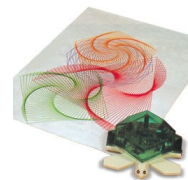


## 8 класс. Практическая работа №6 (Turtle)

### Обработчик событий



### Алгоритм работы

1. В модуле Turtle определены методы обработки событий нажатия на клавишу мыши или одну из клавиш клавиатуры. Обработчики событий реализуют интерактивное взаимодействие программы с пользователем. Существуют следующие обработчики событий:

1. `onclick(fun, btn, add)` - нажатие на клавишу мыши, где `fun` — вызываемая функция, `btn` — номер клавиши мыши (1 — левая, по умолчанию), `add` (значения `True` или `False`, если `True` (по умолчанию), то к кнопке будет добавлена новая привязка, иначе она заменяет прежнюю);
2. `onkey(fun, key)` - нажатие на клавишу клавиатуры, где `fun` — вызываемая функция, `key` — название клавиши, например, «space», «up», «a»;
3. `ontimer(fun, t)` - устанавливает таймер, где `fun` — вызываемая функция, `t` — число  $t \geq 0$ , время в миллисекундах, после истечения которого будет вызвана функция `fun`.

Все выше перечисленные обработчики событий являются членами класса `Screen()` (см. предыдущую работу).

*Примечание. Для класса Turtle также имеются обработчики событий, например, `onclick()` и `ondrag()`. Подробности об этих и других обработчиках вы можете узнать в документации по адресу: <https://docs.python.org/3/library/turtle.html>*

Если применяется обработчик события `onkey()`, то холст должен поддерживаться в фокусе (быть активным объектом). Для этого применяется метод `listen()` после каждого вызова обработчика.

2. Составим программу рисования окружности в указанном (кликом ЛКМ) месте холста. Добавим случайности к цвету, размеру и толщине пера. Определим объекты `t` и `window` (как в предыдущем

занятии). Для определения позиции начала рисования применяется, ранее вам неизвестный, метод `setpos(x, y)`.

### Программа 6.1

```
import turtle
from random import *

window = turtle.Screen()
window.title('Круги')
window.bgcolor('black')
window.setup(width=800, height=800)

t = turtle.Turtle()
t.speed(10)
t.hideturtle()

def cir(x, y):
    t.fillcolor(random(), random(), random())
    t.pencolor(random(), random(), random())
    t.pensize(randint(2, 8))
    t.up()
    t.setpos(x, y) # Фиксируем позицию черепахи
    t.down()
    t.begin_fill()
    t.circle(randint(10, 100))
    t.end_fill()

window.onclick(cir)

window.mainloop()
```

3. Составим программу рисования на холсте в произвольной (случайной) позиции окружности, треугольника или квадрата. Для обработки событий задействуем клавиши «left», «right» и «up». Обратите внимание, что аргумент-функция (в обработчике событий) передаётся по имени без аргументов (и круглых скобок).

## Программа 6.2

```
import turtle
from random import *

window = turtle.Screen()
window.title('Обработка событий')
window.setup(width=800, height=800)

t = turtle.Turtle()
t.speed(10)
t.hideturtle()

def envt(): # Для настроек черепахи
    t.fillcolor(random(), random(), random())
    t.pencolor(random(), random(), random())
    t.pensize(randint(2, 8))
    t.up()
    t.goto(randrange(-300, 300), randrange(-300, 300))
    t.down()

def cir(): # Рисование окружности
    envt()
    t.begin_fill()
    t.circle(randint(10, 100))
    t.end_fill()

def sqv(): # Рисование квадрата
    envt()
    d = randint(10, 100) # Будут разного размера
    t.begin_fill()
    for j in range(4):
        t.fd(d)
        t.lt(90)
    t.end_fill()

def tri(): # Рисование треугольника
```

```

envt()
d = randint(10, 100)
t.begin_fill()
for j in range(3):
    t.fd(d)
    t.lt(120)
t.end_fill()

```

```

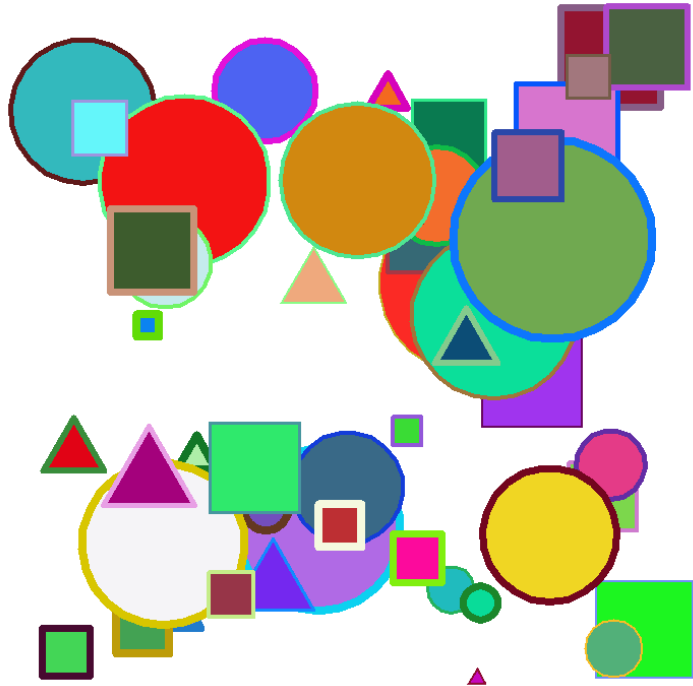
# Обработчики событий
window.onkey(tri,"Up")
window.listen()

window.onkey(sqv,"Left")
window.listen()

window.onkey(cir,"Right")
window.listen()

window.exitonclick()

```



4. Составьте программу рисования звезд с нечетным количеством лучей посредством обработчика событий нажатия на клавиши клавиатуры «5», «7» и «9».

*Примечание: Для рисования звезд с нечетным количеством лучей воспользуйтесь правилом для угла поворота черепахи:  $(180-360/(2*n))$ , где  $n$  — количество лучей.*

### Домашнее задание:

Составить программу использующую обработчик события клика (левой) кнопки мыши для рисования звезд (с нечетным числом лучей, см. выше) в определенном месте холста.