PYTHON – TURTLE 07 – OBIEKTY – ZDARZENIA (22)

Obiekty

Typowe programowanie polega na tworzeniu funkcji (wykonują jakieś działania) i zmiennych (dostarczają danych funkcjom). Jednak funkcje i zmienne nie są ze sobą powiązane.
***Programowanie obiektowe** pozwala połączyć w jedną całość funkcje (nazywane **metodami**) i zmienne (nazywane **atrybutami**). W naszych przykładach posługujemy się jedynie niepełną i uproszczoną wersją programowania obiektowego.*
***Zdarzenie** są również elementem obiektów. Obsługa zdarzeń umożliwia pisanie programów interaktywnych i gier komputerowych. Uruchomiony program oczekuje zwykle na naciskane przez użytkownika klawisze lub przyciski myszki i wykonuje odpowiednie operacje.*

Biblioteka Turtle

*W typowych programach Python-Turtle bibliotekę importowano za pomocą: from turtle import **
Jeżeli posługujemy się obiektami należy zaimportować bibliotekę turtle za pomocą: import turtle

- Otwórz nowe okno poleceń
- **Zapisz plik na pulpicie**, nazwa: **ZDARZENIA - *Nazwisko Imię***
  UWAGA – ten plik również podlega ocenie

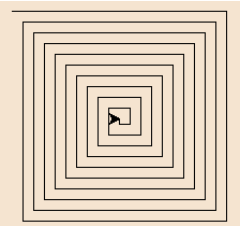
Obiekt żółw (2)

Po zaimportowaniu biblioteki nie działają zwykłe polecenia fd(), rt() itd. Należy je połączyć (kroką) z odpowiednim obiektem

- Przepisz i uruchom program:
  pomiń komentarze

```
import turtle                  # import biblioteki

T1=turtle.Turtle()             # tworzenie obiektu
T1.speed(0)                    # szybkość obiektu
for bok in range(200,0,-5):    # bok zmniejsza się co 5
    T1.fd(bok)                 # obiekt do przodu o bok
    T1.rt(90)                  # obiekt w prawo o 90
```



- Wklej do ramki zrzut ekranu z programem i rysunkiem

Drugi obiekt (2)

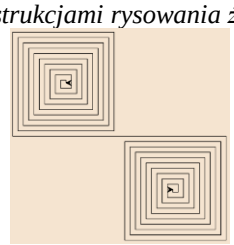
Definiujemy kolejny obiekt z podobnymi parametrami

- Przepisz instrukcje – definiowanie kolejnego żółwia:
  pod definicją żółwia T1, przed pętlą FOR

```
T2=turtle.Turtle()
T2.speed(0)
T2.rt(180)
```

Przepisz instrukcje – rysowanie spirali
w pętli FOR, pod instrukcjami rysowania żółwia T1

```
T2.fd(bok)
T2.rt(90)
```



- Uruchom program
  oba żółwie rysują jednocześnie
- Wklej do ramki zrzut ekranu z programem i rysunkiem

Trzeci i czwarty obiekt (2)

- Zdefiniuj jeszcze dwa obiekty-żółwie o nazwach T3 i T4
- Obiekt T3 obróć o 90 w lewo
- Obiekt T4 obróć o 90 w prawo
- Wstaw do pętli instrukcja rysowania obiektów T3 i T4
- Uruchom program
- Wklej do ramki zrzut ekranu z programem i rysunkiem

- **Wszystkie instrukcje usuń lub weź w komentarz**

TENIS **********

Program (2)

- Wklej do Python program z obu ramek

```
import turtle

ekr=turtle.Screen()
rys=turtle.Turtle()
pil=turtle.Turtle()
rak=turtle.Turtle()

rys.speed(0)
pil.speed(0)
rak.speed(0)
pil.shape('circle')
rak.shape('square')
rak.shapesize(4,0.5,1)
pil.pu()
rys.ht()
rak.pu()

L=-200
P=200
D=-200
G=200

def PROST(lew,pra,dol,gor):
    rys.pu();rys.goto(lew,gor);rys.pd()
    rys.goto(pra,gor)
    rys.goto(pra,dol)
    rys.goto(lew,dol)
    rys.goto(lew,gor)
    rys.pu()

rakX=P-30
rakY=0
rakD=40
rakS=5

def rakGOR():
    global rakY,rakS
    rakY=rakY+rakS
```

```
def rakDOL():
    global rakY,rakS
    rakY=rakY-rakS
def KONIEC():
    turtle.bye()

ekr.onkeypress(rakGOR,'Up')
ekr.onkeypress(rakDOL,'Down')
ekr.onkey(KONIEC,'Escape')
ekr.listen()

X=0
Y=0
dx=5
dy=5
licznik=0

PROST(L,P,D,G)

while True:

    X=X+dx
    Y=Y+dy
    if X<L: dx=-dx
    if X>P: X=L
    if Y<D: dy=-dy
    if Y>G: dy=-dy
    #czy uderza w rakietę
    if X>rakX:
        if Y>=rakY-rakD and Y<=rakY+rakD:
            dx=-dx
        else:
            licznik=licznik+1
            print(licznik)
            X=L
    rak.goto(rakX,rakY)
    pil.goto(X,Y)
```

