

ЗЗС

спец. Информатика

Задачи за задължителна самоподготовка

11.11.2014

Внимание: *Задачите не изискват реализация на съответните методи print (...) за новите класове от йерархията. Реализацията на print за всеки нов клас можете да разглеждате като (важна) допълнителна задача за упражнение.*

Задача. Разработеният на лекции интерпретатор да се разшири с поддръжка на операторите – и /.

Задача. Разработеният на лекции интерпретатор да се разшири с поддръжка на оператор if. БНФ на операцията е:

```
<ifExpr> ::= if <expr> then <expr> else <expr>
```

Семантиката на оператора е следната: ако стойността на първия операнд е различна от 0, стойността на целия израз е равна на стойността на първия операнд. В противен случай, стойността на целия израз е равна на стойността на втория операнд. Например if 0 then 1 else 2 се оценява като 2, а if 1 then 1 else 2 се оценява като 1. Внимание: както всички други оператори, този оператор може да е операнд на друг оператор, например:

```
+ 1 if 0 then 1 else 2
```

е правилен израз, който има стойност 3.

Задача. Разработеният на лекции интерпретатор да се разшири с поддръжка на операцията k+. БНФ на операцията е:

```
<kPlusExpr> ::= k+ <number> <expr> ... <expr>
```

Приемаме, че броят на изразите е равен на стойността на числото, следващо оператора k+. Например: k+ 4 1 2 3 4 е правилно построен израз. Семантиката на оператора е да намира сумата на всичките си операнди.

Задача. Разработеният на лекции интерпретатор да се разшири с поддръжка на операцията `max`. БНФ на операцията е:

`<maxExpr> ::= max (<expr> ... <expr>)`

Броят на изразите в скобите може да е 1 или повече. Например: `max (1 2 3 4)` е правилно построен израз. Семантиката на оператора е да намира най-голямата измежду стойностите на изразите в скобите.

Задача. Йерархията от възли на синтактичното дърво, разработена на лекции, да се обогати със следните чисто виртуални методи в базовия клас `Expression` и съответните им реализации за всеки наследник на `Expression`:

- `int countConstants ()`: намира броя на всички константи, участващи в израз. Например, в следния израз `* + 1 2 + 4 5` има 4 константи.
- `int treeHeight ()`: Намира височината на синтактичното дърво на израза, т.е. максималния брой вложения. Следва списък с примерни изрази и височините на съответните им синтактични дървета.
 - `1` : височина 1
 - `+ 1 2` : височина 2
 - `+ + 1 2 3` : височина 3
 - `+ + 1 2 + 1 4` : отново височина 3
 - `+ 1 + 1 + 1 2` : височина 4