

Programowanie funkcyjne w języku Scheme

Podręcznik dr Racket Scheme: <https://docs.racket-lang.org/guide/index.html>

- Uruchom program DrRacket z plikiem `obj_kuli.rkt`. Przeanalizuj definicję funkcji `obj_kuli`. Uruchom (Run), a następnie oblicz objętość kul o kilku różnych promieniach - w dolnej części okna wpisz np : `(obj_kuli 1)`.
- Wykonaj funkcję `row_kwadrat` (`row_kwadrat.rkt`) dla równania x^2-1
- Zapisz/oblicz w Scheme wyrażenia (bezpośrednio w dolnym oknie – obliczeniowym):

a. $\frac{3 \cdot (7-5)}{(2+1)}$ b. $7 \cdot (3+5+2+1)/8-7 \cdot 3$

- Przypisz wartości symbolom: `pi`: (`define pi 3.14159`) (`define r 4`). Użyj tych wartości do policzenia
- pola koła o promieniu 4
- wartości $\pi/2$;
- Zdefiniuj funkcję $f3param(x, y, z) = \frac{x^2 + y^2}{2z}$, oblicz `f3param(2,3,1)`; `f3param(2.5, 3.0, 1.5)`. Gdzieś były definiowane typy danych?
- Uzupełnij swoją funkcję o podanie wartości 10000, gdy wartość parametru $z = 0$. Postać funkcji `if`:
(`if warunek wyr prawda wyrfalsz`) – wypróbuj definiując funkcję
(`define (jesli a b) (> a b) a b`)
- Zdefiniuj funkcję `pole_trapezu` obliczającą pole trapezu o podanych podstawach i wysokości
- Przeanalizuj program `rowkwadrat.rkt`. Korzystając z operacji LET oblicz rozwiązanie układu równań liniowych:
 $a_{11}x + a_{12}y = c_1$
 $a_{21}x + a_{22}y = c_2$
metodą wyznaczników
- Rekurencja: Zdefiniuj funkcję (`silnia n`)= $n!$
- Wypróbuj operacje na listach z poniższej tabeli

funkcja	przykład	Wynik
Zapis listy	<code>'(1 3 5 8)</code> <code>'('a' 'b' 'c' 'd')</code> dla a, b c użyj cudzysłowów (<code>list 1 3 6 4</code>)	
list	(<code>list 1 3 6 4</code>) (<code>list 'a' 'b' 'c' 'd'</code>) – lista z danych elem	<code>'(1 2 3 4)</code> <code>'('a' 'b' 'c' 'd')</code>
car/first	(<code>first '(2 5 3 5)</code>)	1 - pierwszy element listy
cdr/rest	(<code>rest '(1 5 3 6)</code>) (<code>rest '((1 2 3) 5 (3 4))</code>)	<code>'(5 3 6)</code> ogon listy <code>'(5 (3 4))</code>
cons/	(<code>cons 2 '(3 4)</code>) (<code>cons 'a' 'b'</code>)	<code>'(2 3 4)</code> dołóż do listy (<code>'a' 'b'</code>)
append	(<code>append '(1 2) '(3 4 5)</code>)	<code>'(1 2 3 4 5)</code> składa listy

- Napisz funkcję (`drugi lista`) zwracającą drugi element zadanej listy.
Użyteczne funkcje do badania cech różnych (funkcje predykatywne)

funkcja	przykład	Znaczenie
<code>=, >, <,>, <=</code>	(<code>= x 7</code>) (<code>> n 0</code>) (<code><> n m</code>)	Równość, większość, nierówność
<code>even?</code>	(<code>even? n</code>)	Parzystość liczby
<code>odd?</code>	(<code>odd? n</code>)	Nieparzystość liczby
<code>list?</code>	(<code>list? zmienna</code>)	Czy zmienna jest listą
<code>null? empty?</code>	(<code>null? zmienna</code>) (<code>empty? '()</code>)	Czy zmienna jest listą pustą <code>'()</code>
<code>#t, #f</code>		Prawda, fałsz
<code>eq?</code>	(<code>eq? 3 3</code>) ; (<code>eq? 'a 'a</code>) (<code>eq? 'a 3</code>)	<code>#t</code> ; <code>#t</code> ; <code>#f</code> porównywanie
<code>eqv?</code>	(<code>eqv? 3 pi()</code>) (<code>eqv?</code>	<code>#f</code> ; Porównywanie wartości

12. Napisz funkcję sprawdzającą, czy podana liczba znajduje się na podanej liście liczb
13. Napisz funkcję wyliczającą długość podanej listy
14. Napisz funkcję, która dla podanego $n > 2$ obliczy n -ty wyraz ciągu Fibonacciego ($a_n = a_{n-1} + a_{n-2}$)
15. Napisz funkcję, która dla podanej listy liczb wyprowadzi listę złożoną z najmniejszego i największego elementu listy) *Uwaga: nikt nie powiedział, że nie możesz używać funkcji pomocniczych.*
16. Napisz funkcję podającą listę czynników pierwszych liczby n