

Zadatak

Inline2018 je interpretirani programski jezik čija je svrha prikaz tekstualnog sadržaja sa interpretiranim izvršavanjem ugrađenog programskog koda. Inline2018 sintaksa je takva da izvorni fajlovi primarno sadrže tekst, uz dozvoljene istolinijske (eng. inline) simbole za formatiranje, te istolinijske simbole za izvorni kod. U nastavku je data specifikacija dozvoljenih elemenata kroz BNF notaciju:

```
<validan_red> ::= <tekst> | <deklaracija> | <ukljucivanje>
<tekst> ::= <string> | <string><rezervisani_simbol> |
<string><razmak><otvoren_tag><tekst><zatvoren_tag><string>
<otvoren_tag> ::= "<all_caps>"
<zatvoren_tag> ::= "</all_caps>"
<rezervisani_simbol> ::= "="<promjenljiva> | "@{"<razmak><izraz><razmak>"}"
<ukljucivanje> ::= "include"<razmak><putanja>
<putanja> ::= "naziv_druge_datoteke"
<deklaracija> ::= <promjenljiva><razmak> "="<razmak><izraz><razmak>
<izraz> ::= <izraz_num> | <izraz_str>
<izraz_num> ::= <promjenljiva>|<oznaceni_literal>|"("<razmak><izraz_num><razmak>")" |
<izraz_num><razmak><operator_num><razmak><izraz_num>
<izraz_str> ::= <promjenljiva>|<literal_str> |
<izraz_str><razmak><operator_str><razmak><izraz_str> |
<izraz_str><razmak><operator_str><razmak><izraz_num> |
<izraz_num><razmak><operator_str><razmak><izraz_str>
<oznaceni_literal> ::= <literal_num> | <znak><literal_num>
<znak> ::= "+" | "-"
<literal_num> ::= <cifra> | <cifra><literal_num>
<literal_str> ::= ""<string>""
<string> ::= <slovo> | <spec_karakter> | <cifra> | <slovo><razmak><string> |
<spec_karakter><razmak><string> | <cifra><razmak><string>
<operator_num> ::= "+"|"*"|" / "|"-"|"%"
<operator_str> ::= "+"
<promjenljiva> ::= "["<naziv_promjenljive>"]"
<naziv_promjenljive> ::= <slovo>|<slovo><ostatak_naziva>
<ostatak_naziva> ::= <cifra> | <slovo> | <spec_karakter> | <cifra><ostatak_naziva> |
<slovo><ostatak_naziva> | <spec_karakter><ostatak_naziva>
<slovo> ::=
"a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m"|"n"|"o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"
w"|"x"|"y"|"z"|"A"|"B"|"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M"|"N"|"O"|"P"|"Q"|"R"|"S
"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z"
<cifra> ::= "0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"
<spec_karakter> ::= "_|"."|" $"|"#"
<razmak> ::= <razmak><tab>|<tab>razmak|" " | ""
```

Pojašnjenja nekih kompleksnijih elemenata programskog jezika:

- program se sastoji iz proizvoljnog broja konstrukcija <validan_red>,
- ne postoji karakter za kraj reda, tj. redovi završavaju "newline" karakterom,
- prvo pojavljivanje promjenljive predstavlja njenu deklaraciju, a tip promjenljive određuje izraz sa desne strane operatora dodjele (=),
- nakon prve dodjele, nije dozvoljena promjena tipa promjenljive (ako dođe do promjene tipa, onda se sa lijeve strane mora naći nova promjenljiva,

- dozvoljen je proizvoljan broj kasnijih dodjela vrijednosti promjenljivoj, pod uslovom da su vrijednosti istog tipa kao i tip promjenljive (koji je određen prvom dodjelom),
- operator + je operator konkatencije za stringove. Konkatenacija bilo koje vrijednosti sa stringom daje string,
- prioriteta operatora su definisani na isti način kao kod drugih viših programskih jezika,
- ključna riječ "include" omogućava uključivanje izvornog koda iz druge datoteke koja mora biti dostupna u istom direktorijumu. Efekat uključivanja je isti kao kada je kompletan sadržaj druge datoteke naveden na mjestu uključivanja. To ujedno znači da promjenljiva može biti korištena u datoteci u kojoj je ranije uključena druga datoteka gdje se nalazi deklaracija te promjenljive,
- "tekst" je bilo koji oblik literala koji ne počinje deklaracijom ili nekom od rezervisanih riječi. Program sve redove koji označavaju tekst ispisuje na konzolu. Tekst može sadržavati sljedeće specijalne konstrukcije:
 - jednolinijske (engl. *inline*) blokove – bilo koji počinje sa @{ i završava sa }. Između ovih elemenata se može naći validan izraz,
 - ispis promjenljivih – oznaka promjenljive ispred koje se nalazi znak jednakosti (**bez razmaka**). Na ovom mjestu se u tekstu ispisuje trenutna vrijednost promjenljive,
 - tagovi – sadržaj između elemenata <all_caps> i </all_caps> mora biti ispisan svim velikim slovima. Da bi program bio ispravan, ovi tagovi moraju biti ispravno formirani (za svaki otvoren tag mora postojati i odgovarajući zatvoren).

Primjer ispravnog programskog koda na osnovu prethodne specifikacije:

Ovo je prva linija ulaznog teksta.

Ovo je druga linija ulaznog teksta.

```
[var1] = "tekst"
```

```
[var2] = 5
```

```
[var3] = 2*([var2]-3)+[var2]
```

```
[var4] = "konkatenacija" + [var3]
```

Ovo je =[var4] i <all_caps>ukljucuje</all_caps> i @[var2]=[var2]*5+3} blok inline koda

```
Ispis =[var2]
```

Ispis za prethodni program:

Ovo je prva linija ulaznog teksta.

Ovo je druga linija ulaznog teksta.

Ovo je konkatenacija9 i UKLJUCUJE i blok inline koda

```
Ispis 28
```

Ostale napomene

- projektni zadatak je potrebno implementirati u formi konzolne aplikacije kojoj se kao jedini argument komandne linije proslijeđuje naziv prve datoteke sa izvornim kodom (koja u sebi može sadržavati uključene druge datoteke)
- kao rješenje projektnog zadatka je potrebno predati kompletan izvorni kod aplikacije (ako je korišteno razvojno okruženje, onda je potrebno predati kompletan direktorijum projekta),
- dozvoljena je upotreba bilo kojeg programskog jezika,
- odbranjen projektni zadatak je uslov za izlazak na pismeni dio ispita. Studenti koji su prijavili ispit su dužni da se nekoliko dana prije termina ispita jave predmetnom asistentu kako bi se utvrdio termin odbrane projektnog zadatka,
- jednom odbranjen projektni zadatak važi do objavljivanja teksta sljedećeg projektnog zadatka