

Abstract: *The concept of "Black box" in the information analysis of systems is considered, the approaches and methods in the study of complex systems, where the internal mechanisms remain unknown or inaccessible for direct analysis, are analyzed. Attention is focused on the interaction between the input and output processes of the system, ignoring the internal structure.*

Keywords: *Black box, system, software, information analysis.*

Науковий керівник: Жебко О.О.,
*асистент кафедри економічної кібернетики,
комп'ютерних наук та інформаційних технологій,
Миколаївський національний аграрний університет
м. Миколаїв, Україна*

УДК 004.04

ВИКОРИСТАННЯ МОДУЛЯ TURTLE ДЛЯ СТВОРЕННЯ ГРАФІКИ ТА АНІМАЦІЇ В PYTHON

Русанов Юрій Олександрович

здобувач вищої освіти спеціальності 122 «Комп'ютерні науки»
Миколаївський національний аграрний університет
м. Миколаїв, Україна

Анотація: *Дослідження присвячено аналізу можливостей модуля Turtle для створення графіки та анімації у мові програмування Python. Вона охоплює історію розвитку модуля, його основні функції та методи, а також демонструє практичний приклад використання. Доповідь спрямована на підтримку освітнього процесу в галузі комп'ютерних наук і програмування, а також надає рекомендації щодо ефективного використання модуля для розвитку навичок у візуалізації даних і комп'ютерній графіці.*

Ключові слова: *Python, модуль Turtle, графіка, анімація, програмування.*

Модуль Turtle у Python дозволяє користувачам виконувати команди, які переміщують "черепашку" по екрану, залишаючи за собою лінії, щоб створити графічне зображення. Користувачі можуть контролювати напрямок, колір, товщину ліній та інші параметри, щоб створити різноманітні візуальні ефекти. Також модуль підтримує розробку простих ігор та інтерактивних проектів, що робить його ідеальним інструментом для освітніх цілей.

Для демонстрації можливостей модуля Turtle наведемо приклад коду, який ілюструє створення базових графічних фігур, кольорових малюнків і простих анімацій. Цей приклад може слугувати як вихідний пункт для розробки власних проектів та дослідження можливостей модуля.

Код створює анімаційний ефект, де дві черепашки обертаються навколо центральної точки, кожна з них зафарбована різним кольором. Давайте розберемо кожну частину коду докладніше:

*from turtle import **. Імпортуємо всі символи (функції, класи, змінні) з графічної бібліотеки *turtle*, роблячи їх доступними без префіксу *turtle*.

from time import sleep. Імпортуємо функцію *sleep* з модуля *time*, що дозволяє вам призупинити виконання програми на певний час.

bgcolor("black"). Встановлюємо колір фону вікна *turtle* у чорний.

t = [Turtle(), Turtle()]. Створюємо список *t*, що містить два об'єкти *Turtle* з бібліотеки *turtle*. Ці об'єкти будуть використовуватися для малювання фігур на екрані.

x = 6. Визначаємо змінну *x* і присвоюємо їй значення 6. Це значення буде використовуватися пізніше для контролю розміру фігур.

colors = ["red", "yellow", "blue", "lime"]. Створюємо список *colors*, що містить чотири назви кольорів: *"red"*, *"yellow"*, *"blue"*, *"lime"*. Ці кольори будуть використовуватися для малювання спіральних візерунків.

Цикл *for* ітерує список *t* (що містить дві черепахи):

i.speed(0). Встановлює швидкість малювання поточної черепахи (*i*) на максимальну (*0*).

i.color("white"). Встановлює колір пера поточної черепахи (*i*) у білий.

i.shape("circle"). Встановлює форму поточної черепахи (*i*) у коло.

i.shapesize(0.3). Встановлює розмір кола для поточної черепахи (*i*) у 0,3 від стандартного.

i.width(3). Встановлює товщину лінії пера, що використовується поточною черепахою (*i*), у 3 пікселі.

i.penup(). Піднімає перо поточної черепахи (*i*), щоб вона не малювала при переміщенні.

i.seth(90). Встановлює напрямок (*heading*) поточної черепахи (*i*) у 90 градусів (вгору).

i.fd(350). Переміщує поточну черепаху (*i*) вперед на 350 пікселів.

i.seth(-180). Встановлює напрямок поточної черепахи (*i*) у -180 градусів (вниз).

i.pendown(). Опускає перо поточної черепахи (*i*), щоб вона почала малювати.

t[0].penup(). Піднімає перо першої черепахи (*t[0]*).

delay(0). Вимикає анімацію між кроками.

speed(0). Встановлює загальну швидкість малювання на максимальну (*0*).

ht(). Скриває самі черепахи, залишаючи лише лінії.

sleep(4). Пауза на 4 секунди перед початком анімації.

Цикл *for* ітерує список *colors*:

color(i). Встановлює колір пера для поточної ітерації у колір (*i*) з списку *colors*.

Цикл *for* виконується 360 разів (по одному на градус).

t[0].fd(x). Переміщує першу черепаху (*t[0]*) вперед на *x* (6 пікселів), малюючи лінію.

t[0].lt(1). Повертає першу черепаху.

t[0].lt(1). Повертає першу черепаху (*t[0]*) ліворуч на 1 градус на кожному кроці. Це створює ефект спіралі, що поступово звужується.

