



Методические рекомендации по теме

«Модуль turtle: модель RGB, понятие объект, основные команды управления»

Цель:

- дать представление о рисовании замкнутых многоугольников с помощью модуля turtle в языке Python.

Задачи:

- практика применения функций векторной графики для рисования в Python;
- прикладное применение знаний по расчету градусной меры углов замкнутого многоугольника;
- 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 об использовании модуля turtle для создания векторной графики в Python при рисовании замкнутых многоугольников.

Метапредметные: обучающиеся получают возможность владения общепредметными понятиями «многоугольник», «градус», «окружность», «исполнитель»; 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:

- Что такое градусы в графике и для чего они используются?
- Что такое поворот на 90, 180, 360 градусов?
- Что такое прямой угол?
- Что мы обозначаем в графике числом пикселей?
- Попросите учеников подумать и записать свои идеи чему равна сумма углов в замкнутом многоугольнике
- Какую фигуру мы получим если сделаем программу в которой исполнитель поворачивает 360 раз на 1 градус?

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

- Градусы используются для определения угла поворота исполнителя
- Сумма поворотов при рисовании замкнутого многоугольника равна 360 градусам
- Пиксели обозначают длину линии, которую проходит (рисует) исполнитель

**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.

Многоугольники

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
- Рисование квадрата и равностороннего треугольника
- Цвет линии и заливка цветом фигуры
- Организация случайного выбора цвета заливки
- Организация случайного выбора количество сторон правильного многоугольника

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 11 – 25 (кроме сцен с теорией)*

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

- Правильные многоугольники
- Вычисление угла поворота
- Функция выбора цвета
- Цикл случайного выбора

**см. сцены 8 – 10, 15, 19*

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 26.*

Приложение 1

Сценарный план видеоролика

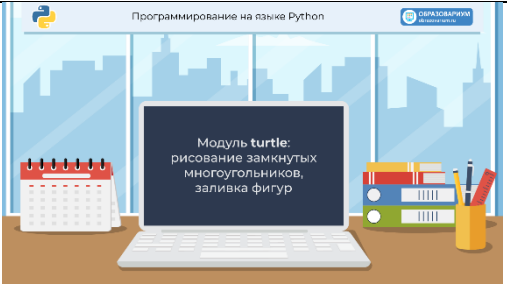
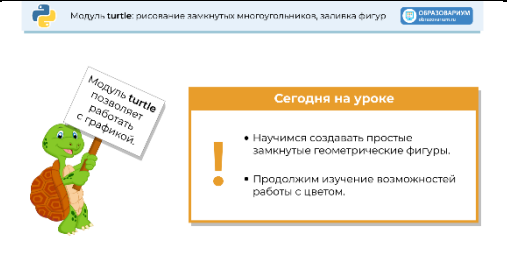
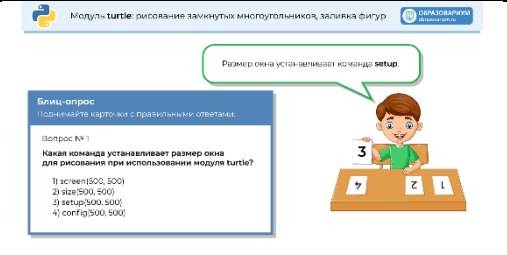
В таблице «Содержание видеоролика» представлены:

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

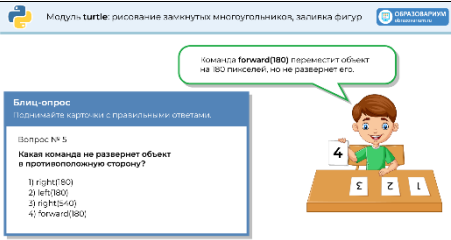
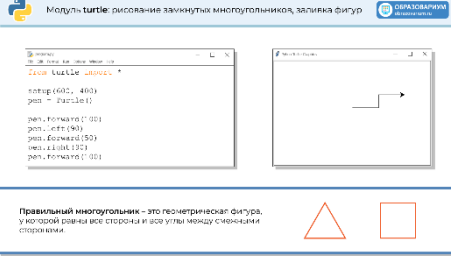
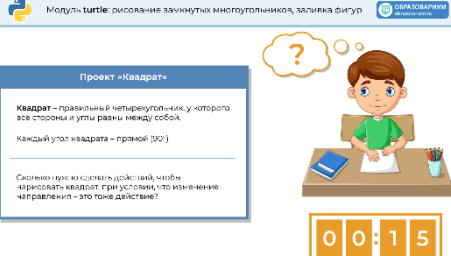
Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала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

