

Kurs języka Lua 2017

Lista zadań nr 4

Na zajęcia 27–28.03.2017

Za zadania z tej listy można uzyskać maksymalnie 6 punktów.
Styl kodu ma wpływ na ocenę jakości rozwiązania.

Zadanie 1. (3p) Napisz funkcję która konwertuje napis kodujący wyrażenie lispowe do tablicy Lua. Zakładamy, że wyrażenia lispowe składają się z nawiasów, symboli, liczba całkowitych, napisów oraz symbolu `nil`. Nawiasy powinny być przekształcane do sekwencji, napisy do napisów, liczby do liczb, `nil` do `nil`, natomiast symbole `S` do tabel postaci `{symbol='S'}`.

Maksymalnie wykorzystaj mechanizm `captures`, nie używaj metod konwertujących napis do np. liczby w celu sprawdzenia czy liczba jest poprawnie sformatowana.

Przykład:

```
(if nil
  (list 1 2 "foo")
  (+ 1 2 var 4))

-->
{{symbol='if'}, nil,
  {{symbol='list'}, 1, 2, 'foo'},
  {{symbol='+'}, 1, 2, {symbol='var'}, 4}}
```

Zadanie 2. (3p) Napisz funkcję która dla zadanego napisu sprawdzi czy koduje on poprawne wyrażenie arytmetyczne, składające się z liczba całkowitych i zmiennoprzecinkowych ze znakiem, działań `+`, `-`, `*`, `/` oraz nawiasów.

Maksymalnie wykorzystaj mechanizm `captures`, w szczególności nie można próbować interpretować fragmentów napisu jako wartości, np. korzystając z metody `tonumber`.

Przykładowe wyrażenia:

```
- 2+ 4.503
(2*3.5*4)- (+12)/3
```

Zadanie 3. (3p) Napisz wyrażenie regularne które dla zadanej ścieżki do pliku przechwyci jego nazwę i rozszerzenie.

```
path = 'K:\hidden-name\Teaching\2016_Lua\[Lab]\Lecture 04.pdf'
string.match(path, ???) --> 'Lecture 04', 'pdf'
string.match('nazwa.pliku.txt', ???) --> 'nazwa.pliku', 'txt'
```

Wykorzystaj je jako fragment programu, który rozłoży ścieżkę na sekwencję kolejnych folderów zakończoną informacją o pliku. Separator właściwy dla swojego systemu operacyjnego możesz zdobyć korzystając z wyrażenia `package.config:sub(1,1)`.

Maksymalnie wykorzystaj możliwości jakie daje mechanizm dopasowywania wzorców.

```
unpath(s) = {'K:', 'hidden-name', 'Teaching', '2016_Lua',
            '[Lab]', {'Lecture 04', 'pdf'}}
```

Zadanie 4. (3p) Zaprojektuj kalendarz przechowujący zdarzenia których czas startu i końca ma typ `date`. Napisz funkcję `add (calendar, event)` która dodaje wydarzenie do kalendarza i zwraca wskaźnik na to wydarzenie w kalendarzu jeśli operacja się powiedzie lub `nil` i wskaźnik na wydarzenie kolidujące w przeciwnym przypadku.

Napisz funkcję `show (calendar [, all])` która wypisuje wydarzenia z kalendarza w porządku rosnącym, domyślnie od aktualnej chwili, chyba, że argument `all` jest prawdą, wtedy wypisz cały kalendarz.