

Mój język będzie wzorowany na języku Latte z mrjp. Dla ustalenia uwagi nazwę go Flatter - z francuskiego "schlebiać" co sugeruje że wszystko powinno się udać.

Wyrażenia są obliczane od lewej do prawej

Funkcje print wypisują argument i znak nowej linii.

Funkcje muszą kończyć się return-em i musi to być ostatnia instrukcja w funkcji.

Przykłady w folderach good/bad.

FUNKCJONALNOŚĆ	STAN	PUNKTY
----------------	------	--------

Na 15 punktów

1) trzy typy: int, string, bool	✓	15 pkt
2) literały, arytmetyka, porównania	✓	
3) zmienne, przypisanie	✓	
4) wypisywanie na wyjście: println, printString, printBool	✓	
5) while, if	✓	
6) funkcje, rekurencja	✓	
7) przekazywanie przez zmienną / przez wartość, przez zmienną oznaczamy przez ref	✓	

Na 20 punktów

8) przesłanianie i statyczne wiązanie	✓	10 pkt
9) obsługa błędów wykonania	✓	
10) Funkcje przyjmujące i zwracające wartość dowolnych	✓	

obsługiwanych typów (zasadniczo to co na 15)		
--	--	--

Na 30 punktów

11) Statyczne typowanie (tj. zawsze terminująca faza kontroli typów przed rozpoczęciem wykonania programu) – 4pkt	✓	4 pkt
12) Dowolnie zagnieżdżone definicje funkcji z zachowaniem poprawności statycznego wiązania identyfikatorów (jak w Pascalu) – 2 pkt	✓	2 pkt
13) Funkcje jako parametry, zwracanie funkcji w wyniku, domknięcia à la JavaScript. funkcje anonimowe – 4pkt,	✓	4 pkt

RAZEM	30 pkt
-------	--------

funkcje anonimowe mają syntax np: (ref int x, bool b) => string { ... };
funkcja która jako argument przyjmuje int-a przez reference i bool-a przez wartość i zwraca string-a