



Методические рекомендации по теме «Модуль turtle: рисование сложных циклических фигур»

Цель:

- дать представление о рисовании циклических фигур с помощью векторной графики модуля Turtle в языке Python.

Задачи:

- знакомство с приемами рисования циклических фигур в векторной графике;
- программирование действий исполнителя 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 рисовании циклических фигур с помощью модуля Turtle в языке Python.

Метапредметные: обучающиеся получают возможность владения общепредметными понятиями «градусы», «исполнитель», «векторная графика», «пиксель», «циклическая фигура»; 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 команды поднять перо и опустить перо?
- Что такое циклическая фигура?
- Связаны ли понятия цикл и циклическая фигура? Как?
- Как можно рассчитать градусную меру угла для рисования звезд и снежинок?

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

- Команды поднять и опустить перо могут потребоваться для рисования прерывистых линий или промежутков в изображениях
- Для рисования циклических фигур мы как правило применяем циклы
- Существует формулы, которые позволяют рассчитать угол поворота при рисовании циклических фигур

**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
- Рисование снежинки
- Рисование звезды

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 – 20** (*кроме сцен с теорией*)

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

- Определение циклической фигуры
- Вычисление угла поворота сложной фигуры
- Формула для нечетного количества лучей

*см. сцены **8, 14, 17**

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 зафиксированному 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 21.

Приложение 1

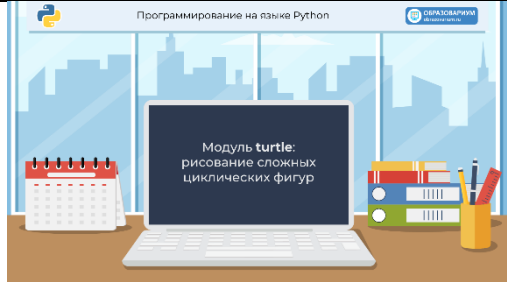
Сценарный план видеоролика

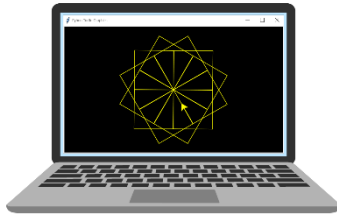
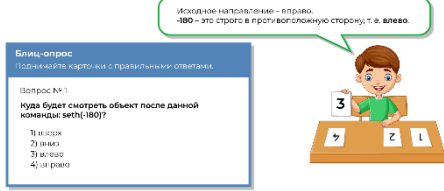
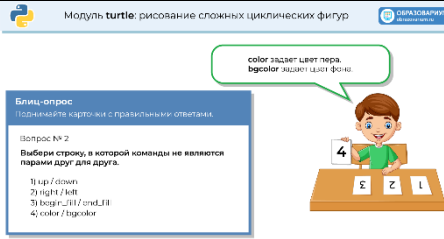
В таблице «Содержание видеоролика» представлены:

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

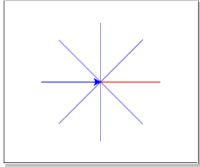

Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала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

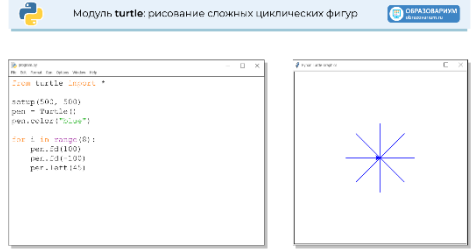
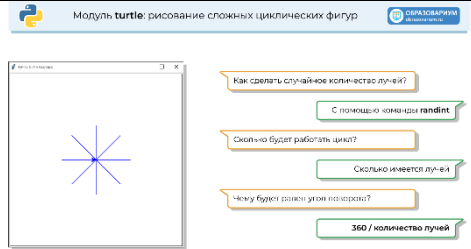
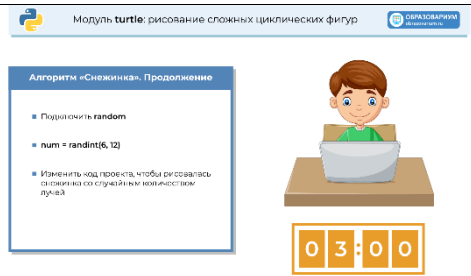
Таблица. Содержание видеоролика

