

# Kurs języka Ruby

## Lista 4.

**Zadanie 1.** Zaprogramuj hierarchię klas reprezentujących wyrażenia arytmetyczne. Hierarchia powinna zawierać reprezentację przynajmniej takich wyrażen jak stała liczbowa, zmienna, cztery podstawowe operacje arytmetyczne. Dla każdej klasy zaprogramuj metodę

- `to_s`
- `oblicz` obliczającą wartość wyrażenia. W razie potrzeby można przyjąć, że metoda ta może mieć dodatkowe argumenty;
- `uproszczenie` upraszczająca wyrażenie korzystając ze znanych reguł typu  $x + 0 = x$ ,  $expr_1 * \frac{1}{expr_1} = 1$ . Zaprogramuj przynajmniej 3 takie reguły.

Zadanie nie wymaga tworzenia wyrażen z napisów, wystarczy jeśli będziemy tworzyć wyrażenia konstruując obiekty

```
Dodawanie.new(Stala.new(5), Zmienna.new("x"))
```

**Zadanie 2.** Korzystając z poprzedniego zadania zaprogramuj klasy reprezentujące podstawowe instrukcje prostego języka programowania: instrukcja podstawienia, instrukcja warunkowa, instrukcja pętli i sekwencja instrukcji. Zaprogramuj w każdej klasie metodę `to_s` oraz `wykonaj`. Można przyjąć, że jeśli wartość wyrażenia arytmetycznego jest zero to jest to interpretowane jako fałsz.

Jako przykład podaj program obliczający np. silnię.

**Zadanie 3.** Zaprogramuj klasę *DrzewoBinarne* implementującą drzewo poszukiwań binarnych wraz z operacjami `wstaw`, `istnieje?` i `usun`. Przyjmij, że elementy drzewa są obiektami klasy *Element*. Podaj przykład wykorzystania tych klas. Zwykle do implementacji takiej klasy konieczne są operatory `>`, `<` czy `==`, jednak zamiast tego wygodniej jest zaimplementować tylko metodę `<=>` i dołączyć *mix-in* *Comparable*. Zaprogramuj metodę `to_s` zwracającą string opisujący drzewo w postaci infiksowej.

Zaprogramuj *StringBT* jako podklasę *DrzewoBinarne* przechowującą obiekty klasy *String*.

Każde zadanie jest warte 4 punkty. Na pracowni oddaje się dwa zadania.

*Marcin Młotkowski*