоятельность, ответственность за порученное дело. Несомненно, большую роль в воспитании моральных качеств, обучающихся играет личный пример педагога.

СПИСОК ЛИТЕРАТУРЫ

1. Бизли Д.М., Г. Ван Россум. Язык программирования Python. Справочник. (пер. с англ.) Киев: ДиаСофт., 2000.
2. Вордерман Кэрол. Программирование для детей. 2016. – 357 с.
3. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.: ил
4. Информатика: Учебник для 10-11 класса / Н.Д. Угринович - М.: Бином, 2009. - 512 с.
5. Лутц М. Программирование на Python. (пер. с англ.) СПб.: Символ-Плюс., 2002.
6. Россум, Г. и др. Язык программирования Python. 2001. 454 с.
7. Сузи Р.А. Python. Наиболее полное руководство. СПб.: БХВ- Петербург., 2002.
8. Сузи Р.А. Язык программирования Python. М: Бином. Лаборатория знаний. - 2006.
9. Фридланд А.Я., Ханамирова Л.С., Фридланд И.А. Информатика и компьютерные технологии. Основные термины. Толковый словарь. М..Издательство Астрель., 2003.
10. Хахаев И.А. Практикум по алгоритмизации и программированию на Python: / И.А. Хахаев - М.: Альт Линукс, 2010. - 126 с.
11. Чаплыгин А. Н. Учимся программировать вместе с Питоном. Revision: 226.

Приложение

Календарно-тематический учебный график на 2025 – 2026 учебный год Место проведения занятий: Школа № 3 Центр профнавигации и развития карьеры - Центр цифрового образования детей «IT-Куб»

№п/п	Планируемая дата	Фактическая дата	Наименование темы, раздела	Кол-во часов	Место проведения	Форма занятия	Форма контроля
			Вводное занятие	2	Аудитория	Групповая	Стартовая диагностика
			Раздел 1. Введение в Python и модуль turtle				
1.1.			Установка Python и настройка среды разработки	2	Аудитория	Групповая	Лекция, творческая работа
1.2.			Основы работы с модулем Turtle: импорт, создание объектов черепахи	2	Аудитория	Групповая	Лекция, творческая работа
1.3.			Примеры простых команд: forward(), left(), right(), pendown()	2	Аудитория	Групповая	Лекция, творческая работа
			Раздел 2. Типы данных и структуры				
2.1			Обзор базовых типов данных: int, float, string, bool.	2	Аудитория	Групповая	Лекция, творческая работа
2.2.			Работа со структурами данных: списки, кортежи, множества, словари.	2	Аудитория	Групповая	Лекция, творческая работа
2.3.			Применение структур данных для хранения координат и параметров фигур в Turtle	4	Аудитория	Групповая	Лекция, творческая работа
2.4			Практические задания по созданию и манипуляции данными.	4	Аудитория	Групповая	Лекция, творческая работа
			Раздел 3. Условные операторы и циклы				
3.1			Использование операторов ветвления if, elif, else	4	Аудитория	Групповая	Лекция, творческая работа
3.2			Введение в циклы for и while.	2	Аудитория	Групповая	Лекция, творческая работа

3.3		Применение циклов для рисования фигур с использованием Turtle	2	Аудитория	Групповая	Лекция, творческая работа
3.4		Создание графиков и узоров с помощью условий и циклов	4	Аудитория	Групповая	Лекция, творческая работа
		Раздел 4. Функции и анонимные функции				
4.1		Определение и использование функций. Введение в анонимные функции (lambda)	2	Аудитория	Групповая	Лекция, творческая работа
4.2		Создание функций для рисования фигур с параметрами	2	Аудитория	Групповая	Лекция, творческая работа
4.3		Практическое задание на написание функций для различных задач	2	Аудитория	Групповая	Лекция, творческая работа
		Раздел 5. Функции высшего порядка		Аудитория	Групповая	Лекция, творческая работа
5.1		Понятие и использование функций высшего порядка	2	Аудитория	Групповая	Лекция, творческая работа
5.2		Применение в контексте создания сложных рисунков в Turtle	2	Аудитория	Групповая	Лекция, творческая работа
5.3		Задание на использование функций высшего порядка в проектах	2	Аудитория	Групповая	Лекция, творческая работа
		Раздел 6. Объектно-ориентированное программирование (ООП)				
6.1		Основы ООП: классы и объекты	4	Аудитория	Групповая	Лекция, творческая работа
6.2		Инкапсуляция, наследование и полиморфизм	4	Аудитория	Групповая	Лекция, творческая работа
6.3		Создание собственных классов для расширения функциональности Turtle	4	Аудитория	Групповая	Лекция, творческая работа
		Раздел 7. Проектная работа				
7.1		Разработка индивидуальных проектов с использованием всех пройденных тем	2	Аудитория	Групповая	Лекция, творческая работа
7.2		Примеры проектов: создание интерактивных игр или анимации с использованием Turtle	2	Аудитория	Групповая	Лекция, творческая работа
		Раздел 8. Введение в Minecraft и библиотеку MCPi				

