

Autocompleter

1 second, 32 MB

1 Background

ภาษา TUMSO (The Untyped Microlanguage without Strings and Objects) เป็นภาษาที่ใช้สำหรับการคำนวณจำนวนเต็ม

1.1 ภาษา TUMSO

ไวยากรณ์ของภาษา TUMSO มีดังนี้ (ให้อ่าน `::=` ว่า "คือ" และอ่าน `|` ว่า "หรือ")

```
<program> ::= <expression>

<expression> ::= <number>
                | TRUE
                | FALSE
                | <identifier>
                | (CALL <expression> <expression>)
                | (FUNCTION (<identifier>) <expression>)
                | (LET <identifier> = <expression> IN <expression>)
                | (IF <expression> THEN <expression> ELSE <expression>)
                | (PLUS <expression> <expression>)
                | (MINUS <expression> <expression>)
                | (IS_EQUAL <expression> <expression>)
```

- `<number>` เป็นจำนวนเต็ม ตัวอย่างเช่น `-1234`
- `<identifier>` เป็นชื่อตัวแปร ประกอบไปด้วยอักขระในช่วง `a` ถึง `z` และเครื่องหมาย `_` เช่น `the_sum_of_two_numbers`

โทเคนในภาษานี้ได้แก่ ตัวเลข `TRUE` `FALSE` `CALL` `FUNCTION` `LET` `=` `IN` `IF` `THEN` `ELSE` `PLUS` `MINUS` `IS_EQUAL` และตัวแปร โดยโทเคนจะต้องถูกแยกออกจากกันด้วยช่องว่าง วงเล็บ หรืออักขระขึ้นบรรทัด

1.2 ความหมายของโครงสร้างทางภาษา

- สำหรับตัวเลข และ `TRUE` กับ `FALSE` จะได้ค่าผลลัพธ์เป็นเป็นค่าตัวเลข ค่า `TRUE` หรือค่า `FALSE` นั้น ๆ
- สำหรับตัวแปร จะได้ค่าผลลัพธ์เป็นค่าที่ตัวแปรนั้นเก็บอยู่
- สำหรับ `(CALL <f> <a>)` ค่าผลลัพธ์ของ `<f>` ควรจะเป็นค่าฟังก์ชัน และได้ค่าผลลัพธ์เป็นค่าของการเรียกใช้ฟังก์ชันนั้นด้วยพารามิเตอร์ซึ่งก็คือค่าผลลัพธ์ของ `<a>`
- สำหรับ `(FUNCTION (<a>) <e>)` จะได้ค่าผลลัพธ์เป็นค่าฟังก์ชันนั้น ๆ ที่มีพารามิเตอร์คือ `<a>` และเมื่อมีการเรียกฟังก์ชันด้วยพารามิเตอร์ `<v>` จะสร้างตัวแปร `<a>` ขึ้นมาเก็บค่า `<v>` และได้ค่าผลลัพธ์เป็น `<e>` (โดยที่ `<a>` สามารถถูกอ้างถึงได้ใน `<e>`)
- สำหรับ `(LET <x> = <v> IN <e>)` จะสร้างตัวแปร `<x>` ขึ้นมาเก็บค่าของ `<v>` และได้ค่าผลลัพธ์เป็น `<e>` (โดยที่ `<x>` สามารถถูกอ้างถึงได้ในทั้ง `<v>` และ `<e>`)
- สำหรับ `(IF <cond> THEN <then> ELSE <else>)` หาก `<cond>` มีค่าคือผลลัพธ์คือ `TRUE` จะได้ค่าผลลัพธ์เป็น `<then>` แต่หาก `<cond>` มีค่าคือผลลัพธ์คือ `FALSE` จะได้ค่าผลลัพธ์เป็น `<else>`
- สำหรับ `(PLUS <a>)` ค่าผลลัพธ์ของ `<a>` และ `` ควรจะเป็นตัวเลข และได้ค่าผลลัพธ์เป็น `<a> + `

- สำหรับ (MINUS <a>) ค่าผลลัพธ์ของ <a> และ ควรจะเป็นตัวเลข และจะได้ค่าผลลัพธ์เป็น <a> –
- สำหรับ (IS_EQUAL <a>) ค่าผลลัพธ์ของ <a> และ ควรจะเป็นตัวเลข และจะได้ค่าผลลัพธ์เป็น TRUE หาก <a> = และจะได้ค่าผลลัพธ์เป็น FALSE หาก <a> ≠

