Algoritmi in podatkovne strukture 1

Naloga 1: Programabilni kalkulator

V programskem jeziku java napišite programabilni kalkulator, ki omogoča napredno računanje (programiranje) s celimi (int) števili in nizi. Kalkulator omogoča:

- računanje aritmetičnih izrazov v obratnem poljskem zapisu (angl. reverse polish notation)
- preprosto konkatenacijsko programiranje (s pomočjo sklada)
- izvedbo preprostih programskih konstruktov (pogojev, zank, funkcij).

Kalkulator naj izraze bere s standardnega vhoda. Vsaka vrstica predstavlja en izraz, sestavljen iz več delov (nizov) - vsak predstavlja bodisi veljaven ukaz, celo število ali niz znakov. Med posameznimi dela izraza je vedno (vsaj) en presledek. Predpostavite lahko, da so pri testiranju vsi vhodi sintaktično in semantično pravilni.

Rešitev implementirajte s pomočjo 42 skladov: glavnega (indeks 0) in pomožnih (z indeksi od 1 do 41). Velika večina ukazov deluje nad glavnim skladom (z indeksom 0. Pri ukazih, ki se lahko izvajajo nad poljubnim skladom, je indeks sklada posebej določen s številom na vrhu glavnega sklada.

Kalkulator naj podpira naslednje ukaze oz. operacije (nad glavnim skladom):

- `echo` - v vrstici izpiše vrh sklada (sklad pusti nespremenjen); če je sklad prazen, izpiše prazno vrstico
- `pop` - odstrani vrh sklada
- `dup` - podvoji vrh sklada (x -> x x)
- `dup2` - podvoji par na vrhu sklada (x y -> x y x y)
- `swap` - zamenja vrhnja dva elementa sklada (x y -> y x)

Naslednje operacije zamenjajo vrh glavnega sklada z ustreznim rezultatom (x -> y):

- `char` - vrh sklada zamenja z znakom, ki ima ASCII/Unicode kodo vrha sklada
- `even` - vrh sklada zamenja z 1, če je vrh sod, sicer z 0
- `odd` - vrh sklada zamenja z 1, če je vrh lih, sicer z 0
- `!` - vrh sklada zamenja s faktorielo vrha
- `len` - vrh sklada zamenja z dolžino elementa na vrhu

Naslednje operacije zamenjajo vrhnja dva elementa glavnega sklada z ustreznim rezultatom (x y -> r):

- `<>` - primerja zgornja dva elementa (x y) sklada in na sklad porine 1 (če x <> y) ali 0 (če x == y)
- `<` - primerja zgornja dva elementa sklada in na sklad porine 1 (če x < y) ali 0 (sicer)
- `<=` - primerja zgornja dva elementa sklada in na sklad porine 1 (če x <= y) ali 0 (sicer)
- `==` - primerja zgornja dva elementa sklada in na sklad porine 1 (če x == y) ali 0 (sicer)
- `>` - primerja zgornja dva elementa sklada in na sklad porine 1 (če x > y) ali 0 (sicer)
- `>=` - primerja zgornja dva elementa sklada in na sklad porine 1 (če x >= y) ali 0 (sicer)
- `+` - na sklad porine vsoto vrhnjih dveh elementov sklada
- `-` - na sklad porine razliko vrhnjih dveh elementov sklada
- `*` - na sklad porine zmnožek vrhnjih dveh elementov sklada
- `/` - na sklad porine kvocient (celoštevilsko deljenje) vrhnjih dveh elementov sklada
- `%` - na sklad porine ostanek po deljenju elementa pod vrhom z elementom na vrh
- `.` - stakne (združi, zlepi) vrhnja dva elementa sklada v en element (x y -> xy)
- `rnd` - na sklad porine naključno število, ki ima vrednost >= x in <= y

Naslednje operacije omogočajo izvedbo pogojnega stavka (izpolnjenost pogoja hranimo v interni spremeljivki):

- `then` – z glavnega sklada vzame vrhnje število; če je to različno od 0, nastavi izpolnjenost pogoja na `true`
- `else` – zanika izpolnjenost pogoja
- `?...` – vsak ukaz, ki se začne z `?`, se izpolni (ali pa ne) glede na prednastavljeno izpolnjenost pogoja

Za delo s poljubnim skladom (glavnim ali pomožnimi) imamo na voljo spodnje ukaze. Pri tem velja, da število na vrhu glavnega sklada določa indeks sklada, nad katerim se izvaja ukaz:

