
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Programski prevodioci 1
Nastavnik: dr Dragan Bojić, red. prof.
Asistenti: mast. inž. Maja Vukasović, mast. inž. Kristijan Žiža
Školska: 2020/2021.
Ispitni rok: Avgustovsko-septembarski ispitni rok
Datum: 01.08.2021.

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. Nabranjanje (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

Specificirane izmene i dopune projekta se odnose na postavku projekta iz januarskog roka.

- Ukloniti ternarni operator:

$\text{Expr} = \text{Condition} \text{ "?" Expr ":" Expr} \mid \text{CondFact} \text{ "?" Expr ":" Expr}.$

I Nivo A

- Dopuniti podršku za if-then-else kontrolnu strukturu:

$\text{"if" "(" Condition ")" Statement ["else" Statement]}.$

Potrebno je implementirati optimalno izračunavanje složenih kondicionala. Ukoliko je na osnovu parcijalnog dela složenog kondicionala poznato da će ceo uslov biti prihvaćen ili odbijen, ne izračunava se ostatak kondicionala.

II Nivo B

- Proširiti programski jezik Microjava dodavanjem operacija za rad sa podacima jednodimenzionalnog vektorskog tipa ($\text{int}[]$). Operacije ne menjaju svoje operande, nego stvaraju nove. Ukoliko vektori nisu iste dužine, za operacije koje to zahtevaju, potrebno je generisati izuzetak u vreme izvršavanja. Potrebno je dodati sledeće operacije (notaciono se pišu unutar para duplih navodnika).
 - Operacija množenja dva vektora $"**"$ proizvodi skalarni rezultat. Vektori moraju imati isti tip elementata i istu dužinu. Za vektore $A = [a_1, a_2, \dots, a_n]$ i $B = [b_1, b_2, \dots, b_n]$ rezultat operacije $A "**" B$ je skalarna veličina koja se formira prema jednakosti $A "**" B = a_1 *$

$b_1 + a_2 * b_2 + \dots + a_n * b_n$, gde operacija $*$ na desnoj strani jednakosti predstavlja standardno množenje celih brojeva.

- Operacija množenja vektora izrazom skalarnog tipa $"*"$ množi svaki element vektora vrednošću datog izraza. Tip skalara mora odgovarati tipu elemenata vektora. Za vektor $A = [a_1, a_2, \dots, a_n]$ i izraz s skalarnog tipa, operacija $s "*" A$ daje novi vektor koji se formira prema jednakosti $s "*" A = [s * a_1, s * a_2, \dots, s * a_n]$.

III Nivo C

- Potrebno je implementirati podršku za pristup statičkim poljima klase.

ClassVarDecl = [static] Type ident ["[" "]"] {", " ident ["[" "]"]} ";

Statičkim poljima se može pristupiti iz metoda. Ukoliko im se pristupa unutar metoda klase u okviru kojih su definisani, može im se pristupiti ili navođenjem imena ili u obliku Klasa.polje. Ukoliko se statičkom polju pristupa van klase u kojoj je definisano, pristupa se isključivo u obliku Klasa.polje.

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.