virt0.cpp

```
#include <iostream>
using namespace std;

class ratio {
    private: long zaehl, nenn;
    public:  ratio(long z=0L, long n=0L):zaehl(z),nenn(n){};
             virtual ~ratio(){ cout<<"Destruktor ratio\n"; }
             friend ostream &operator<<(ostream &, ratio &);
};

class object {
    public: virtual void aus(){};
            virtual ~object(){ cout<<"Destruktor object\n"; }
};

ostream &operator<<(ostream & cout, ratio &z){
            return cout<<z.zaehl<<"  "<<z.nenn;
}

class intObject : public object {
    public: intObject(int x=0):wert(x){};
            int wert;
            void aus(){ cout<<wert<<endl; }
            virtual ~intObject(){ cout<<"Destruktor intObject\n"; }
};

class charptrObject : public object {
    public: charptrObject(char *s=""):
                          adr(s?strcpy(new char[strlen(s)+1], s):0){}
            char *adr;
            void aus(){ cout<<(adr?adr:"0")<<endl;
                        cout<<typeid(*this).name()<<endl;
            }
            virtual ~charptrObject(){ delete [] adr; adr = 0;
                                      cout<<"Destruktor charptrObject\n";
            }
};

class ratioObject : public ratio, public object {
    public: ratioObject(long z=0L, long n=1L):ratio(z,n){};
            void aus(){ cout<<*this<<endl; }
            virtual ~ratioObject(){ cout<<"Destruktor ratioObject\n"; }
};

void main(){
            object *feld[3];

            feld[0]=new intObject(4711);
            feld[1]=new charptrObject("HTW");
            feld[2]=new ratioObject(3,4);

            feld[0]->aus();           // 4711
            feld[1]->aus();           // HTW   class charptrObject
            feld[2]->aus();           // 3  4

            delete feld[0]; feld[0]=0; delete feld[1]; feld[1]=0;
            delete feld[2]; feld[2]=0;
            cin.get();
}
```