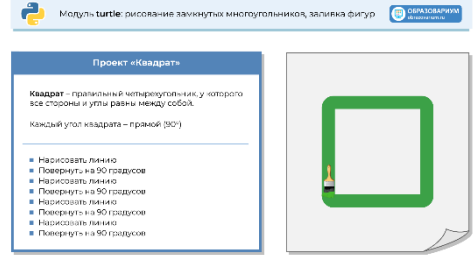
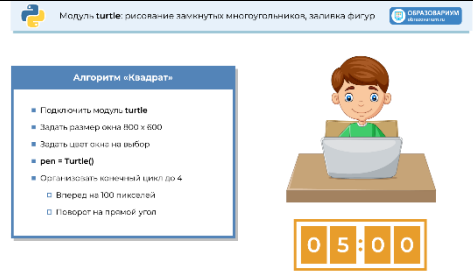
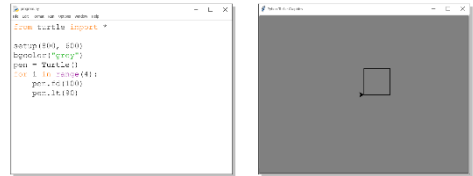
Таблица. Содержание видеоролика

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

<p>Вводный блок. Мы узнаем</p>	<p><i>Обозначаем ученикам тему и цели урока.</i></p> <p>Модуль turtle: рисование замкнутых многоугольников, заливка фигур</p>	 <p>Сцена 1</p>	<p>1 2</p>
	<p>Мы уже знаем, что модуль turtle позволяет работать с графикой, для чего создается специальная информационная структура, которая называется объект класса.</p> <p>Сегодня мы научимся создавать простые замкнутые геометрические фигуры, а также продолжим работу с цветом.</p>	 <p>Сцена 2</p>	
<p>Блок повторения. 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 устанавливает размер окна для рисования при использовании модуля turtle?</p> <ol style="list-style-type: none"> 1) screen(500, 500) 2) size(500, 500) 3) setup(500, 500) 4) config(500, 500) <p><i>Ответ № 3: Размер окна устанавливает команда setup.</i></p>	 <p>Сцена 3</p>	<p>3 4 5 6 7</p>

	<p>Второй вопрос. В какой строке правильно создан объект для рисования?</p> <ol style="list-style-type: none"> 1) имя.turtle() 2) имя.Turtle() 3) имя(Turtle) 4) имя.Turtle <p><i>Ответ 2. Правильный синтаксис операции таков: имя.Turtle()</i></p>	<p>Модуль turtle: рисование замкнутых многоугольников, заливка фигур</p> <p>Правильный синтаксис операции таков: <code>имя = Turtle()</code></p> <p>Блиц-опрос Подчеркните карточки с правильными ответами.</p> <p>Вопрос № 2 В какой строке правильно создан объект для рисования?</p> <ol style="list-style-type: none"> 1) имя = turtle() 2) имя = Turtle() 3) имя = turtle 4) имя = turtle <p>Сцена 4</p>	
	<p>Третий вопрос. Какой из перечисленных цветов не входит в состав базовых цветов модели RGB?</p> <ol style="list-style-type: none"> 1) синий 2) желтый 3) красный 4) зеленый <p><i>Ответ 2. В цветовой модели RGB основными цветами являются красный, зеленый и синий.</i></p>	<p>Модуль turtle: рисование замкнутых многоугольников, заливка фигур</p> <p>В цветовой модели RGB основными цветами являются красный, зеленый и синий.</p> <p>Блиц-опрос Подчеркните карточки с правильными ответами.</p> <p>Вопрос № 3 Какой из перечисленных цветов не входит в состав базовых цветов модели RGB?</p> <ol style="list-style-type: none"> 1) синий 2) желтый 3) красный 4) зеленый <p>Сцена 5</p>	
	<p>Четвертый вопрос. Какой цвет окна для рисования будет если прописать команду bgcolor(0, 0, 0)?</p> <ol style="list-style-type: none"> 1) черный 2) белый 3) красный 4) ошибка <p><i>Ответ 1. В команде используется цветовая модель RGB. RGB(0, 0, 0) соответствует черному цвету.</i></p>	<p>Модуль turtle: рисование замкнутых многоугольников, заливка фигур</p> <p>В команде используется цветовая модель RGB. RGB(0, 0, 0) соответствует черному цвету.</p> <p>Блиц-опрос Подчеркните карточки с правильными ответами.</p> <p>Вопрос № 4 Какой цвет окна получится, если прописать команду bgcolor(0, 0, 0)?</p> <ol style="list-style-type: none"> 1) черный 2) белый 3) красный 4) ошибка <p>Сцена 6</p>	

