Programowanie I

Zajęcia nr 10

Rafał Masełek

12. maja 2022r.

Zadania 1-3 przygotowane przez B. Zglinickiego

Zadanie 0. – Figury geometryczne

Napisz klasę Figure, która reprezentuje figurę geometryczną. Konstruktor Figure może przyjąć dowolną ilość argumentów, które wpisuje do zmiennej "sides" będącej listą. 'sides" przechowuje długości boków figury. Zaimplementuj metodę "circumference" liczącą obwód figury. Dodatkowo, klasa Figure posiada pole statyczne liczące ile figur utworzono w trakcie działania programu. Dla klasy Figure zdefiniuj metodę __repr__.

Klasa Figure posiada dwie klasy pochodne: Triangle i Rectangle, reprezentujące trójkąt i prostokąt. Konstruktory tych klas przyjmują odpowiednio 3 i 2 argumenty, będące długościami boków. W przypadku otrzymania innej liczby argumentów należy zgłosić wyjątek. Dla obu klas proszę przeciążyć metodę __repr__ oraz napisać metodę "area" liczącą pole figury.

Napisz również klasę Square dziedziczącą po klasie Rectangle. Napisz dla niej kosntruktor przyjmujący tylko jeden argument. Przeciąż metodę __repr__ . Nie musisz przeciążać metody "area" (dlaczego?).

Pisząc klasy pochodne skorzystaj z konstruktorów klas pierwotnych.

W programie stwórz trzy figury: trójkąt, prostokąt i kwadrat. Dla każdej z nich wypisz obwód i pole, oraz wyprintuj obiekty na standardowe wyjście. Na koniec wyświetl liczbę stworzonych instancji klasy Figure, korzystając z pola statycznego.

Zadanie 1. – Relatywistyczne składanie prędkości

Niech $\beta = v/c$ będzie prędkością wyrażoną w jednostkach prędkości światła c. Zgodnie z relatywistycznym prawem składania prędkości w przypadku jednowymiarowym, jeżeli β_2 jest prędkością ciała drugiego względem ciała pierwszego, zaś β_1 – prędkością ciała pierwszego w pewnym układzie odniesienia, to prędkość ciała drugiego w tym układzie odniesienia wynosi

$$\beta = \frac{\beta_1 + \beta_2}{1 + \beta_1\beta_2} \tag{1}$$

Napisz klasę Velocity reprezentującą prędkość w ruchu jednowymiarowym. Zaimplementuj:

- konstruktor, który można wywołać bez argumentów lub z jednym argumentem; w pierwszym przypadku tworzony obiekt powinien być inicjalizowany wartością zero, w drugim zaś – wartością podaną jako argument,
- metodę gamma, zwracającą wartość czynnika

$$\gamma = \frac{1}{\sqrt{1 - \beta^2}} \tag{2}$$

- operator dodawania +, działający zgodnie z relatywistycznym prawem składania prędkości,
- operator +=, działający zgodnie z relatywistycznym prawem składania prędkości,

- metody __str__ i __repr__, formatujące opis instancji klasy.

Korzystając z tej klasy, napisz program velocity, który wczytuje ze standardowego wejścia dwie prędkości wyrażone w jednostkach prędkości światła i wypisuje na standardowe wyjście ich relatywistyczną sumę, również wyrażoną w jednostkach prędkości światła, oraz odpowiadający jej czynnik γ.
Przykładowe wykonanie
Wywołanie:
python3 velocity
Wejście:
0.5 0.7
Wyjście:
beta = 0.888889
gamma = 2.18282

Zadanie 2 – Liczby całkowite modulo 17

Napisz klasę Rest, reprezentującą liczby całkowite od 0 do 16 z dodawaniem i mnożeniem modulo 17. Zaimplementuj:

- konstruktor, który można wywołać bez argumentów lub z jednym argumentem; w pierwszym przypadku tworzony obiekt powinien być inicjalizowany wartością 0, zaś w drugim – resztą z dzielenia podanej liczby przez 17,
- operatory dodawania + i mnożenia * modulo 17,
- operatory += i *= działające modulo 17,
- metody __str__ i __repr__, formatujące opis instancji klasy.

Korzystając z tej klasy, napisz program rest, który wczytuje ze standardowego wejścia dwie liczby z prze- działu od 0 do 16 i wypisuje na standardowe wyjście ich sumę oraz iloczyn modulo 17.
Przykładowe wykonanie
Wywołanie:
python3 rest
Wejście:
9 12
Wyjście:
Suma modulo 17: 4
Iloczyn modulo 17: 6

Zadanie 3 – Liczby wymierne

Napisz klasę RationalNumber, reprezentującą liczbę wymierną. Klasa powinna przechowywać liczby całkowite p i q, których iloraz jest równy reprezentowanej liczbie wymiernej, przy czym p i q powinny być względnie pierwsze oraz $q > 0$. Zaimplementuj:

- konstruktor, który można wywołać bez argumentów lub z jednym albo dwoma argumentami całkowitymi; w pierwszym przypadku tworzony obiekt powinien być inicjalizowany wartością 0, w drugim – zadaną liczbą całkowitą, zaś w trzecim – ilorazem argumentów konstruktora; konstruktor wywołany z dwoma argumentami powinien działać poprawnie również wtedy, gdy wartości jego argumentów nie są względnie pierwsze lub gdy drugi argument jest ujemny; konstruktor powinien ponadto móc być wywołany z jednym argumentem będącym łańcuchem tekstowym, wówczas powinien inicjalizować odpowiednio wartości p i q, gdy łańcuch jest postaci p/q, np. -2/3, w przeciwnym razie tworzony obiekt powinien być inicjalizowany wartością 0; konstruktor wywołany w ten sposób powinien działać poprawnie również wtedy, gdy wartości p i q nie są względnie pierwsze lub gdy q jest ujemne, np. 4/-2,
- metody numerator i denominator, zwracające odpowiednio licznik p i mianownik q reprezentowanej liczby wymiernej,
- operator konwersji do typu float,

- jednoargumentowy operator zmiany znaku -,
- operator porównania $<$,
- operatory dodawania $+$ i mnożenia $*$,
- operatory $+=$ i $*=$,
- metody __repr__ i __str__; druga z nich powinna przedstawiać liczbę w postaci p/q, np. -5/7.

Korzystając z tej klasy, napisz program rational, który wczytuje ze standardowego wejścia dwie liczby wymierne, a następnie wypisuje na standardowe wyjście w kolejnych liniach:

- podane liczby w postaci dziesiętnej,
- liczby przeciwne do podanych,
- podane liczby w kolejności niemalejącej,
- sumę i iloczyn podanych liczb.

Przykładowe wykonanie
Wywołanie:
python3 rational
Wejście:
2/4 1/-3
Wyjście:
0.5 -0.3333333
-1/2 1/3
-1/3 1/2
1/6 -1/6