Animacja piłki wygląda identycznie jak w przykładzie z poprzedniej lekcji, ale działamy na obiektach
EKR – okno żółwia w którym sprawdzamy klawiaturę
RYS – obszar w którym porusza się żółw – prostokąt narysowany za pomocą funkcji PROST
PIL – okrągła piłka
RAK – prostokątna rakieta tenisowa, narysowany za pomocą przeskalowanego prostokąta - rak.shapesize(4,0.5,1)
Rakietka porusza się tylko w górę i w dół – funkcje rakGORA i rakDOL
Jeżeli piłka osiągnie prawy koniec obszaru, to wylatuje z lewej strony X=L
Jeżeli piłka trafi w rakietkę (sprawdzamy poziomo i pionowo) `if X>rakX: if Y>=rakY-rakD and Y<=rakY+rakD:` *to zmienia się kierunek odbicia*

w przeciwnym razie na ekranie pojawia się kolejna liczba punktów i piłka wylatuje z lewej strony
Funkcje ONKEYPRES opisują reakcję na wciskanie klawiszy

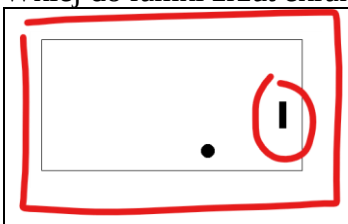
- Uruchom program
- Klawisze ↑ i ↓ sterują rakietką
- ESC – kończy działanie programu
- Wklej do ramki zrzut ekranu z programem i rysunkiem (**gdy trafiasz piłką w rakietkę**)

Kolory (2)

- Wstaw do programu instrukcję `pil.color('red')` aby kulka miała kolor czerwony
- Ustaw rakietkę w kolorze niebieskim
- Wklej do ramki zrzut ekranu z programem i rysunkiem

Rozmiar (2)

- Zmień kwadratowe boisko na prostokątne
- Zmniejsz długość rakiety
- Wklej do ramki zrzut ekranu z programem i rysunkiem



- **Usuń wszystkie instrukcje lub weź w komentarz**

RYSOWANIE **********

Program (2)

Na ekranie dwa żółwie, którymi sterujemy za pomocą klawiszy ze strzałkami. Spacja resetuje program.

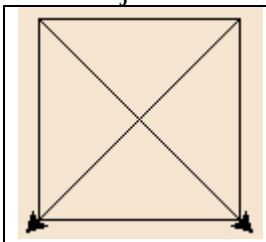
- **Wklej program** z ramki

```
import turtle
# tworzenie obiektów
EKR=turtle.Screen()
T1=turtle.Turtle()
T2=turtle.Turtle()
# definiowanie przesuwania i obrotu
def DoPrzodu():
    T1.fd(10)
    T2.fd(10)
def DoTylu():
    T1.bk(10)
    T2.bk(10)
def WLewo():
    T1.lt(15)
    T2.rt(15)
def WPrawo():
    T1.rt(15)
    T2.lt(15)
# resetowanie ustawień
def OdNowa():
    EKR.reset()
    T2.rt(180)
# przypisanie funkcji do zdarzeń klawiatury
EKR.onkey(DoPrzodu,'Up')
```

```
EKR.onkey(WLewo,'Left')
EKR.onkey(WPrawo,'Right')
EKR.onkey(DoTylu,'Down')
EKR.onkey(OdNowa,' ')
# na początku programu reset
OdNowa()
# uruchomienie nasłuchiwania zdarzeń
EKR.listen()
# program działa w nieskończoność
EKR.mainloop()
```

