



Методические рекомендации по теме

«Модуль turtle: перемещение объекта по координатам, работа с костюмами»

Цель:

- дать представление о рисовании по координатам и с помощью костюмов модуля turtle в языке Python.

Задачи:

- знакомство с приемами рисования по координатам в векторной графике;
- программирование действий исполнителя с помощью костюмов модуля turtle на языке Python;
- анализ программного кода с целью определения, что выведет программа при конкретных исходных данных;
- исправление ошибок и дописывание программного кода;
- написание программного кода.

Планируемые результаты

Личностные: обучающиеся получают навыки активной коммуникации в группе, осознанной ориентировки в мире ИТ профессий, постановки собственных образовательных целей и задач, владение первичными навыками анализа и критичной оценки получаемой информации.

Предметные: обучающиеся получают представления о рисовании по координатам и использовании костюмов исполнителя модуля turtle в языке Python.

Метапредметные: обучающиеся получают возможность владения общепредметными понятиями «градусы», «исполнитель», «векторная графика», «пиксель», «координаты»; информационно-логическими умениями; умениями самостоятельно планировать пути достижения целей; владения умениями принятия решений и осуществления

осознанного выбора; повышения уровня ИКТ – компетентности и расширение кругозора в области информатики и программирования; знакомство с профессиональной деятельностью программиста в рамках ранней профориентации; развитие интеллектуальных способностей, а также логического и критического мышления.

Материалы к занятию

Приложение 1: Сценарный план видеоролика

Приложение 2: Домашние задание и практика

Приложение 3: Краткие организационно-методические рекомендации по организации работы на занятии

Ход проведения урока

1. Организационный момент.

Мотивация на учебную деятельность.

Приветствие учащихся, сообщение темы и целей занятия.

2. Вводный блок.

Тема.

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия.

Проблемная дискуссия по вопросам:

- Для чего в программировании графики могут понадобиться координаты?
- Как связаны костюм и рисование?
- Удалось ли узнать, что такое фракталы и подумать о том с помощью какой программы можно их нарисовать в векторной графике?

Итоги дискуссии (обобщаются преподавателем и фиксируются ответы учеников на доске, чтобы вернуться к ним и оценить правильность предположений учеников на этапе рефлексии):

- Координаты при рисовании позволяют задать точное положение исполнителя на экране
- Исполнитель может менять форму (переодевать костюм)

**см. сцены 1 – 2 (здесь и далее приводится Таблица «Содержание видеоролика». Приложение 1).*

3. Блок повторения.

Блиц-опрос.

Преподаватель предлагает ученикам ответить на 5 вопросов по предыдущей теме; задания выполняются в сопровождении видеоролика с использованием таймера; ученики выполняют задания, голосуют, обсуждают результаты. Процедура голосования определяется инструкцией; учитель должен убедиться, что всем понятна процедура голосования. Преподаватель может поставить ролик на паузу и обсудить результаты голосования; объяснить правильный ответ руководствуясь материалами предыдущего занятия

**см. сцены 3 – 7*

4. Практический блок.

Векторная графика

Работа с модулем turtle подразумевает в большей степени выполнение практических графических проектов, поэтому освоение нового материала организовано в формате выполнения и разбора заданий с теоретическими вставками для объяснения основных понятий.

Для организации **практической работы** ученики занимают рабочие места и запускают Python (терминал IDLE) на своих компьютерах. Для выполнения практической работы используются материалы видеоролика:

- Подключение модуля turtle
- Костюм/форма пера «черепаха» и изменение размер пера
- Поворот «черепахи»
- Убегающий квадрат
- Как оставить отпечаток от пера

После демонстрации каждого задания запускается таймер. Время работы таймера определяется сложностью задания. До завершения работы таймера ученики выполняют задания на компьютерах.

После завершения работы таймера демонстрируется разбор задания. Ученики останавливают работу и обсуждают разбор задания.

**см. сцены 9 – 21 (кроме сцен с теорией)*

Практические задания разделены **теоретическими вставками**, необходимыми для работы над проектами урока:

- Форма пера
- Возможные формы пера
- Координаты объекта

**см. сцены 8, 13, 16*

По итогам работы ученики получают объекты, созданные с помощью векторной графики.

При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу; если ответы на вопросы вызывают у учеников затруднения, преподаватель может вывести нужную сцену ролика на экран для помощи ученикам.

5. Рефлексия. Сообщение домашнего задания.

Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Преподаватель возвращается к зафиксированным в ходе дискуссии в начале урока предположениям учеников и обсуждает насколько их предположения были правильными, делаются выводы.

Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*).

**см. сцена 22.*

Приложение 1

Сценарный план видеоролика

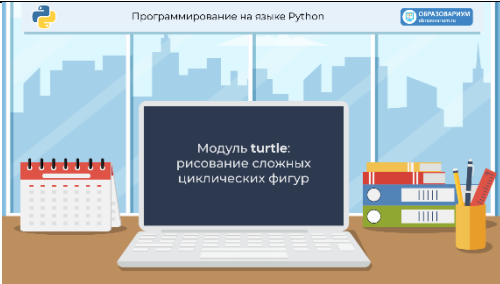
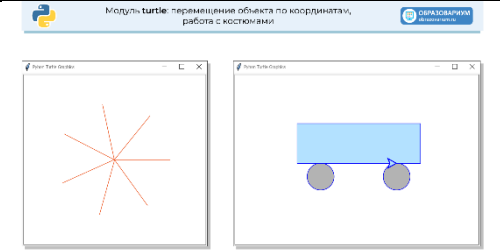
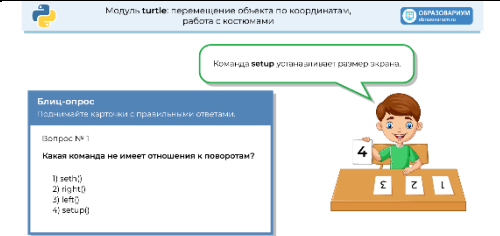
В таблице «Содержание видеоролика» представлены:

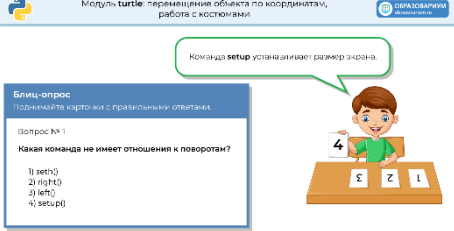
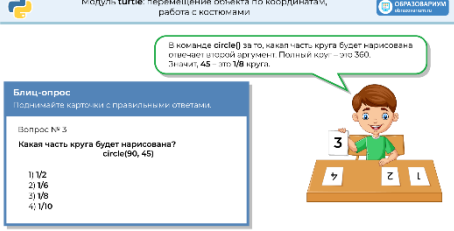
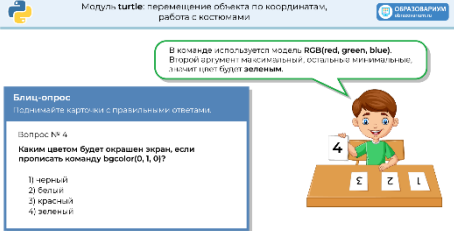
- название блоков видеоролика (тайминг);
- краткое описание содержания в каждом блоке;
- фрагменты из видеоролика, относящиеся к соответствующему блоку;
- номера сцен в каждом блоке.

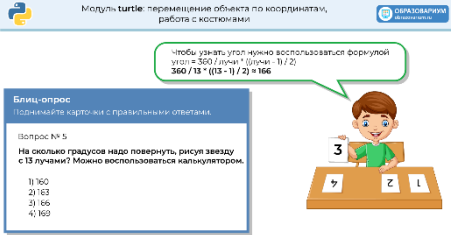
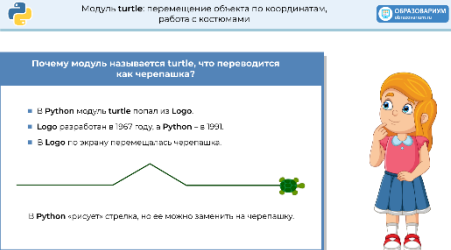
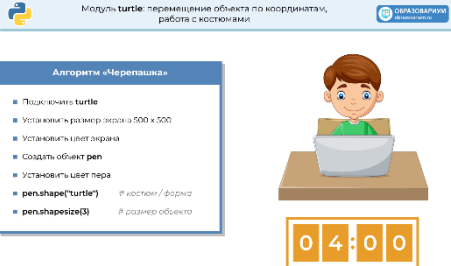
Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала как на уроке, так и для домашних заданий.

