Funkcije

•

Kada imamo više ponavljanja istih blokova naredbi (operacija) u različitim delovima programa

Opis argumenata funkcije je lista argumenata funkcije koja sadrži tipove argumenata i identifikatore argumenata. Npr. funkcija Izracunaj(int a, char ch, float

x) u listi su upisana tri argumenta i to: a, ch i x čiji su tipovi podataka navedeni ispred.

ili prikazano na sledećem dijagramu:

(time skraćujemo tekst programa)

•

|  |  |
| --- | --- |
| Upotrebom funkcija, program se deli na odvojene blokove. Svaki blok radi određeni posao. |  |
| Ovako olakšavamo čitljivost/razumevanje programa. Osmišljavanje, razumevanje i održavanje |
| manjih blokova koda je lakše. |  |

•

Kada se deo programa moze izvršiti sa različitim parametrima. Ovi parametri se prosleđuju

funkciji kao argumenti funkcije.

**Definisanje funkcije**

Funkcija se definše na sledeći način:

|  |  |
| --- | --- |
| tip\_rezultata ime\_funkcije(opis argumenata funkcije) |  |
|  |
| { |  |
|  |
| /\*telo funkcije\*/ |  |
|  |
| } |  |

|  |
| --- |
| tip\_rezultata je neki od već poznatih tipova |
| podataka i on označava kakav je rezultat |  |
| koji funkcija vraća. |  |

|  |
| --- |
| Često se kaže i "tip funkcije", a tada se misli |
| upravo na tip njenog rezultata. |  |
| ime\_funkcije je identifikator funkcije |  |

Telo funkcije je niz naredbi koje počinju sa deklaracijama promenljivih kao u programu tj. funkciji main().

U telu funkcije naredba za vraćanje rezultata koji funkcija daje je: return rezultat;

|  |
| --- |
| Svaka funkcija se sastoji od skupa naredbi koje određuju šta i kako funkcija radi. Taj skup naredbi čini |
| telo funkcije. Telo funkcije započinje nakon otvorene vitičaste zagrade ({), a završava zatvorenom |  |
| vitičastom zagradom (}). |  |

|  |  |
| --- | --- |
| **Prilikom definicije funkcije navodi se:** |  |
| **– tip povratne vrednosti funkcije (ako se ne navede podrazumeva se int)** |
| **– ime funkcije** |  |
| **– lista argumenata** |  |
| **– telo funkcije.** |  |

**(Obratiti pažnju gde treba koristiti deklaraciju a gde definiciju funkcije)**

**U deklaraciji (prototipu) funkcije je sve isto kao i u definiciji, osim što izostaje telo funkcije.**

**Primer definicije funkcije:**

**tip\_rezultata ime\_funkcije (tip param1, tip param2, ..., tip paramN)**

**{**

**/\*telo funkcije\*/**

**}**

**Primer deklaracije funkcije:**

**tip\_rezultata ime\_funkcije (tip param1, tip param2, ..., tip paramN);**

**Poziv funkcije**

|  |
| --- |
| Funkcija se izvršava tako što se „poziva“ u (glavnom) programu ili drugoj funkciji. Funkcija se poziva |
| navođenjem njenog naziva i argumenata zapisanih u redosledu koji je zadan deklaracijom, kao u |  |
| primeru StaImasIz("kung-fu-a") gde učenika pitamo koju ocenu ima iz Kung-Fu-a. |  |

**Povratna vrednost funkcije**

|  |
| --- |
| U programskom jeziku C++ očekujemo da funkcija vrati neku vrednost. Ova povratna vrednost ima svoj |
| tip kao i ostale vrednosti u C++, pa može biti tipa integer, float, char ili bilo šta drugo. Vrednost funkcije |
| vraćamo sa komandom return. |  |

{

}

return rezultat;

rezultat = x \* x;

float rezultat;

float Kvadriraj(float x)

Funkcija iz gornjeg primera vraća vrednost promenljive “rezultat” kao povratnu vrednost funkcije.

|  |  |
| --- | --- |
| Takođe, možemo koristiti i izraz u return komandi. Na primer možemo zameniti poslednje dve linije |  |
| funkcije sa ‘return (x \* x);’ Ako zaboravite povratnu vrednost u nekoj funkciji dobićete poruku |  |
| upozorenja (eng. warning) od C++ kompajlera da funkcija mora da vrati vrednost. Upozorenja |  |
| kompajlera ne zaustavljaju izvršavanje programa ali greške zaustavljaju. Ipak, dobra je praksa obraćati |
| pažnju i na ova upozorenja i popraviti kod tako da se prevodi bez upozorenja. |  |

|  |
| --- |
| Poziv funkcije u programu se ostvaruje navođenjem: |
| – imena funkcije |  |
| – liste stvarnih argumenata funkcije |  |
| npr. faktorijel(5); |  |

|  |
| --- |
| Mehanizam za vraćanje vrednosti iz funkcije predstavlja naredba return |
| return izraz; |  |
| Tip izraza se evenutualno konvertuje u tip rezultata. |  |

**Funkcije koje ne vraćaju vrednost**

|  |
| --- |
| Prema pravilima pisanja programa na jezicima C i C++, svaka funkcija ima tip. Ako nam nije potrebno |
| da funkcija na mesto poziva vrati vrednost, deklarišemo (objavljujemo) da je tip funkcije, to jest njene |
| povratne vrednosti, tip void (srpski: nevažeći, prazan): |  |

|  |  |
| --- | --- |
| void test () |  |
|  |
| { |  |
|  |
| /\* kod funkcije ali bez povratne vrednosti \*/ |  |
|  |
| } |  |

nećemo koristiti. Zato se pozivi funkcija tipa void pišu u programu kao naredbe, a ne kao izrazi.

Tip void se ne može koristiti nizašta, njegovom upotrebom u deklaraciji samo saopštavamo da vrednost

Funkcija tipa void može (i ne mora) da sadrži naredbu return, a ako je sadrži, onda bez ikakvog izraza.

Evo kako to izgleda na primeru:

|  |  |
| --- | --- |
| void ispisUpozorenja() |  |
|  |
| { |  |
|  |
| cout << "Upozorenje!"; |  |
|  |
| } |  |

Poziv ove funkcije sa nekog drugog mesta u programu bi izgledao ovako:

|  |  |
| --- | --- |
| … |  |
|  |
| ispisUpozorenja(); |  |
|  |
| … |  |
|  |
| Iza return može i da se ne stavi ništa, ali u tom slučaju se ni jedna vrednost |  |

|  |  |
| --- | --- |
| ne vraća pozivaocu. |  |
|  |
| * Funkcija koja nema povratnu vrednost deklariše se da ima povratni tip void.
 |  |
|  |
| * Slično, ako funkcija nema argumente, u deklaraciji se umesto liste argumenata
 |  |
|  |
| navodi void. |  |

**Deklaracija funkcije**

**Argumenti funkcije**

mogu koristiti u telu funkcije.

Funkcije mogu da prihvataju ulazne parametre u formi promenljivih i izraza i ove ulazne promenljive se

moguće je preneti i vrednost iz funkcije (tako da parametri budu izlazni), ali o tome kasnije.

Do sad su parametri služili da prenesu vrednost sa mesta poziva u funkciju i to su ulazni parametri, a