- `print` - v vrstici izpiše vsebino sklada (z indeksom, ki je podan na vrhu glavnega sklada) od dna do vrha
- `clear` – izprazne sklad (z indeksom, ki je podan na vrhu glavnega sklada)
- `run` – izvede vse ukaze na (pomožnem) skladu (z indeksom, ki je podan na vrhu glavnega sklada) od dna do vrha (sklad ostane nespremenjen)
- `loop` - izvede vse ukaze na (pomožnem) skladu (z indeksom, ki je podan na vrhu glavnega sklada) od dna do vrha (sklad ostane nespremenjen), pri čemer to ponovi tolikokrat, kot je podano s številom pod vrhom sklada
- `fun` – na pomožni sklad (z indeksom, ki je podan na vrhu glavnega sklada) zapiše toliko naslednjih ukazov, kolikor določa število pod vrhom glavnega sklada
- `move` – z glavnega sklada prenese na pomožni sklad (z indeksom, ki je podan na vrhu glavnega sklada) toliko elementov, kolikor določa število pod vrhom glavnega sklada (elementi se prenesejo eden za drugim)
- `reverse` - obrne vrstni red vseh elementov na skladu (z indeksom, ki je podan na vrhu glavnega sklada) - u v x y z -> z y x v u

Če ukaz ni na seznamu zgoraj naštetih, potem gre za število ali niz, ki se porine na vrh glavnega sklada (`push`).

Primeri izvajanja

V vseh primerih so vrstice, ki se začnejo z `>`, vnešene v program, ostale pa so izpis programa. Program zaženemo z

```
java Naloga1
```

Nekaj primerov izvajanja operacij:

```
> 0 -12 103 3131 -100 53 111 dup2 0 reverse 0 print
111 53 111 53 -100 3131 103 -12 0

> 41051 141 + echo -100 50 - echo
41192
-150

> 5 11 17 + + 10 * 0 print
330

> 5 ! echo even 0 print
120
1

> 70 90 rnd echo char echo
66
B

> 0 1 2 3 4 4 1 fun dup 0 reverse swap 2 2 move 0 print 1 print 2 print
0 1 2
dup 0 reverse swap
4 3

> 0 1 2 3 1 fun 0 reverse dup 1 run 0 print 2 1 loop 0 print
2 1 0 0

2 2 1 0 0 0

> 5 3 1 2 5 1 fun == then ?dup2 else ?+ 1 run 0 print
8

> 9 1 fun dup 0 reverse swap % dup then ?1 ?run 24 10 0 print 1 run pop echo
24 10
2

> 3 1 fun 0 100 rnd 3 2 fun 5 1 loop  7 3 fun dup2 <= then ?pop else ?swap ?pop 3 4 fun 4 3 loop

1 print 2 print 3 print 4 print 2 run 0 print 4 run 0 print
0 100 rnd

5 1 loop

dup2 <= then ?pop else ?swap ?pop

4 3 loop

34 96 12 48 25
12
```

Drugi napotki in namigi

Pri izvedbi lahko uporabite le javansko knjižnico `java.util.Scanner`. Vse ostale knjižnice niso dovoljene oz. potrebne - izrecno niso dovoljene knjižnice, ki vsebujejo javanske zbirke (npr. `ArrayList`) iz Collection Framework-a

Za predstavitev skladov najprej zapišite vmesnik (abstraktni podatkovni tip) `Stack` z ustreznimi operacijami, nato pa ga implementirajte s pomočjo javanskega polja. Velikost sklada lahko omejite na 64. Za izvedbo več skladov napišite še vnesnik `Sequence`.

Upoštevajte, da je celoten izraz (program) napisan v eni vrstici, torej so pred obdelavo vsake vrstice vhoda vsi skladi prazni ter izpolnjenost pogoja nastavljena na FALSE.

Kot aritmetične operacije lahko uporabljate operacije nad javanskimi primitivnimi tipi (`int`), za določanje naključnega števila pa metodo `Math.random()`.

Oddaja naloge

Vse podrobnosti za avtomatsko ocenjevanje in oddajo so (bodo) predstavljene v navodilih za oddajo domačih nalog.

Veliko uspeha pri delu!

Status oddaje naloge

Status oddaje naloge	Pri tej nalogi vam ni treba oddati ničesar.
Stanje ocen	Neocenjeno
Rok za oddajo	nedelja, 2. december 2018, 23:55
Preostali čas	Rok za oddajo naloge je potekel
Zadnja sprememba	-
Komentar oddaje	Komentarji (0)