

# JPP zadanie 2 - deklaracja języka

Bartłomiej Karwowski  
385713

## 1 Wstęp

Opisany w tym dokumencie język programowania jest 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**.

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

## 4 Opis języka

Program w tym języku (tak jak w *Latte*) jest listą definicji funkcji, a także dodatkowo definicji zmiennych globalnych. Pozostałe części głównej struktury programu są identyczne jak w wymienionym języku, włącznie z wzajemną rekurencją funkcji (zmienne globalne także mogą być deklarowane w dowolnej kolejności).

Wszystkie instrukcje i wyrażenia które są w *Latte* zachowują się w tym języku identycznie. Typy i predefiniowane funkcje także są identyczne. Obsługa parametrów funkcji także jest analogiczna (przekazywanie argumentów przez wartość).

### 4.1 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++.

## 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. Przesłanie 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.