

Paradygmaty programowania

Programowanie generyczne w C++

Dr inż. Andrzej Grosser

Częstochowa, 2014

Spis treści

1. Zadanie 7	5
1.1. Wprowadzenie	5
1.2. Podstawy Scheme	5
1.2.1. Typy danych	5
1.2.2. Wyrażenia	5
1.2.3. Definiowanie funkcji	5

1. Zadanie 7

1.1. Wprowadzenie

Scheme jest dialektem języka Lisp, wspiera wiele paradygmatów programowania w tym szczególnie istotny dla przedmiotu jest paradygmat programowania funkcyjnego. Jest językiem dynamicznym.

Podstawowym środowiskiem programowania będzie na zajęciach drscheme (nowsza wersja tego środowiska drracket).

1.2. Podstawy Scheme

1.2.1. Typy danych

Liczby całkowite 1, 2, 3

Liczby wymierne $7/12$, $4/5$

Liczby zmiennoprzecinkowe 1.23, 1.4

Liczby zespolone $4.0 + 5.0i$

Typ logiczny #t, #f

Typ znakowy #\a

Łańcuch znaków "Witaj świecie"

Symbole 'x

1.2.2. Wyrażenia

Wyrażenia w języku Scheme zapisuje się za pomocą notacji polskiej, oznacza to, że najpierw zapisywane są operatory a dopiero potem ich argumenty, całość ujmuje się nawiasami Np. $4 + 3$ w języku Scheme to

(+ 4 3)

$2 * 4 + 3$ w języku Scheme to

(+ (* 2 4) 3)

1.2.3. Definiowanie funkcji

Definicja funkcji obejmuje nazwę z listą parametrów formalnych zapisanych w nawiasach okrągłych :

```
(define (kw x) (* x x ))
```

Wywołanie funkcji (kw 4)

Wyrażenie warunkowe if (if predykat wartość1 wartość2) Jeśli predykat jest spełniony używana jest wartość1, w przeciwnym razie wartość2 np. (define (wabs x) (if (< x 0) (- x) x)) Wyrażenie cond

```
(cond
```

```
(predykat_1 odp_1)
```

```
(predykat_2 odp_2)
```

```
.....
```

```
(predykat_n odp_n)
```

```
(else odp_else))
```

Np:

```
(define (stawka wiek)
```

```
(cond
```

```
((and (>= wiek 18) (<= wiek 21)) 2.0)
```

```
((or(< wiek 18) (>= wiek 70)) 3.0)
```

```
(else 1.0)))
```