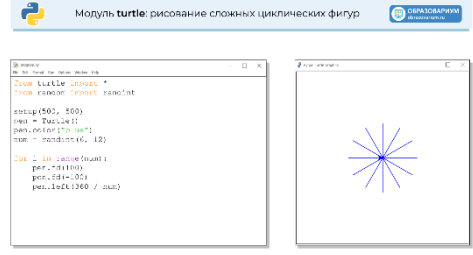
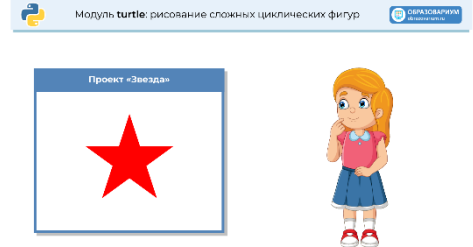
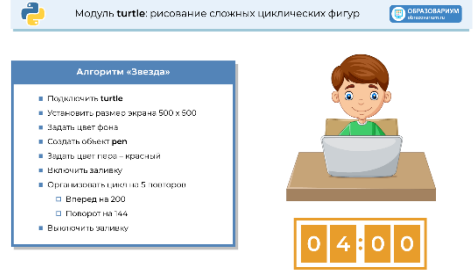
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока. Модуль turtle: рисование сложных циклических фигур	 Сцена 1	1 2

	<p>Продолжаем знакомство с графическим модулем turtle, рассмотрим алгоритмы, которые позволят нам создавать сложные циклические 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. Куда будет смотреть объект после данной команды: seth(-180)?</p> <ol style="list-style-type: none"> 1) вверх 2) вниз 3) влево 4) вправо <p><i>Ответ № 3: Исходное направление – вправо. Минус 180, это строго в противоположную сторону. Также как и просто 180.</i></p>	 <p>Сцена 3</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p>Второй вопрос. Выбери строку, в которой команды не являются парами друг для друга</p> <ol style="list-style-type: none"> 1) up – down 2) right – left 3) begin_fill – end_fill 4) color – bgcolor <p><i>Ответ 4. Команда color относится к цвету пера, а bgcolor – к цвету экрана</i></p>	 <p>Сцена 4</p>	

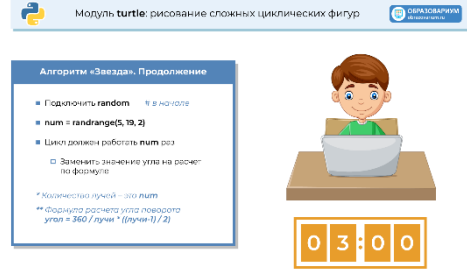
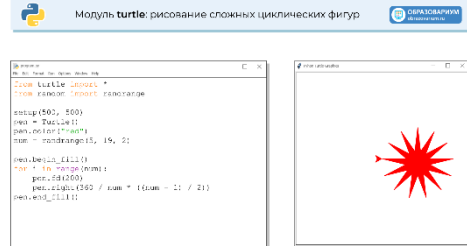
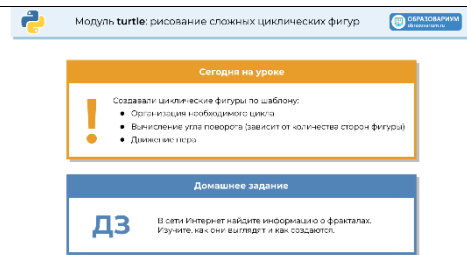
	<p>Третий вопрос. Какая часть круга будет нарисована: circle(180, 120)</p> <ol style="list-style-type: none"> 1) $\frac{1}{2}$ 2) $\frac{1}{3}$ 3) $\frac{1}{4}$ 4) $\frac{18}{12}$ <p><i>Ответ 2. Первый аргумент — это радиус, он имеет отношение лишь к размеру. Нам важен второй аргумент. 120 — это третья часть полного круга</i></p>	<div data-bbox="1509 236 1962 466"> <p>Модуль turtle: рисование сложных циклических фигур</p> <p>В команде circle(180, 120) первый аргумент описывает радиус, а второй – какая часть круга будет нарисована. 120 – это $\frac{1}{3}$ часть полного круга.</p> <p>Блиц-опрос Подчеркните карточки с правильными ответами.</p> <p>Вопрос № 3 Какая часть круга будет нарисована: circle(180, 120)?</p> <ol style="list-style-type: none"> 1) $\frac{1}{2}$ 2) $\frac{1}{3}$ 3) $\frac{1}{4}$ 4) $\frac{18}{12}$ </div> <p>Сцена 5</p>
--	--	---

