Języki i paradygmaty programowania

Interpreter

Mateusz Perlik

Język LatteFun jest językiem imperatywnym. Jest on wersją języka Latte (lekko zmodyfikowanego w podstawowych aspektach) 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 typ `void` oraz typy funkcyjne zdefiniowane indukcyjnie jako postaci $(T_1, ..., T_n) \to T_0$, gdzie T_i dla $i = 0, ..., n$ są typami (potencjalnie funkcyjnymi).
2. Standardowo (jak w Latte).
3. Występują wbudowane funkcje print i println. Przyjmują jeden argument podstawowego typu. Funkcja println po wypisaniu wartości przechodzi do nowej linii.
4. Zmienne należy deklarować pojedynczo. Niezainicjalizowane zmienne mają domyślną wartość dla swojego typu, tj. 0 dla `int`, `false` dla `bool`, `""` (puste słowo) dla `string`, zaś dla typu funkcyjnego $(T_1, ..., T_n) \to T_0$ domyślną wartością jest funkcja stale równa domyślnej wartości typu T_0. Składniowo typ ten reprezentowany jest jako `[(T1, ..., Tn) -> T0]`.
5. Instrukcje `if`, `else` oraz `while` po warunku wymagają bloku otoczonego klamrami dla jednoznaczności.
6. Funkcje z rekurencją, funkcje zwracające void jako procedury.
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. 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 (domknięcia). Ponadto występują lambdy. Składnia jest następująca:
    - `lambda (T1 x1, ..., Tn xn) -> T0 { B }`, gdzie `B` jest blokiem będącym poprawnym ciałem funkcji o typie $(T_1, ..., T_n) \to T_0$ lub
    - `lambda (T1 x1, ..., Tn xn) -> T0 . E`, gdzie `E` jest wyrażeniem typu T_0, będący wartością funkcji.

    Słowo kluczowe `lambda` można również zastąpić unicode'owym symbolem λ.

    Aby wywołać lambdę nieprzypisaną do zmiennej, trzeba otoczyć ją nawiasem, np. `(lambda (int x, int y) -> int . x + y)(2, 3)` jest poprawnym wyrażeniem.