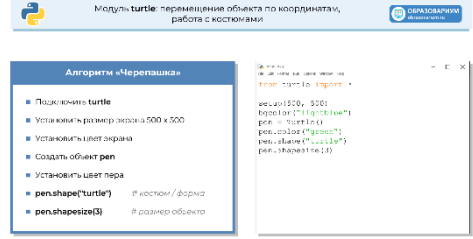
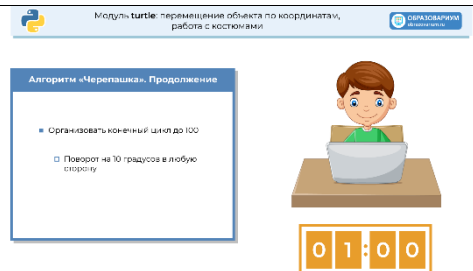
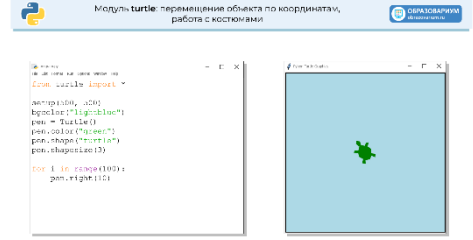
Таблица. Содержание видеоролика

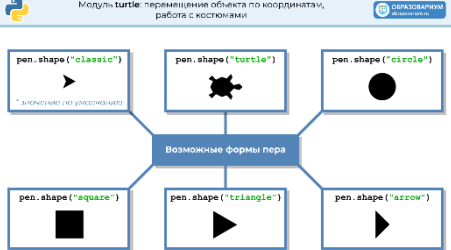

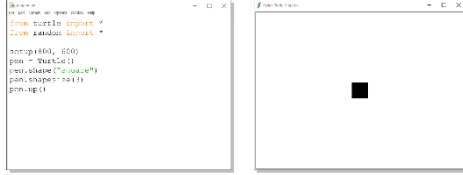
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
----------------	--------------------------------	--------------------------	--------

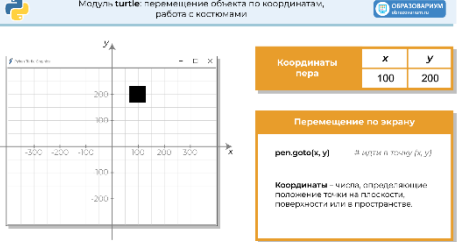
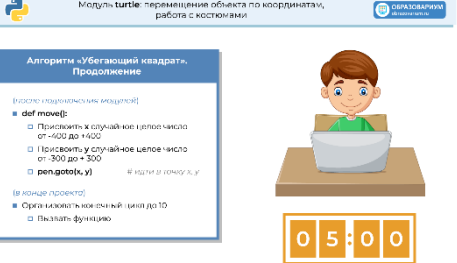
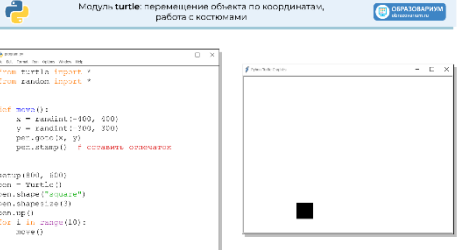
<p>Вводный блок.</p> <p>Мы узнаем</p>	<p><i>Обозначаем ученикам тему и цели урока.</i></p> <p>Модуль turtle: перемещение объекта по координатам, работа с костюмами</p>	 <p>Сцена 1</p>	<p>1 2</p>
	<p>Мы уже знаем, как создавать циклические фигуры разной сложности, как работать с окружностями и цветом. Умеем перемещать объект перо на известное расстояние в пикселях и поворачивать на определенное количество градусов.</p> <p>Сегодня нам предстоит узнать каким образом можно перемещать объект перо, не думая о том, в какую сторону он повернут, а также познакомимся с еще одним необычным способом создавать простые геометрические фигуры.</p>	 <p>Сцена 2</p>	
<p>Блок повторения.</p> <p>Блиц-опрос</p>	<p><i>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</i></p> <p>Первый вопрос. Какая команда не имеет отношения к поворотам?</p> <ol style="list-style-type: none"> 1) seth() 2) right() 3) left() 4) setup() <p><i>Ответ № 4: Команда setup устанавливает размер экрана.</i></p>	 <p>Сцена 3</p>	<p>3 4 5 6 7</p>


