

# ВИКОРИСТАННЯ АЛГОРИТМІВ РОЗВ'ЯЗУВАННЯ ГРАФІЧНИХ ЗАДАЧ ЗАСОБАМИ PYTHON

*Таганова Діана Василівна*

*студентка 2-го курсу*

*МДПУ ім. Б. Хмельницького*

*Чорна Альона Віталіївна*

*асистент кафедри інформатики і кібернетики*

*МДПУ ім. Б. Хмельницького*

**Анотація.** В статті розглянуті основні принципи та методи побудови примітивних графічних елементів засобами Python. Визначені особливості використання черепашкової графіки. Наведені приклади побудови різноманітних геометричних фігур.

**Ключові слова:** мова програмування Python, графічні фігури, черепашкова графіка, графічний модуль.

Одією із сучасних мов програмування, що набула популярності стала мова Python (за даними українського IT-порталу DOU.UA) [3]. Python є об'єктно-орієнтованою мовою програмування високого рівня. Мова має підтримку процедурну, функціональну, аспектно-орієнтовану та об'єктно-орієнтовану парадигму програмування. Мова програмування Python придатна вирішити велике коло задач, зокрема графічних [2, с. 114].

В Україні існує низка шкіл, які навчають основи програмування Python. До безкоштовних належать: «Code Club UA» знаходиться майже у всіх областях України; «IT2School» знаходиться у Одесі і підходить дітям від 7 до 18 років.

До платних шкіл і курсів відносяться: «BroBots – Brovary IT Hub» знаходиться у Броварах; «Inforpulse Univer» - курси від компанії «Інфопульс Україна»; «IT Univer» - Харківський клуб; «Robo House» ця школа має 14 філій у Києві та 1 в Одесі; «Мала комп'ютерна академія «Крок»» - це одна з найбільших академій де навчається більше 20 000 студентів.

Мова Python використовує так звану черепащачу графіку для побудови нескладних фігур, контурів ліній. Черепашка у Python - це стрілочка в центрі вікна, так, на справжню черепаху вона не дуже схожа. Рухом черепашки можна управляти, даючи їй різні команди [1, с. 49].

Найважливіший інструмент у Python який імітує «черепащачу графіку», створений спеціально для навчання модуль *turtle*.

Наведемо приклад використання черепашої графіки засобами Python. Він у нас буде називатися SquareSpiral (Квадратна спіраль).

*#SquareSpiral - Квадратна спіраль*

```
import turtle
```

```
t = turtle.Pen()
```

```
for x in range(100):
```

```
    t.forward(x)
```

```
    t.left(90)
```

Розберемо зміст коду:

# - означає коментар у коді. Коментарі пишуться, щоб зробити якусь замітку, але компілятори і інтерпретатори ігнорують його. Ми у даному випадку написали, те що повинна зробити програма.

*1 рядок* - Підключаємо модуль *turtle*, який дозволяє малювати на екрані нескладні малюнки.

*2 рядок* - Спростимо написання черепашки, щоб не прописувати *turtle.Pen*, його можна коротко прописати як *t*.

*3 рядок* - Створюємо цикл і вказуємо скільки разів він буде повторюватися. Буква *x* називається змінною, але дітям її можна назвати "невідомою" (Змінна зберігає значення, яке може змінюватися в міру виконання програми).

Прописуємо дії в циклі. Які будуть повторно виконуватися до тих пір, поки програма не прорахує число задане змінною *x*.

*4 рядок* - *forward* вказує початок циклу і повідомляє ручці черепашки пересунути на *x* точок на екрані. Так як *x* дорівнює 0, ручка не переміститься.

Далі комп'ютер додає 1, щоб пересунути змінну  $x$  на наступну позицію в діапазоні, і, так як 1 все ще знаходиться в діапазоні від 0 до 99, цикл продовжує виконання. Тепер змінна  $x$  дорівнює 1, тому ручка пересувається вперед на 1 точку, а потім виконує наступну дію.

*5 рядок* - Вказуємо поворот наліво кожні 90 градусів.

Результатом роботи стане побудована спіраль на екрані. Змінюємо одне з чисел в програмі для створення перекрученого квадрату. Для цього внесемо зміни в останній рядок програми, у *t.left* змінимо 90 на 91, збережемо під іншою назвою, запусимо і побачимо, що наш квадрат став перекрученим, тобто спіральним (рис. 1).

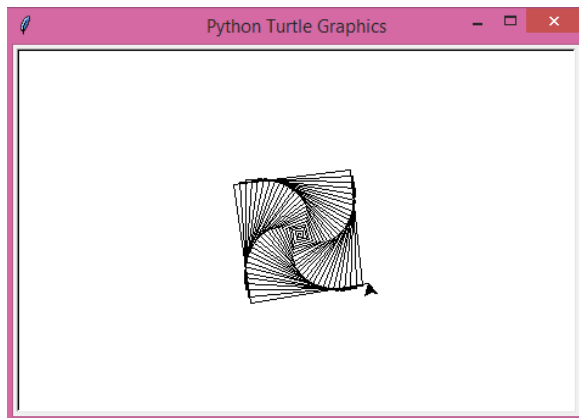


Рис. 1. Квадратна спіраль

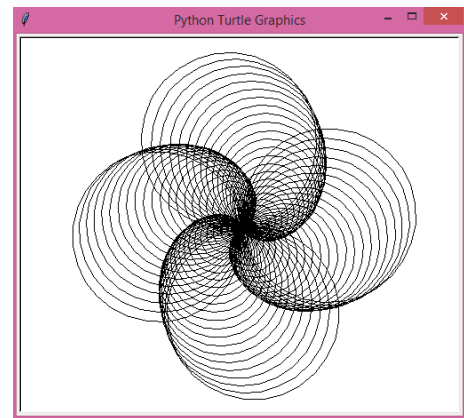


Рис. 2. Кругова спіраль

До інших можливостей бібліотеки *turtle* є створення кругової спіралі. Змінюємо рядок переміщення черепашки *t.forward(x)* на *t.circle(x)* (промалювати окружність з радіусом  $x$ ) (рис. 2).

