



Методические рекомендации по теме

«Модуль turtle: обработка событий клавиатуры для управления анимацией»

Цель:

- дать представление об управлении исполнителем при помощи клавиатуры в модуле turtle языка Python.

Задачи:

- знакомство с программированием управления исполнителем при помощи клавиатуры;
- программирование событий при нажатии различных кнопок клавиатуры;
- 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 программировании клавиатуры для управления исполнителем модуля turtle в языке Python.

Метапредметные: обучающиеся получают возможность владения общепредметными понятиями «событие», «исполнитель», «клавиатура», «клавиша»; 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 обработчик событий в программировании?
- Какие события могут быть при использовании клавиатуры?
- Как может себя вести исполнитель при управлении с клавиатуры?
- Что такое лямбда-функция?

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

- При управлении с клавиатуры событием может быть нажатие на любую кнопку или комбинацию кнопок
- Исполнитель может выполнить программу, переместиться в определенную сторону, перейти в координаты, повернуться, опустить или поднять перо и т.д.
- Лямбда функция – функция, которая может выполнять только одну команду

**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

- Управление клавиатурой
- Команда onkeypress
- Функции вверх, вниз, влево, вправо
- Поднять и опустить перо

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 – 18 (кроме сцен с теорией)*

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

- Обработчик событий клавиатуры
- Команда onkeypress
- Управление пером
- Координаты «мыши»

**см. сцены 8, 11, 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 19.*

Приложение 1

Сценарный план видеоролика

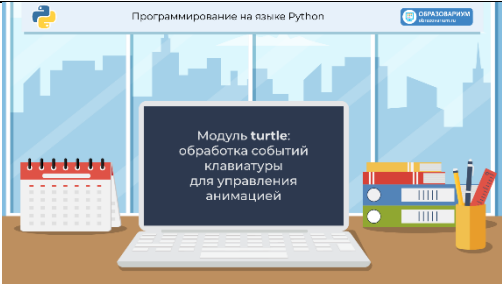
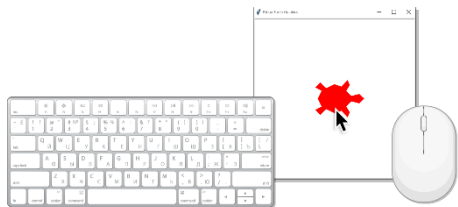
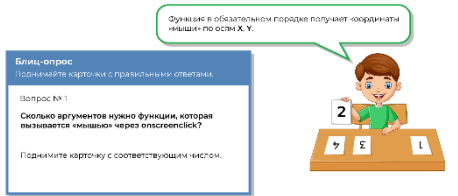
В таблице «Содержание видеоролика» представлены:

- 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>Обозначаем ученикам тему и цели урока.</p> <p>Модуль turtle: обработка событий клавиатуры для управления анимацией</p>	 <p>Сцена 1</p>	<p>1 2</p>
	<p>На прошлом занятии мы познакомились с обработчиком событий мыши – командой, которая вызывается при возникновении определенных событий в программе, связанных с кнопками мыши.</p> <p>Теперь мы научимся активировать ещё и клавиши клавиатуры и связывать их с необходимыми функциями, чтобы управлять нашей программой</p>	 <p>Сцена 2</p>	
<p>Блок повторения. Блиц-опрос</p>	<p>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</p> <p>Первый вопрос. Сколько аргументов нужно функции, которая вызывается «мышью» через onclickclick?</p> <p>Поднимите карточку с соответствующим числом.</p> <p><i>Ответ № 2: Функция в обязательном порядке получает координаты «мыши» по осям X, Y</i></p>	 <p>Сцена 3</p>	<p>3 4 5 6 7</p>

Второй вопрос. Какая кнопка мыши активирована onclick (one_2, 3)

- 1) левая
- 2) средняя
- 3) правая
- 4) никакая

Ответ 3. Первый аргумент — это имя функции, а второй — это номер кнопки.

1 – левая кнопка, 2 – «колесо», 3 – правая кнопка.

Третий вопрос. Какая команда не имеет отношения к поворотам?

- 1) seth ()
- 2) right ()
- 3) left ()
- 4) goto ()

Ответ 4. Команда goto перемещает объект в заданную точку, но не поворачивает.

Четвертый вопрос. Если setup (800,600) - какая точка имеется на экране?

