Programowanie obiektowe

Zadania lab1

Piotr Błaszyński

2 lutego 2017

Wszystkie klasy zapisywać w oddzielnych plikach i **zachować** na kolejne zajęcia (niektóre z nich się przydadzą). Wszystkie klasy przetestować, tworząc kilka obiektów klasy i uruchamiając każdą z dostępnych metod przynajmniej raz.

1. Przygotować klasę do przechowywania imion i nazwisk.
   a) Utworzyć trzy nazwane obiekty tej klasy.
2. Przygotować klasę opisującą kwadrat,
   a) metoda __init__ powinna przyjmować długość boku,
   b) napisać metodę zwracającą pole powierzchni,
   c) utworzyć trzy kwadraty,
   d) wypisać pole powierzchni poszczególnych kwadratów.
3. Przygotować klasę opisującą prostokąt (na bazie kopii kwadratu),
   a) metoda __init__ powinna przyjmować długości boków,
   b) napisać metodę zwracającą pole powierzchni,
   c) utworzyć trzy prostokąty,
   d) utworzone prostokąty dodać do listy,
   e) wypisać pole powierzchni poszczególnych prostokątów, korzystając z listy,
   f) przygotować funkcje (nie metodę), która przyjmie prostokąt jako parametr i wypisze długości jego boków i pole powierzchni, do wykonania tego skorzysta z metod z klasy, wywołać funkcję dla wszystkich wcześniej utworzonych prostokątów.

4. Przygotować klasę do zliczania (dodawania i odejmowania wartości) liczb. Oprócz inicjacji dowolną wartością, klasa ma umożliwiać dodawanie, odejmowanie i wypisywanie wartości (jako metody).

5. Przygotować klasę do prostych obliczeń statystycznych (można ją nazwać Statystyka).
   a) inicjalizacja przy pomocy listy,
   b) metoda zwracająca wartość sumy całej listy (można zrobić wariant pozwalający na obliczanie sumy fragmentu listy),
   c) metoda obliczająca średnią (suma/liczbę elementów), metoda powinna skorzystać z poprzedniej metody,
   d) metoda obliczająca medianę (Dla listy uporzadkowanej jest to wartość, która jest w połowie listy w wypadku nieparzystej liczby elementów. Dla parzystej liczby elementów – średnia arytmetyczna dwóch środkowych liczb). W związku z tym trzeba skorzystać z funkcji sorted.
   e) metoda obliczająca minimum,
   f) metoda obliczająca maximum. Wykonać testy wszystkich metod na co najmniej 3 listach po co najmniej 5-6 wartości.

6. Przygotować klasę reprezentującą kalendarz na dany miesiąc.
   a) metoda `__init__` przyjmuje rok i miesiąc,
   b) metoda pokaz wyświetla odpowiednią liczbę dni w blokach po 7,
      i. rok przestępny to taki, którego liczba określająca rok dzieli się bez reszty przez 4, ale nie dzieli się bez reszty przez 100, ale dzieli się bez reszty przez 400,
   c) wersja dla **zaawansowanych**: ustalić w jaki dzień wypada pierwszy dzień tego miesiąca i odpowiednio podpisać dni tygodnia (lub rozpocząć wyświetlanie kalendarza od odpowiedniego dnia). Wykorzystać dzielenie modulo 7 (x % 7).

7. Przygotować klasę reprezentującą figure geometryczną (na chwilę obecną klasa ma przechowywać nazwę i położenie figury i je wypisywać.

8. Przygotować klasę reprezentującą planetę, atrybuty wydedukować z poniższej tabeli:

Planeta	odległość od Słońca w j.a.	rzeczywista
Wulkan	0.03	false
Merkury	0.38	true
Wenus	0.72	true
Ziemia	1.0	true
Mars	1.52	true
Faeton	2.7	false
Jowisz	5.2	true
Saturn	9.53	true
Uran	19.19	true
Neptun	30.06	true
Pluton	39.48	false/true

Utworzyć listę, przechować w niej wszystkie planety z powyższej tabeli. j.a. - jednostka astronomiczna to 149 597 870 700 m. W przybliżeniu średnia odległość Ziemi od Słońca. Wyświetlić listę planet, odległość od Słońca podając w tys. kilometrów. Jakie pytanie powinno paść przy ostatniej pozycji w tabelce (Pluton)? Wyświetlić jeszcze dwie listy planet: rzeczywiste i nie. Zapisać do pliku (Przykładowy kod do obsługi plików na końcu listy zadań). Następnie dorobić odczyt danych z pliku i zastąpić nim część tworzącą obiekty (obiekty mają być tworzone z pliku). Czy obsługa obiektów w ten sposób jest wygodna?

9. Przygotować klasę udostępniającą metody konwersji jednostek:
   a) cali na cm,
   b) cm na cale,
   c) kg na lbs (funty),
   d) lbs na kg,

1 lbs	0.45359237 kg
1 kg	2.20462262 lbs
1 cm	0.393700787 cal
1 cal	2.54 cm

Czy to powinna być typowa klasa zawierająca metodę __init__, czy może należy wykonać ją inaczej. Przetestować wszystkie konwersje na kilku różnych danych wejściowych.

Dla przypomnienia, przykładowa klasa ręka i korzystanie z niej:

```
class Reka:
    def __init__(self, ktora):
        self.ktora=ktora
    def pokaz(self):
        print(self.ktora)

prawa = Reka("PRAWA")
lewa = Reka("LEWA")

prawa.pokaz()
lewa.pokaz()
```

Odczyt pliku:

```
plik = open('nazwa_pliku')
try:
   tekst = plik.read()
finally:
   plik.close()

print (tekst)
```

Zapis do pliku:

```
plik = open('plik.txt', 'w')
try:
    plik.write("tresc")
finally:
    plik.close()
```