*# Circular Spilar - Кругова спіраль*

```
import turtle
```

```
t = turtle.Pen()
```

```
for x in range(100):
```

```
    t.circle(x)
```

```
    t.left(91)
```

Для добавлення кольору після рядка *t = turtle.Pen()*, щоб наш малюнок повністю був тільки в одному кольорі, необхідно прописати *t.pencolor ("blue")*, після цього ручка черепашки буде синього кольору.

Колір можна прописати який завгодно, головною умовою – він має бути вказаний в лапках. Для вказування системи кольорів будемо використовувати шістнадцятковою системою. Колір у зазначеній системі можна поставити абсолютно будь-який, вибравши його в палітрі фотошоп, наприклад `#fbbc05`. Для того щоб черепашка змінювала колір кожного разу, коли програма проходить цикл промальовування слід перенести функцію `t.pencolor()` в групу дій, що знаходяться під циклом `for`. У середині цієї команди варто прописати параметр `(colors [x% 4])`, де `x` – та ж сама змінна, яку ми використовували і в інших програмах і вона так само буде збільшуватися. Висловлювання в квадратних дужках вказує, що буде використовуватися перші 4 кольорів списку, і те що при кожній зміні значення `x` варто вибрати один з кольорів. Символ `%` робить операцію взяття по модулю, результат якого буде залишок від ділення. Наведемо лістинг коду кольорової квадратної спіралі.

*#ColorSquareSpiral – Кольорова квадратна спіраль*

```
import turtle
t = turtle.Pen()
colors = ["blue", "yellow", "red", "green"]
for x in range(100):
    t.pencolor(colors[x%4])
    t.forward(x)
    t.left(91)
```

Так за весь цикл 100 кроків вираз `colors[x% 4]` пройшло за всіма чотирма кольорами 25 разів (рис. 3).

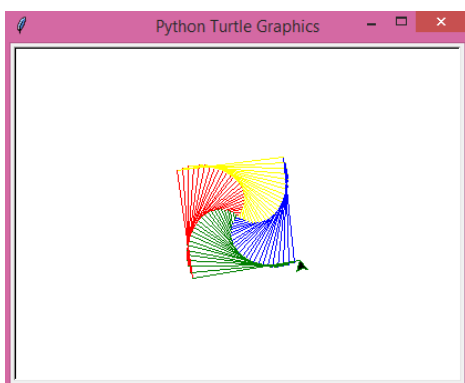


Рис. 3. Кольорова спіраль

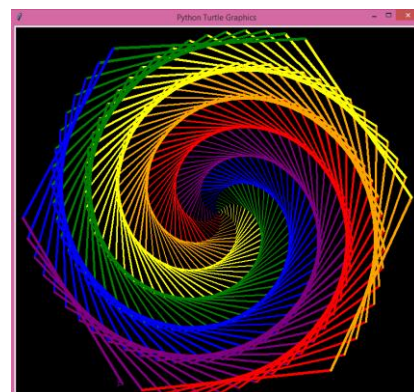


Рис. 4. Незвичайна квадратна спіраль

Для зміни фону зображення дописуємо рядок з кодом `turtle.bgcolor ("black")`. До інших популярних функцій можна віднести зміну кількості сторін фігури (функція `sides`); зміну товщини лінії фігури (`t.width()`); зміну кута фігури (`t.left`) та ін. Використовуючи зазначені функції можна побудувати незвичайну кольорову спіраль (рис. 4).

```
#NiceSpiral - Кольорова спіраль на чорному фоні  
  
import turtle  
  
t = turtle.Pen()  
  
turtle.bgcolor("black")  
  
colors = ["red", "orange", "yellow", "green", "blue", "purple"]  
  
for x in range(360):  
    t.pencolor(colors[x%6])  
    t.width(x/100 + 1)  
    t.forward(x)  
    t.left(59)
```

Таким чином, використовуючи різноманітні графічні функції мови Python можна створювати прості графічні фігури (квадрат, коло, спіраль), змінювати колір, характер побудови.

### *Література*

1. Бриггс Джейсон Python для детей. Самоучитель по программированию / Джейсон Бриггс; пер. с англ. Станислава Ломакина; [науч. ред. Д. Абрамова]. — М. : Манн, Иванов и Фербер, 2017. — 320 с.

2. Осадча К. П. Аналіз методів розробки алгоритмів розв'язання математичних задач засобами мови Python / К.П Осадча, О.В Хромишев // Системи обробки інформації: збірник наукових праць. — Х.: Харківський університет Повітряних Сил імені Івана Кожедуба, 2016. — № 2(139). — С. 114-117.

3. Рейтинг языков программирования № 6: новые лошадки на коммерческом рынке [Електронний ресурс]. — Режим доступу до ресурсу: URL: <http://dou.ua/lenta/articles/language-rating-jan-2015>.