

BASIC-lite interpreter - Specifikáció

Írta: Szenes Márton Miklós, Neptun kód: KTZRDZ

Rövid leírás

A program egy **BASIC**-szerű programozási nyelv butított, egyszerűsített változatát valósítja meg, továbbiakban **BASIC-lite**-nak nevezve. Biztosít a programkód írásához egy interfészt, alap parancsokat a kód szereksztéséhez, mentéséhez, beolvasásához és futtatásához.

Az értelmező képes regiszterekben számértékeket eltárolni és azokkal műveleteket végezni, feltételes utasításokat végrehajtani, és ugrani a programkódon belül, kiírni a standard kiementre, és olvasni a standard bementről.

Alapértelmezetten két regisztere van az értelmezőnek, melyek nevei: `a` , `i` . A regiszterek számát dinamikusan növeli az értelmező, ha szükséges.

Példaprogram

Az alábbi **BASIC-lite** program például kiírja az első 5db páratlan számot.

```
10 let a=1
20 let i=5
25 print "Első 5 paratlan szám: "
30 print a
35 print " "
40 let a=a+2
50 let i=i-1
60 if i>0
70 goto 30
```

IDE - Interfész

A program indulásakor egy CLI-s felület fogadja a felhasználót. Itt az alábbi parancsok adhatóak ki:

- `HELP` : Kiírja az interfész parancsait, és működésüket
- `RUN` : Futtatja a betöltött programot.
- `END` : Lezárja az aktuális interfészt (kód szerkesztő/alkalmazás).

- **LIST** : Kiírja a betöltött programot sorszám szerint növekvő sorban.
- **NEW** : Új programot hoz létre.
- **LOAD** <fájlnév> : Beolvassa fájlból a programot a kapott fájl névvel.
- **SAVE** <fájlnév> : Elmenti a betöltött programot a megadott fájl névvel.

Kódolás

Egy sor begépelésekor a sor végén `Enter` -t leütve a program kiértékeli az utasítást.

- Ha interfész utasítás, akkor végre hajtja az adott utasítás szerint az IDE-ben a változtatásokat.
- Ha program kódsor, akkor eltárolja azt az értelmező a memóriájában.
- Ha a program kódsor sorszáma negatív, akkor az annak a sorszámnak vett abszolút értékű utasítást törli az értelmező memóriájából, ha van ilyen.

Az interfész utasítás abban különbözik a program kódsortól, hogy a kódsor első argumentuma egy sorszám, míg az IDE utasítás első argumentuma nem tartalmazhat számot.

BASIC-lite szintaxis

Egy program kódsornak 3 argumentuma van mindig: `sorszám` , `utasítás` , `paraméter` .

Ezen paraméterek egymástól legalább egy szóközzel kell legyenek elválasztva.

A paraméteren belül tetszőleges 'whitespace' lehet, mivel az értelmező törli majd ezeket.

Ezért fontos, hogy ha két karaktersorozatot egymás mellé írunk egy szóközzel elválasztva, úgy azt az értelmező egy szóként fogja kezelni.

Ezalól kivétel ha sztringet írunk be a `print` utasításhoz, aminél természetesen nem törlődnek a 'whitespace' karakterek.

Így például a `10 let a = 4 * (b - c)` sort így bontja fel:

Sorszám	Utasítás	Praméter
10	let	$a=4*(b-c)$

Ahol az `a` lesz a balérték, és a $4*(b-c)$ az értékadás jobbértéke, ahol `b` és `c` regiszterneveket jelölnek, és annak értékeire hivatkoznak.

Sorszám:

Egy program kódsor sorszám egy 0-nál nagyobb pozitív egész szám mindig.

Amennyiben a sorszám 0, úgy az a sor kommentnek tekintendő, és nem kerül kiértékelésre a futtatás

során.

Ha a sorszám negatív, úgy a fent említett módon törlődik az utasítás a program memóriából. Minden más esetben, ha az első argumentum nem egy egész szám, úgy a program hibát dob.

Utasítás és paraméter:

A második paraméter az utasítás kulcsszó. Ezután következik a harmadik paraméter, ami egészen a sor végéig tart.

A program 5 féle utasítást tud értelmezni. Ezek a következők, és a szintaktikájuk:

- `let <regiszter> = <érték>` : Regiszternek értékadás. Az érték tartalmazhat matematikai alapműveleteket és zárójeleket. (`+` , `-` , `*` , `/`)
- `print <regiszter>/<string>` : Kiírja a regiszter vagy a kapott idézőjelek közé tett sztring értékét a szabványos kimenetre. A sztring tartalma kizárólag az angol abc nagy- és kisbetűit tartalmazhatja, illetve `\n` (sortörés), `\t` (tab), `\"` (idézőjel) speciális karaktereket.
- `if <feltétel>` : Feltételes elágazás. Ha a feltétel igaz, akkor végrehajtja a következő utasítást a sorban, ellenkező esetben az következő utáni utasításra ugrik a program. A feltétel tartalmazhat számokat, regisztereket, összehasonlító operátorokat, és/vagy/nem logikai kapukat és zárójeleket. (`>` , `>=` , `<` , `<=` , `==` , `!=` , `&&` , `||` , `!`)
- `goto <sorazonosító>` : Ha létezik a sorazonosító, akkor a megjelölt sorazonosítóhoz ugrik a program. Ha nincs ilyen, akkor hibát dob az értelmező.
- `read <regiszter>` : Beolvas a szabványos bemenetről egy számot és eltárolja az értéket a regiszterben.

Hibakezelés

Az **BASIC-lite** értelmező minden lehetséges kód elírásra dobjon kivételt, mely tartalmazza a hiba indokát.