	<p>Пятый вопрос. Какая команда не развернет объект в противоположную сторону?</p> <ol style="list-style-type: none"> 1) right(180) 2) left(180) 3) right(540) 4) forward(180) <p><i>Ответ 4. Команда forward(180) переместит объект на 180 пикселей, но не развернет его.</i></p>	 <p>Модуль: turtle: рисование замкнутых многоугольников, заливка фигур</p> <p>Блиц-опрос Подчеркните карточку с правильными ответами.</p> <p>Вопрос № 5 Какая команда не развернет объект в противоположную сторону?</p> <ol style="list-style-type: none"> 1) right(180) 2) left(180) 3) right(540) 4) forward(180) <p>Команда forward(180) переместит объект на 180 пикселей, но не развернет его.</p> <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p>Правильные многоугольники</p>	<p>На прошлом занятии созданный нами объект рисовал ломанные линии, используя команды forward и повороты на определенное количество градусов.</p> <p>Мы научимся создавать правильные многоугольники. Напомним, что правильный многоугольник – это геометрическая фигура, у которой равны все стороны и все углы между смежными сторонами.</p>	 <p>Модуль: turtle: рисование замкнутых многоугольников, заливка фигур</p> <pre> import turtle t = turtle.Turtle() t.forward(400) t.right(90) t.forward(400) t.right(90) t.forward(400) t.right(90) t.forward(400) t.right(90) </pre> <p>Правильный многоугольник – это геометрическая фигура, у которой равны все стороны и все углы между смежными сторонами.</p> <p>Сцена 8</p>	8 9 10
	<p><i>После окончания дикторского текста запускается таймер на 15 сек.</i></p> <p>Вопрос. Наш первый проект будет посвящен рисованию квадрата: замкнутой фигуры, состоящей из четырех одинаковых сторон, расположенных под углом 90 градусов друг к другу.</p> <p>Сколько нужно сделать действий, чтобы нарисовать квадрат, при условии, что изменение направления – это тоже действие?</p>	 <p>Модуль: turtle: рисование замкнутых многоугольников, заливка фигур</p> <p>Проект «Квадрат»</p> <p>Квадрат – правильный четырехугольник, у которого все стороны и углы равны между собой. Каждый угол квадрата – прямой (90°)</p> <p>Сколько нужно сделать действий, чтобы нарисовать квадрат, при условии, что изменение направления – это тоже действие?</p> <p>00:15</p> <p>Сцена 9</p>	

