JPP zadanie 2 – deklaracja języka

Bartłomiej Karwowski
385713

1 Wstęp

Opisany w tym dokumencie język programowania został stworzony na podstawie języka *Latte* ze strony:

`https://www.mimuw.edu.pl/~ben/Zajecia/Mrj2018/Latte/`

2 Gramatyka języka

Opis gramatyki w notacji EBNF znajduje się w pliku **gram.cf**

3 Przykładowe programy

Przykładowe programy znajdują się w folderze **prog** w podfolderach **good** (poprawne programy) oraz **bad** (niepoprawne programy).

- latte1/2/3.in — przykładowe programy z *Latte*.
- *.out — output przykładowych programów

4 Opis języka

Opis głównej części języka jest analogiczny do *Latte*:
„Program w języku Latte jest listą definicji funkcji. Na definicję funkcji składa się typ zwracanej wartości, nazwa, lista argumentów oraz ciało. Funkcje muszą mieć unikalne nazwy. W programie musi wystąpić funkcja o nazwie main zwracająca int i nie przyjmująca argumentów (od niej zaczyna się wykonanie programu). Funkcje o typie wyniku innym niż void muszą zwracać wartość za pomocą instrukcji return. Funkcje mogą być wzajemnie rekurencyjne; co za tym idzie mogą być definiowane w dowolnej kolejności (użycie funkcji może występować przed jej definicją)."

4.1 Wywołania funkcji i przesłanianie identyfikatorów

Wszystkie parametry są przekazywane przez wartość. Wewnątrz funkcji parametry formalne zachowują się jak zmienne lokalne. Przesłanianie identyfikatorów odbywa się ze statycznym ich wiązaniem, zaś reguły redeklaracji zmiennych są wzorowane na języku C. Jedynym uchyleniem od tej zasady jest możliwość deklarowania funkcji i zmiennych o tej samej nazwie.

4.2 Zmienne globalne

Zmienne globalne można deklarować w dowolnej kolejności (tak jak funkcje). Oznacza to, że poza ciałem funkcji, na początku działania programu, można przypisać im tylko stałe wyrażenia (tzn. nie można przypisać jednej zmiennej wartości drugiej). W ciele funkcji przypisania są dowolne (tak jak dla zmiennych lokalnych).

4.3 Instrukcje i wyrażenia

Wszystkie instrukcje i wyrażenia które są w *Latte* zachowują się w tym języku identycznie.

4.4 Napisy

Napisy, jak w *Latte*, mogą występować tu jako: literały, wartości zmiennych, argumentów i wyników funkcji. Różnica jest taka, że tutaj napisy jako parametry funkcji przekazywane są przez wartość. Nie można także konkatenować napisów operatorem +.

4.5 Predefiniowane funkcje

W języku występują następujące predefiniowane funkcje, których działanie identyczne jak w *Latte* (w przypadku wypisywania outputu w dwóch wersjach (odpowiednio — bez znaku końca linii oraz z nim):

- void printInt(int), void printIntLn(int)

- void printString(string), void printStringLn(string)
- void error()

4.6 Inne instrukcje

- Pętla for (for i = a to b) — tak jak w wymaganiu punktowym nr 7, zmienna i jest *read-only*, zaś wartość b jest wyliczana tylko raz na początku pętli.
- break/continue — analogicznie jak np. w C++.

4.7 Błędy wykonania

Język obsługuje następujące błędy wykonania programu:

- dzielenie przez 0
- wyrażenie modulo 0
- wyżej wymieniona funkcja void error()

5 Oczekiwana liczba punktów

Spodziewana ilość punktów za poprawne wykonanie wszystkich poniższych zagadnień — **25 pkt.**

5.1 Na 20 pkt.

1. Co najmniej trzy typy wartości: int, bool i string.
2. Literały, arytmetyka, porównania.
3. Zmienne, operacja przypisania
4. Jawne wypisywanie wartości na wyjście (instrukcja lub wbudowana procedura print).
5. while, if (z else i bez).
6. Funkcje lub procedury (bez zagnieżdżania), rekurencja.
7. b) Zmienne „read-only" i użycie ich np. w implementacji pętli for w stylu Pascala.
8. Przesłanianie identyfikatorów ze statycznym ich wiązaniem (zmienne lokalne i globalne).
9. Obsługa błędów wykonania, np. dzielenie przez zero.
10. Funkcje przyjmujące i zwracające wartość dowolnych obsługiwanych typów.

5.2 Dodatkowe punkty

- 4 pkt. – Statyczne typowanie (tj. zawsze terminująca faza kontroli typów przed rozpoczęciem wykonania programu).
- 1 pkt. – Operacje przerywające pętlę while – break i continue.