

Министерство образования и науки Нижегородской области
Государственное бюджетное образовательное учреждение высшего образования
«Нижегородский государственный инженерно-экономический университет»

Центр цифрового образования детей IT-куб

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«Программирование на языке Python в Minecraft»

Направленность: техническая

Уровень: стартовый

Возраст обучающихся: 7-12 лет

Срок реализации: 1 год

Авторы-составители:

Николаева Е.А., заместитель начальника по учебной части

Лапаев А.С., педагог дополнительного образования

Сивожелезова В.В., методист

Княгинино

2022

I. Комплекс основных характеристик программы

1. Пояснительная записка

Программа «Программирование на языке Python в Minecraft» построена таким образом, чтобы заинтересовать учащихся программированием и найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни при работе с большим объемом информации, научиться общаться с компьютером, который ничего не умеет делать, если не умеет человек.

Программа «Программирование на языке Python в Minecraft» разработана на основе педагогического опыта работы составителей и нормативно-правовых документов:

- Федеральный Закон «Об образовании в Российской Федерации» (№273-ФЗ от 29.12.2012);

- Приказ Министерство образования и науки Российской Федерации №196 от 09.11.2018 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

- «Санитарно-эпидемиологические правила и нормативы СанПиН 2.4.4.3172-14» (утв. Главным государственным санитарным врачом РФ 4 июля 2014 г. N 41);

Программа «Программирование на языке Python в Minecraft» **имеет техническую направленность**, ориентированную на развитие у учащихся технических и научных способностей.

Актуальность программы. Актуальность программы состоит в том, что среда Minecraft позволяет сформировать у учащихся интерес к программированию, а язык программирования Python отвечает всем современным требованиям объектно-ориентированного программирования. Связь игры и языка программирования позволяет сформировать навыки программирования, раскрыть технологию программирования. Изучение Python значительно облегчает переход к изучению других языков программирования. Преимуществом такой связи: игры и языка программирования, наличие версий различных операционных систем, к тому же программы являются свободно распространяемыми, что немало важно для образовательных учреждений.

Отличительная особенность дополнительной общеразвивающей программы «Программирование на языке Python в Minecraft» в том, что она является практико-ориентированной. Освоение навыков программирования происходит в процессе практической работы, с применением игровых технологий. Это позволяет обучающимся получать не только теоретические знания, но и развивать логическое и творческое мышление, навыки решения задач, которые пригодятся в разных областях деятельности.

Программа состоит из пяти разделов: «Основы языка программирования», «Телепортация с помощью переменных», «Математика,

моментальное строительство и супер-прыжки», «Общаемся с помощью строк», ««Истина» и «Ложь» булевых значений». Модули различаются по сложности и реализуются по принципу «от простого к сложному».

Программа «Программирование на языке Python в Minecraft» реализуется на стартовом уровне. Данный уровень позволяет подготовить детей к изучению программирования и сформировать мотивацию к изучению языков программирования.

Адресат программы. Возраст учащихся, участвующих в реализации данной программы - от 7 до 12 лет. Количество обучающихся в группе – 8-12 человек.

Режим занятий. Занятия проводятся два раза в неделю по 2 занятия. Продолжительность одного занятия составляет 2 академических часа, продолжительность одного академического часа для детей младше 10 лет составляет 30 минут. После одного занятия организовывается перерыв 10 мин.

Срок освоения общеразвивающей программы – 1 учебный год, объем программы - 140 часов.

Формы обучения и виды занятий. Форма обучения очная с возможностью применения дистанционных технологий.

Основной вид занятий – комбинированный, включающий в себя теоретическую и практическую части. Задания курса выполняются с помощью персонального компьютера и необходимого программного обеспечения. Задания являются обязательными для выполнения обучающимися. При выполнении заданий обучающиеся формируют теоретические и практические знания, а также приобретают навыки использования компьютерной техники.

В ходе обучения предполагается использование здоровьесберегающих технологий, таких как:

- динамические паузы;
- гимнастика для глаз;
- соблюдение обучающимися правил работы на ПК.

В процессе обучения проводится промежуточная аттестация по модулям программы для определения усвоения знаний обучающимися и позволяет отследить процесс закрепления знаний.

Педагогическая целесообразность. Программа «Программирование на языке Python в Minecraft» реализуется с применением игровой технологии. Изучение языка программирования посредством любимой игры позволяет, как качественно сформировать предметные навыки (работа с языком программирования), так и поддерживать на высоком уровне познавательный интерес обучающихся, готовность к творческой деятельности.

