Podstawy Programowania Obiektowego w Pythonie

Wprowadzenie do OOP

Programowanie Obiektowe (OOP) to paradygmat programowania, który skupia się na tworzeniu obiektów,
czyli instancji klas. Klasy definiują strukturę i zachowanie obiektów. Programowanie obiektowe pozwala
na lepsze zorganizowanie i modularność kodu.

Podstawowe pojęcia

1. Klasa: Szablon, według którego tworzone są obiekty. Klasa może zawierać atrybuty (dane) oraz metody (funkcje).
2. Obiekt: Instancja klasy, która posiada wszystkie jej atrybuty i metody.
3. Atrybuty: Zmienne należące do klasy lub obiektu, które przechowują dane.
4. Metody: Funkcje należące do klasy lub obiektu, które określają zachowanie obiektu.

Tworzenie klasy

W Pythonie klasa definiowana jest za pomocą słowa kluczowego 'class'. Przykładowa klasa może wyglądać tak:

```
class Pies:
    def __init__(self, imie, rasa):
        self.imie = imie
        self.rasa = rasa

    def szczekaj(self):
        print(f"{self.imie} szczeka!")
```

Metoda __init__ jest konstruktorem, który pozwala zainicjalizować atrybuty obiektu, takie

jak 'imie' oraz 'rasa'.

Tworzenie obiektu

Aby utworzyć obiekt z klasy, wystarczy wywołać klasę z odpowiednimi argumentami. Przykład:

```
mój_pies = Pies("Reksio", "Owczarek")
mój_pies.szczekaj()   # Wynik: Reksio szczeka!
```

Atrybuty i metody

Obiekty mogą mieć atrybuty oraz metody. Atrybuty przechowują dane, a metody definiują zachowania obiektów. W powyższym przykładzie 'imie' i 'rasa' to atrybuty, a 'szczekaj()' to metoda.

Dziedziczenie

Dziedziczenie pozwala tworzyć nowe klasy bazujące na istniejących klasach. Klasa dziedzicząca otrzymuje wszystkie atrybuty i metody klasy bazowej, a także może definiować swoje własne.

Przykład:

```
class PiesObronny(Pies):
  def strzez(self):
    print(f"{self.imie} strzeże!")
```

Klasa PiesObronny dziedziczy po klasie Pies, co oznacza, że posiada wszystkie jej atrybuty i metody, a także wprowadza nową metodę 'strzez()'.

Zakończenie

Programowanie obiektowe to potężny paradygmat, który pozwala na tworzenie bardziej modularnego, elastycznego i czytelnego kodu. Dzięki zrozumieniu podstawowych pojęć, takich jak klasy, obiekty, atrybuty, metody oraz dziedziczenie, można łatwo zarządzać bardziej złożonymi aplikacjami.