

## ФУНКЦИИ – ЧЕНКИ НА СТРУКТУРАТА

Покрај податоци, членови на структурата може да бидат и функции.

Пр. Да се напише програма во која со функции се внесуваат и печатат податоци за студент при што се креира класа student со податоци: index – int, ime – string, prezime – string, godnastudii – int.

```
#include <iostream>
using namespace std;
typedef struct student
{
    int index;
    string ime;
    string prezime;
    int godnastudii;

void vnesi()
{
    cout<<"vnesi broj na indeks"<<endl;
    cin>>index;
    cout<<"vnesi ime"<<endl;
    cin>>ime;
    cout<<"vnesi prezime"<<endl;
    cin>>prezime;
    cout<<"vnesi godina na studii"<<endl;
    cin>>godnastudii;
}

void pecati()
{
    cout<<"vneseno e"<<endl;
    cout<<index<<" "<<ime<<" "<<prezime<<" "<<godnastudii;
}
};

int main()
{
    student k;
    cout << "vnesi podatoci za student" << endl;
    k.vnesi();
    k.pecati();
    return 0;
}
```

Пр. да се напише програма со која се составува список од n студенти. Студентите да се внесат како структура составена од име, презиме и број на бодови. Да се подреди списокот според бројот на бодови во опаѓачки редослед.

```

#include <iostream>
#include <string>
using namespace std;
typedef struct student
{
    int index;
    char ime[50];
    char prezime[50];
    int bodovi;

    void vnesi()
    {
        cout<<"vnesi broj na indeks"<<endl;
        cin>>index;
        cout<<"vnesi ime"<<endl;
        cin>>ime;
        cout<<"vnesi prezime"<<endl;
        cin>>prezime;
        cout<<"vnesi bodovi"<<endl;
        cin>>bodovi;
    }

    void pecati()
    {
        cout<<index<<" "<<ime<<" "<<prezime<<" "<<bodovi<<endl;
    }
};

void podredi(student spisok[1000], int n)
{
    student p;
    int i,j;
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(spisok[i].bodovi<spisok[j].bodovi)
            {
                p=spisok[i];
                spisok[i]=spisok[j];
                spisok[j]=p;
            }
        }
    }
}

int main()

```

```

{
    int i,n;
    struct student spisok[1000];
    cout << "vnesi broj na studenti" << endl;
    cin>>n;
    for(i=0;i<n;i++)
    {
        spisok[i].vnesi();
    }
    podredi(spisok, n);
    cout<<"podreden spisok "<<endl;
    for(i=0;i<n;i++)
    {
        spisok[i].pecati();
    }
    return 0;
}

```

Втор начин за истата задача:

```

#include <iostream>
#include <string>
using namespace std;
typedef struct student
{
    int index;
    char ime[50];
    char prezime[50];
    int bodovi;
};
void vnesi(student spisok[1000], int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        cout<<"vnesi broj na indeks"<<endl;
        cin>>spisok[i].index;
        cout<<"vnesi ime"<<endl;
        cin>>spisok[i].ime;
        cout<<"vnesi prezime"<<endl;
        cin>>spisok[i].prezime;
        cout<<"vnesi bodovi"<<endl;
        cin>>spisok[i].bodovi;
    }
}

void pecati(student spisok[1000], int n)
{

```

```

int i;
for(i=0;i<n;i++)
{
    cout<<spisok[i].index<<" "<<spisok[i].ime<<" "<<spisok[i].prezime<<" "<<spisok[i].bodovi<<endl;
}
}

void podredi(student spisok[1000], int n)
{
    student p;
    int i,j;
for(i=0;i<n-1;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(spisok[i].bodovi<spisok[j].bodovi)
        {
            p=spisok[i];
            spisok[i]=spisok[j];
            spisok[j]=p;
        }
    }
}
}

int main()
{
    int i,n;
    student spisok[1000];
    cout << "vnesi broj na studenti" << endl;
    cin>>n;
    vnesi(spisok, n);
    podredi(spisok, n);
    cout<<"podreden spisok "<<endl;
    pecati(spisok, n);
    return 0;
}

```

Пр. Да се напише програма која ќе го пресметува векторски производ на два вектори. Векторите се претставени со координати во тродимензионален координатен систем. Скаларниот производ да се пресмета со посебна функција. За вектор да се дефинира посебна структура `vektor`.

```

#include <iostream>
using namespace std;
struct vektor
{
    float x;
    float y;

```

```

float z;
};

float scalaren(vektor v1, vektor v2)
{
    return v1.x*v2.x+v1.y*v2.y+v1.z*v2.z;
}

int main()
{
    vektor p1,p2,p3;
    cout<<"vnesi koordinati na prviot vektor" << endl;
    cin>>p1.x>>p1.y>>p1.z;
    cout << "vnesi koordinati na vtoriot vektor" << endl;
    cin>>p2.x>>p2.y>>p2.z;
    cout<<"skalarniot proizvod e "<<scalaren(p1,p2)<<endl;
    return 0;
}

```

Пр. претходната задача да се прошири со пресметување на векторски производ на два вектори.

*Прашања поврзани со наставните единици може да се испраќаат на email:  
 anetastojceska@gmail.com*