Python Turtle

opracowanie Piotr Kotarski (na podstawie: https://docs.python.org/3/library/turtle.html#methods-of-turtlescreen-screen)

Spis treści

[Uwagi wstępne 1](#_Toc95430188)

[Turtle - metody 1](#_Toc95430189)

[Przesuwanie i rysowanie żółwia 1](#_Toc95430190)

[Stan żółwia 1](#_Toc95430191)

[Sterowanie piórem i wypełnieniem 2](#_Toc95430192)

[Wygląd żółwia 2](#_Toc95430193)

[Rysowanie tekstu 2](#_Toc95430194)

[Sterowanie animacją 2](#_Toc95430195)

[Instrukcje wejściowe 2](#_Toc95430196)

[Obsługa zdarzeń 2](#_Toc95430197)

[Ustawienia, konfiguracja 3](#_Toc95430198)

[Okno i ekran, obiekty 3](#_Toc95430199)

[Sterowanie programem 3](#_Toc95430200)

# Uwagi wstępne

oznaczenia parametrów:

n,m 🡪 liczby całkowite  
x,y,r 🡪 liczby rzeczywiste, współrzędne, rozmiary, promień  
-n,-x 🡪 liczba ujemna  
k, kąt 🡪 kąt w stopniach lub radianach (zależnie od ustawień)  
⭯, ⭮ 🡪 kierunek kątowy   
kolor 🡪 ustawienie koloru:  
 nazwa 🡪 "red"  
 kod HEX 🡪 "#FF88a2"  
 w zależności od colormode():  
 r,g,b 🡪 0.5, 0.9, 1 lub 128, 200, 255  
 (r,g,b) 🡪 (0.5, 0.9, 1) lub (128, 200, 255)  
nazwa 🡪 nazwa ustalona w bibliotece

# Przesuwanie i rysowanie żółwia

|  |  |
| --- | --- |
| fd(n)  forward(n) | przesuń do przodu o n |
| bk(n)  backward(n)  back(n) | przesuń do tyłu o n |
| rt(kąt)  right(kąt) | obróć o kąt w prawo |
| lt(kąt)  left(kąt) | obróć o kąt w lewo |
| goto(x,y)  setpos(x,y)  setposition(x,y) | przenieś do pozycji (x,y). nie zmieniaj orientacji żółwia. |
| setx(n) | przesuń w poziomie |
| sety(n) | przesuń w pionie |
| seth(kąt)  setheading() | ustaw kąt żółwia |
| home() | ustaw żółwia w stan początkowy |
| circle(r)  circle(-r)  circle(r,kąt)  circle(r,kąt,ile) | narysuj okrąg o promieniu r ⭯  narysuj okrąg o promieniu r ⭮  narysuj łuk (część okręgu)  jw. z podaniem ilości odcinków |
| dot()  dot(r)  dot(r,"kolor") | narysuj okrągłą kropkę  jw. o promieniu r  jw. w podanym kolorze |
| stamp() | zostaw stempel żółwia, zwraca numer stempla |
| clearstamp(n) | usuń stempel o numerze n |
| clearstamps()  clearstamps(n)  clearstamps(-n) | usuń wszystkie stemple  usuń pierwsze n stempli  usuń ostatnie n stempli |
| undo() | cofnij ostatnią akcję |
| begin\_poly() | rozpocznij rejestrowanie wierzchołków wielokąta. Obecna pozycja żółwia to pierwszy wierzchołek wielokąta. |
| end\_poly() | zatrzymaj rejestrowanie wierzchołków wielokąta. Bieżąca pozycja żółwia to ostatni wierzchołek wielokąta. Zostanie połączony z pierwszym wierzchołkiem. |
| get\_poly() | zwróć ostatnio zarejestrowany wielokąt |

# Rysowanie tekstu

|  |  |
| --- | --- |
| write(txt,  move=False,  align='left',  font='Arial', 8, 'normal') | napisz tekst txt na pozycji żółwia  move 🡪 czy przenieść żółwia  align 🡪 wyrównanie ("left","center","right")  font 🡪 czcionka: nazwa, rozmiar, typ |

