

Osnovi računarskih sistema

Milena Vujošević–Janičić

Čas 1

Vežbe 22 03 2005

1.1 Prostor imena

Svaki objekat, funkcija, tip ili šablon koji su deklarirani u globalnoj oblasti važenja uvode neki globalni entitet. Svaki globalni entitet koji je uveden u globalnoj oblasti važenja prostora imena **mora imati jedinsten naziv**.

To znači da ukoliko želimo da koristimo neku biblioteku u svom programu moramo voditi računa da nazivi globalnih entiteta u našem programu ne dođu u koliziju sa nazivima globalnih entiteta iz biblioteke.

Kako možemo da budemo sigurni da nazivi globalnih entiteta u našem programu neće doći u sukob sa već postojećim nazivima koji su deklarirani u bibliotekama koje koristi naš program?

Ovaj problem sukobljavanja imena rešava se uz pomoć prostora imena. Autor biblioteke može da definiše prostor imena kako bi sklonio nazive u biblioteci iz globalnog prostora imena.

Primer 1

```
#include <iostream>

//using namespace std;

// Posto nismo ukljucili prostor imena std
// mozemo da deklariseemo globalnu promenljivu
// cije je ime cout
double cout = 3.14;

// Pisemo novi prostor imena proba1
// i u okviru njega mozemo sada da definisemo
// novu promenljivu sa istim imenom
namespace proba1 {
    int cout = 5;
    float cin = 4;
    //...
```

```
};

// ovo je nastavak prostora imena
namespace proba1 {
    // Ovde mozemo da dodamo neke nove
    // definicije. Prostor imena moze
    // da sadrzi i druge ugnezdene definicije
    // prostora imena kao i deklaracije ili
    // definicije funkcija, objekata, klasa,
    // sablona i tipova.
};

main()
{
    // Ovo cout je loklana promenljiva
    char* cout = "cout";

    // Referisemo se na cout iz standardnog
    // prostora imena sa std::cout
    // Operator :: naziva se operator
    // oblasti vazenja
    // Ovde se vrsi stampanje lokalne
    // promenljive cout
    std::cout << cout << std::endl;
    // cout

    // Sa proba1:cout i proba1::cin koristimo cout
    // i cin iz prostora imena proba1
    std::cout << proba1::cout << std::endl;
    std::cout << proba1::cin << std::endl;
    // 5
    // 4

    // Sa ::cout koristimo globalnu promenljivu
    // cout (definisanu van prostora imena)
    std::cout << ::cout << std::endl;
    // 3.14

    return 0;
}
```

Ukoliko ne želimo da uključimo ceo prostor imena ali želimo da često koristimo neko posebno ime iz tog prostora tada je moguće uključiti samo dato ime iz prostora imena čime se eliminiše potreba za stalnim korišćenjem operatora oblasti važenja.

Primer 2

```
#include <iostream>

//using namespace std;

// Koristicemo samo cout i endl iz standardne
// biblioteke
using std::cout;
using std::endl;

main()
{
    cout << "... " << endl;

    return 0;
}
```


Sadržaj

1	Vežbe 22 03 2005	3
1.1	Prostor imena	3