Педагогическая целесообразность также связана с реализацией следующих возможностей для развития ребёнка:

- создание максимального количества ситуаций успеха;
- возможность длительного влияния на формирование личности обучающегося, выявление и стимулирование проявлений положительных личностных качеств ребенка, для постижения самооценности собственной личности;

- практическая значимость (расширение кругозора, использование приобретаемых качеств, знаний в повседневной жизни), предоставление обучающемуся широких возможностей для самовыражения средствами программирования.

1.2. Цель и задачи общеразвивающей программы

Цель программы – обучение программированию на языке python через создание творческих проектов в игре Minecraft.

Для успешной реализации программы необходимо решить ряд задач:

Задачи:

1. Сформировать навыки разработки, тестирования и отладки несложных программ;
2. Овладеть понятиям основ языка программирования.
3. Сформировать представление о профессии «программист»;
4. Развивать способности коммуникации в работе и во взаимодействии со сверстниками;
5. Развивать умение работать с компьютерными программами и дополнительными источниками информации.

2. Содержание общеразвивающей программы

2.1. Учебный план

№ п/п	Наименование раздела, темы	Количество часов			Формы контроля
		всего	теор ет.	практ.	
Раздел 1. Основы языка программирования		28	8	20	
1	Вводное занятие	10	4	6	Устный опрос
2	Основные операторы	12	2	10	Конкурс
3	Библиотеки	6	2	4	Творческая работа
Раздел 2. Телепортация с помощью переменных		28	8	20	
1	Основные моменты	12	2	10	Устный опрос
2	Телепортация игрока	6	2	4	Устный опрос
3	Перемещение в точности туда, куда надо	6	2	4	Фронтальный опрос
4	Телепортационный тур	4	2	2	Творческая работа
Раздел 3. Математика, моментальное строительство и супер-прыжки		28	12	16	
1	Основные моменты	6	2	4	Устный опрос
2	Башенка из блоков	4	2	2	Беседа
3	Супер-прыжок	4	2	2	Устный опрос
4	Изменение блока под ногами	4	2	2	Фронтальный опрос
5	Быстрое строительство	4	2	2	Устный опрос
6	Супер-прыжок в неизвестность	6	2	4	Творческая работа
Раздел 4. Общаемся с помощью строк		28	9	19	
1	Основные моменты	10	2	8	Устный опрос
2	Привет, мир Minecraft	4	2	2	Беседа
3	Отправьте в чат сообщение	8	2	6	Беседа
4	Добавьте перед сообщениями имена	2	1	1	Фронтальный опрос
5	Позвольте пользователю выбрать тип блока	4	2	2	Творческая работа
Раздел 5. «Истина» и «Ложь» булевых значений		28	8	20	
1	Основные моменты	16	4	12	Устный опрос
2	Оставить разрушение блоков	6	2	4	Фронтальный опрос
3	Игрок в воде?	4	2	2	Фронтальный опрос
4	Итоговое занятие	2	-	2	Тестирование
	ИТОГО:	140	45	95	

2.2. Содержание учебного плана

Модуль 1. Основы языка программирования

Тема 1. Вводное занятие

Теория: что такое язык программирования, редактор кода, как с этим работать.

Практика: решение задачи

Тема 2. Основные операторы

Теория: Команды `print()`, `input()`, условный оператор, циклы. Их назначение, использование.

Практика: Составление игр.

Тема 3. Библиотеки.

Теория: Подключение библиотек.

Практика: подключить к игре Minecraft стороннюю библиотеку

Модуль 2. Телепортация с помощью переменных

Тема 4. Основные моменты.

Теория: изучение вопросов: что такое программа, хранение информации в переменных, синтаксис для переменных, изменение значений применений, целые числа.

Практика: применение полученных знаний в игре Minecraft.

Тема 5. Телепортация игрока

Теория: Использование переменных и работа с целыми числами. Знакомство с вещественными числами.

Практика: Создание программы телепортации игрока.

Тема 6. Перемещение в точности туда, куда надо

Теория: Изучение телепортации игрока в место с вещественными значениями координат

Практика: Изменение кода с прошлой миссии в игре Minecraft.

Тема 7. Телепортационный тур

Теория: Изучение перемещения в разные области Minecraft.

Практика: применение теории в игре Minecraft.

Модуль 3. Математика, моментальное строительство и супер-прыжки

Тема 8. Основные моменты

Теория: Изучение, что такое выражения и команды, операции.

