

Вопросы

Что такое конструктор?

- метод класса в объектно-ориентированном программировании, который используется для создания новых объектов этого класса
- метод класса, который используется для копирования объектов класса
- метод класса, который используется для сохранения объектов класса
- метод класса, который используется для удаления объектов класса

Что такое позиционные аргументы?

- любые аргументы класса
- аргументы, которые передаются в том же порядке, в котором они определены при создании функции
- аргументы, передаваемые в конструктор или метод класса по имени параметра
- числовые аргументы

Что такое self?

- интерфейс класса
- постоянный аргумент
- параметр особого класса в Python
- обязательный параметр, который передается во все методы класса в Python

Самостоятельная работа:

Создайте текстовую игру, в которой есть 2 разных класса персонажей и реализуется сражение между двумя бойцами данных классов (нанесение урона и порядок действий в бою).

Задание 1

- Создайте класс, который будет создавать случайное появление фигуры.
- Реализуйте очистку после нажатия на круг.

Задание 2

- Создайте класс «Человек», добавьте 3 метода к этому человеку.
- Создайте два класса-наследника (киберспортсмен, геймдизайнер) от класса «Человек». Добавьте в каждый из созданных классов еще по два метода.

```

import turtle
import random

# turtle speed to slowest
turtle.speed(2)
turtle.shape('turtle')

turtle.penup()
turtle.goto(0,330)
turtle.write("Кликните по Черепашке", move=False, align="center", font=("Comic Sans MS",30,))
turtle.penup()

turtle.goto(random.randint(-200, 200), random.randint(-200, 200))
turtle.pendown()

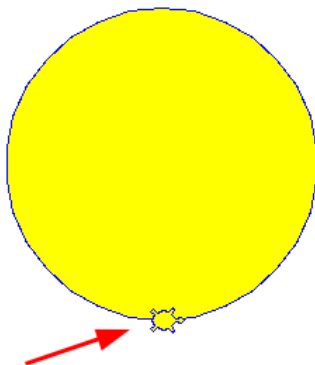
turtle.color('blue')
turtle.fillcolor('yellow')
turtle.begin_fill()
turtle.circle(random.randint(50, 200))
turtle.end_fill()

# method to action
def fxn(x, y):
    turtle.clear()

# allow user to click
# for some action
turtle.onclick(fxn)
turtle.done()

```

Кликните по Черепашке



С классом

```

import turtle
import random

```

```

turtle.speed(2) # turtle speed to slowest
turtle.shape('turtle')

turtle.penup()
turtle.goto(0, 330)
turtle.write("Кликните по Черепашке", move=False, align="center", font=("Comic Sans MS", 30,))
turtle.penup()

class Circle:
    def __init__(self):
        pass
    def draw(self):
        turtle.goto(random.randint(-200, 200), random.randint(-200, 200))
        turtle.pendown()
        turtle.color('blue')
        turtle.fillcolor('yellow')
        turtle.begin_fill()
        turtle.circle(random.randint(50, 200))
        turtle.end_fill()

# method to action
def fxn(x, y):
    turtle.clear()

a = Circle()
a.draw()
print(a)

turtle.onclick(fxn) # allow user to click for some action
turtle.done()

```

Как порисходит наследование в простом примере:

<https://younglinux.info/oopython/inheritance>

```

class Table:
    def __init__(self, l=10, w=10, h=10):
        self.length = l
        self.width = w
        self.height = h

class KitchenTable(Table):
    def __init__(self, l=1, w=1, h=0.7, p=4):
        Table.__init__(self, l, w, h)
        self.places = p

a = KitchenTable()
print(a.__dict__)

```

