

Повратне вредности

Најједноставнији начин да се размене подаци са функцијом јесте коришћење повратних вредности.

Функције које имају повратне вредности су процењене (evaluate) на ту вредност као што су промењиве процењене на вредност на коју указују када се користе у изразима.

Ако постоји функција UzmiString() чија је повратна вредност стринг, може се писати код:

```
string mojString;
mojString = UzmiString();
```

У другом случају, може постојати функција UzmiPromenjivu() која враћа вредност типа double:

```
double mojaPromenjiva;
double multiplikator = 5.3;
mojaPromenjiva = UzmiPromenjivu() * multiplikator;
```

Када функција враћа вредност, мора се модификовати функција на два начина:

- Специфицирати тип враћене вредности у декларацији функције уместо коришћења void службене речи
- Користити службену реч return на крају функције чиме се враћена вредност враћа у позивајући под

Ограничења повратних вредности функција

У смислу кодова, ово изгледа као следећи пример:

```
static <povratniTip> <ImeFunkcije>()
{
    ...
    return <povratnaVrednost>
}
```

Једино ограничење је то што <povratnaVrednost> мора бити која је или типа <povratnaVrednost> или се може имплицитно конвертовати у тај тип.

Ипак, <povratnaVrednost> може бити било којег типа, укључујући још компликованије типове података.

Може бити на пример једноставно као:

```
static double UzmiPromenjivu()
{return 3.2;}
```

Ипак, повратне вредности су обично резултат математичког рачунања унутар функције, док би се приказан пример најлакше извео увођењем константе.

Функције са више return исказа

Када се дође до return исказа, извршавање програма се пребацује на позивајући код.

Ниједна линија у телу функције после тог исказа се неће извршити, иако ово не значи да return исказ мора бити смештен у последњу линију кода функције.

Исказ return се може користити унутар тела функције, најчешће после извођења структуре гранања.

Смештање return унутар for петље, унутар if блока или било које друге структуре изазваће да се та структура моментално заврши а функција прекине са радом:

```
static double UzmiPromenjivu()
{
    double proveripromenjivu;
    //proveripromenjivu dobija vrednost preko koda
    if (proveripromenjivu < 5)
        return 4.7;
    return 3.2;
}
```

У примеру се враћа једна од две могуће вредности у зависности од вредности у proveripromenjivu.

Једино ограничење у овом случају је то што return исказ мора бити обрађен пре него што се дође до затварајуће витичасте заграде.

Следећи код није добар:

```
static double UzmiPromenjivu()
{
    double proveripromenjivu;
    //proveripromenjivu dobija vrednost preko koda
    if (proveripromenjivu < 5)
        return 4.7;
}
```

У примеру, ако је услов proveripromenjivu < 5 нетачан, онда се једини return исказ неће реализовати, што није допуштено.

Сви делови обраде унутар функције морају достићи некакав return исказ.

У највећем броју случајева, компјлер то детектује и даје грешку “not all code paths return a value.”

Методе са телом у исказу

Функције које извршавају једну линију кода могу користити могућност звану методе са телом у исказу (expression-bodied methods).

Следећа шема у функцији користи => (ламбда стрелу) да би се имплементирала ова метода:

```
static <povratniTip> <ImeFunkcije>() => <mojaPromenjiva> * <mojaPromenjiva>;
```

Функција Mnozenje() се може написати у оба формата:

```
static double Mnozenje(double mojaProm1, double mojaProm2)
{
    return mojaProm1 * mojaProm2;
}
```

Или преко =>:

```
static double Mnozenje(double mP1, double mP2) => mP1 * mP2;
```