# Stan żółwia

|  |  |
| --- | --- |
| st()  showturtle() | pokaż zółwia |
| ht()  hideturtle() | ukryj żółwia |
| isvisible() | czy żółw widoczny |
| pos()  position() | zwróć bieżącą pozycję żółwia (x,y) |
| towards(x,y) | zwróć kąt linii od żółwia do (x,y) |
| xcor() | zwróć współrządną x |
| ycor() | zwróć współrzędną y |
| heading() | zwróć bieżący kurs (kąt) |
| distance() | zwróć odległość od żółwia do (x,y) |

# Sterowanie piórem i wypełnieniem

|  |  |
| --- | --- |
| pd()  pendown(),down() | pisak w dół |
| pu()  penup(),up() | pisak w górę |
| width(x)  pensize(x) | ustaw rozmiar pióra |
| isdown() | czy pióro opuszczone |
| filling() | zwraca stan wypełnienia (True/False) |
| begin\_fill() | rozpoczęcie rysowania z wypełnieniem |
| end\_fill() | wypełnij kształt narysowany po ostatnim begin\_file() |
| color() | zwraca kolory pióra i wypełnienia |
| color(kolor) | ustaw kolor jednakowy dla pióra i wypełnienia |
| color(kolorp,kolorf) | ustaw kolory dla pióra i wypełnienia |
| pencolor(kolor) | jw., tylko kolor pióra |
| fillcolor(kolor) | jw., tylko kolor wypełnienia |

# Wygląd żółwia

|  |  |
| --- | --- |
| shape()  shape(nazwa=None) | zwróć nazwę kształtu żółwia  Ustaw kształt żółwia ("arrow”, “turtle”, “circle”, “square”, “triangle”, “classic”) |
| resizemode()  resizemode(tryb) | zwróć tryb zmiany rozmiaru żółwia  ustaw tryb zmiany rozmiaru żółwia (“auto”, “user”, “noresize”) |
| shapesize(  stretch\_wid=None,  stretch\_len=None, outline=None)  turtlesize() | zmiana rozmiaru żółwia  stretch\_wid – rozciąganie po szerokości stretch\_len – rozciąganie po szerokości zarys – szerokość konturu |
| shearfactor(wsp) | ustaw wsp. ścinania kształtu żółwia. Jest to tg kąta ścinania |
| tilt() | obróć kształt żółwia o kąt |
| settiltangle() | obróć kształt żółwia do podanego kąta |
| tiltangle() | ustaw lub zwróć aktualny kąt nachylenia |
| shapetransform(t11,t12,t21,t22) | ustaw lub zwróć bieżącą macierz transformacji kształtu żółwia (t11,t12,t21,t22), t11\*t22-t12\*t21<>0 |
| get\_shapepoly() | zwróć bieżący wielokąt kształtu jako krotkę par współrzędnych. Można to wykorzystać do zdefiniowania nowego kształtu lub składników kształtu złożonego. |

# Sterowanie animacją

|  |  |
| --- | --- |
| **Metoda** | **opis** |
| speed(n) | ustaw szybkość  0 - bez opóźnień, najszybciej  1🡪10: wolno 🡪 szybko |
| delay(czas) | ustaw lub zwróć opóźnienie rysowania w ms. (Jest to w przybliżeniu odstęp czasu między dwiema kolejnymi aktualizacjami kanwy) |
| tracer(n=None,  delay=None) | włącz/wyłącz animację żółwia i ustaw opóźnienie aktualizacji rysunków.  Jeśli podano n to wykonywana jest tylko każda n-ta regularna aktualizacja ekranu.  Po wywołaniu bez argumentów, zwraca aktualnie przechowywaną wartość n.  delay - patrz delay() |
| update() | aktualizuj TurtleScreen. Stosowane, gdy tracer wyłączony. |

# Instrukcje wejściowe

|  |  |
| --- | --- |
| textinput(tytuł, monit) | zwraca tekst wprowadzony w oknie dialogowym |
| numinput(tytuł, monit, default=None, min=None, max=None) | zwraca liczbę wprowadzoną w oknie dialogowym |