<p>Теоретический блок.</p> <p>Циклическая фигура</p>	<p>Знакомство со сложными циклическими фигурами мы начнем с восьмиконечной снежинки. Каким образом сделать наш код максимально простым?</p> <p>Разумеется, через цикл. Но какой?</p> <p>На первый взгляд – нужно 4 раза под определенным углом, нарисовать линию, по принципу: в одну сторону – в противоположную сторону – в центр</p> <p>Однако это не самое удачное решение: получается минимум три команды. Лучше сделать цикл не 4, а 8 раз, каждый раз рисуя луч в сторону и возвращаясь в центр.</p> <p>Всего две команды. А угол поворота? Его мы можем вычислить просто: 360 разделить на количество лучей, то есть на 8. Получится 45 градусов</p>	<p>Модуль turtle: рисование сложных циклических фигур</p> <p>Проект «Снежинка»</p> <ul style="list-style-type: none"> Будем использовать цикл Количество повторов цикла: 8 Угол поворота: $360 / 8 = 45$  <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 6 мин.</i></p> <p>Задание 1. Алгоритм «Снежинка»</p> <ul style="list-style-type: none"> Подключить turtle Установить размер экран 500×500 Задать цвет фона Создать объект pen Задать цвет пера – синий Организовать цикл на 8 повторов <ul style="list-style-type: none"> Вперед на 100 Назад на 100 Поворот на 45 	<p>Модуль turtle: рисование сложных циклических фигур</p> <p>Алгоритм «Снежинка»</p> <ul style="list-style-type: none"> Подключить turtle Установить размер экрана 500×500 Задать цвет фона Создать объект pen Задать цвет пера – синий Организовать цикл на 8 повторов <ul style="list-style-type: none"> Вперед на 100 Назад на 100 Поворот на 45  <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("blue") for i in range(8): pen.fd(100) pen.bk(100) pen.left(45)</pre>	 <p>Сцена 10</p>	
	<p>Немного усложним наш проект. Пусть это будет снежинка со случайным количеством лучей- от шести до двенадцати.</p> <p>Этот выбор мы доверим random. Соответственно и цикл будет работать не 8 раз, а сколько укажет команда randint. Угол поворота пусть вычисляет программа.</p> <p>Он равен 360 разделить на количество лучей.</p>	 <p>Сцена 11</p>	11
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1.</p> <p>Продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 1. Алгоритм «Снежинка». Продолжение</p> <ul style="list-style-type: none"> Подключить random num = randint(6,12) Изменить код проекта, чтобы рисовалась снежинка со случайным количеством лучей 	 <p>Сцена 12</p>	12 13

	<p>Разбор задания 1. Код программы может выглядеть так:</p> <pre> from turtle import * from random import randint setup(500, 500) pen = Turtle() pen.color("blue") num = randint(6, 12) for i in range(num): pen.fd(100) pen.fd(-100) pen.left(360 / num) </pre>	 <p>Сцена 13</p>	
<p>Теоретический блок.</p> <p>Рисование фигуры Звезда</p>	<p>А теперь разберемся как создавать очень интересную фигуру, под названием звезда.</p> <p>Это тоже циклическая фигура, и в нашем случае имеет пять лучей. Но вся сложность заключена не в коде, а в вычислении угла поворота</p>	 <p>Сцена 13</p>	14
<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 «Звезда»</p> <ul style="list-style-type: none"> ■ Подключить turtle ■ Установить размер экран 500 x 500 ■ Установить цвет фона ■ Создайте объект pen ■ Установить цвет пера – красный ■ Включить заливку ■ Организовать цикл на 5 повторов <ul style="list-style-type: none"> ○ Вперед на 200 ○ Поворот на 144 	 <p>Сцена 14</p>	15 16

