

Projekat: Razvoj interpretera

Pravila

1. **Zadatak timskog rada:** Projekat se mora raditi isključivo u paru. Nijedan student ne sme raditi individualno na projektnom zadatku. Studenti koji nisu uspeali da nađu partnera moći će putem napravljene tabele da pronađu slobodne studente sa kojima mogu raditi projekat.
2. **Implementacioni detalji:** Implementacija projektnog zadatka može biti realizovana u bilo kojem programskom jeziku po izboru studenata, u granicama razuma. Za izradu je dozvoljeno koristiti bilo koji generator parsera i leksera, ili ručno napisan rekurzivni silaz.
3. **Dizajn jezika:** Studenti su u obavezi da dizajniraju sopstveni programski jezik. Dizajn jezika mora biti funkcionalan i razumljiv, s tim da je zabranjeno dizajniranje ezoteričnih jezika (jezika koji su namerno teški za upotrebu ili razumevanje).
4. **Razumevanje projekta:** Svaki student koji radi na projektu u paru mora pojedinačno razumeti svaki deo projekta, uključujući funkcionalnosti i svrhu implementiranih delova. Studenti moraju biti u stanju da samostalno pokažu razumevanje gradiva.
5. **Dodatna pitanja i izmene na odbrani:** Na odbrani projekta, studentima će biti postavljena dodatna pitanja u vezi sa projektom, a od njih se može zahtevati i da izmene određene delove implementacije kako bi dokazali svoje razumevanje i prilagodljivost rešenja.
6. **Neakademsko ponašanje:** Svaka forma neakadenskog ponašanja (prepisivanje, plagijat, korišćenje "generativnih" veštačkih inteligencija poput ChatGPT-a) je striktno zabranjena i može dovesti do disciplinskog postupka.
7. **Predaja rada:** Od studenata se očekuje da prilikom izrade projekta koriste platformu GitHub Classroom za definisanje timova i postavljanje radova. Ime tima treba da bude u formatu `tim_iinterpretatorić1222_pprevodilić4423`. Prijave se vrše na sledećem assignment-u: <https://classroom.github.com/a/t1i21Sr2>

Ako niste sigurni u neko od pravila, ili želite dalju elaboraciju, kontaktirajte sa asistentima.

Specifikacija

Projekat se sastoji iz više etapa, koje su predstavljene u sledećoj tabeli:

NAZIV ETAPE	BROJ POENA
Leksička analiza	5
Sintaksna analiza	10
Semantička analiza	10
Generisanje međukoda	15
Interpretiranje međukoda	10

Da bi studenti mogli da izrađuju navedene etape, potrebno je da definišu programski jezik za koji će pisati projekat. U nastavku su date osnovne strukture i ograničenja programskog jezika koji studenti treba da smisle radi izrade projekta:

1. **Tipovi podataka:** Programski jezik treba da podrži celobrojni i logički tip podatka, ali i nizove, s tim što istinitosne vrednosti kod logičkih tipova podataka mogu biti numerički tipovi. Takođe je potrebno da tip podatka promenljive bude poznat programu već pri prevođenju (statička tipizacija).
2. **Ulaz i izlaz iz programa:** Interpreter (i programski jezik) treba da podržavaju dobijanje argumenata sa komandne linije. Jezik treba da podržava ispis na standardni izlaz.
3. **Operatori:** Programski jezik treba da poznaje sledeće operatore: operator dodele, bar jedan aritmetički operator, relacione i logičke operatore. Logički operatori treba da vrše *short-circuiting* (t.j. $A \ || \ B$ ne treba da izračunava podizraz B kada je A tačno, i $A \ \&\& \ B$ ne treba da izračunava podizraz B kada je A netačno).
4. **Naredbe grananja:** Programski jezik treba da poznaje naredbu grananja (u većini viših programskih jezika nazvanoj *if*), kao i naredbu alternative (slično naredbi *else*).
5. **Petlje:** Programski jezik treba da poznaje bar dve vrste petlje.
6. **Funkcije:** Programski jezik treba da dozvoljava implementaciju funkcija. Funkcija treba da sadrži ulaz (parametre), izlaz (povratni tip) i telo (listu naredbi koje treba da izvrši).

Rokovi i predaja projekta

U dogovoru sa asistentom, možete braniti gradivo pre roka, tj. pre kolokvijumske nedelje, ali isključivo za gradivo koje je određeno za datu kontrolnu tačku.

U nastavku je dat spisak kontrolnih tačaka:

1. **Prva kontrolna tačka:** Podrazumeva izradu prve dve etape. Brani se najkasnije u prvoj kolokvijumskoj nedelji.
2. **Druga kontrolna tačka:** Podrazumeva izradu etapa koje nisu pređene u prvoj kontrolnoj tački. Brani se najkasnije u drugoj kolokvijumskoj nedelji.