
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Programski prevodioci 1
Nastavnik: dr Dragan Bojić, vanr. prof.
Asistenti: Maja Vukasović, mast. inž.
Kristijan Žiža, mast. inž.
Školska: 2020/2021.
Ispitni rok: Julski ispitni rok

Projekat

– kompajler za Mikrojavu –

- izmene i dopune -

Važne napomene: Pre čitanja ovog teksta, **obavezno** pročitati opšta pravila predmeta i pravila vezana za izradu domaćih zadataka! Pročitati potom ovaj tekst **u celini i pažljivo**, pre započinjanja realizacije ili traženja pomoći. Ukoliko u zadatku nešto nije dovoljno precizno definisano ili su postavljeni kontradiktorni zahtevi, može se koristiti diskusiona lista za razjašnjavanje nejasnoća u zahtevima, van onoga što se može samostalno rešiti uvođenjem razumnih pretpostavki. **Srećan rad!**

1. Dodatni obavezni zahtevi za sve nivoe projektnog zadatka

Potrebno je implementirati interfejs *Compiler* koji se nalazi u priloženoj arhivi. Interfejs deklarise samo jednu metodu – *compile*. Metoda prima kao argumente putanju do ulaznog fajla koji sadrži kod koji se kompajlira i putanju do izlaznog fajla u koji je potrebno smestiti generisani bajtkod. Povratna vrednost metode je lista *CompilerError* objekata. U slučaju uspešnog kompajliranja i generisanja koda, vraća se *null* ili prazna lista. Klasa *CompilerError* opisuje greške do kojih je moguće doći prilikom kompajliranja. Nabrojanje (*enum*) *CompilerErrorType* predviđa postojanje leksičke, sintaksne i semantičke greške. Prilikom kreiranja greške zadaje se i linija na kojoj je greška otkrivena, kao i opis greške.

Na sajtu predmeta, u sekciji za domaće zadatke postavljena je arhiva u kojoj se nalaze opisani interfejs i klasa za definiciju grešaka. Sadržaj arhive je potrebno uključiti u projekat, a pomenuti interfejs implementirati prema uputstvu. Nije dozvoljeno menjanje sadržaja arhive.

2. Izmene i dopune zahteva po nivoima

I Nivo A

- Ukloniti ternarni operator:
Expr = Condition "?" Expr ":" Expr
/ CondFact "?" Expr ":" Expr.
- Dodati podršku za *if-then-else* kontrolnu strukturu:
"if" "(" Condition ")" Statement ["else" Statement]

II Nivo B

- *Switch* statement koji vraća rezultat. Povratna vrednost se specificira ključnom rečju *yield*.
- Povratne vrednosti se moraju slagati po tipu sa promeljivom u koju se vrši upis (kao i prilikom bilo koje druge dodele).
- Kada se naidje na ključnu reč *yield*, izlazi se iz *switch*-a.
- *default* grana mora postojati i mora sadržati *yield* sa odgovarajućom vrednošću koja se vraća u slučaju da nijedna druga grana nije uparena sa uslovom *switch*-a.
- Primer:

```
int result = switch(val) {
    case 0:  yield 1;
    case 1:  yield 0;
    default: yield 5;
};
```

$Expr = ["-"] Term \{ Addop Term \}$
 / "switch" "(" Expr ")" "{" {"case" numConst ": " {Statement} } "}"

Statement = DesignatorStatement ";"
 | "if" "(" Condition ")" Statement ["else" Statement]
 | "do" Statement "while" "(" Condition ")" ";"
 / "yield" Expr ";"
 | "break" ";"
 | "continue" ";"
 | "return" [Expr] ";"
 | "read" "(" Designator ")" ";"
 | "print" "(" Expr [", " numConst] ")" ";"
 | "{" {Statement} "}"

III Nivo C

Iste izmene i dopune kao za nivo B (switch koji vraća rezultat).

NAPOMENA: Zahteve navedene u prvom poglavlju ovog dokumenta **obavezno** svi rade. Za svaki nivo se rade **isključivo i obavezno** oni dodatni zahtevi iz drugog poglavlja ovog dokumenta koji su predviđeni za taj nivo.
