

Języki i paradygmaty programowania

Interpreter

Mateusz Perlik

Język LatteFun jest językiem imperatywnym. Jest on wersją języka Latte rozbudowaną o aspekty funkcyjne. Dokładniej, wspiera zagnieżdżone definicje funkcji, lambdy i zmienne o typach funkcyjnych. Ponadto język będzie statycznie typowany.

Doprecyzowanie podpunktów z tabelki:

1. Są trzy podstawowe typy, tj. `int`, `bool` i `string`, a ponadto typy funkcyjne zdefiniowane indukcyjnie jako postaci $(T_1, \dots, T_n) \rightarrow T_0$, gdzie T_i dla $i = 0, \dots, n$ są typami (potencjalnie funkcyjnymi).
2. Standardowo (jak w Latte).
3. Standardowo (`printInt` i `printString`).
4. Zmienne należy deklarować pojedynczo. Niezainicjalizowane zmienne mają domyślną wartość dla swojego typu, tj. 0 dla `int`, `false` dla `bool`, `""` dla `string`, zaś dla typu funkcyjnego $(T_1, \dots, T_n) \rightarrow T_0$ domyślną wartością jest funkcja stale równa domyślnej wartości T_0 . Składnia dla typu funkcyjnego jest następująca: jeśli chcemy zadeklarować f o typie $(T_1, \dots, T_n) \rightarrow T_0$, to piszemy $[(T_1, \dots, T_n) \rightarrow T_0]$ f .
5. Instrukcje `if` oraz `while` wymagają bloków po warunku dla jednoznaczności.
6. Funkcje z rekurencją, procedury jako funkcje zwracające `void`.
7. Przekazywanie przez wartość i przez zmienną. Domyślne przekazywanie przez wartość. Przekazywanie przez zmienną wymaga dopisania między typem zmiennej i jej identyfikatorem słowa kluczowego `ref`.
9. Standardowo zmienne lokalne i globalne oraz zagnieżdżone funkcje.
10. Jakiś komunikat o błędzie i zatrzymanie interpretera.
11. jw.
12. Standardowo
13. jw.
17. Zmienne o typach funkcyjnych jw., mogą być przekazywane jako parametry i mogą być wartościami funkcji. Ponadto występują lambdy. Składnia jest następująca: `lambda (T1 id1, ..., Tn idn) → T0 { Stmt }`, gdzie `Stmt` jest ciągiem instrukcji zawierającym instrukcję `return`, lub `lambda (T1 id1, ..., Tn idn) → T0 . Expr`, gdzie `Expr` jest wartością funkcji. Słowo kluczowe "lambda" można też zamienić na unicode'owy symbol λ .