`pu()`. Піднімає перо першої черепахи (`t[0]`), щоб вона не залишала слідів під час переміщення до наступної позиції.

`goto(t[0].pos())`. Переміщує другу черепаху (`t[1]`) у поточну позицію першої черепахи (`t[0]`). Це встановлює точку відліку для малювання спіралі другою черепахою.

`pd()`. Опускає перо другої черепахи (`t[1]`), щоб вона почала малювати свою спіраль.

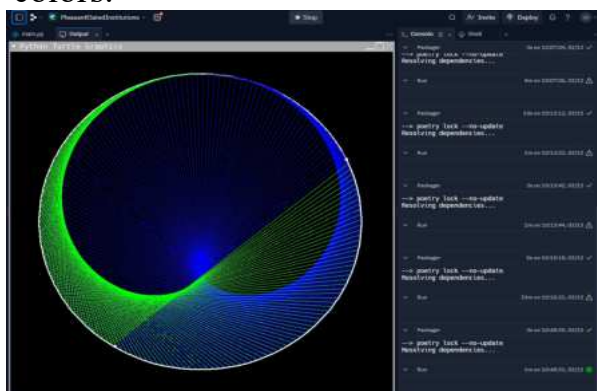
`t[1].fd(2 * x)`. Переміщує другу черепаху (`t[1]`) вперед на $2 * x$ (12 пікселів), малюючи лінію. Оскільки друга черепаха повертається на більший кут (2 градуси), її спіраль буде більш розширена, ніж спіраль першої черепахи.

`t[1].lt(2)`. Повертає другу черепаху (`t[1]`) ліворуч на 2 градуси на кожному кроці. Це створює ефект спіралі, що розширюється.

`goto(t[1].pos())`. Переміщує першу черепаху (`t[0]`) у поточну позицію другої черепахи (`t[1]`). Це встановлює точку відліку для наступного кола спіралі першої черепахи.

Завершення програми: `done()`: Ця функція завершує роботу вікна `turtle` і зупиняє програму.

Після виконання цього коду ми побачимо на екрані анімацію двох спіралей, що обертаються одна навколо одної. Перша спіраль буде звужуватися до центру, а друга – розширюватися назовні. Колір спіралей буде змінюватися щоразу після завершення повного оберту на 360 градусів, перебираючи кольори з визначеного списку `colors`.



Цей код демонструє основні принципи роботи з графічною бібліотекою `turtle` в Python. За допомогою простих команд можна створювати цікаві візуальні ефекти.

Список використаних джерел:

1. Основи програмування: методичні вказівки до виконання комп'ютерних практикумів з дисципліни «Основи програмування». Основи програмування мовою Python. / Уклад.: А. В. Яковенко. К.: НТУУ «КПІ ім. І. Сікорського», 2017. 87 с.

2. Костюченко А.О. Основи програмування мовою Python. Чернігів : ФОП Баликіна С.М., 2020. 180 с.

Abstract: The study is devoted to the analysis of the capabilities of the Turtle module for creating graphics and animation in the Python programming language. It covers the development history of the module, its main functions and methods, and

also demonstrates a practical example of use. The report aims to support the educational process in the field of computer science and programming, and also provides recommendations for the effective use of the module to develop skills in data visualization and computer graphics.

Keywords: Python, Turtle module, graphics, animation, programming.

Науковий керівник: Тищенко С.І.,

*канд. пед., наук, доцент, доцент кафедри економічної кібернетики,
комп'ютерних наук та інформаційних технологій,
Миколаївський національний аграрний університет
м. Миколаїв, Україна*

УДК 004.04

РОЗРОБКА НАЙПРОСТІШОГО КАЛЬКУЛЯТОРА В PYTHON

Сиротенко Віктор Дмитрович,

*здобувач вищої освіти спеціальності 122 «Комп'ютерні науки»
Миколаївський національний аграрний університет
м. Миколаїв, Україна*

Анотація: *У цій науковій доповіді розглядається процес розробки найпростішого калькулятора з використанням мови програмування Python. Акцент робиться на поясненні кожного кроку в процесі створення програми, забезпеченні зрозумілого коду для початківців і представленні основних концепцій програмування, таких як змінні, функції та обробка виключень. Крім того, наводяться приклади коду з детальними коментарями, щоб полегшити розуміння. Ця доповідь може бути корисною як для новачків, які хочуть навчитися основам програмування на Python, так і для досвідчених розробників, які шукають чітке та зрозуміле пояснення створення простого калькулятора.*

Ключові слова: *Python, калькулятор, програмування, змінні, функції, обробка виключень, код, коментарі.*

Програмування – це захоплюючий і творчий процес, який дозволяє створювати різноманітні програмні рішення для вирішення повсякденних проблем. Одним з найпопулярніших завдань для початківців у програмуванні є створення калькулятора. Хоча це може здатися простим завданням, воно насправді дозволяє ознайомитися з багатьма основними концепціями програмування, такими як змінні, функції, обробка виключень та взаємодія з користувачем.