Практика: Решение задач.

Тема 9. Башенка из блоков

Теория: Изучение создания блочных построек.

Практика: Создание блоков и помещение их в заданное место.

Тема 10. Супер-прыжок

Теория: Операции сложения, вычитания.

Практика: Изменение позиции игрока с помощью операций сложения и вычитания.

Тема 11. Изменение блока под ногами

Теория: Функции `getTilePos()` и `setBlock()`.

Практика: Изменение блока под ногами игрока на блок «текущая лава», используя функции `getTilePos()` и `setBlock()`.

Тема 12. Быстрое строительство

Теория: Изучение мгновенной постройки зданий.

Практика: применение теории в игре Minecraft.

Тема 13. Супер-прыжок в неизвестность

Теория: Изменение координат на случайные.

Практика: Перемещение игрока, изменяя его координаты.

Модуль 4. Общаемся с помощью строк.

Тема 14. Основные моменты

Теория: Функции `print()`. Функция `input()`.

Практика: Решение задач.

Тема 15. Привет, мир Minecraft

Теория: Функции `postToChat()`.

Практика: Отправление сообщения в чат с помощью функции `postToChat()`.

Тема 16. Отправьте в чат сообщение

Теория: Сохранение сообщения в переменной `message`. Склейка строк. Преобразование числа в строку.

Практика: применение теории в игре Minecraft.

Тема 17. Добавьте перед сообщениями имена

Теория: Преобразование строки в целое число.

Практика: Указание в начале сообщения имя пользователя в игре Minecraft.

Тема 18. Позвольте пользователю выбрать тип блока

Теория: Функции `setBlock()` и `setBlocks()`.

Практика: Создание любых типов блоков всех типов, с помощью функций `setBlock()` и `setBlocks()`.

Модуль 5. «Истина» и «Ложь» булевых значений

Тема 19. Основные моменты

Теория: Булевы значения: основы.

Практика: Решение задач.

Тема 20. Оставить разрушение блоков

Теория: Неуязвимые блоки. Склейка строк и булевых значений.

Практика: Создание неуязвимых блоков. Решение задач.

Тема 21. Игрок в воде?

Теория: Операции сравнения.

Практика: Определение, находится ли игрок в воде, с помощью операций сравнения.

Тема 22. Итоговое занятие

Практика: Проверка и оценка полученных знаний, умений и навыков.

1.4. Планируемые результаты

Предметные:

1. Владение понятиями: переменная, условный оператор, цикл, строки.
2. Владение навыками разработки, тестирования и отладки несложных программ на языке python с помощью Minecraft.
3. Знание алгоритмов разработки.
4. Владение основами языка программирования python.

Личностные:

1. Сформировано представление о профессии программиста.
2. Сформированы навыки аргументации своего мнения.
3. Сформированы навыки нахождения единого решения в малой группе.
4. Сформировано ответственное отношение к формированию и решению новых задач и достижению поставленных целей.

Метапредметные:

1. Развитие алгоритмического и логического мышления.
2. Развитие творческого мышления
3. Способность к коммуникации и взаимодействию со сверстниками.
4. Умение работать с компьютерными программами и дополнительными источниками информации в соответствии с возрастом.

II. Комплекс организационно-педагогических условий реализации общеразвивающей программы

2.1 Календарный учебный график на 2022-2023 учебный год

№ п/п	Основные характеристики образовательного процесса	Кол-во/ сроки
1.	Количество учебных недель	35
2.	Количество часов в неделю	4
3.	Количество часов	140
4.	Недель в I полугодии	17
5.	Недель в II полугодии	18
6.	Начало учебного года	5 сентября
7.	Окончание учебного года	4 июня

2.2 Условия реализации общеразвивающей программы

Материально-техническое обеспечение

Требования к помещению:

- помещение для занятий, отвечающие требованиям СанПин для учреждений дополнительного образования;
- качественное освещение;
- столы, стулья по количеству обучающихся и 1 рабочим местом для педагога;
- шкаф для оборудования.

Оборудование:

1. Компьютер в сборе тип 1 PowerCool PC52ITS-0432 – 13 шт.
2. Моноблочное интерактивное устройство 65W21K – 1 шт.
3. Маркерная доска

Кадровое обеспечение

Реализовывать программу могут педагоги дополнительного образования, обладающие достаточными знаниями в области педагогики и программирования на языке Python.

