

# Programowanie obiektowe

## Lista 8.

Poniższa lista zadań jest do zrobienia w języku Ruby. Każde zadanie to 4 punkty. Wybierz 2 zadania.

### Zadanie 1

Na wzór klasy *Temperatura* z wykładu zaprogramuj klasy reprezentujące *masę* i *długość*, które implementują atrybuty wirtualne reprezentujące zarówno jednostki podstawowe układu SI<sup>1</sup> jak i jednostki podstawowe anglosaskiego układu jednostek miar<sup>2</sup>.

Korzystając z tych klas zaprogramuj klasy reprezentujące *powierzchnię* i *ciśnienie* (są to tzw. *jednostki pochodne*). Klasa reprezentująca powierzchnię powinna posiadać atrybuty reprezentujące wielkość wyrażoną w jednostkach *hektar* i *cal kwadratowy*, natomiast klasa reprezentująca ciśnienie powinna mieć atrybut wyrażający ciśnienie w *barach* i *psi* (funt na cal kwadratowy).

Jako ilustrację do swojego programu wydrukuj dwie tabelki jedna związana z powierzchnią, druga z ciśnieniem, zawierające w jednej kolumnie wartość wyrażoną w jednostkach SI, w drugiej w jednostkach anglosaskich.

### Zadanie 2

(Zadanie podobne do poprzedniego) Na wzór klasy *Temperatura* z wykładu zaprogramuj klasy reprezentujące *długość* (przynajmniej z kilometrami i milami morskimi) i *czas* (przynajmniej z godzinami i sekundami). Jako klasy reprezentujące jednostki pochodne zaprogramuj *prędkość* (wyrażoną w km/h i węzłami) i *przyspieszenie* (w km/s<sup>2</sup> i mm/h<sup>2</sup>).

Jako ilustrację do swojego programu wydrukuj dwie tabelki jedna związana z prędkością, druga z przyspieszeniem, zawierające w jednej kolumnie wartość wyrażoną w jednostkach SI, w drugiej w jednostkach spoza układu SI.

### Zadanie 3

Zaprogramuj klasę reprezentującą wyrażenia arytmetyczne w postaci ONP. Zakładamy, że wyrażenia są zbudowane z liczb całkowitych, zmiennych i czterech podstawowych operatorów arytmetycznych<sup>3</sup>.

Wyrażenia są podawane w konstruktorze obiektu w postaci tablicy zawierającej liczby, operatory i zmienne. Zaprogramuj metodę *.oblicz*, która wylicza i zwraca wyliczoną wartość wyrażenia; argumentem tej metody powinien być słownik zawierający nazwy zmiennych i ich wartości.

Dodatkowo metoda *.oblicz* w trakcie działania powinna wypisywać wygląd stosu podczas kolejnych etapów obliczeń.

### Zadanie 4

Jedną z najprostszych metod szyfrowania jest szyfr podstawieniowy, w którym za literę podstawia się inną literę. Zaprogramuj dwie klasy:

- klasę *Jawna* przechowującą napis w postaci jawnej i implementującą metodę *zaszyfruj(key)* zwracającą obiekt klasy *Zaszyfrowane*;
- klasę *Zaszyfrowana* przechowującą napis zaszyfrowany i implementującą metodę *oszyfruj(key)* zwracającą obiekt klasy *Jawna*.

<sup>1</sup>[https://pl.wikipedia.org/wiki/Układ\\_SI](https://pl.wikipedia.org/wiki/Układ_SI)

<sup>2</sup>[https://pl.wikipedia.org/wiki/Anglosaski\\_układ\\_jednostek\\_miar](https://pl.wikipedia.org/wiki/Anglosaski_układ_jednostek_miar)

<sup>3</sup>Można się ograniczyć tylko do dzielenia całkowitego

Obydwie klasy winne implementować metodę `to_s` zwracającą przechowywany tekst (zaszyfrowany lub nie). Argument `key` jest słownikiem. Na przykład dla klucza postaci

```
{  
  'a' => 'b',  
  'b' => 'r',  
  ...  
  'r' => 'y',  
  'y' => 'u',  
  'u' => 'a'  
}
```

słowo *'ruby'* jest szyfrowane jako *'yaru'*. Dla uproszczenia można przyjąć, że szyfrowane będą jedynie teksty zawierające małe litery alfabetu angielskiego, cyfry, odstęp i znaki interpunkcyjne.

*Marcin Młotkowski*