1.3 ตัวอย่างโปรแกรมในภาษา TUMSO

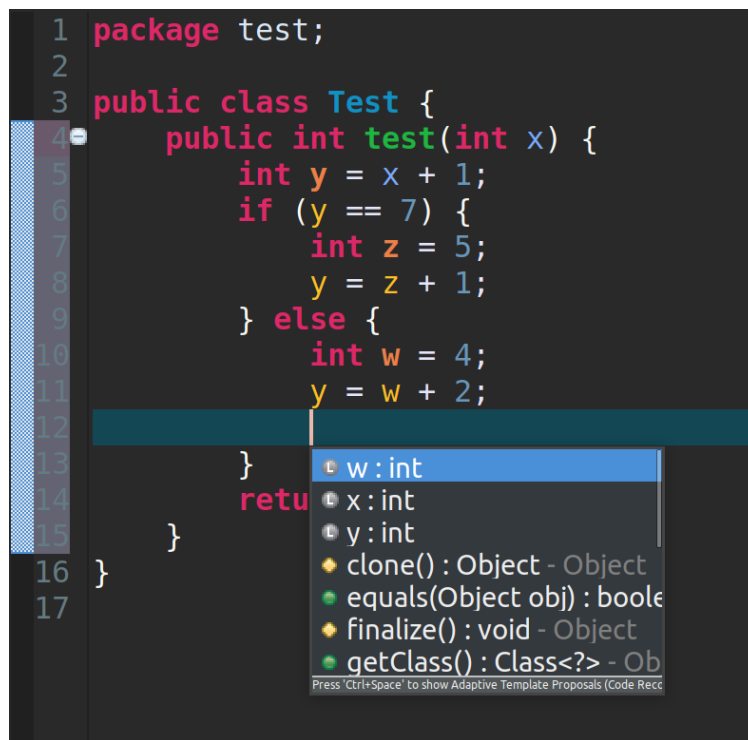
โปรแกรม	ผลลัพธ์
FALSE	FALSE
<pre>(LET a = 1 IN (PLUS (LET a = 2 IN a) a))</pre>	3
<pre>(LET mult = (FUNCTION (a) (FUNCTION (b) (IF (IS_EQUAL 0 a) THEN 0 ELSE (PLUS b (CALL (CALL mult b) (MINUS a 1)))))) IN (LET fact = (FUNCTION (n) (IF (IS_EQUAL 0 n) THEN 1 ELSE (CALL (CALL mult n) (CALL fact (MINUS n 1)))))) IN (CALL fact 5)))</pre>	120
a	Error: 'a' is undefined
(CALL 1 2)	Error: type mismatch

2 Task

ปัญหาการเรียกใช้ตัวแปรที่ไม่ได้ถูกนิยามไว้เป็นปัญหาที่พบได้ในภาษาโปรแกรมส่วนใหญ่ รวมถึงภาษา TUMSO ด้วย (ดูตัวอย่างโปรแกรมที่ 4 เป็นต้น) บาง IDE (Integrated development environment) เช่น Eclipse ของภาษา Java มีเครื่องมือแนะนำตัวแปรที่สามารถใช้ได้ ในตำแหน่งที่เคอร์เซอร์กำลังอยู่ เพื่อที่คุณจะได้ไม่เขียนโปรแกรมผิดตั้งแต่แรก

หน้าที่ของคุณคือให้เขียนโปรแกรมรับโค้ดที่ไม่สมบูรณ์ในภาษา TUMSO โดยมีสัญลักษณ์ # อยู่หนึ่งที่ในตำแหน่ง <expression> (แสดงถึงเคอร์เซอร์ในโปรแกรมที่กำลังเขียนอยู่) และแสดงผลตัวแปรทั้งหมดที่สามารถใช้ได้ทั้งตำแหน่ง #

```
1 package test;
2
3 public class Test {
4     public int test(int x) {
5         int y = x + 1;
6         if (y == 7) {
7             int z = 5;
8             y = z + 1;
9         } else {
10            int w = 4;
11            y = w + 2;
12        }
13    }
14    return y;
15 }
16 }
17 }
```

A screenshot of the Eclipse IDE interface. The main editor shows a Java file named 'Test.java' with the following code:

```
package test;

public class Test {
    public int test(int x) {
        int y = x + 1;
        if (y == 7) {
            int z = 5;
            y = z + 1;
        } else {
            int w = 4;
            y = w + 2;
        }
    }
    return y;
}
```

 The cursor is positioned at the end of the line 'y = w + 2;'. A tooltip is visible, showing the variable 'w' is of type 'int'. The tooltip also lists several methods: 'clone() : Object - Object', 'equals(Object obj) : boolean', 'finalize() : void - Object', and 'getClass() : Class<?> - Object'. At the bottom of the tooltip, it says 'Press \'Ctrl+Space\' to show Adaptive Template Proposals (Code Recommender)'.

รูปที่ 1: ตัวอย่างโปรแกรม Eclipse ที่แนะนำตัวแปรที่สามารถใช้งานได้ในตำแหน่งที่เคอร์เซอร์อยู่สำหรับภาษา Java

3 Input

ประกอบไปด้วยโค้ดที่ไม่สมบูรณ์ในภาษา TUMSO ซึ่งมีอักขระ # อยู่หนึ่งที่ในตำแหน่ง `<expression>` และหากแทนที่อักขระ # ในโค้ดนี้ด้วย `<expression>` ใด ๆ (เช่น 0) จะทำให้กลายเป็นโปรแกรมที่มีไวยากรณ์ถูกต้องในภาษา TUMSO โค้ดนี้จะมีความยาวที่บรรทัดก็ได้

4 Output

มี n บรรทัด โดย n คือจำนวนตัวแปรที่สามารถใช้ได้ตำแหน่ง # และแต่ละบรรทัดมีชื่อของตัวแปรที่สามารถใช้ได้ดังกล่าวในลำดับพจนานุกรม (lexicographic order)

5 Constraints

โค้ดที่จะมีขนาดไม่เกิน 10^6 ไบต์

6 Sample

Input	Output
<pre>(LET mult = (FUNCTION (a) (FUNCTION (b) (IF (IS_EQUAL 0 a) THEN 0 ELSE (PLUS b (CALL (CALL mult b) (MINUS a 1)))))) IN (LET fact = (FUNCTION (n) (IF (IS_EQUAL 0 n) THEN 1 ELSE (CALL (CALL mult n) (CALL fact (MINUS n 1)))))) IN (CALL fact #)))</pre>	factorial mult
<pre>(IF TRUE THEN a ELSE (LET x = 1 IN #))</pre>	x