- 1) 800, 0
- 2) 600, 600
- 3) -300, 300
- 4) 0, 600

Ответ 3. Экран будет иметь точки с координатами по оси X от -400 до 400 и по оси Y от -300 до 300. В этот диапазон попадает только точка (-300, 300).

Модуль turtle: обработка событий клавиатуры для управления анимацией

За кнопку мыши отвечает второй аргумент:
1 - левая кнопка, 2 - колесо, 3 - правая кнопка.

Блиц-опрос
Подчеркните карточки с правильными ответами.

Вопрос № 2
Какая кнопка мыши активирована обработчиком событий?
`onclick(one_2, 3)`

1) левая
2) средняя
3) правая
4) никакая

Сцена 4

Модуль turtle: обработка событий клавиатуры для управления анимацией

Команда goto перемещает объект в заданную точку, но не поворачивает.

Блиц-опрос
Подчеркните карточки с правильными ответами.

Вопрос № 3
Какая команда не имеет отношения к поворотам?

1) seth()
2) right()
3) left()
4) goto()

Сцена 5

Модуль turtle: обработка событий клавиатуры для управления анимацией

Экран будет иметь точки с координатами по оси X от -400 до 400 и по оси Y от -300 до 300. В этот диапазон попадает только точка (-300, 300).

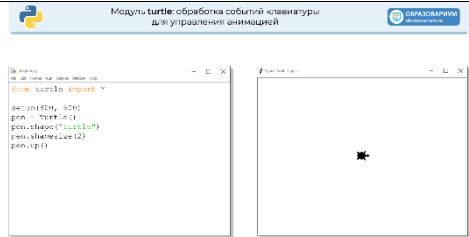

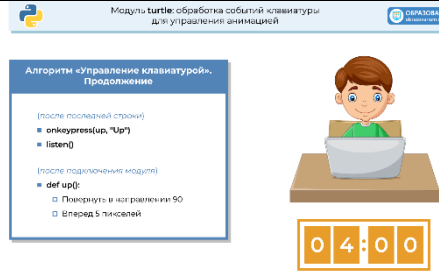
Блиц-опрос
Подчеркните карточки с правильными ответами.

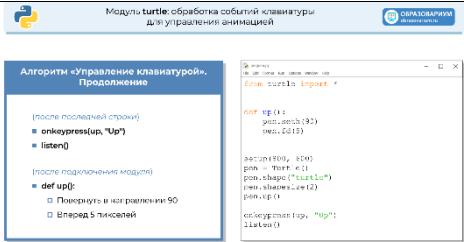
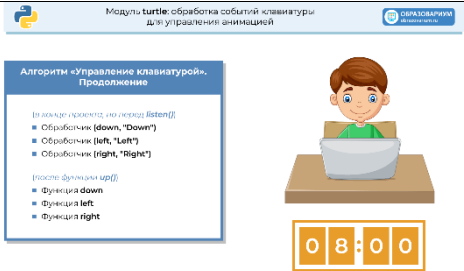
Вопрос № 4
Какая точка имеется на экране с заданными параметрами?
`setup(800, 600)`

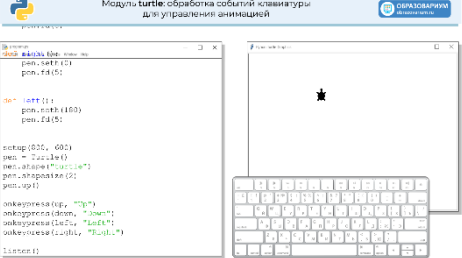
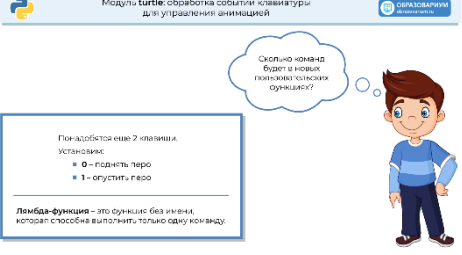
1) (800, 0)
2) (600, 600)
3) (-300, 300)
4) (0, 600)

