Конструктори и деструктори

1. Конструктори и деструктори

При креирање на објект во некоја класа, за негово користење потребно е да се предвиди (дефинира) т.н. конструктор, кој е всушност функција – членка во класата и треба да го има истото име како што е името на класата. Основната разлика меѓу конструкторите и другите функции во класата е тоа што конструкторот не враќа никаква вредност и секогаш до него има пристап од типот на public.
Ако во класата не е дефиниран конструктор од самиот програмер, тогаш компајлерот во C++ набавува default constructor – без параметри, кој се повикува при самото дефинирањето на објект во некоја класа. (пр. ucenik objucenik;)

Бидејќи со самото дефинирање на класата-објектот се зафаќа меморија и доколку не ни треба класата –објектот, тој треба да се избриши, а тоа се прави со функција – **деструктор,** а тоа се прави со следната команда: **име на класата::~име на класата().** Пример ако сакаме да ја избришиме класата ucenik тогаш деструкторот ќе биде: **ucenik::~ucenik()** како наредба/команда во главниот дел од програмата.

Пр.1. #include <iostream>
using namespace std;
class Proizvod
{ public:
Proizvod(float x , float y) - **конструктор**
{ cout<<x*y;
}
};
int main ()
{
float br1,br2;
cout<<"Vnesi dva broja"<<endl;
cin>>br1>>br2;
Proizvod mojProizvod(br1 , br2);
return 0;
}

Пр. 2. Програма која користи класа артикл со податоци ime(string), kolicina, cena(float) функции set и get и со ф-ја iznos
#include <iostream>

using namespace std;
class artikal
{
public:
string ime;
float kol, cena;
void set(string i, float k, float c)
{
ime=i;
kol=k;
cena=c;

```
    }
    void get()
    {
       cout<<ime;
    }
    float iznos(float kol, float cena)
    {
       return cena*kol;
    }
};
int main()
{
  artikal obj;
  string i;
  float k,c;
  cout<<"vnesi ime cena i kolicina"<<endl;
  cin>>i>>k>>c;
  obj.set(i,k,c);
  obj.get();
  cout<<obj.iznos(k,c);
   return 0;
}
```

Пр.3. Класа со податочни членки- (тоа се податоци – атрибути за сите објекти што припаѓа на класата, кои се дефинираат внатре во класата) и функциите-членки во класата set и get
Пример:

```
#include <iostream>
#include <string>
using namespace std;
class  ucenik
{
public:
string Prezime,Ime;
float mat,mak,ang;
void  setPodatoci(string P,string I,float o1,float o2, float o3)
{
  Prezime=P;
  Ime=I;
  mat=o1;
  mak=o2;
  ang=o3;
}
string getPrezimeIme()
{
  return Prezime+"  "+Ime;
}
 float  Prosek(float  mat,float mak,float ang)
{
 float p;
 p=((mat+mak+ang)/3);
 return p;
}
};
```

```
int  main()
 {
  string P,I;
  float o1,o2,o3;
  ucenik  objucenik;
  cout<<"Vnesi Prezime i ime na ucenikot"<<endl;
  cin>>P>>I;
  cout<<"Vnesi gi ocenkite po matematika, makedoski i angliski jazik"<<endl;
  cin>>o1>>o2>>o3;
  objucenik.setPodatoci(P,I,o1,o2,o3);
  cout<<"Ucenikot  "<<objucenik.getPrezimeIme()<<"  ima  "<<objucenik.Prosek(o1,o2,o3)<< "
prosek";
  return 0;
  }
```

Пр.4. Програма со која ќе се дефинира класа со име vraboten што содржи податочна членка (set и get функции) со следните податоци: Prezime (string), Ime(string), Godvrab(int), и функција членка што ќе пресметува работен стаж на вработениот во години.

```
#include <iostream>
#include <string>
using namespace std;
class  vraboten
{
public:
string Prezime,Ime;
float godvrab;
void  setPodatoci(string P,string I,int g)
{
   Prezime=P;
   Ime=I;
   godvrab=g;

}
void getPodatoci()
{
   cout<<Prezime<<"  "<<Ime;
}
  int  Staz(int godvrab)
{
  int p;
  p=2013-godvrab;
  cout<<p;
}
};
int  main()
 {
  string P,I;
  int g;
  vraboten  objvraboten;
  cout<<"Vnesi Prezime i ime na vraboteniot"<<endl;
  cin>>P>>I;
  cout<<"Vnesi ja godinata na vrabotuvanje"<<endl;
```

```
cin>>g;
objvraboten.setPodatoci(P,I,g);
objvraboten.getPodatoci();
cout<<" ima staz ";
objvraboten.Staz(g);
return 0;
}
```