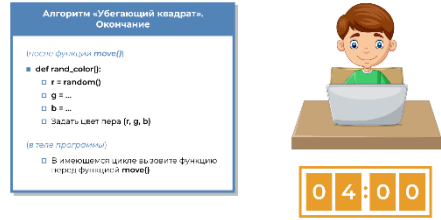
	<p>Второй вопрос. В какой угол будет смотреть объект после данной команды: <code>seth(135)</code></p> <ol style="list-style-type: none"> 1) правый верхний 2) правый нижний 3) левый верхний 4) левый нижний <p><i>Ответ 3. Направление 90 – это вверх, 180 – влево. Данное направление будет между ними</i></p>	 <p>Модуль: turtle: перемещение объекта по координатам, работа с костюмами</p> <p>Команда <code>setup</code> устанавливает размер экрана.</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 1 Какая команда не имеет отношения к поворотам?</p> <ol style="list-style-type: none"> 1) <code>seth()</code> 2) <code>right()</code> 3) <code>left()</code> 4) <code>setup()</code> <p>Сцена 4</p>	
	<p>Третий вопрос. Какая часть круга будет нарисована: <code>circle(90, 45)</code></p> <ol style="list-style-type: none"> 1) 1/2 2) 1/6 3) 1/8 4) 1/10 <p><i>Ответ 3. Первый аргумент — это радиус, он имеет отношение лишь к размеру. Нам важен второй аргумент. 45 — это восьмая часть полного круга</i></p>	 <p>Модуль: turtle: перемещение объекта по координатам, работа с костюмами</p> <p>В команде <code>circle()</code> за то, какая часть круга будет нарисована отвечает второй аргумент. Полный круг — это 360. Значит, 45 — это 1/8 круга.</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 3 Какая часть круга будет нарисована? <code>circle(90, 45)</code></p> <ol style="list-style-type: none"> 1) 1/2 2) 1/6 3) 1/8 4) 1/10 <p>Сцена 5</p>	
	<p>Четвертый вопрос. Каким цветом будет окрашен экран, если прописать команду <code>bgcolor(0, 1, 0)</code></p> <ol style="list-style-type: none"> 1) черный 2) белый 3) красный 4) зеленый <p><i>Ответ 4. Второй параметр - green — имеет максимальное значение, остальные минимальные.</i></p>	 <p>Модуль: turtle: перемещение объекта по координатам, работа с костюмами</p> <p>В команде используется модель RGB(<code>red, green, blue</code>). Второй аргумент максимальный, остальные минимальные, значит цвет будет зеленым.</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 4 Каким цветом будет окрашен экран, если прописать команду <code>bgcolor(0, 1, 0)</code>?</p> <ol style="list-style-type: none"> 1) черный 2) белый 3) красный 4) зеленый <p>Сцена 6</p>	

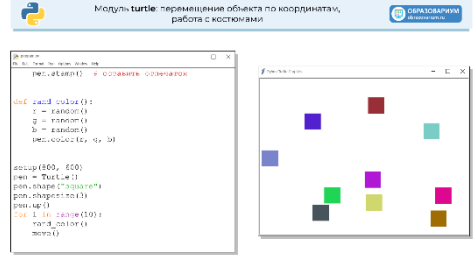
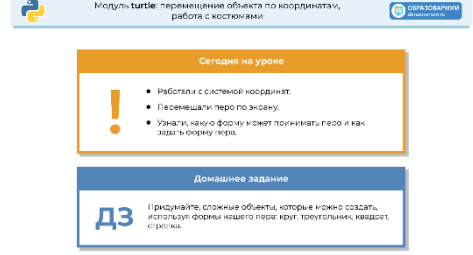
	<p>Пятый вопрос. На сколько градусов надо повернуть, рисуя звезду с 13ю лучами? Можно воспользоваться калькулятором</p> <ol style="list-style-type: none"> 1) 160 2) 163 3) 166 4) 169 <p><i>Ответ 3. Чтобы узнать угол надо 360 разделить на 13 и умножить на шесть.</i></p>	 <p>Модуль turtle: перемещение объекта по координатам, работа с костюмами</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 5 На сколько градусов надо повернуть, рисуя звезду с 13 лучами? Можно воспользоваться калькулятором.</p> <ol style="list-style-type: none"> 1) 160 2) 163 3) 166 4) 169 <p>Чтобы узнать угол нужно воспользоваться формулой: $360 / 13 * (13 - 1) / 2 = 166$</p> <p>3</p> <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p>Форма пера</p>	<p>Вероятно, у многих из вас возникал вопрос – а почему данный модуль называется так странно? turtle -то есть – черепашка.</p> <p>Дело в том, что в Python этот модуль попал из другого языка – Лого, который был разработан еще в далеком 1967 году. За 25 лет до появления Python.</p> <p>Там этот модуль был представлен небольшой черепашкой, которая ползала по экрану и создавала рисунки</p> <p>В Python по экрану движется небольшая стрелка. Но если вам хочется – можно заменить ее на черепашку.</p>	 <p>Модуль turtle: перемещение объекта по координатам, работа с костюмами</p> <p>Почему модуль называется turtle, что переводится как черепашка?</p> <ul style="list-style-type: none"> В Python модуль turtle попал из Logo. Logo разработан в 1967 году, а Python - в 1991. В Logo по экрану перемещалась черепашка. <p>В Python «рисует» стрелка, но ее можно заменить на черепашку.</p> <p>Сцена 8</p>	8
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1.</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p>Задание 1. Алгоритм «Черепашка»</p> <ul style="list-style-type: none"> Подключить turtle Установить размер экран 500 x 500 Установить цвет экрана Создать объект pen Установить цвет пера pen.shape("turtle") # костюм/форма pen.shapesize(3) # размер объекта 	 <p>Модуль turtle: перемещение объекта по координатам, работа с костюмами</p> <p>Алгоритм «Черепашка»</p> <ul style="list-style-type: none"> Подключить turtle Установить размер экрана 500 x 500 Установить цвет экрана Создать объект pen Установить цвет пера pen.shape("turtle") # костюм/форма pen.shapesize(3) # размер объекта <p>Сцена 9</p>	9 10

