

Измене и допуне пројектног задатка из Програмских преводилаца за јулски рок 2017/2018.

Рок за слање решења 24.6.2018.

Ниво А

Проширити имплементацију преводиоца за програмских језик Микројава додавањем подршке за дефинисање метода са формалним аргументима које могу да враћају вредности произвољног уграђеног типа. Имплементирати семантичку анализу позива функција са аргументима (и типом резултата) провером стварних аргумената (израза) по броју и типу са формалним. Преводилац треба да генерише код за прослеђивање стварних аргумената, позив функције и адекватну обраду повратне вредности.

Ниво Б

Проширити програмски језик Микројава додавањем операција за рад са подацима једнодимензионалног векторског типа (нпр. `int[]`). Операције не мењају своје операнде, неко стварају нове. Уколико вектори нису исте дужине, за операције које то захтевају, потребно је генерисати изузетак у време извршавања. Потребно је додати следеће операције (нотационо се пишу унутар пара дуплих наводника):

- Операција „+“ сабира два вектора који морају имати исти тип елемената и исту дужину. За векторе $A = [a_1, a_2, \dots, a_n]$ и $B = [b_1, b_2, \dots, b_n]$, резултата операције $A + B$ је нови вектор који се формира према једнакости $A + B = [a_1 + b_1, a_2 + b_2, \dots, a_n + b_n]$.
- Операција множења два вектора „*“ производи скаларни резултат. Вектори морају имати исти тип елемената и исту дужину. За векторе $A = [a_1, a_2, \dots, a_n]$ и $B = [b_1, b_2, \dots, b_n]$ резултат операције $A * B$ скаларна величина које се формира према једнакости $A * B = a_1 * b_1 + a_2 * b_2 + \dots + a_n * b_n$, где је операција „*“ на десној страни једнакости дефинисана за тип података елемената вектора.
- Операција множења вектора изразом скаларног типа „*“ множи сваки елемената вектора вредношћу датог израза. Тип скалара мора бити једнак типу елемената вектора. За вектор $A = [a_1, a_2, \dots, a_n]$ и израз s скаларног типа, операција $s * A$ даје нови вектор који се формира према једнакости $s * A = [s * a_1, s * a_2, \dots, s * a_n]$.

Ниво Ц

Имплементирати оптимизацију полиморфног позива виртуелних метода у специјалном случају чији опис следи. Посматрају се класе B, D_1, D_2, \dots, D_n и бинарна релација наслеђивања $>$. Уколико класа D_1 наслеђује класу B то се представља нотацијом $D_1 > B$.

Посматра се хијерархија наслеђивања $H \equiv D_n > D_{n-1} > \dots > D_2 > D_1 > B$. Нека класа B има виртуелну методу f која је редефинисана у последњој класи D_n хијерархије H . Нека је у програму написаном на језику Микројава декларисана референца $T obj$, и иницијализована адресом објекта неког од типова у H , где $T \in \{B, D_1, D_2, \dots, D_{n-1}\}$. Тада се позив операције $obj.f(\dots)$ преводи (оптимизује) на следећи начин: уколико се провером у време извршавања добије да је D_n динамички тип референце obj , тада се врши статички позив методе f из класе D_n (уместо инструкције `invokevirtual` користи се `call`). У супротном врши се динамичко везивање инструкцијом `invokevirtual`.

Напомена: за сваки ниво се раде искључиво они додатни захтеви који су предвиђени за тај ниво. Додатни захтеви нису условног карактера (осим за ниво А) и носе 4 поена за сваки ниво. Нпр. уколико се не уради додатни захтев за ниво Ц, може се добити максимално 36 поена.