

# Programowanie funkcyjne 2014

## Lista 9

kzi

27 stycznia 2015

**Uwaga:** Jeśli zadanie jest niedospecyfikowane, to braki w specyfikacji można uzupełnić w dowolny sensowny sposób.

### Zadania w Racket'cie (PLT Scheme)

1. Napisz moduł do obsługi odwrotnej notacji polskiej, zawierający
  - a) funkcję, która bierze wyrażenie w ONP i zwraca wynik jego obliczenia, (5ptk)
  - b) funkcję, która bierze wyrażenie w notacji infiksowej (może zawierać nawiasy) i przekształca je do wyrażenia w ONP. (10ptk)

Reprezentacja wyrażeń może być dowolna. W szczególności jeśli wyrażenia są napisami, można przyjąć, że wszystkie literały są rozdzielone spacjami.

Funkcje powinny obsługiwać operacje dodawania, odejmowania, mnożenia i dzielenia.

*Wskazówka: Algorytmy są na Wikipedii.*

2. Napisz aplikację internetową, która oblicza wyrażenia arytmetyczne. Aplikacja powinna mieć pole tekstowe i przycisk. Po wprowadzeniu poprawnego wyrażenia arytmetycznego w polu tekstowym i naciśnięciu przycisku powinien pojawić się wynik obliczenia tego wyrażenia.

Jeśli nie udało nam się zrobić punktu 1b), to nasz kalkulator może działać tylko na wyrażeniach w ONP. Jeśli nam się udało, oczywiście powinien przyjmować wyrażenia w notacji infiksowej. Tak jak poprzednio można przyjąć, że w poprawnym wyrażeniu literały rozdzielone są spacjami.

Aplikacja może robić cokolwiek dla niepoprawnych wyrażeń. (5ptk)

*Wskazówka: Skorzystać z tutoriala <http://docs.racket-lang.org/continue/> (wystarczy pierwsze 6 sekcji). Tak na dobrą sprawę to zadanie to CtrlC, CtrlV z tutoriala i skorzystanie z zadania 1, dlatego jest za nie mało punktów. Ale to nie znaczy, że kod może być śmietnikiem. Niepotrzebne fragmenty kodu z tutoriala należy usunąć.*

3. Napisz moduł do obsługi prostych wyjątków za pomocą kontynuacji. Moduł powinien zawierać
  - a) funkcję (`try/with body handler`)  $\rightarrow$  any, która symuluje wyrażenie `try body () with Exception exn_val -> handler exn_val` w OCaml'u,
  - b) funkcję (`throw exp`)  $\rightarrow$  any, która symuluje wyrażenie `raise (Exception exp)` w OCaml'u.

Moduł powinien działać poprawnie dla programów jednowątkowych. (8ptk)

Uwaga. W OCaml'u nie da się zdefiniować wyjątku `Exception`, bo jego argument może być dowolnego typu, a system typów na takie wyjątki nie pozwala. Ale na potrzeby zadania możemy sobie wyobrazić, że się da.

*Wskazówka: Zdefiniować globalną zmienną, w której będziemy trzymać kontynuację wracającą do ostatniego `try/with`.*