

Programowanie 1R – Ćwiczenia 6

Klasy (podstawy)

Mikołaj Marcinkowski

14 kwietnia 2026

Zadania możesz wykonywać jako oddzielne skrypty `.py` lub w notatniku jupytera `.ipynb`, chyba że treść zadania wskazuje inaczej.

Zadania wzorowane na Serii 6 zadań autorstwa Bartłomieja Zglinickiego (<https://www.fuw.edu.pl/~bzglinicki/teaching/plr/zadania/seria-6/>), udostępnionej na licencji CC BY 4.0.

Zadanie 1. Okręgi

Napisz klasę `Okrag`, która reprezentuje okrąg w dwuwymiarowym kartezjańskim układzie współrzędnych. Klasa ta powinna zawierać:

- pole `x`, przechowujące współrzędną x środka okręgu,
- pole `y`, przechowujące współrzędną y środka okręgu,
- pole `r`, przechowujące promień okręgu,
- konstruktor przyjmujący położenie środka okręgu i jego promień,
- metodę `obwod`, zwracającą obwód okręgu,
- metodę `przeciecia`, która przyjmuje inny obiekt klasy `Okrag` i zwraca liczbę punktów przecięcia z tym okręgiem.

Napisz program, który tworzy dwa okręgi i zwraca ich obwody oraz liczbę punktów przecięcia między nimi.

Zadanie 2. Oporniki

Napisz klasę `Opornik` reprezentującą opornik, która zawiera:

- pole `_R`, przechowujące opór opornika w omach,

- konstruktor przyjmujący jako argument opór opornika z domyślną wartością 0,
- publiczną¹ metodę `get_opor`, zwracającą opór opornika,
- publiczną metodę `set_opor`, zmieniającą opór opornika na wartość podaną jako argument.

Napisz funkcję `szeregowo`, która przyjmuje dwa obiekty klasy `Opornik` i zwraca obiekt klasy `Opornik` o oporze równym zastępczemu oporowi wejściowych oporników połączonych szeregowo.

Napisz funkcję `rownolegle`, która przyjmuje dwa obiekty klasy `Opornik` i zwraca obiekt klasy `Opornik` o oporze równym zastępczemu oporowi wejściowych oporników połączonych równolegle.

Napisz program, który zwraca opory zastępcze dwóch oporników połączonych szeregowo i równolegle.

BONUS: Przerób funkcje `szeregowo` i `rownolegle` tak, by umiały przyjąć dowolną liczbę argumentów.

Zadanie 3. Stos

Napisz klasę `Stos` implementującą strukturę danych stosu. Klasa ta powinna zawierać:

- pole będące listą przechowującą elementy stosu,
- konstruktor, który inicjalizuje tę listę jako pustą,
- metodę `push`, która przyjmuje dowolny obiekt i doczepia go na końcu listy,
- metodę `pop`, która usuwa ostatni element listy i go zwraca.

Użyj obiektu klasy `Stos`, by odwrócić napis podany przez użytkownika.

¹W niektórych językach programowania (np. C++) istnieje podział na prywatne i publiczne pola i metody klasy. Metody i pola prywatne są widoczne tylko dla innych metod danej klasy, a publiczne mogą być wywołane z zewnątrz. Jeśli np. zmienienie wartości pewnego pola wymaga dodatkowych zmian innych pól, zamiast udostępniać je publicznie, zostawia się je jako prywatne, a wszystkie potrzebne zmiany przeprowadza się w publicznej metodzie `set`. Python nie posiada stricte mechanizmu prywatnych i publicznych metod/pól, jeśli jednak zaczniemy nazwę metody/pola od dwóch podkreśleń `__` i nie zakończymy jej dwoma podkreśleniami, Python utrudni do niej dostęp. Zwyczajowo poza klasą nie powinno się odwoływać do metod/pól zaczynających się od podkreślenia.

Dla chętnych oryginalna seria zadań Bartłomieja Zglinickiego zawiera jeszcze zadanie 4, w którym symulujemy rozprzestrzenianie się epidemii wśród ludzi chodzących po dwuwymiarowej planszy.