

Tempest

Opis języka

Jakub Grondziowski
jg417886

Język Tempest jest językiem imperatywnym w pełni zainspirowany językiem latte. Ma standardową składnię i nie zawiera żadnych niestandardowych konstrukcji. Program napisany w języku Tempest powinien mieć rozszerzenie .tms. Pełny opis wybranych przeze mnie podpunktów do zrealizowania:

01. trzy typy:

Tak jak w latte, język Tempest posiada 3 typy - int, boolean oraz string.

02. literały, arytmetyka, porównania:

Żadnych zmian względem latte.

03. zmienne, przypisanie:

Standardowa składnia postaci Typ Nazwa "=" Wyrażenie. W Tempeście nie ma możliwości deklarowania zmiennych bez podania ich wartości.

04. print:

W Tempest istnieją 3 rodzaje printów: printString, printBool oraz printInt, każdy przewidziany na odpowiedni typ.

05. while, if:

Z uwagi na możliwe konflikty Tempest wymaga stosowania bloków zamiast instrukcji (blok to instrukcja otoczona nawiasami wąsatymi) zarówno w instrukcjach while, jak i w instrukcjach if (nie tyczy się warunków).

06. funkcje lub procedury, rekurencja:

Standardowa składnia deklaracji funkcji postaci
ZwracanyTyp Nazwa(ListaArgumentów) {Instrukcja}. ZwracanyTyp może być jednym z 3 dostępnych typów lub "void" dla procedur. ListaArgumentów to lista postaci Typ Nazwa, gdzie każdy element listy jest rozdzielony przecinkami.

07. przez zmienną / przez wartość / in/out:

Przekazywanie argumentów do funkcji przez wartość to standardowa deklaracja funkcji opisana w 06. Aby przekazać argument przez zmienną, należy dodać między Typem a Nazwą argumentu znak "@" (w deklaracji funkcji).

08. –

09. przesłanie i statyczne wiązanie:

Standardowe.

10. obsługa błędów wykonania:

Standardowe.

11. funkcje zwracające wartość:

Opisane w 06.

12. statyczne typowanie:

Standardowe, z Type Checkerem.

13. funkcje zagnieżdżone ze statycznym wiązaniem:

Wewnątrz funkcji będzie możliwe tworzenie funkcji zagnieżdżonych o składni tej samej co ta opisana w 06.