# Obsługa zdarzeń

|  |  |
| --- | --- |
| mainloop()  done() | rozpoczyna pętlę zdarzeń.  Musi być ostatnią instrukcją w programie. |
| listen() | ustaw fokus na TurtleScreen |
| onclick(fun, btn=1, add=None) | ustawienie obsługi kliknięcia,  fun - funkcja z dwoma argumentami: fun(x,y)  btn - 1-LPM  add=True - dodanie nowej obsługi, False - zastąpienie istniejącej |
| onclick(None) | usunięcie obsługi kliknięcia |
| onrelease() | zwolnienie przycisku myszy, parametry jw |
| ondrag() | przeciąganie myszą, parametry jw |
| onkey(fun,"klawisz")  onkeyrelease() | ustaw zwolnienie klawisza  fun - funkcja bezargumentowa lub None (usunięcie)  klawisz - znak lub symbol klawisza |
| onkeypress() | ustaw naciśnięcie klawisza, parametry jw |
| ontimer(fun,czas) | cykliczne uruchamianie funkcji  fun - funkcja bezargumentowa  czas w ms |
| exitonclick() | powiąż metodę bye() z kliknięciami myszy na ekranie |

# Ustawienia, konfiguracja

|  |  |
| --- | --- |
| degrees(kąt=360) | ustaw jednostkę miary kątów |
| radians() | ustaw jednostki kąta radian |
| mode(tryb) | ustaw tryb żółwia (kurs początkowy i kierunek dodatni kąta)  “standard” - 🡒⭯  “logo” - 🡑⭮  “world” - własne ustawienia |
| pen(par=None) | odczytaj lub ustaw parametry stanu pióra  par - słownik zawierające kilka lub wszystkie parametry:  “shown”: True/False  “pendown”: True/False  “pencolor”: kolor  “fillcolor”: kolor  “pensize”: x  “speed”: 0, 1..10  “resizemode”: “auto” or “user” or “noresize”  “stretchfactor”: (n, m)  “outline”: n  “tilt”: kąt |
| colormode(tryb) | ustaw tryb kodowania koloru  tryb – jedna z wartości 1 lub 255 |
| setundobuffer(n) | ustaw rozmiar bufora dla undo |
| undobufferentries() | zwróć rozmiar bufora |
| getshapes() | zwróć nazwy dostępnych kształtów żółwia |
| addshape("plik")  addshape("nazwa",kształt)  addshape("nazwa",obiekt)  register\_shape(…) | rejestracja nowego kształtu żółwia  plik - graficzny, nie obraca się  nazwa - zarejestrowana nazwa kształtu  kształt - krotka par współrzędnych  obiekt - typu Shape |
| turtles() | zwróć listę żółwi na ekranie |

# Okno i ekran, obiekty

|  |  |
| --- | --- |
| title("tytuł") | ustaw tytuł okna |
| window\_height() | zwróć wysokość okna |
| window\_width() | zwróć szerokość okna |
| bgcolor() | zwróć kolor tła |
| bgcolor(kolor) | ustaw kolor tła |
| bgpic(plik) | ustaw tło obrazkowe, nopic - usuń |
| screensize(dx,dy,  kolor=None) | ustaw rozmiary płótna (i tło) |
| clone() | utwórz i zwróć klon żółwia o tych samych parametrach |
| getturtle()  getpen() | zwraca własny obiekt żółwia |
| getscreen() | zwróć obiekt TurtleScreen, na którym rysuje żółw. |
| getcanvas() | zwróć płótno należące do TurtleScreen |
| setworldcoordinates  (x1,y1,x2,y2) | ustaw współrzędne użytkownika, włącza tryb "world".  x1,y1 - lewy dolny róg  x2,y2 - prawy górny róg  Kąty mogą być zniekształcone |
| setup(  width=…,  height=…,  startx=…,  starty=…) | ustaw rozmiar i położenie okna głównego.  Domyślne wartości w pliku turtle.cfg.  width, height: domyślnie 50% i 70% ekranu  – liczba całkowita - rozmiar w pikselach,  – liczba rzeczywista - ułamek ekranu,  startx, starty: współrzędne pozycji początkowej (home)  – jeśli >0, w px od lewej/górnej krawędzi ekranu,  – jeśli <0, w px od prawej/dolnej krawędzi ekranu,  – jeśli NONE, wyśrodkowanie |

# Sterowanie programem

|  |  |
| --- | --- |
| clearscreen() | kasuj ekran i wszystkie żółwie |
| resetscreen() | resetuj wszystkie żółwie do ich stanu początkowego |
| reset() | usuń rysunki żółwia z ekranu, ponownie wyśrodkuj żółwia i ustaw zmienne na wartości domyślne. |
| clear() | usuń rysunki żółwia z ekranu. Nie ruszaj żółwia. Nie ma to wpływu na stan i pozycję żółwia oraz rysunki innych żółwi. |
| bye() | zamknij okno programu |