	<div><div>■ Выключить заливку</div><div>Разбор задания 2. Код программы будет выглядеть так:<pre>from turtle import * setup(500, 500) pen = Turtle() pen.color("red") pen.begin_fill() for i in range(5): pen.fd(200) pen.right(144)</pre></div></div>	<div><div>Модуль turtle: рисование сложных циклических фигур</div><div><div>Алгоритм «Звезда»</div><div><ul style="list-style-type: none">■ Подключить turtle■ Установить размер экрана 500 x 500■ Задать цвет фона■ Создать объект pen■ Задать цвет пера – красный■ Начинать заливку■ Организовать цикл из 5 повторений<ul style="list-style-type: none">□ Вперед на 200□ Поворот на 144■ Выключить заливку</div><div><pre>from turtle import * setup(500, 500) pen = Turtle() pen.color("red") pen.begin_fill() for i in range(5): pen.fd(200) pen.right(144) pen.end_fill()</pre></div></div></div> <div>Сцена 15</div>																
<div>Теоретический блок.</div> <div>Рисуем окружностями</div>	<div>Вот дополнительные данные: чтобы нарисовать звезду в 7 лучей – угол поворота равен 154; в 9 лучей – 160 градусов</div> <div>Некая закономерность прослеживается, что чем больше лучей, тем больше угол, но все равно не особо понятно. Посмотрим на формулу. Как видите она достаточно необычная. Хотя в ней, по факту, всего одна переменная: количество лучей. А 360 градусов – это полный круг.</div> <div>Сразу оговоримся, что эта формула не работает с четным количеством лучей!</div>	<div><div>Модуль turtle: рисование сложных циклических фигур</div><div><div>Формула угла поворота для звезды с нечетным количеством лучей</div><div>угол = 360 / лучи * (лучи-1) / 2</div><div><table><tr><th>Лучи</th><th>Градусы</th><th>Расчет угла поворота</th></tr><tr><td>5</td><td>144</td><td>5 лучей 360 / 5 * 2 = 144</td></tr><tr><td>7</td><td>154</td><td>7 лучей 360 / 7 * 5 = 154</td></tr><tr><td>9</td><td>160</td><td>9 лучей 360 / 9 * 4 = 160</td></tr><tr><td>11</td><td>163</td><td>11 лучей 360 / 11 * 5 = 163</td></tr></table></div></div></div> <div>Сцена 16</div>	Лучи	Градусы	Расчет угла поворота	5	144	5 лучей 360 / 5 * 2 = 144	7	154	7 лучей 360 / 7 * 5 = 154	9	160	9 лучей 360 / 9 * 4 = 160	11	163	11 лучей 360 / 11 * 5 = 163	17
Лучи	Градусы	Расчет угла поворота																
5	144	5 лучей 360 / 5 * 2 = 144																
7	154	7 лучей 360 / 7 * 5 = 154																
9	160	9 лучей 360 / 9 * 4 = 160																
11	163	11 лучей 360 / 11 * 5 = 163																
<div>Блок заданий.</div> <div>Практические задания:</div> <div>Задание 3.</div>	<div>После окончания дикторского текста запускается таймер на 1 мин.</div> <div>Задание. Проверка формулы</div> <div>Поменяйте в проекте данные цикла и угла поворота, чтобы убедиться в правильности формулы.</div> <div><div>ЛУЧИ</div><div>ГРАДУСЫ</div><div>5144</div><div>7154</div><div>9160</div><div>11163</div></div>	<div><div>Модуль turtle: рисование сложных циклических фигур</div><div><div>Проверка формулы</div><div>Поменяйте в проекте данные цикла и угла поворота, чтобы убедиться в правильности формулы.</div><div><table><tr><th>Лучи</th><th>Градусы</th></tr><tr><td>5</td><td>144</td></tr><tr><td>7</td><td>154</td></tr><tr><td>9</td><td>160</td></tr><tr><td>11</td><td>163</td></tr></table></div><div>01:00</div></div></div> <div>Сцена 17</div>	Лучи	Градусы	5	144	7	154	9	160	11	163	18 19 20					
Лучи	Градусы																	
5	144																	
7	154																	
9	160																	
11	163																	

