Druga domaća zadaća iz kolegija Interpretacija programa – dokumentacija

Tim M – Iva Dimnjaković, Lovro Gorup, Filipa Lovrić

Uvod

U ovoj domaćoj zadaći odabrana je tema kineziologije, točnije, dio kineziologije koji se bavi numeričkim aspektom rezultata sportaša. Ideja je, razvojem specijaliziranog jezika, omogućiti kineziolozima jednostavno definirianje sportaša s njegovim fizičkim karakteristikama (visina i težina) te rezultata u proizvoljnoj aktivnosti. Nakon toga, moguće je podacima manipulirati i izvoditi različite analize koje mogu pomoći u raznim istraživanjima. Omogućen je interaktivni način rada tako da korisnik izvodi naredbu po naredbu, a moguće je i izvršavanje programa iz neke datoteke čiji se ispis sprema u neku drugu datoteku.

Opis jezika

Pretpostavimo da postoji sportaš John o kojem imamo podatke njegove visine i težine te njegovih rezultata duljine skoka u dalj mjerenih u centimetrima. Jezik se sastoji od sljedećih komponenti:

- sportas John{188:78} – definira se sportaš imena John visine 188 cm i težine 78 kg
- rezultati John{219.5;217;221.1;216.4}
- a = 5; - definira varijablu a s vrijednošću 5
- @John.height ILI @John->height – vraća Johnovu visinu (u ovom primjeru 188)
- @John.weight ILI @John->weight – vraća Johnovu težinu (u ovom primjeru 78)
- ~John.count ILI ~John->count – vraća broj spremljenih Johnovih rezultata (u ovom primjeru 4)
- ~John[i] ILI ~John.(i) – dohvaća Johnov i-ti rezultat ako je i iz {0, 1, …, ~John.count – 1}, inače javlja grešku
- +~~John 215; ILI ++~John 215; – ubacuje broj 215 kao zadnji rezultat
- -~~John 215; ILI --~John 215; - izbacuje broj 215 iz rezultata
- + - * / su standardne operacije s brojevnim varijablama
- loop(n) …<neki kod> … endloop; - definicija petlje čiji se unutrašnji kod izvodi točno n puta
- print "*neki tekst*"; - ispisuje tekst unutar navodnika (u ovom slučaju *neki tekst*)
- print x; – ispisuje vrijednost x-a, ako je x brojevna varijabla
- print @John – ispisuje Johnovu visinu i težinu
- print ~John – ispisuje Johnove rezultate
- function ime(param1, param2, …) … <tijelo funkcije> endfuction; – definira funkciju s imenom ime i parametrima koji mogu biti brojevnog tipa, ali i primjerice @John ili ~John
- function ime(param1, param2, …); - poziv funkcije imena ime
- $$ komentar – jednolinijski komentar

- dato datoteka.txt ILI dato datoteka – naredba koja otvara datoteku datoteka.txt izvodi program koji je zapisan u toj datoteci i vrši ispis u datoteku kinesiology.txt

Analiza zahtjeva

U ovom je projektu bilo potrebno zadovoljiti sljedeći niz zahtjeva:

- ✓ interaktivni način rada (korisnik utipkava naredbu po naredbu) – korisnik utipkava liniju po liniju te pritiskom na tipku *enter* uneseno se izvodi (oprez: kod unošenja npr. loop(n) … endloop; potrebno je sve unijeti u istom redu
- ✓ aritmetičke izraze s višemjesnim operatorima te varijable – ostvarene su standardne aritmetičke operacije te, zbog fokusa na numerički aspekt, brojevne varijable
- ✓ definicije funkcija i funkcijske pozive – jednostavna definicija funkcija pomoću ključnih riječi function i endfunction te s imenom i popisom parametara, a slično i poziv (vidi gore). Međutim, funkcije ne mogu vraćati rezultate (sve su tipa void)
- ✓ posebna tipa podataka neuobičajenih u (poznatijim) programskim jezicima; jedan tip smije biti konačan), a drugi tip mora biti potencijalno beskonačan– tip sportaša koji se definira sa sportas ime{<visina>:<tezina>} i kojem se pristupa s @ime je konačan tip, dok je tip rezultata koji se definira s rezultati ime{<1. broj>, <2. broj>, …} i kojem se pristupa s ~ime je traženi beskonačni tip (rezultata može biti proizvoljno mnogo)
- ✓ barem tri korisna operatora za rad s gornjim posebnim tipovima podataka te barem jedan alternativni način pisanja (alias) za svaki od tih operatora – operatori koji su implementirani su operator pristupa i-tom rezultatu ~ime[i] (alias ~ime.(i)), operator dodavanja rezultata +~~ime <broj> (alias ++~ime <broj>) te operator izbacivanja pojedinog rezultata -~~ime <broj> (alias --~<broj>)
- ✓ unos iz datoteke i ispis u datoteku – omogućeno je učitavanje i izvršavanje programa koji se nalazi u tekstualnoj datoteci, a to se postiže naredbom „dato ime_datoteke“
- ✓ jednolinijske komentare – početak komentara je definiran s $$ te se proteže do kraja

Dodatno, zbog smislenosti i lakše upotrebe, implementiran je i pristup varijablama visine i težine te broju spremljenih rezultata kao @ime.height, @ime.weight i ~ime.count (aliasi @ime->height, @ime->weight i ~ime->count).

Upute za pokretanje i testiranje

Program se nalazi u datoteci *kinesiology.py* te se pokreće njezinim izvršavanjem. Nakon toga će se pojaviti ispis „Unesite naredbu:“ te se može unijeti proizvoljna naredba, pritom je bitno naglasiti da su petlja i funkcije jedna naredba te se moraju u tom načinu upisati u jednom retku.

Međutim, naredbe su međusobno povezane, tj. ono što se u nekoj liniji definira, može se koristiti i u svakoj sljedećoj.

Unos naredbi se prekida upisivanje ključne riječi **exit**.

Postoje i tri primjera programa napisanih u definiranom jeziku, a nalaze se u datotekama primjer1.txt, primjer2.txt i primjer3.txt. Programi se mogu izvesti upisom, primjerice, „dato primjer1“ kao naredbe u interaktivnom načinu rada. Time će se prikazati pripadni AST-ovi u terminalu, dok će se ispis ostvariti u datoteci kinesiology.txt (pri prvom pokretanju će se stvoriti automatski takva datoteka, a nakon toga će se appendati svako novo izvođenje).