

Paradygmaty programowania - studia stacjonarne, lista 8, 30.11.2017, godzina 15:15
Czas na rozwiązanie: 70 min.

WAŻNE1: Do każdego zadania przygotuj 3-5 testów sprawdzających poprawność działania!

WAŻNE2: Nie wolno stosować funkcji bibliotecznych i konstrukcji innych niż te przedstawione na wykładach 1-6.

1. Napisz funkcję eval realizującą prosty ewaluator operacji wektorowych oparty na maszynie stosowej.

Rozkazy do ewaluatora podane są jako lista elementów typu:

type eval = Scal of float | Vect of float*float | Neg | Add | Sub | Mul | Prod;;

Do reprezentacji stosu użyj typu "tstack list", gdzie tstack jest albo skłarem, albo wektorem.

Dla uproszczenia dla operacji Mul można przyjąć zadaną kolejność argumentów, na przykład skalar, wektor.

Przykład:

```
let instr1 = [Scal 5.; Vect (3.,4.); Mul; Vect( 1., 1.); Prod];;  
daje wynik: 35.
```

```
let instr2 = [Scal 5.; Vect (3.,4.); Mul; Mul];;  
zgłasza błąd (wyrażenie jest niepoprawne, druga operacja Mul nie ma wystarczającej liczby argumentów)
```

```
let instr3 = [Scal 5.; Scal 5.; Mul];;  
zgłasza błąd (wyrażenie jest niepoprawne, typ drugiego argumentu jest niepoprawny)
```

Punkty: 12 (język Java).

2. Napisz funkcję powielającą elementy w kolekcji na podstawie drugiej kolekcji określającej ile razy elementy mają być powielone.

Funkcję napisz w języku Java w sposób imperatywny przy użyciu tablic i w języku Scala używając list i stylu funkcyjnego.

przykład: duplicate [1;2;3] oraz [0;3;1;4] daje wynik [2;2;2;3]

Punkty: 8 (język Java, Scala)