	<p>Разбор ответа. Решение выглядит так: Правильный ответ – восемь. Причем оригинальных действий всего два: нарисовать линию и повернуть карандаш вправо или влево на 90 градусов. Эту последовательность надо сделать четыре раза. Следовательно, при создании кода мы можем воспользоваться конечным циклом, в котором будут прописаны повторяющиеся команды</p>	 <p>Сцена 10</p>	
<p>Блок заданий. Практические задания: Задание 1.</p>	<p><i>После окончания дикторского текста запускается таймер на 5 мин.</i></p> <p>Задание 1. Проект «Квадрат».</p> <ul style="list-style-type: none"> ■ Подключить модуль turtle ■ Задать размер окна 800x600 ■ Задать цвет окна на выбор ■ pen = Turtle() ■ Организовать конечный цикл до 4 <ul style="list-style-type: none"> ○ Вперед на 100 пикселей ○ Поворот на прямой угол 	 <p>Сцена 11</p>	<p>11 12</p>
	<p>Разбор задания 1. Код программы будет выглядеть так:</p> <pre>from turtle import * setup(800, 600) bgcolor("grey") pen = Turtle() for i in range(4): pen.fd(100) pen.lt(90)</pre>	 <p>Сцена 12</p>	

Блок заданий.
Практические задания:
Задание 2.

После окончания дикторского текста запускается таймер на 1 мин.

Задание 2. Алгоритм «Равносторонний треугольник».

- Измените код так, чтобы получился треугольник.

Модуль turtle: рисование замкнутых многоугольников, заливка фигур

```
from turtle import *\n\nbgcolor("grey")\npen = Turtle()\nfor i in range(3):\n    pen.fd(100)\n    pen.lt(120)
```

Проект «Равносторонний треугольник»

Измените код так, чтобы получился равносторонний треугольник.

01:00

Сцена 13

Разбор задания 2. Код программы может выглядеть так:

```
from turtle import *\nsetup(800, 600)\nbgcolor("grey")\npen = Turtle()\nfor i in range(3):\n    pen.fd(100)\n    pen.lt(120)
```

Модуль turtle: рисование замкнутых многоугольников, заливка фигур

```
from turtle import *\n\nbgcolor("grey")\npen = Turtle()\nfor i in range(3):\n    pen.fd(100)\n    pen.lt(120)
```

Сцена 14

Теоретический блок.

Углы поворота

Почему при рисовании квадрата угол поворота был 90, а для треугольника вдруг стал 120?

Все дело в том, что мы имеем дело с замкнутой фигурой, которую можно представить как деформированный круг. А круг это 360 градусов. Соответственно если разделить это число на количество сторон, то получится угол поворота

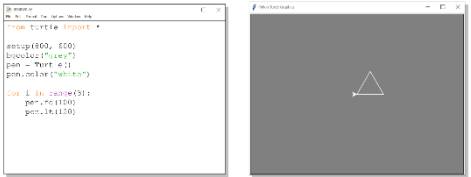
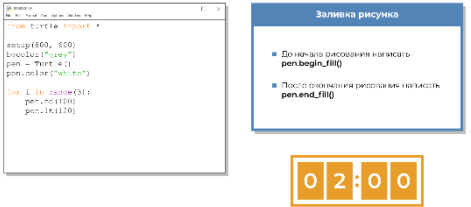
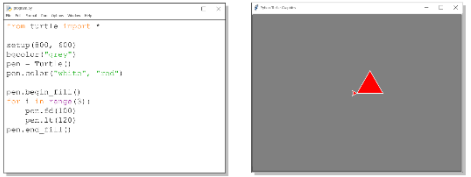
Модуль turtle: рисование замкнутых многоугольников, заливка фигур



Углы поворота	
Квадрат	90°
Равносторонний треугольник	120°
Равносторонний многоугольник	360° / количество сторон






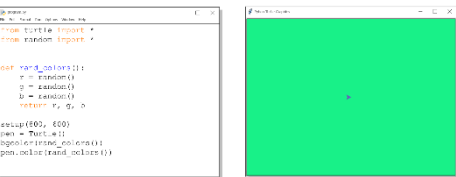



Сцена 15

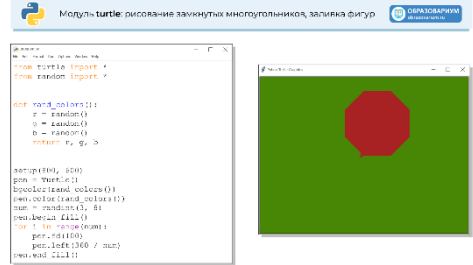
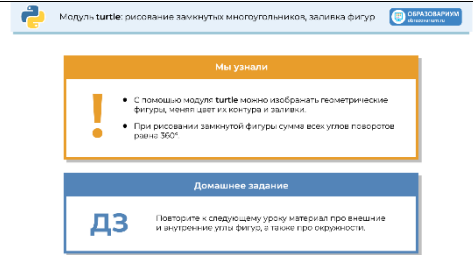
13
14