Сцена 6

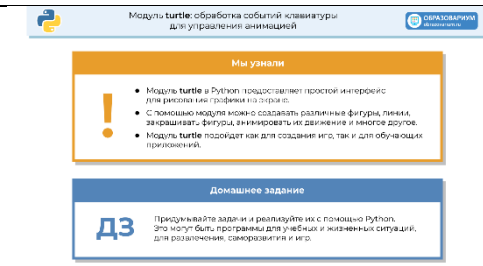
	<p>Пятый вопрос. В какой угол будет повернут объект после данной команды: <code>seth(-45)</code></p> <ol style="list-style-type: none"> 1) правый верхний 2) правый нижний 3) левый верхний 4) левый нижний <p><i>Ответ 2. Направление 0 – это вправо, -90 – вниз. <code>seth(-45)</code> будет указывать между ними, т. е. в правый нижний угол.</i></p>	<p>Модуль turtle: обработка событий клавиатуры для управления анимацией</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 5 В какой угол будет повернут объект после данной команды? <code>seth(-45)</code></p> <p>1) правый верхний 2) правый нижний 3) левый верхний 4) левый нижний</p> <p>Направление 0 – это вправо, -90 – вниз. <code>seth(-45)</code> будет указывать между ними, т. е. в правый нижний угол.</p> <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p>Обработчик событий</p>	<p>Управлять программой при помощи мыши очень удобно, однако функционал при этом достаточно ограничен. Ведь в нашем распоряжении всего три кнопки.</p> <p>Поэтому сегодня мы узнаем, как в модуле turtle обрабатывать события, заданные с клавиатуры.</p> <p>Или проще говоря – как управлять программой при помощи клавиш</p>	<p>Модуль turtle: обработка событий клавиатуры для управления анимацией</p> <p>Как в модуле turtle обрабатывать события, заданные с клавиатуры?</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 «Управление клавиатурой»</p> <ul style="list-style-type: none"> ■ Подключить turtle ■ Установить размер экрана 800 x 600 ■ Создать объект pen ■ Установить костюм черепашки ■ Установить размер, равный 2 ■ Поднять перо 	<p>Модуль turtle: обработка событий клавиатуры для управления анимацией</p> <p>Алгоритм «Управление клавиатурой»</p> <ul style="list-style-type: none"> ■ Подключить turtle ■ Установить размер экрана 800 x 600 ■ Создать объект pen ■ Установить костюм «черепашки» ■ Установить размер, равный 2 ■ Поднять перо <p>Сцена 9</p>	9 10

	<p>Разбор задания 1. Код программы может выглядеть так:</p> <pre>from turtle import *</pre> <pre>setup(800, 600) pen = Turtle() pen.shape("turtle") pen.shapesize(2) pen.up()</pre>	 <p>Сцена 10</p>	
<p>Команда onkeypress</p>	<p>Теперь давайте активируем обработчик событий клавиатуры. Это команда onkeypress, которая имеет два аргумента: имя вызываемой функции и название клавиши. А еще нам понадобится команда-активатор listen – слушать. Без нее обработчик клавиатуры работать не будет. Эта команда пишется один раз, после всех обработчиков.</p> <p>Обработчик клавиатуры onkeypress (имя функции, имя клавиши) listen() – активатор</p>	 <p>Сцена 11</p>	11
<p>Блок заданий. Практические задания: Задание 1. Продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p>Задание 1. Алгоритм «Управление клавиатурой». Продолжение <i>после последней строки</i></p> <ul style="list-style-type: none"> onkeypress(up, "Up") listen() <p><i>после подключения модуля</i></p> <ul style="list-style-type: none"> def up(): <ul style="list-style-type: none"> Повернуть в направлении 90 Вперед 5 пикселей 	 <p>Сцена 12</p>	12 13

	<p>Разбор задания 1. Код программы может выглядеть так:</p> <p><i>Дописывается к существующему коду</i></p> <pre> from turtle import * def up(): pen.seth(90) pen.fd(5) setup(800, 600) pen = Turtle() pen.shape("turtle") pen.shapesize(2) pen.up() onkeypress(up, "Up") listen() </pre>	 <p>Сцена 13</p>	
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1.</p> <p>Продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 10 мин.</i></p> <p>Задание 1. «Управление клавиатурой». Продолжение</p> <p><i>В конце проекта, но перед listen()</i></p> <ul style="list-style-type: none"> ■ обработчик (down, "Down") ■ обработчик (left, "Left") ■ обработчик (right, "Right") <p><i>После функции up()</i></p> <ul style="list-style-type: none"> ■ Функция down ■ Функция left ■ Функция right 	 <p>Сцена 14</p>	<p>14 15</p>