	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 3. Алгоритм «Звезда». Продолжение</p> <ul style="list-style-type: none"> Подключить random # в начале num = randrange(5, 19, 2) Цикл должен работать num раз <ul style="list-style-type: none"> Заменить значение угла на расчет по формуле <p>* Количество лучей – это num ** Формула расчета угла поворота $\text{угол} = 360 / \text{лучи} * ((\text{лучи} - 1) / 2)$</p>	 <p>Сцена 18</p>	
	<p>Разбор задания 3. Код программы может выглядеть так:</p> <pre>from turtle import * from random import randrange setup(500, 500) pen = Turtle() pen.color("red") num = randrange(5, 19, 2) pen.begin_fill() for i in range(num): pen.fd(200) pen.right(360 / num * ((num - 1) / 2)) pen.end_fill()</pre>	 <p>Сцена 19</p>	20
<p>Блок завершения занятия. Рефлексия. Сообщение домашнего задания</p>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p>Подведем итоги.</p> <p>Мы узнали: Как создаются циклические фигуры по шаблону:</p> <ul style="list-style-type: none"> Организация необходимого цикла Вычисление угла поворота Движение пера <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 21</p>	21

Домашнее задание

Поищите в интернете информацию о фракталах: как они выглядят и как создаются. Подумайте можем ли мы запрограммировать рисование фракталов в векторной графике.

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

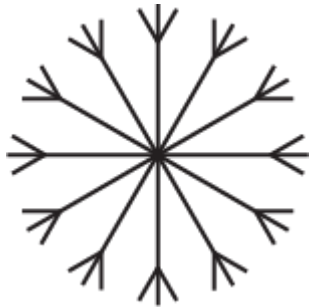
Практика

Проект «Сложная снежинка»

Доработайте проект «Снежинка», созданный на уроке. На конце каждого луча должна быть нарисована еще «веточка».

Размер и угол наклона «веток» можете выбрать на свое усмотрение.

Например, сложная снежинка может выглядеть так:



Проект «Четная звезда»

С помощью модуля turtle создайте изображение звезды с четным количеством лучей.

Случайным образом выберите

- четное количество лучей в диапазоне от 8 до 20,
- угол поворота луча обратно в диапазоне от 130 до 190.

В конечном цикле организуйте рисование одного луча. Для этого проведите линию, поверните на выбранный угол, проведите линию обратно. Для рисования следующего луча сделайте поворот в другую сторону на такое значение: **выбранный угол - 360 / количество лучей.**

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Модуль turtle: рисование сложных циклических фигур».

Перед началом занятия необходимо повторить понятия, с которыми нам предстоит работать (градусы, пиксели), а также новые команды управления: поднять и опустить перо, выбор направления объекта.

Также можно разобрать домашнее задание. К числу узоров, которые создаются при помощи окружностей можно отнести: олимпийские кольца, светофор, мишень в тире. А с помощью неполной окружности – радугу.

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.

На уроке будет создано несколько проектов с похожими установочными данными: размер экрана, цвет фона. Поэтому можно переносить данный код из одного проекта в другой ради экономии времени. Единственно, в проекте «снежинка» рекомендуется синий цвет, а в проекте «звезда» - красный.

Сами фигуры создаются на основе однотипного кода: цикл- движение-поворот. Однако, если вычисление угла поворота в первом проекте не вызовет никаких сложностей, то во втором проекте этому надо уделить отдельно время. Для этого можно поставить ролик на паузу и дать практические задания на вычисления: например, какие цифры будут в формуле если речь идет о звезде в 45 лучей? $(360 / 45 * 22)$

Новый метод модуля random – **randrange** -позволяет генерировать целые числа используя не сплошной числовой ряд, а применяя шаг пропуска чисел. Логично, что если шаг = 2 и первый аргумент – нечетное число, то и все остальные сгенерированные числа тоже будут нечетными.