	<p>Разбор задания 1. Код программы может выглядеть так:</p> <pre>from turtle import * setup(500, 500) pen = Turtle() pen.color("green") pen.shape("turtle") pen.shapesize(3)</pre>	 <p>Сцена 10</p>	
<p>Блок заданий. Практические задания: Задание 1. Продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 1 мин.</i></p> <p>Задание 1. Алгоритм «Черепашка». Продолжение</p> <ul style="list-style-type: none"> Организовать конечный цикл до 100 <ul style="list-style-type: none"> Поворот на 10 градусов в любую сторону 	 <p>Сцена 11</p>	11 12
	<p>Разбор задания 1. Код программы может выглядеть так: <i>Дописывается к существующему коду</i></p> <pre>for i in range(100): pen.right(10)</pre>	 <p>Сцена 12</p>	

<p>Теоретический блок.</p> <p>Возможные формы пера</p>	<p>Давайте узнаем – какие еще формы может принять наше перо?</p> <p>Увы, но их не много: помимо стрелки и черепашки – квадрат, круг и треугольник.</p> <p>Но даже этого нам хватит, чтобы сделать новый интересный проект, под названием «убегающий квадрат».</p> <p>Возможные формы пера:</p> <pre>pen.shape("classic"); pen.shape("turtle"); pen.shape("circle"); pen.shape("square"); pen.shape("triangle"); pen.shape("arrow").</pre>	 <p>Сцена 13</p>	<p>13</p>
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 2.</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p>Задание 2. Алгоритм «Убегающий квадрат»</p> <ul style="list-style-type: none"> Подключить turtle и random Задать размеры экрана 800 x 600 Создать объект pen Поднять перо pen.shape("square") pen.shapesize(3) 	 <p>Сцена 14</p>	<p>14 15</p>
	<p>Разбор задания 2. Код программы будет выглядеть так:</p> <pre>from turtle import * from random import * setup(800, 600) pen = Turtle() pen.shape("square") pen.shapesize(3) pen.up()</pre>	 <p>Сцена 15</p>	

<p>Теоретический блок.</p> <p>Координаты объекта</p>	<p>Теперь давайте научим его двигаться по экрану в заданную точку. Для этого, у объекта есть команда goto. Она перемещает объект, ориентируясь на указанные координаты. Кстати, а что такое координаты? Координаты – числа, определяющих положение точки на плоскости, поверхности или в пространстве.</p> <p>По умолчанию, он находится в центре экрана и его икс и игрек равны нулю. Это значит, что при движении влево его икс будет уменьшаться, а вправо – увеличиваться. Игрек же будет увеличиваться при движении вверх и уменьшаться – при перемещении вниз. Впрочем, вам это должно быть знакомо, ведь это ни что иное как декартова система координат на плоскости.</p>	 <p>Сцена 16</p>	<p>16</p>
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 3.</p>	<p>После окончания дикторского текста запускается таймер на 5 мин.</p> <p>Задание 2. Алгоритм «Убегающий квадрат». Продолжение (после подключения модулей)</p> <ul style="list-style-type: none"> def move(): <ul style="list-style-type: none"> ○ Присвоить x случайное целое число от -400 до +400 ○ Присвоить y случайное целое число от -300 до + 300 ○ pen.goto(x,y) # идти в точку x, y <p>(В конце проекта)</p> <ul style="list-style-type: none"> ○ Организовать конечный до 10 <ul style="list-style-type: none"> ○ Вызвать функцию 	 <p>Сцена 17</p>	<p>17 18 19</p>
	<p>Разбор задания 3. Код программы может выглядеть так:</p> <p>Дописывается к уже существующему коду</p> <pre> from turtle import * from random import * def move(): x = randint(-400, 400) y = randint(-300, 300) pen.goto(x, y) setup(800, 600) pen = Turtle() </pre>	 <p>Сцена 18</p>	