	<p>Разбор задания 1. Код программы будет выглядеть так:</p> <pre> from turtle import * def up(): pen.seth(90) pen.fd(5) def down(): pen.seth(270) pen.fd(5) def right(): pen.seth(0) pen.fd(5) def left(): pen.seth(180) pen.fd(5) setup(800, 600) pen = Turtle() pen.shape("turtle") pen.shapesize(2) pen.up() onkeypress(up, "Up") onkeypress(down, "Down") onkeypress(left, "Left") onkeypress(right, "Right") listen() </pre>	 <p>Сцена 15</p>	
<p>Теоретический блок.</p> <p>Управление пером</p>	<p>Однако лучше сделаем из нее простейший графопостроитель. То есть научим ее еще поднимать и опускать перо. И для этого активируем еще две клавиши с цифрами ноль – поднять перо и один – опустить перо. Однако прежде ответим на вопрос: сколько команд будет в наших пользовательских функциях? Ответ – одна: поднять или опустить перо. Делать функцию ради одной команды не очень правильно. Давайте вспомни про лямбда-функцию. Лямбда-функция – это функция без имени, которая способна выполнить только одну команду.</p>	 <p>Сцена 16</p>	16

<p>Блок заданий. Практические задания: Задание 1.</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 1. Алгоритм «Управление клавиатурой». Продолжение</p> <p><i>В конце проекта, но перед listen()</i></p> <ul style="list-style-type: none"> ■ onkeypress(lambda: pen.down(), '1') ■ onkeypress(lambda: pen.up(), '0') 	<div data-bbox="1496 228 1962 499"> </div> <p>Сцена 17</p>	<p>17 18</p>
	<p>Разбор задания 1. Код программы может выглядеть так:</p> <pre> from turtle import * def up(): pen.seth(90) pen.fd(5) def down(): pen.seth(270) pen.fd(5) def right(): pen.seth(0) pen.fd(5) def left(): pen.seth(180) pen.fd(5) setup(800, 600) pen = Turtle() pen.shape("turtle") pen.shapesize(2) pen.up() onkeypress(up, "Up") onkeypress(down, "Down") onkeypress(left, "Left") onkeypress(right, "Right") onkeypress(lambda: pen.down(), "1") </pre>	<div data-bbox="1496 579 1962 850"> </div> <p>Сцена 18</p>	

	<pre>onkeypress(lambda: pen.up(), "0") listen()</pre>		
Блок завершения занятия. Рефлексия. Сообщение домашнего задания	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p>Подведем итоги.</p> <ul style="list-style-type: none"> Модуль turtle в Python предоставляет простой интерфейс для рисования графики на экране. С помощью модуля можно создавать различные фигуры, линии, закрашивать фигуры, анимировать их движение и многое другое. Модуль turtle подойдет как для создания игр, так и для обучающих приложений. <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 19</p>	19

Приложение 2

Домашнее задание

Всем желающим продолжить самостоятельное изучение языка Python рекомендован замечательный интерактивный учебник Питонтьютор. Его адрес в интернете: <https://pythontutor.ru/>

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

Практика

Проект «Непрерывное движение»

С помощью модуля turtle создайте программу непрерывного передвижения черепашки.

Черепашка должна двигаться непрерывно до тех пор, пока не коснется краев экрана. Реализуйте смену направления движения по нажатию на стрелки.

Проект «Смена цвета»

Дополните проект «Управление клавиатурой», созданный на уроке. Добавьте возможность смены цвета для черепашки нажатием на кнопки клавиатуры.

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии

«Модуль turtle: обработка событий клавиатуры для управления анимацией».

Перед началом занятия о том, что собой представляет обработчик событий – команда, вызываемая при возникновении определенных событий. В чем синтаксические особенности вызова пользовательских функций (без указания скобок) и почему необходимы аргументы для событий «мыши» (передаются координаты)?

Также желательно повторить материал про лямбда-функции, поскольку их использование также запланировано.

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.

Создаваемый проект позволит понять принцип работы обработчика событий клавиатуры – команды `onkeypress`. Проект состоит из нескольких частей, каждую из которых можно тестировать после создания.

Особенное внимание необходимо уделить функциям, создаваемым обработчиком. В данном случае никаких аргументов не передается, однако скобки при вызове функции также не ставятся.

В случае необходимости обработчика выполнить только одну команду – мы можем прибегнуть к помощи лямбда-функции. Команда `listen()` – активатор клавиатуры – идет после всех обработчиков. Обратите на это внимание.