

Programowanie w języku Python – ćwiczenia 12

Zagadnienia: programowanie obiektowe: dziedziczenie

We wszystkich zadaniach stworzysz klasę nadrzędną oraz klasę (lub klasy) potomną. W klasie potomnej przesłonisz konstruktor oraz co najmniej jedną z metod klasy nadrzędnej. Dodatkowo klasa potomna odziedziczy co najmniej jedną z metod zdefiniowanych przez Ciebie w klasie nadrzędnej i zdefiniujesz w niej co najmniej jedną, nową metodę.

We wszystkich zadaniach stwórz obiekty wszystkich zdefiniowanych przez siebie klas i wypróbuj działanie wszystkich zdefiniowanych przez siebie metod.

Zadanie 1

Zdefiniuj klasę *Prostokat*. Każdy obiekt (instancja) klasy prostokąt ma mieć dwa atrybuty (długości boków *a* i *b*) zdefiniowane w konstruktorze `__init__`. Zdefiniuj metody instancji:

- `__repr__` - wyświetli dany obiekt klasy *Prostokat* w odpowiedni sposób,
- `pole` - zwróci pole danego prostokąta,
- `obwod` - zwróci obwód danego prostokąta oraz
- `wypisz` - dany prostokąt przedstawi się w dowolny sposób.

Następnie zdefiniuj klasę *Kwadrat*, która dziedziczy po klasie *Prostokat*.

- Przesłoń konstruktor `__init__`, tak aby tworząc kwadrat można było podać tylko jeden argument, długość boku *a*.
- Przesłoń metodę `wypisz` w ten sposób, aby dodatkowo (poza tym wszystkim co wypisuje metoda `wypisz` klasy *Prostokat*) na ekranie pojawił się napis „Jestem kwadratem”. W definicji metody `wypisz` wywołaj metodę `wypisz` klasy *Prostokat*.
- Zdefiniuj nową metodę `__repr__`, odpowiednią dla kwadratu.
- Dodaj do klasy *Kwadrat* metodę `rysuj`, która narysuje kwadrat w dowolny sposób.

Zadanie 2

Zdefiniuj klasę *Osoba* z trzema atrybutami instancji zdefiniowanymi w konstruktorze `__init__`: *imie*, *nazwisko* oraz *wiek*. Zdefiniuj metody instancji:

- `przedstaw_sie` – wypisze na ekranie odpowiednie przedstawienie się danej osoby.
- `urodziny` – zwiększy liczbę lat danej osoby o 1. Zwróci wiek osoby przed urodzinami.
- `__repr__` - wyświetli obiekt klasy *Osoba* w odpowiedni sposób.

Zdefiniuj dwie klasy potomne klasy *Osoba*, *Student* oraz *Pracownik*. W klasie *Student*:

- Przesłoń konstruktor `__init__` dodając atrybut (pole) *rok*, który będzie przechowywał rok studiów danego studenta.
- Przesłoń metodę `przedstaw_sie` wywołując w jej definicji metodę `przedstaw_sie` klasy *Osoba* i coś do niej dodając.
- Przesłoń metodę `__repr__` w dowolny sposób (z wykorzystaniem lub nie metody `__repr__` klasy *Osoba*).
- Dodaj metodę `kolejny_rok`, w której zwiększysz *rok* studiów danego studenta o 1. Metoda ta nic nie zwraca.

W klasie `Pracownik`:

- Przesłoń konstruktor `__init__` dodając atrybuty (pola) `ile`, który będzie przechowywał wielkość zarobków danego pracownika, oraz `zawod`, który będzie przechowywał zawód pracownika.
- Przesłoń metodę `przedstaw_sie` wywołując w jej definicji metodę `przedstaw_sie` klasy `Osoba` i coś do niej dodając.
- Przesłoń metodę `__repr__` w dowolny sposób (z wykorzystaniem lub nie metody `__repr__` klasy `Osoba`).
- Dodaj metodę `podwyzka(n)`, w której zwiększysz wartość atrybutu `ile` danego pracownika o `n` procent. Metoda ta zwraca wysokość zarobków danego pracownika przed podwyżką.