8.1		Установка PyCharm, скачивание необходимых файлов, знакомство со средой и её командами	2	Аудитория	Групповая	Лекция, творческая работа
8.2		Установка Minecraft и библиотеки MCPi	2	Аудитория	Групповая	Лекция, творческая работа
8.3		Настройка сервера. Основы работы с MCPi: подключение к игре, создание объекта игрока	2	Аудитория	Групповая	Лекция, творческая работа
		Раздел 9. Основные команды MCPi				
9.1		Использование команд setBlock(), getBlock(), getTilePos()	4	Аудитория	Групповая	Лекция, творческая работа
9.2		Создание блоков различных типов и манипуляция с ними	4	Аудитория	Групповая	Лекция, творческая работа
9.3		Примеры использования команд для создания простых объектов	4	Аудитория	Групповая	Лекция, творческая работа
		Раздел 10. Работы с координатами и позициями		Аудитория	Групповая	Лекция, творческая работа
10.1		Простой цикл for. Функция range()	4	Аудитория	Групповая	Лекция, творческая работа
10.2		Эксперименты с функции range(). Проект: «Постройка колонн и пирамид»	4	Аудитория	Групповая	Лекция, творческая работа
10.3		Выход из цикла с помощью break. Проект: «Алмазоискатель»	4	Аудитория	Групповая	Лекция, творческая работа

		Раздел 11. Создание простых построек с помощью циклов				
11.1		Использование циклов для автоматизации строительства	4	Аудитория	Групповая	Лекция, творческая работа
11.2		Установка Minecraft и библиотеки MCPi	4	Аудитория	Групповая	Лекция, творческая работа
11.3		Настройка сервера. Основы работы с MCPi: подключение к игре, создание объекта игрока	4	Аудитория	Групповая	Лекция, творческая работа
		Раздел 12. Работа с блоками и их свойствами				
12.1		Использование команд getBlock() и setBlock() для изменение состояние блоков	4	Аудитория	Групповая	Лекция, творческая работа

12.2		Изучение различных типов блоков и их взаимодействие	4	Аудитория	Групповая	Лекция, творческая работа
12.3		Создание интерактивных объектов на основе свойств блоков	4	Аудитория	Групповая	Лекция, творческая работа
		Раздел 13. Создание игровых механик				
13.1		Разработка простых игр и квестов внутри Minecraft	4	Аудитория	Групповая	Лекция, творческая работа
13.2		Использование логики условий для создания интерактивных элементов	4	Аудитория	Групповая	Лекция, творческая работа
13.3		Примеры проектов: автоматизированные фермы, системы управления, умные дома	4	Аудитория	Групповая	Лекция, творческая работа
		Раздел 14. Проектная работа в Minecraft				
14.1		Разработка индивидуальных или групповых проектов с использованием всех изучения тем	4	Аудитория	Групповая	Лекция, творческая работа
14.2		Работа с файлами на Python для сохранения кода в txt документов, а также использование для модификаций	4	Аудитория	Групповая	Лекция, творческая работа
14.3		Знакомство с модулями csv файлами для хранения координат игрока и блоков	4	Аудитория	Групповая	Лекция, творческая работа
14.4		Защита проектов				
		Итоговое занятие	2			