

## Paradygmaty programowania

### Lista 1 (treningowa)

**Zad. 1.** Napisać funkcję rozwiązującą problem możliwości budowy trójkąta z trzech odcinków o długościach  $a$ ,  $b$ ,  $c$ . Aby trójkąt mógł istnieć, musi być spełniony następujący warunek:  $|b-c| < a < b+c$ . Pole trójkąta można wyznaczyć wtedy z następującej zależności:  $\text{pole}^2 = p(p-a)(p-b)(p-c)$  gdzie  $p = (a+b+c)/2$ . Funkcja ma zwracać pole trójkąta jeśli jego budowa jest możliwa i wyjątek jeśli nie. Zadanie należy wykonać w OCaml i w Scala.

**Zad. 2.** Dane są dwie liczby:  $n$  całkowite i  $x$  rzeczywiste. Zdefiniować rekurencyjną funkcję (rekurencja ogonowa) wyznaczającą wartość  $S$  zdefiniowaną poniżej. Zadanie należy wykonać w OCaml i w Scala.

$$S = \sum_{i=1}^n (-1)^i x^i / i!$$

**Zad. 3.** Zdefiniować rekurencyjną funkcję (rekurencja ogonowa) wyznaczającą wartość  $n$ , dla której suma

$$S = \sum_{i=1}^n 1/i$$

przekroczy zadaną wartość  $R$ . Zadanie należy wykonać w OCaml i w Scala.

**Zad. 4.** Napisać rekurencyjną funkcję (rekurencja ogonowa), która dla danej listy liczb rzeczywistych znajdzie najmniejszą liczbę w liście. Zadanie należy wykonać w OCaml i w Scala.

**Zad. 5.** Napisać rekurencyjną funkcję (rekurencja ogonowa), która daną listę liczb całkowitych podzieli na dwie listy: listę liczb mniejszych od zera i listę liczb większych równych zero. Wynik przedstawić w postaci listy dwóch list wynikowych. Zadanie należy wykonać w OCaml i w Scala.