15

<p>Практические задания: Задание 2. Продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 2. Алгоритм «Равносторонний треугольник». Продолжение Цвет объекта. После создания объекта pen напишите: pen.color("white")</p>	 <p>Сцена 16</p>	<p>16 17 18</p>
	<p>Но это еще не все. Давайте его закрасим. Для этого используется блок из двух команд: начать заливку – она пишется до начала рисования и закончить заливку – после рисования. Обе команды пишутся после имени нашего объекта. Заливка рисунка</p> <ul style="list-style-type: none"> До начала рисования написать pen.begin_fill() После окончания рисования написать pen.end_fill() 	 <p>Сцена 17</p>	
	<p>Разбор задания 2. Код программы может выглядеть так:</p> <pre>from turtle import * setup(800, 600) bgcolor("grey") pen = Turtle() pen.color("white") pen.begin_fill() for i in range(3): pen.fd(100) pen.lt(120) pen.end_fill()</pre>	 <p>Сцена 18</p>	

<p>Теоретический блок.</p> <p>Выбор фона</p>	<p>Теперь сделаем финальный проект, в котором компьютер будет случайным образом выбирать цвет фона, цвет объекта, а также какую фигуру нарисовать: от треугольника до восьмиугольника.</p> <p>Нам понадобится</p> <ul style="list-style-type: none"> • Подключение модулей • Создание функции выбора цвета • Создание окна и объекта • Установки цвета фона и цвета объекта • Выбор фигуры • Цикл рисования с заливкой 	<p>Модуль: turtle: рисование замкнутых многоугольников, заливка фигур</p> <p>Проект «Случайная цветная фигура»</p> <div> <div> <p>Будем рисовать фигуру, выбирая случайным образом:</p> <ul style="list-style-type: none"> • цвет фона • цвет объекта • какую фигуру нарисовать </div> <div> <p>Нам понадобятся:</p> <ul style="list-style-type: none"> • подключение модулей • создание функции выбора цвета • создание окна и объекта • установка цвета фона и цвета объекта • выбор фигуры • цикл рисования с заливкой </div> </div> <p>Сцена 19</p>	<p>19</p>
<p>Практические задания:</p> <p>Задание 3.</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p>Задание 3. Алгоритм «Случайная цветная фигура»</p> <ul style="list-style-type: none"> • Подключить turtle, random • Функция rand_color <ul style="list-style-type: none"> o r=random() o g... o b.... o return r, g, b 	<p>Модуль: turtle: рисование замкнутых многоугольников, заливка фигур</p> <p>Алгоритм «Случайная цветная фигура»</p> <ul style="list-style-type: none"> • Подключить модули turtle, random • Функция rand_color <ul style="list-style-type: none"> o r=random() o g... o b.... o return r, g, b  <p>Сцена 20</p>	<p>20 21</p>
	<p>Разбор задания 3. Код программы может выглядеть так:</p> <pre>from turtle import * from random import * def rand_colors(): r = random() g = random() b = random() return r, g, b</pre>	<p>Модуль: turtle: рисование замкнутых многоугольников, заливка фигур</p> <p>Алгоритм «Случайная цветная фигура»</p> <ul style="list-style-type: none"> • Подключить модули turtle, random • Функция rand_color <ul style="list-style-type: none"> o r=random() o g... o b.... o return r, g, b  <p>Сцена 21</p>	

