

Задатак 1.

Написати програм на језику C++ који:

- дефинише структуру типа `mobilni` чији су чланови
`model` (низ карактера)
`kolicina` (цео број)
`cena` (цео број)
- декларише динамички (ев. статички) низ за 10 структура типа `mobilni`, чита са тастатуре и уписује у текст датотеку `katalog.txt` запис о свакој структури чија је `kolicina` различита од 0,
- чита структуре из датотеке `katalog.txt` и приказује на екрану само оне које имају вредност податка `model` тражену са тастатуре, понавља овај поступак за задате највише две међусобно различите вредности податка `model` и након тога се завршава.

Решење 1.

```
#include <iostream>
#include <fstream>
#include <stdlib.h>
#include <string.h>
#define MAX 10
using namespace std;
struct mobilni{ char model[MAX+1]; int kolicina, cena; };
int main()
{
    int i, j, n=0, istih=0, razlicitih=0;
    char model[MAX+1], *modeli[MAX];
    fstream f;
    mobilni *niz;

    niz=new mobilni[MAX]; if(niz==NULL){cout<<"Nije uspela dodela";exit(1);}
    f.open("katalog.txt", ios::out | ios::in | ios::trunc); if(!f){cout<<"Greska datoteke";exit(1);}

    cout<<"\nUneti za svaki mobilni u posebnom redu: model, kolicinu,cenu"<<endl;
    for(i=0; i<MAX; i++)
    {
        cin>>niz[i].model>>niz[i].kolicina>>niz[i].cena;
        if(niz[i].kolicina)
        {
            f<<niz[i].model<<' '<<niz[i].kolicina<<' '<<niz[i].cena<<endl;
            n++;
        }
    } delete niz;

    niz=new mobilni[n]; if(niz==NULL){cout<<"Greska dodele";exit(1);}
    for(i=0;i<n;i++)
    {
        modeli[i]=new char[MAX+1]; if(modeli[i]==NULL){cout<<"Greska dodele";exit(1);}
        f.seekg(0);
        for(i=0; i<n; i++)
        {
            cout<<"\nmodel: ";cin>>model; strcpy(modeli[i],model);

            for(j=0;j<i;j++) if(strcmp(modeli[j],model)==0) {istih=1; continue;}
            if(istih) {istih=0; continue;}

            f>>niz[i].model>>niz[i].kolicina>>niz[i].cena;
            if(strcmp(niz[i].model,model)==0)
                cout<<niz[i].model<<' '<<niz[i].kolicina<<' '<<niz[i].cena<<endl;
            if(++razlicitih==2) break;
        }
    }
    f.close(); delete niz; for(i=0;i<n;i++) delete modeli[i]; return 0;
}
```

Задатак 2.

Написати програм на језику C++ који:

- a) чита са тастатуре $n \leq 10$ редова знакова,
- b) уписује редове знакова са тастатуре у текст датотеку dat1.txt,
- c) уписује исте редове знакова, без празних редова, у бинарну датотеку dat2.bin,
- d) чита редове из датотеке dat2.bin и само редове у задатом са тастатуре опсегу редних бројева редова, уписује у текст датотеку dat3.txt.

Решење 2.

```
#include <iostream>
#include <fstream>
#include <stdlib.h>
#include <string.h>
#define BROJ 10
#define MAX 80
using namespace std;
int main()
{
    int i, l, a, b, n, n2=0; char red[MAX+1];
    fstream f1, f2, f3;

    f1.open("dat1.txt",ios::out);if(!f1){cout<<" Neuspelo otvaranje dat!";exit(1); }
    f2.open("dat2.bin",ios::out | ios::binary);if(!f2){cout<<" Neuspelo otvaranje dat!";exit(1); }

    do{
        cout<<"\nUnesite n: ";cin>>n;cin.get();
    }while(n<1 || n>BROJ);

    cout<<"\nUnesite redove:\n";
    for(i=0; i<n; i++)
    {
        cin.getline(red,MAX+1);
        f1<<red<<endl;
        l=strlen(red);
        if(l!=0)
        {
            f2.write((char *)&l, sizeof(int));
            f2.write((char *)red, l+1);
            n2++;
        }
    }
    f1.close();f2.close();

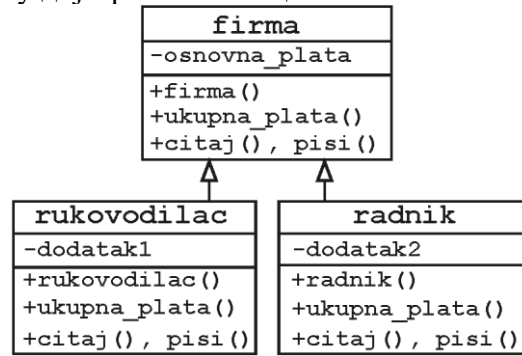
    f2.open("dat2.bin",ios::in | ios::binary);if(!f2){cout<<" Neuspelo otvaranje dat!";exit(1); }
    f3.open("dat3.txt",ios::out);if(!f3){cout<<" Neuspelo otvaranje dat!";exit(1); }
    do
    {
        cout<<"\nUnesite opseg: \n"; cin>>a>>b;
    }while(a<1 || a>n2 || b<1 || b>n2 || a>b);

    for(i=0; i<(a-1); i++)
    {
        f2.read((char *)&l, sizeof(int));
        f2.read((char *)&red, l+1);
    }
    for(i=0; i<(b-(a-1)); i++)
    {
        f2.read((char *)&l, sizeof(int));
        f2.read((char *)&red, l+1);
        f3<<red<<endl;
    }
    f2.close(); f3.close(); return 0;
}
```