2.3 Формы аттестации и оценочные материалы

Система контроля знаний и умений учащихся представляется в виде учёта результатов по итогам выполнения заданий отдельных кейсов и посредством наблюдения, отслеживания динамики развития учащегося.

Система отслеживания результатов обучающихся выстроена следующим образом:

- промежуточный контроль по первому модулю в форме творческой работы, оценивается степень выполнения задания. Создано менее половины игры – низкий уровень освоения; Игра создана, но не выполнено дополнительное задание – средний уровень освоения; Игра создана, выполнено дополнительное задание – высокий уровень освоения.

- промежуточный контроль по второму модулю в форме творческой работы, оценивается степень выполнения задания. Создано менее половины игры – низкий уровень освоения; Игра создана, но не выполнено дополнительное задание – средний уровень освоения; Игра создана, выполнено дополнительное задание – высокий уровень освоения.

- промежуточный контроль по третьему модулю в форме творческой работы, оценивается степень выполнения задания. Создано менее половины игры – низкий уровень освоения; Игра создана, но не выполнено дополнительное задание – средний уровень освоения; Игра создана, выполнено дополнительное задание – высокий уровень освоения.

- промежуточный контроль по четвертому модулю в форме творческой работы, оценивается степень выполнения задания. Создано менее половины игры – низкий уровень освоения; Игра создана, но не выполнено дополнительное задание – средний уровень освоения; Игра создана, выполнено дополнительное задание – высокий уровень освоения.

- промежуточный контроль по пятому модулю в форме творческой работы, оценивается степень выполнения задания. Создано менее половины игры – низкий уровень освоения; Игра создана, но не выполнено дополнительное задание – средний уровень освоения; Игра создана, выполнено дополнительное задание – высокий уровень освоения.

- Итоговый контроль осуществляется в форме тестирования. Тест состоит из 15 вопросов. Каждый вопрос оценивается в 1 балл. 0–5 баллов – низкий уровень освоения; 6–13 баллов – средний уровень освоения; 14-15 баллов – высокий уровень освоения.

Текущий контроль осуществляется путём определения качества выполнения заданий. Способы проверки уровня освоения тем: опрос, решение задач.

2.4 Методические материалы

В рамках реализации программы применяются следующие *методы обучения*:

- словесный: рассказ, беседа;
- практический: показ, выполнение практических работ и т.д.;
- объяснительно-иллюстративный: рассказ, показ, фильм и т.п.;
- репродуктивный: воспроизведение, действие по алгоритму;
- эвристический: частично-поисковый, самостоятельное нахождение ответов на поставленные педагогом вопросы;
- проблемный: постановка проблемных вопросов, создание проблемных ситуаций);
- проектный метод: создание творческих работ;
- метод игры: игры дидактические, развивающие, деловые;

Большую часть при реализации образовательной деятельности занимают активные и интерактивные методы в соответствии с возрастными особенностями обучающихся.

При преподавании курса используются следующие *педагогические технологии*: группового обучения, специальные технологии, соответствующие технической направленности; коллективного взаимообучения, дифференцированного обучения, проблемного обучения, игровой деятельности, коллективной творческой деятельности, критического мышления и др. Особое внимание уделяется использованию в учебном процессе *здоровьесберегающих технологий*, способствующих предотвращению состояний переутомления, гиподинамии (физминутки, зарядки для глаз и т.д.).

Используются следующие формы занятий: комбинированное занятие, практикум, урок-презентация, олимпиада, мастер-класс, игра и т.д. По дидактической цели занятия делятся на вводные, занятия по углублению знаний, практические занятия, занятия по систематизации и обобщению знаний, по контролю знаний, умений и навыков.

Структура учебного занятия строится в рамках технологии развития критического мышления и включает следующие этапы: вызов (мотивация к изучению материала), осмысление (изучение, повторение, закрепление учебного материала), рефлексия (подведение итогов, рефлексия эмоционального состояния, саморефлексия и т.д.).