	<pre> pen.shape("square") pen.shapesize(3) pen.up() for i in range(10): move() </pre>		
	<p>Задание 2. Алгоритм «Убегающий квадрат». Продолжение</p> <p>Давайте добавим внутрь функции, после команды goto, вот такую строчку. pen.stamp() # оставить отпечаток</p> <p>Команда stamp заставит наш объект оставить отпечаток своего костюма в той точке, куда он придет.</p> <p>Давайте теперь сделаем эти квадраты цветными.</p>	 <p>Сцена 19</p>	
	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p>Задание 2. Алгоритм «Убегающий квадрат». Продолжение <i>(после функции move ())</i></p> <ul style="list-style-type: none"> def rand_color(): <ul style="list-style-type: none"> o r = random () o g = o b = o Задать цвет пера (r, g, b) <p><i>(в теле программы)</i></p> <ul style="list-style-type: none"> o В имеющемся цикле вызовите функцию ПЕРЕД функцией move() 	 <p>Сцена 20</p>	20 21

	<p>Разбор задания 3. Код программы может выглядеть так:</p> <p><i>Дописывается к уже существующему коду</i></p> <pre>def rand_color(): r = random() g = random() b = random() pen.color(r, g, b) setup(800, 600) pen = Turtle() pen.shape("square") pen.shapesize(3) pen.up() for i in range(10): rand_color() move()</pre>	 <p>Сцена 21</p>	
<p>Блок завершения занятия.</p> <p>Рефлексия.</p> <p>Сообщение домашнего задания</p>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p>Подведем итоги.</p> <p>Мы узнали:</p> <ul style="list-style-type: none"> ■ Как работать с системой координат. ■ Как перемещали перо по экрану. ■ Узнали, какую форму может принимать перо и как задать форму пера. <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 22</p>	22

Приложение 2

Домашнее задание

Какие сложные объекты можно создать, используя формы пера исполнителя: круг, треугольник, квадрат, стрелка?

Задание можно выполнить на компьютере и представить результат и код в виде файла или снимка экрана, или распечатки.

Практика

Проект «Светофор»

С помощью модуля turtle создайте изображение светофора.

Важное условие: в проекте нельзя использовать команду circle.

Проект «Олимпийские кольца»

С помощью модуля turtle создайте изображение колец с олимпийского флага.

Примечание: не забудьте, что цвета колец на флаге разные.

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Модуль turtle: перемещение объекта по координатам, работа с костюмами».

Перед началом занятия необходимо повторить понятия «градусы и пиксели», а также команды управления: поднять и опустить перо, выбор направления объекта, создание окружности.

Если имеется техническая возможность - посвятить немного времени домашнему заданию - фракталам: что это за объекты, какие принципы лежат в основе создания и показать, как они могут выглядеть.

Перед просмотром блока повторения из ролика необходимо раздать дидактический материал для выполнения заданий (по 4 пронумерованных карточки). Во время голосований карточками можно останавливать ролик и вести учет правильных ответов. По окончании блока – отметить тех, у кого наилучший результат. Далее карточки необходимо собрать.

Первый проект показывает возможности Python при работе с формой объекта. Код, который создается в качестве первоначальных установок, достаточно стандартный и может быть, по большей части, использован во втором проекте для экономии времени. Размер объекта и продолжительность циклов носят условный характер.

Второй проект потребует вспомнить школьный курс математики, посвященный координатам на плоскости. Можно поставить ролик на паузу и проговорить этот материал дополнительно. Важно, чтобы ребята поняли – как изменяются при движении координаты объекта. А также что размер экрана — это общее количество точек от края до края, а не от центра до края.

Второй проект, в силу своего размера, делается в три этапа. После каждого из них обязательно тестируйте код на работоспособность.

Если останется время – можете заменить костюм на «triangle» или «circle».