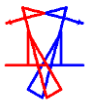
*Obiekt **EKR** jest ekranem, na którym rysujemy*
*Obiekty **T1** i **T2** są żółwiami.*
*Funkcje **DoPrzodu** i **DoTylu** przesuwają żółwia o 10 pikseli*
*Funkcje **WPrawo** i **Wlewo** obracają żółwie o 15° w różne kierunki*
*Funkcja **OdNowa** czyści ekran i ustawia żółwie w pozycjach początkowych (na przeciwko siebie)*
*Zdarzenie **onkey** przypisuje funkcję do naciśnięcia klawisza*
*Funkcja **listen** włącza nasłuchiwanie zdarzeń*
*Funkcja **mainloop** powoduje, że program nie kończy swojego działania*

- Uruchom program
- Klawisze ↑ i ↓ poruszają żółwiem co 10
- Klawisze ← i → obracają żółwiem co 15 stopni
- Spacja – resetuje program
- Spróbuj narysować kopertę
- Wklej do ramki zrzut ekranu z rysunkiem



Kolory (2)

- Do funkcji **OdNowa** wstaw polecenia:
  - zmień kolor jednego żółwia na niebieski a drugiego na czerwony – **color**()
  - zmień grubość rysowanych linii na 5 – **pensize**()
- Spróbuj narysować kolorowy „bazgrołek”
- Wklej do ramki zrzut ekranu z rysunkiem



ARMATA **********

Program (2)

Kursorami sterujemy armatą i symulujemy rzut ukośny

- **Wklej program** z ramki

```
import turtle
o=turtle.Screen()
a=turtle.Turtle()

X=-500
Y=-300
kat=45
lufa=50
```

```
def armata():
   a.reset()
   a.speed(0);a.pensize(10)
   a.pu();a.goto(X,Y);a.pd();a.ht()
   a.lt(kat)
   a.fd(lufa);a.bk(lufa)
def DoPrzodu():
   global lufa
   lufa=lufa+10; armata()
def DoTylu():
   global lufa
   lufa=lufa-10; armata()
def WLewo():
   global kat
   kat=kat+5; armata()
def WPrawo():
   global kat
   kat=kat-5; armata()

def Strzelaj():
   p=turtle.Turtle()
   p.speed(0)
   p.pu(); p.goto(X,Y)
   p.shape("circle")
   a.fd(lufa)
   dx=-(X-a.xcor())
   dy=-(Y-a.ycor())
   a.bk(lufa)
   x=X
   y=Y
   while y>=Y:
      x=x+dx
      y=y+dy
      dy=dy-9.81
      o.delay(50)
      p.goto(x,y)

o.onkey(DoPrzodu,'Up')
o.onkey(WLewo,'Left')
o.onkey(WPrawo,'Right')
o.onkey(DoTylu,'Down')

o.onkey(Strzelaj,' ')

armata()
o.listen()
o.mainloop()
```

*Początkowe ustawienie armaty opisują współrzędne **X, Y, LUFA, KAT***
*Funkcja **armata** wykonywana jest na początku programu i po każdej zmianie długości i kąta*
*Funkcje **DoPrzodu** i **DoTylu** zmieniają wielkość zmiennej **ARM** i rysują od nowa armatę*
*Funkcje **WPrawo** i **WLewo** zmieniają wielkość zmiennej **KAT** i rysują od nowa armatę*
*Jeżeli funkcje zmieniają wartości zmiennych, muszą być opisane za pomocą słowa **global***
*Funkcja **Strzelaj** definiuje nowy obiekt żółwia, który będzie w postaci koła i ustawiony w punkcie X, Y*
*W zmiennych **dx** i **dy** wyliczamy długości armaty na osiach X i Y – prędkość w poziomie i w pionie*
*W pętli **while**, która skończy się, gdy koło będzie poniżej poziomu **Y** wyliczamy*
*kolejne położenie **x** i **y***
nową prędkość pinową pomniejszoną o wartość przyspieszenia ziemskiego
koło jest przesuwane do nowego położenia

- Uruchom program
- Klawisze ← i → zmieniają kąt armaty
- Klawisze ↑ i ↓ zmieniają siłę armaty
- Spacja - strzał
- Wystrzel kilka razy
- Wykonaj zrzut ekranu z rysunkiem i wklej do ramki

Kolory (2)

- Zmień kolor armaty na niebieski
- Zmień kolor kuli na czerwony
- Wystrzel kilka razy
- Wykonaj zrzut ekranu z rysunkiem i wklej do ramki

Strzelamy (2)

- Ustaw armatę tak, aby kula trafiła dokładnie w koniec twojego okna
- Wykonaj zrzut ekranu z rysunkiem i wklej do ramki