ohmref.cpp

```
#include <iostream>        // ohmref.cpp
using namespace std;

class ohm {
             double u,i; // [v], [A], private
      public:
             ohm(double pu=0.0, double pi=1.0);

// Kopierkonstruktor, hier nicht explizit notwendig:
  ohm(ohm &);
        //
  ~ohm(){cout<<"Instanz Typ ohm geloescht"<<endl;}
 double get_r(){ return i?u/i:0.0;} //bei i=0.0 unendl.
  double get_u(){ return u;}         //inline
  double get_i(){ return i;}         //inline
  void set_u(double pu=0.0){u=pu;}   //inline
  void set_i(double pi=1.0){i=pi;}   //inline
  ohm &kopieren(ohm &o){ this->u=o.u; this->i=o.i;
                          return *this;}
};

// Explizite Definitionen Konstruktor
ohm::ohm(double pu, double pi):u(pu),i(pi){};

//Kopierkonstruktor:
ohm::ohm(ohm &o):u(o.u),i(o.i){//this->u=u; this->i=i;
                 cout<<"Kopierkonstruktor\n";
}

void main(){
    ohm &o1=*new ohm(220.0,5.0),o2=o1;//Initialis.Ref.
//                           ohm o2(o1);//Aequivalent
 int i(0), j(i), k=j; // 3 moegliche Initialisierungen

 (o1).set_u(240.0);
 o1.set_i(20.0);
 cout<<"o1.R = "<<o1.get_r()<<" o1.U = "<<o1.get_u()
 <<" o1.I = "<<o1.get_i()<<'\n';
     o2.kopieren(o1);
    cout<<"o2.R = "<<o2.get_r()<<" o2.U = "<<o2.get_u()
 <<" o2.I = "<<o2.get_i()<<'\n';
 delete &o1;
 cin.get();
}
```

```
Kopierkonstruktor
o1.R = 12 o1.U = 240 o1.I = 20
o2.R = 12 o2.U = 240 o2.I = 20
Instanz Typ ohm geloescht
Instanz Typ ohm geloescht
```