Задатак 3.

Написати на језику C++:

а) дефиницију класа на основу дијаграма на слици



- б) дефиниције конструктора класа, који иницијализују сваки атрибут на вредност 0, дефиниције метода основне класа као виртуелних, а предефинисаних у изведеним класама, тако да читају са тастатуре вредности тражених атрибута, израчунавају укупну плату и уписују у тражену датотеку вредности одговарајућих атрибута класа,
- с) главну функцију која нуди опције 1.Rukovodilac / 2. Radnik / 3. Крај, у петљи од 20 итерација креира динамички објекте одговарајућих класа (у свакој итерацији у зависности од изабране опције) и позива над креираним објектима методе, да упишу у датотеку zaposleni.txt податке о плати сваког задатог руководиоца и радника.

Решење 3.

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

```
using namespace std;
```

```
class firma {
    int osnovna_plata;
public:
    firma() { osnovna_plata=0; }
    virtual void citaaj(){ cout<<"Osnovna plata? "; cin>>ws; cin>>osnovna_plata; }

    virtual void pisi(char *ime_dat)
    {
        fstream fout;
        fout.open(ime_dat, ios::app); if(!fout)exit(1);
        fout<<"\nOsnovna plata: "<<osnovna_plata<<" ";
        fout.close();
    }
};

class rukovodilac : public firma {
    int dodatak1;
public:
    rukovodilac() : firma() { dodatak1=0; }
    void citaaj(){ firma::citaaj(); cout<<"Dodatak rukovodioca? "; cin>>ws; cin>>dodatak1; }
    void pisi(char *ime_dat)
    {
        fstream fout;
        fout.open(ime_dat, ios::app); if(!fout)exit(1);
        firma::pisi(ime_dat); fout<<"Dodatak rukovodioca: "<<dodatak1<<endl;
        fout.close();
    }
};
```

```

class radnik : public firma {
    int dodatak2;
public:
    radnik() : firma() { dodatak2=0; }
    void citaj(){ firma::citaj(); cout<<"Dodatak radnika? ";    cin>>ws; cin>>dodatak2; }
    void pisi(char *ime_dat)
    {
        fstream fout;
        fout.open(ime_dat, ios::app); if(!fout)exit(1);
        firma::pisi(ime_dat); fout<<"Dodatak radnika: "<<dodatak2<<endl;
        fout.close();
    }
};

//Glavna funkcija: kreiranje objekata, realizacija programa
int main()
{
    firma *objekat[10];
    int i, n=0, izbor;

    cout<< "\n\nNapraviti izbor: ";
    while(1)
    {
        cout << "\n\nIzbor: 1.Rukovodilac, 2.Radnik, 3.Izlaz ";
        cin >> izbor; if(izbor==3) break;

        objekat[n]=0;
        switch(izbor)
        {
            case 1:
                objekat[n]=new rukovodilac;
                if(objekat[n]) //provera dinamicke dodele
                {
                    objekat[n]->citaj();
                    objekat[n++]->pisi("zaposleni.txt");
                }
                break;
            case 2:
                objekat[n]=new radnik;
                if(objekat[n]) //provera dinamicke dodele
                {
                    objekat[n]->citaj();
                    objekat[n++]->pisi("zaposleni.txt");
                }
                break;
            default:
                break;
        }
        if(n==10)
            break;
    }
    for(int i=0; i<n; i++)
        delete objekat[i]; //oslobadjanje dinamicki kreiranih objekata
    return 0;
}

```