

Tytuł ćwiczenia: Jednostka 4 - Paradygmat obiektowy (cz. 1)
Autor: Joanna Dagil
Grupa: TCH-1
Data: 28.10.2025

1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z zasadami paradygmatu obiektowego.

2 Przebieg ćwiczenia

W tym ćwiczeniu posłużę się językiem Python w edytorze Visual Studio Code. W celu wykonania zadań zapoznaję się z tutorialami, a następnie przechodzę do stworzenia plików:

student.py

```
1 class Student:
2     def __init__(self, imie, nazwisko, oceny):
3         self.imie = imie
4         self.nazwisko = nazwisko
5         self.oceny = oceny
6
7     def dodaj_ocene(self, ocena):
8         self.oceny.append(ocena)
9     def srednia(self):
10        return sum(self.oceny) / len(self.oceny) if self.oceny else 0
11    def info(self):
12        print(f"Student {self.imie} {self.nazwisko} ma średnia: {self.srednia()}")
13
14
15 if __name__ == "__main__":
16     s1 = Student("Aaa", "Aaaaa", [60, 70, 95])
17     s2 = Student("Bbb", "Bbbbbb", [40, 10])
18     s3 = Student("Ccc", "Ccccc", [60, 70, 70, 60])
19
20     s1.info()
21     s2.info()
22     s3.info()
23
24     s3.dodaj_ocene(85)
25     s3.info()
```

prostokat.py

```
1 class Prostokat:
2     def __init__(self, szerokosc, wysokosc):
3         self.szerokosc = szerokosc
4         self.wysokosc = wysokosc
5
6     def pole(self):
7         return self.szerokosc * self.wysokosc
8     def obwod(self):
9         return 2 * (self.szerokosc + self.wysokosc)
10    def skaluj(self, faktor):
11        self.szerokosc *= faktor
12        self.wysokosc *= faktor
13
14 if __name__ == "__main__":
15     s1 = Prostokat(3, 4)
16     s2 = Prostokat(5, 6)
17
18     print("Prostokąt 1 - pole:", s1.pole(), "obwód:", s1.obwod())
19     print("Prostokąt 2 - pole:", s2.pole(), "obwód:", s2.obwod())
20
21     s1.skaluj(2)
22     print("Po skalowaniu Prostokąt 1 - pole:", s1.pole(), "obwód:", s1.obwod())
```

bank.py

```
1 class BankKonto:
2     def __init__(self, nr, saldo=0.0):
3         self.numer_konta = nr
4         self.saldo = saldo
5
6     def wplac(self, kwota):
7         if kwota > 0:
8             self.saldo += kwota
9             return True
10        return False
11
12    def wyplac(self, kwota):
13        if 0 < kwota <= self.saldo:
14            self.saldo -= kwota
15            return True
16        return False
17
18    def saldo(self):
19        return self.saldo
20
21 if __name__ == "__main__":
22     konto = BankKonto("1234", 1000.0)
23     print("Saldo początkowe:", konto.saldo)
24     konto.wplac(500.0)
25     print("Saldo po wpłacie 500.0:", konto.saldo)
26     konto.wyplac(200.0)
27     print("Saldo po wypłacie 200.0:", konto.saldo)
28     sukces = konto.wyplac(2000.0)
29     print("Próba wypłaty 2000.0:", "Sukces" if sukces else "Niepowodzenie")
30     print("Saldo końcowe:", konto.saldo)
```

Program uruchamiam poleceniem

```
1 python {nazwa}.py
```

3 Wyniki działania

Dla student.py:

```
1 Student Aaa Aaaaa ma średnia: 75.0
2 Student Bbb Bbbbbb ma średnia: 25.0
3 Student Ccc Ccccc ma średnia: 65.0
4 Student Ccc Ccccc ma średnia: 69.0
```

Dla prostokat.py:

```
1 Prostokąt 1 - pole: 12 obwód: 14
2 Prostokąt 2 - pole: 30 obwód: 22
3 Po skalowaniu Prostokąt 1 - pole: 48 obwód: 28
```

Dla bank.py:

```
1 Saldo początkowe: 1000.0
2 Saldo po wpłacie 500.0: 1500.0
3 Saldo po wypłacie 200.0: 1300.0
4 Próba wypłaty 2000.0: Niepowodzenie
5 Saldo końcowe: 1300.0
```

4 Opis struktury klas

Wszystkie klasy w zadaniach dla tej jednostki nie dziedziczą, ani nie są dziedziczone, więc struktury są pojedyncze.

5 Porównanie paradygmatu proceduralnego i obiektowego

Paradygmat proceduralny oddziela dane od funkcji, które je przetwarzają, podczas gdy paradygmat obiektowy łączy je w jedną całość w postaci obiektów. W podejściu proceduralnym program składa się z sekwencji wywołań procedur, natomiast w obiektowym ze współpracujących obiektów. Proceduralny skupia się na operacjach wykonywanych na danych, a obiektowy na relacjach między bytami i ich zachowaniach. Funkcje przyporządkowane są do obiektów, co poprawia organizację kodu względem podejścia proceduralnego.

6 Wnioski

Paradygmat obiektowy opiera się na strukturach danych zwanymi obiektami. Umożliwia to naturalne podejście modelowania zagadnień. Pozwala nam raz definiować strukturę klasy i tworzyć (z użyciem konstruktora) dowolną ilość jej instancji, zapewniając ich kompatybilność.