Список литературы

1. Программируем с Minecraft. Создай свой мир с помощью Python / Крэйг Ричардсон; пер. с англ. Станислава Ломакина; [науч. ред. Г. Гаджиев]. — М.: Манн, Иванов и Фербер, 2017. — 368 с.: ил.
2. Кнут, Дональд, Эрвин / К53 Искусство программирования, том II. Основные алгоритмы, 3-е изд. : Пер. с англ. - М. : Издательский дом "Вильямс", 2002. -- 720 с. : ил. - Парал. тит. англ.
3. Вордерман, Стили, Квигли: Программирование на Python. Иллюстрированное руководство для детей
4. Андрей Корягин, Алиса Корягина. - Электрон. текстовые дан. (1 файл PDF: 224 с.). - Ростов н/Д: Феникс, 2021. - 222, [1] с.: ил. - (Гений программирования). - ISBN 978-5-222 -35282-3
5. Изучаем Python, 4-е издание. — Пер. с англ. — СПб.: Символ-Плюс, 2011.— 1280 с., ил. ISBN 978-5-93286-159-2Э
6. Эрик Мэтиз. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2016.

Электронные ресурсы:

1. <https://www.python.org/> Официальный сайт языка программирования.
2. <https://www.jetbrains.com/ru-ru/pycharm/> Официальный сайт редактора кода.

Список литературы для детей:

1. Привет, Python! Моя первая книга по программированию. П.Томашевский. Издательство: Наука и Техника, 2018. – 256 с.
2. Программирование для детей: Иллюстрированное руководство по языкам Scratch и Python. К. Вордерман и др. Издательство: Манн, Иванов и Фербер, 2017. – 224 с.

**Оценочные материалы по курсу
«Программирование на языке Python в игре Minecraft».**

Раздел 1. «Основы языка программирования».

Форма контроля – творческая работа.

Создайте мини-игру, где игрок, наступая на блоки земли, булыжника, песка, взлетает в высоту на 10 блоков. Дополнительно: реализуйте возможность проваливаться на 3 блока вниз, если игрок наступает на блок с алмазом.

Раздел 2. «Телепортация с помощью переменных»

Форма контроля – творческая работа.

Вы уже умеете задавать позицию игрока с помощью целых чисел, однако его координаты можно указать куда более точно, если воспользоваться вещественными числами. В ходе этой миссии вы измените код и телепортируете игрока в место с вещественными значениями координат.

Бонусное задание:

Присвойте переменным x , y и z какие-нибудь положительные и отрицательные вещественные значения и запустите программу. Затем измените цифры после точек и снова запустите. Что произойдет?

Раздел 3. «Математика, моментальное строительство и супер-прыжки»

Форма контроля – творческая работа.

Ваша задача — сделать так, чтобы игрок переместился, изменив свои координаты x , y и z на случайные. Для этого сначала нужно узнать текущие координаты игрока и прибавить к каждой из них случайное число. Чтобы изменить координаты x и z , используйте значения от -10 до 10 , а для координаты y — от 0 до 10 .

Бонусное задание:

Давайте сделаем программу `randomJump.py` еще более непредсказуемой! После прыжка в случайное место разместите под ногами игрока блок случайно выбранного типа. А еще вы можете добавить в программу код телепортации, чтобы каждый раз игрок оказывался в случайном месте. Если вы телепортируете игрока туда, откуда он не сможет выбраться, запустите `teleport.py` еще раз, и герой окажется в более безопасном месте.

Раздел 4. «Общаемся с помощью строк»

Форма контроля – творческая работа.

Для прохождения последней миссии из этой главы вам понадобятся все знания о переменных и математических операциях, а также умение отправлять сообщения в чат. Ваша **задача** — отследить, сколько прошел игрок за 10 секунд, и вывести эту информацию в чат.

Бонусное задание:

Сейчас программа отображает расстояние для каждой оси координат. Вместо этого вычислите и выведите на экран расстояние от начальной до конечной точки в виде одного целого числа.

Подсказка: для этого вам понадобится знание теоремы Пифагора.

Раздел 5. «Истина» и «Ложь» булевых значений»

Форма контроля – творческая работа.

Ваш игрок проголодался и хочет знать, есть ли в доме еда. Больше всего ему по вкусу арбузы, которые хранятся всегда в одном и том же месте. Но остались ли они в запасе или следует прихватить арбуз по дороге домой?

К счастью, вы изучаете Python и, немного поразмыслив, сможете узнать, есть ли в доме арбузы.

В ходе этой миссии вам предстоит создать программу, которая будет проверять, находится ли в заданных вами координатах блок «арбуз». Перед тем как приступить к написанию кода, подготовьтесь: обследуйте дом, или ферму, или другое место, где вы обычно храните арбузы, и найдите их там.

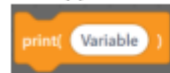
Бонусное задание:

Измените тип блока, наличие которого проверяет программа. Можете выяснить, растет ли у вас на ферме пшеница или не украл ли кто-нибудь входную дверь.

