

```

#include <string.h>
#include <iostream>
using namespace std;

class Adresa {
private:
    char* str;
    int nr;

public:
    Adresa(){

}

    Adresa(char* c, int n):nr(n){ //se apeleaza pseudoconstructorul pentru nr de tip int;
        //pseudoconstructorii se folosesc pentru initializarea tipurilor de date de baza
        if (c==NULL)
            str=NULL;
        else{
            str=new char[strlen(c)+1];
            strcpy(str,c);
        }

    }

    Adresa(const Adresa&p){
        str=NULL;
        *this=p; //reutilizez codul scris in operator=
    }

    Adresa& operator=(const Adresa &p) {
        if(this!=&p){
            if (str!=NULL) delete [] str;
            str=new char[strlen(p.str)+1];
            strcpy(str, p.str);
            nr=p.nr;
        }
        return *this;
    }

    ~Adresa(){
        if (str!=NULL) delete[]str;
    }

    friend ostream& operator<<(ostream &dev,const Adresa &p){
        // in operatorul de << din
        //Carte se foloseste operatorul de << din Pagina
        if (p.str!=NULL) dev<<"Strada: "<<p.str<<endl;
        dev<<"Nr. : "<<p.nr<<endl<<endl;
        return dev;
    }
};

```

```
class Imobil
{
    Adresa a;
    char* descriere; // casa 150 mp, apartament 3 cam, etc
    int pret;
    bool de_inchiriat; // true - daca da, altfel de vanzare - false
}; // folosit de o firma de tranzactii imobiliare/inchirieri, etc
```

- sa implementeze clasa Adresa folosind codul dat dar impartind-o in .h si .cpp
- sa implementeze (constructor cu param, de copiere, op=, op<<, destr)

1. In main sa creeze obiecte de tip Firma/Imobil. Sa explice ce metode din Adresa se folosesc in clasa agregata si unde.
2. Sa creeze un vector pt evidenta imobilelor respectiv firmelor din Bucuresti. Sa sorteze vectorul dupa pret/ nr angajati.
3. Sa afiseze doar imobilele de inchiriat/ firmele cu capital strain de pe strada Florilor