

**Paradygmaty programowania - studia stacjonarne, lista 6, 16.11.2017, 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 1-5.**

1. Napisz funkcję eval realizującą prosty ewaluator operacji na funkcjach liczb rzeczywistych oparty na maszynie stosowej.

Rozkazy do ewaluatora podane są jako lista elementów typu:

```
type eval = Val of float | Fun of (float->float) | Neg | Add | Mul | Comp | Eval;;
```

Do reprezentacji stosu użyj typu "tstack list", gdzie tstack jest albo wartością liczbową, albo funkcją .

Dla uproszczenia dla operacji Eval można przyjąć zadaną kolejność argumentów, na przykład wartość liczbowa, funkcja.

Przykład:

```
let instr1 = [Val 16.; Fun sqrt; Fun sqrt; Comp; Eval];;  
daje wynik: 2.
```

```
let instr2 = [Val 5.; Fun sqrt; Comp; Comp];;  
zgłasza błąd (wyrażenie jest niepoprawne, druga operacja Comp nie ma wystarczającej liczby argumentów)
```

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

Punkty: 11 (język Ocaml). Uważaj na ostrzeżenia kompilatora (utrata punktów)!

2. Napisz funkcję, która zwraca "lustrzane odbicie" drzewa zdefiniowanego na wykładzie 4, strona 17.

Punkty: 5 (język Scala).

3. Napisz funkcję, która generuje w postaci listy leniwej, ciąg kolejnych liczb parzystych począwszy od pierwszej liczby parzystej większej lub równej k, gdzie k jest liczbą całkowitą.

Przykład: lparzyste 12 -> LCons (12, <fun>)

Punkty: 4 (język Ocaml).