Итоговое тестирование по курсу «Программирование на языке Python в Minecraft»

Вопрос 1.

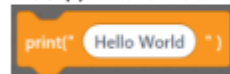
Что делает эта команда?



- 1) Выводит на экран текст,
- 2) Выводит на экран всё, кроме текста,
- 3) Позволяет пользователю что-то ввести в программу.

Вопрос 2.

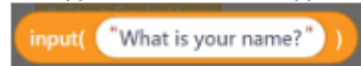
Что делает эта команда?



- 1) Выводит текст на экран,
- 2) Выводит на экран всё, кроме текста,
- 3) Позволяет пользователю что-то ввести в программу.

Вопрос 3.

Что делает эта команда?



- 1) Выводит текст на экран,
- 2) Выводит на экран всё, кроме текста,
- 3) Позволяет пользователю что-то ввести в программу.

Вопрос 4.

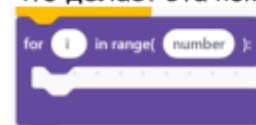
Что делает эта команда?



- 1) Превращает объект в текст,
- 2) Превращает объект в число,
- 3) Позволяет складывать текстовые сообщения.

Вопрос 5.

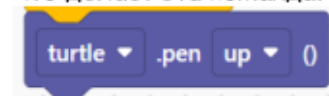
Что делает эта команда?



- 1) Повторяет определенное количество раз команды, которые внутри,
- 2) Рисует фигуры,
- 3) Создает переменные.

Вопрос 6.

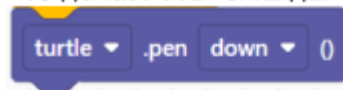
Что делает эта команда?



- 1) Перемещает в определенное место карандаш,
- 2) Опускает карандаш,
- 3) Поднимает карандаш.

Вопрос 7.

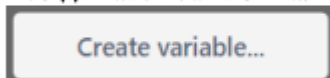
Что делает эта команда?



- 1) Перемещает в определенное место карандаш,
- 2) Опускает карандаш,
- 3) Поднимает карандаш.

Вопрос 8.

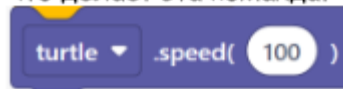
Что делает эта кнопка?



- 1) Создает переменную,
- 2) Создает новый файл,
- 3) Создает новую команду.

Вопрос 9.

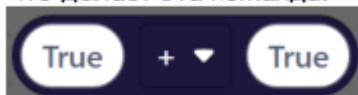
Что делает эта команда?



- 1) Добавляет цвет карандашу,
- 2) Увеличивает размер рисунка,
- 3) Увеличивает скорость отрисовки фигуры.

Вопрос 10.

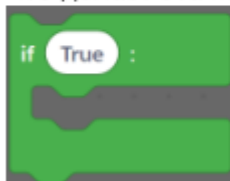
Что делает эта команда?



- 1) Складывает числа,
- 2) Складывает текстовые сообщения.

Вопрос 11.

Что делает эта команда?



- 1) Создает повторы кода,
- 2) Печатает сообщение на экран,
- 3) Что-то проверяет и если условие верно, то выполняет.

Вопрос 12.

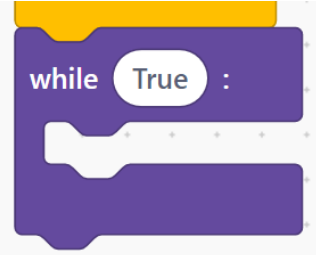
Что делает данная команда?



- 1) Перемещает персонажа в нужные координаты,
- 2) Выводит текст на экран,
- 3) Создает цикл.

Вопрос 13.

Что делает данная команда?



- 1) Создает цикл с определенным количеством повторов,
- 2) Превращает объект в текст,
- 3) Создает бесконечный цикл.

Вопрос 14.

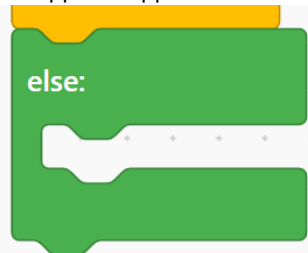
Что делает данная команда?



- 1) Создает цикл,
- 2) Создает функцию,
- 3) Создает условный оператор.

Вопрос 15.

Что делает данная команда?



- 1) Завершает условный оператор,
- 2) Создает условный оператор,
- 3) Сравнивает объекты.