<p>Практические задания: Задание 3. Продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i> Задание 3. Алгоритм «Случайная цветная фигура». Продолжение</p> <ul style="list-style-type: none"> • Задать размер окна 800x600 • Создать объект pen • bgcolor(rand_colors()) • pen.color(rand_colors()) 	<div data-bbox="1509 231 1962 260">  Модуль turtle: рисование замкнутых многоугольников, заливка фигур  </div> <div data-bbox="1509 292 1727 469"> <p>Алгоритм «Случайная цветная фигура». Продолжение</p> <ul style="list-style-type: none"> ■ Задать размер окна 800 x 600 ■ Создать объект pen ■ bgcolor(rand_colors()) ■ pen.color(rand_colors()) </div> <div data-bbox="1778 280 1928 499">  </div> <p>Сцена 22</p>	<p>22 23</p>
	<p>Разбор задания 3. Код программы может выглядеть так: <i>Дописывается к уже существующему коду</i></p> <pre> setup(800, 600) pen = Turtle() bgcolor(rand_colors()) pen.color(rand_colors()) </pre>	<div data-bbox="1509 571 1962 600">  Модуль turtle: рисование замкнутых многоугольников, заливка фигур  </div> <div data-bbox="1509 632 1962 809">  </div> <p>Сцена 23</p>	<p>24 25</p>
<p>Практические задания: Задание 3. Окончание</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i> Задание 3. Алгоритм «Случайная цветная фигура». Окончание</p> <ul style="list-style-type: none"> • num = randint(3, 8) # количество сторон • Начать заливку • Организовать конечный цикл до num <ul style="list-style-type: none"> о Вперед 100 о Поворот на 360 / num • Закончить заливку 	<div data-bbox="1509 890 1962 919">  Модуль turtle: рисование замкнутых многоугольников, заливка фигур  </div> <div data-bbox="1509 951 1727 1128"> <p>Алгоритм «Случайная цветная фигура». Продолжение</p> <ul style="list-style-type: none"> ■ num = randint(3, 8) # количество сторон ■ Начать заливку ■ Организовать конечный цикл до num <ul style="list-style-type: none"> □ Вперед 100 □ Поворот на 360 / num ■ Закончить заливку </div> <div data-bbox="1778 940 1928 1158">  </div> <p>Сцена 24</p>	

	<p>Разбор задания 3. Код программы может выглядеть так:</p> <p><i>Дописывается к уже существующему коду</i></p> <pre> num = randint(3, 8) pen.begin_fill() for i in range(num): pen.fd(100) pen.left(360 / num) pen.end_fill() </pre>	 <p>Сцена 25</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> <ul style="list-style-type: none"> С помощью модуля turtle можно изображать геометрические фигуры, меняя цвет их контура и заливки. При рисовании замкнутой фигуры сумма всех углов поворотов равна 360°. <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 26</p>	26

Приложение 2

Домашнее задание

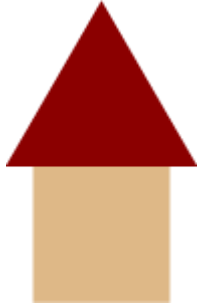
Найдите в различных источниках (интернет, литература) информацию про внешние и внутренние углы фигур, а также про окружности. Проанализируйте и изучите данную информацию.

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

Практика

Проект «Домик»

С помощью модуля turtle создайте рисунок домика. Не забудьте задать цвет стены и крыши.



Проект «Вертикальный флаг»

С помощью модуля turtle создайте рисунок вертикального триколора со случайно выбранными цветами.

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Модуль turtle: модель RGB, понятие объект, основные команды управления».

Перед началом занятия необходимо повторить понятия, с которыми нам предстоит работать (градусы, пиксели), а также новые команды (установка размера, движение, повороты).

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.

Главная идея всех проектов – сумма поворотов нашего объекта, рисующего замкнутую фигуру равна 360 градусов. Поэтому все вычисления должны опираться на этот факт. Можно в качестве разминки посчитать угол поворота для фигур, состоящих из 3-8 углов.

Создание второго проекта происходит в три этапа – для удобства понимания кода. При этом код пользовательской функции практически полностью повторяет код прошлого урока, а циклическая часть – взята из первого проекта. Делать фигуры больше чем в 8 углов не информативно – чем больше сторон, тем больше места на экране она занимает. По этой же причине не стоит делать длину стороны более 150 пикселей. А еще (и это можно подчеркнуть) с увеличением количества сторон фигура все больше будет похожа на обычный круг.