Zdefiniuj klasę `Nauczyciel` będącą klasą potomną klasy `Pracownik`. W klasie `Nauczyciel`:

- Przesłoń konstruktor `__init__` klasy `Pracownik` dodając atrybut (pole) `przedmioty`, który będzie przechowywał listę z nazwami przedmiotów, których może uczyć dany nauczyciel. Domyślnie lista `przedmioty` jest pusta. W konstruktorze nadaj wartość `'nauczyciel'` atrybutowi `zawod`.
- Przesłoń metodę `przedstaw_sie` wywołując w jej definicji metodę `przedstaw_sie` klasy `Osoba` i coś do niej dodając.
- Przesłoń metodę `__repr__` wywołując w jej definicji metodę `__repr__` klasy `Pracownik`.
- Dodaj metodę `nowy_przedmiot(prz)`, w której dodasz do listy `przedmioty` danego nauczyciela nowy przedmiot `prz`. W metodzie tej wywołaj metodę `podwyzka`, ponieważ nauczyciel za każdy nowy przedmiot, którego może teraz uczyć dostaje podwyżkę 5%. Metoda ta nic nie zwraca.

Zadanie 3

Zdefiniuj klasę `Zespolona` (z dwoma atrybutami instancji `x`, `y` – część rzeczywistą oraz część urojoną liczby zespolonej) wraz z metodami instancji:

- `__init__`
- `modul` - w metodzie tej zostanie obliczony (i zwrócony) moduł danej liczby zespolonej, czyli odległość punktu (x, y) od początku układu współrzędnych.
- `__repr__`
- `dodaj(z)` – metoda ta zwróci liczbę zespoloną (obiekt klasy `Zespolona`) powstałą w wyniku dodania do liczby zespolonej, na rzecz której ta metoda została wywołana, liczby zespolonej `z`. Możesz w zamian zdefiniować metodę `__add__`.
- `razy_skalar(sk)` – metoda ta zwróci liczbę zespoloną (obiekt klasy `Zespolona`) powstałą w wyniku przemnożenia liczby zespolonej, na rzecz której metoda ta została wywołana, poprzez skalar `sk`. Możesz w zamian zdefiniować metody `__mul__` i `__rmul__`.

Zdefiniuj klasę `Rzeczywista`, która dziedziczy po klasie `Zespolona`. Przesłoń metody `__init__`, `__repr__`, oraz `dodaj`. W definicji metody `__init__` wywołaj odpowiednią metodę klasy `Zespolona`. Dodaj metodę `reszta(m)`, która zwróci resztę z dzielenia liczby, na rzecz której ta metoda została wywołana, przez `m`.

Zadanie 4

Zdefiniuj klasę `Prostopadloscian` (z trzema atrybutami instancji będącymi długościami krawędzi prostopadłościanu `a`, `b` oraz `c`) wraz z metodami `__init__`, `__repr__`, `pole`, `objetosc` oraz `wypisz`.

Zdefiniuj klasę `Szescian` będącą klasą potomną klasy `Prostopadloscian`. Przesłoń w odpowiedni sposób metody `__init__`, `__repr__`, oraz `wypisz`. Dodaj metodę `suma_krawedzi`, która zwróci sumę wszystkich krawędzi danego sześcianu.

Zadanie 5

Zdefiniuj klasę `Zwierzak`. Obiekt klasy `Zwierzak` ma mieć trzy atrybuty zdefiniowane w konstruktorze `__init__`, `imie`, `wiek` oraz `gatunek`. W klasie tej zdefiniuj metody:

- `__repr__`
- `urodziny` – zwiększy wiek danego zwierzaka o 1.
- `przedstaw_sie` – zwierzę powie jak ma na imię, ile ma lat i jakiego jest gatunku.
- `powiedz(tekst)` – zwierzę powie tekst podany jako argument.

Zdefiniuj trzy klasy potomne klasy `Zwierzak` odpowiadające trzem różnym gatunkom zwierząt. W każdej dodaj jeden, dodatkowy atrybut instancji. Przesłoń w nich metody:

- `__repr__`
- `przedstaw_sie`
- `powiedz(tekst)` w ten sposób, aby domyślnie (o ile nie podamy żadnego argumentu) zwierzę wydawało odgłos charakterystyczny dla danego gatunku. W definicji tej metody wywołaj metodę `powiedz` klasy `Zwierzak`.
- Dodaj dowolną, wymyśloną przez Ciebie metodę.