Latte♭

Język jest *mniej więcej* Latte bez większości rozszerzeń i kilkoma niewielkimi zmianami.

Opis języka

W opisie używam nazw zdefiniowanych w gramatyce. Nawiasy kwadratowe ([ ]) oznaczają listę, NIE wartość opcjonalną.

Tokeny i literały

Indentyfikatory

Identyfikatorami są nazwy nadawane zmiennym i funkcjom. 1. Identyfikator to niepusty ciąg liter, cyfr lub znaków '_' 2. Zaczyna się od litery 3. Wielkość liter ma znaczenie 4. Nie może być słowem kluczowym

Słowa kluczowe

- if
- else
- while
- int
- string
- boolean
- void
- true
- false
- print
- return

Integer

Niepusty ciąg cyfr, może być poprzedzony jednym znakiem '-'

String

Ciąg dowolnych (z wyłączeniem '"') znaków dowolnej długości ograniczony cudzysłowami.

Komentarze

- Wszystko, co występuje po "//" aż do końca bieżącej linii
- Wszystko, co zawarte między znakami "/*" i "*/"

Typy

Występują typy proste: - int (reprezentujące Integer) - boolean (reprezentujące literał "true" albo "false") - string (reprezentujące String) Oraz dodatkowo: - tablica: [T] gdzie T jest typem prostym lub tablicą - void:

oznaczenie wartości zwracanej przez funkcje, które nic nie zwracają

Typowanie jest statyczne.

Tablice

Tablice są indeksowane liczbami całkowitymi od -N do N-1, gdzie N jest długością tablicy. Indeksami od 0 do N-1 są oznaczone kolejne elementy od lewej, a indeksami od -1 do -N kolejne elementy od prawej (jak w Pythonie). Tablice mają wbudowane metody: - .get(\), która zwraca element mieszczący się w tablicy pod indeksem wartości - .set(\, \), która zapisuje wartość wyliczoną z \pod indeksem \, a gdy indeks nie występuje w tablicy, zwraca błąd w czasie działania programu - .len, która zwraca liczbę elementów przechowywanych w tablicy - .push(\), która dodaje nowy element na koniec tablicy - .remove(\), która usuwa element pod indeksem \

Wyrażenia

Arytmetyka

- Operatory arytmetyczne (+, -, /, %, *) działają standardowo na liczbach całkowitych.
- Operatory logiczne (&&, !, ||) działają standardowo na wartościach boolowskich.
- Operatory (==, !=) działają na dwóch zmiennych należących do tego samego typu.
- Operatory (<, <=, >, >=) działają na zmiennych typu całkowitego i stringach.
- Operator infiksowy ++ służy do konkatenacji dwóch stringów lub dwóch tablic.

Funkcje

Funkcje są wołane w następujący sposób: `<Ident>( [<ExprOrRef>] );` gdzie argumenty są postaci: `<Expr>` lub `& <Ident>`. W pierwszym przypadku do funkcji przekazywany jest argument przez wartość. W drugim przez zmienną.

Instrukcje

Funkcje

Funkcje są deklarowane w następujący sposób: `<TypeOrVoid> <Ident>([<Type> <Ident>]) { [Stmt] }` gdzie to typ prosty, tablica lub void. W ciele funkcji może wystąpić wywołanie jej samej (rekurencja).

Deklaracje/przypisanie

- Wszystkie zmienne muszą zostać zadeklarowane przed ich pierwszym użyciem.
- W przypadku niezgodności typów przy przypisaniu, błąd zostanie zgłoszony w czasie sprawdzania typów, przed uruchomieniem programu.
- Przy deklaracji zmienne typu T przyjmują domyślnie wartość:
    - dla T = int: 0
    - dla T = string: ""
    - dla T = boolean: "false"
    - dla T = array: []
- Występuje przesłanianie identyfikatorów i statyczne wiązanie zmiennych.

if ... else ...

Jest postaci: `if (<BoolExpr>) { [Stmt] }` lub `if (<BoolExpr>) { [Stmt] } else { [Stmt] }` gdzie jest wyrażeniem, które oblicza się do wartości typu boolean.

while ...

Jest postaci: `while (<BoolExpr>) { [Stmt] }` gdzie jest wyrażeniem, które oblicza się do wartości typu boolean.

print

Jest postaci: `print(<StringExpr>);` gdzie jest wyrażeniem, które oblicza się do wartości typu string.

Obsługa błędów

- Błędy związane ze złym typowaniem obsługiwane są przed uruchomieniem programu.
- Błędy związane z wykonaniem (próba odwołania do nieistniejącego indeksu tablicy, dzielenie przez 0 itp.) obsługiwane są w czasie działania programu.

Podsumowanie

Język zawiera:

1. Wymagania na 15 punktów

- trzy typy
- literały, arytmetykę, porównania
- zmienne, przypisanie
- print
- while, if
- funkcje, rekurencja
- przekazywanie argumentu przez zmienną/przez wartość

2. Dodatkowe wymagania na 20 punktów:

- przesłanianie, statyczne wiązanie
- obsługa błędów wykonania
- funkcje zwracające wartości

3. Statyczne typowanie (+4 pkt)
4. Tablice indeksowane int (+2 pkt)
   Razem: 26 pkt