

Vježbe 4.
// reference i copy konstruktor

Pr.

```
#include <iostream.h>
```

```
class KompBr {
    private:
        int a,b;
    public:
        int Re(){return a;}
        int Im(){return b;}
        void UpisiRe(int x){a = x;}
        void UpisiIm(int x){b = x;}

    /*    KompBr(){
            a = 2;
            b = 3;
        }*/
        KompBr(): a(2), b(3) {} // drugaciji način definiranja - inicijalizacija parametrara

    /*    KompBr(int aa, int bb){
            a = aa;
            b = bb;
        }*/
        KompBr(int aa, int bb):a(aa), b(bb)
        {cout<<"\nKonstruktor sa paramet. inicijalizacijom priv. clanova";}

        KompBr(const KompBr &refKomp) { //copy konstruktor
            a = refKomp.a;
            b = refKomp.b;
        }
        ~KompBr(){
            cout<<"\n Pozvan je destruktorktor! ";
        }
};

void main() {

    KompBr komp;
    cout<<"\n Kompleksni broj je: " << komp.Re() <<" + " <<komp.Im() <<"*i ";

    int re, im;
    cout<<"\n Upisi re dio: ";
    cin>> re;
    cout<<"\n Upisi im dio: " ;
    cin >> im;
    komp.UpisiRe(re);
    komp.UpisiIm(im);
    cout<<"\n Kompleksni broj je: " << komp.Re() <<" + " <<komp.Im() <<"*i ";

    KompBr komp2(re, im);
    cout<<"\n Kompleksni broj je: " << komp2.Re() <<" + " <<komp2.Im() <<"*i ";

    cout<<"\n Poziv copy konstruktora - kopiraju se podaci iz objekta komp!";
```

```

KompBr komp3(komp); // poziv copy konstruktora
cout<< "\n Kompleksni broj je: " << komp3.Re() << " + " << komp3.Im() << "*i ";

cout<< "\nDefiniranje reference na objekt"<<endl;
KompBr &reKomp = komp; // definirana ref. na objekt
reKomp.UpisiRe(re+5);
cout<< "\n Referenca na kompleksni broj je: " << reKomp.Re() << " + " << reKomp.Im() << "*i ";
cout<< "\n Kompleksni broj je: " << komp.Re() << " + " << komp.Im() << "*i ";
}

```

Pr. Var su pokazivaci

```

#include <iostream.h>

class KompBr {
private:
    int * a;
    int * b;
public:
    KompBr(){
        a = new int(10);
        b = new int (5);
    }
    KompBr(const KompBr &refKom){ // copy konstruktor
        a = new int;
        b = new int;
        *a = refKom.Re();
        *b= refKom.Im();
    }

    ~KompBr() {
        delete a;
        a = 0;
        delete b;
        b=0;
    }
    int Re()const {return *a;}
    int Im()const {return *b;}
    void UpisiRe(int x){*a = x;}
    void UpisiIm(int x){*b = x;}
};

void main() {
    int re, im;

    KompBr Komp;
    cout<< "\n Kompleksni broj bez upisa var je: " << Komp.Re() << " + " << Komp.Im() << "*i ";

    cout<< "\n Upisi re dio: ";
    cin>> re;
    cout<< "\n Upisi im dio: " ;
    cin >> im;
}

```

```
Komp.UpisiRe(re);  
Komp.UpisiIm(im);
```

```
KompBr komp2 (Komp);  
cout<< "\n Kompleksni broj bez upisa var je: " << komp2.Re() <<" + " <<komp2.Im() << "*i ";  
cout<< "\n Upisi re dio: ";  
cin>> re;  
cout<< "\n Upisi im dio: " ;  
cin >> im;  
komp2.UpisiRe(re);  
komp2.UpisiIm(im);  
cout<< "\n Kompleksni broj je: " << komp2.Re() <<" + " <<komp2.Im() << "*i ";
```

```
KompBr &reKomp = Komp;  
reKomp.UpisiRe(re+5);  
cout<< "\n Kompleksni broj je: " << Komp.Re() <<" + " <<Komp.Im() << "*i ";  
cout<< "\n Referenca je: " << reKomp.Re() <<" + " <<reKomp.Im() << "*i ";
```

```
}
```