

## Programiranje 2

### Računske vježbe 10

Projektovati klasu za obradu vektora (niza) realnih brojeva sa zadatim opsezima indeksa. Za razrješavanje konfliktnih situacija koristiti mehanizam obrade izuzetaka. Napisati glavni program za prikazivanje mogućnosti klase.

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

class vektor
{
    private:
        float *niz;
        int min_ops;
        int max_ops;
    public:
        enum greska //Kodovi gresaka
        {OK,
        OPSEG, //neispravan opseg indeksiranja
        MEMORIJA, //dodjela memorije nije uspjela,
        PRAZAN, //vektor je prazan
        INDEKS, //indeks je izvan opsega
        DUZINA}; //neusaglasene duzine vektora
        vektor(){niz=0;}
        vektor(int, int);
        vektor(const vektor &);
        ~vektor(){delete []niz; niz=0;}
        vektor & operator=(const vektor &);
        float & operator[](int)const;
        friend double operator*(const vektor &, const vektor &);
};

vektor::vektor(int a, int b):min_ops(a),max_ops(b)
{
    if(min_ops>max_ops) throw(OPSEG);
    if(!(niz=new float[max_ops-min_ops+1])) throw MEMORIJA;
    for(int i=0; i<max_ops-min_ops+1; i++) niz[i]=0;
}

vektor::vektor(const vektor &a):min_ops(a.min_ops), max_ops(a.max_ops)
{
    if(!(niz=new float[max_ops-min_ops+1])) throw MEMORIJA;
    for(int i=0;i<max_ops-min_ops+1;i++) niz[i]=a.niz[i];
}

vektor & vektor::operator=(const vektor &a)
{
    if(a.niz==0) throw PRAZAN;
    if(&a!=this)
    {
        delete []niz;
        min_ops=a.min_ops;
        max_ops=a.max_ops;
        if(!(niz=new float[max_ops-min_ops+1])) throw MEMORIJA;
        else for(int i=0;i<max_ops-min_ops+1;i++) niz[i]=a.niz[i];
    }
    return *this;
}
```

```

float & vektor::operator[](int i) const
{
    if(!niz) throw PRAZAN; // u niz nije nista upisano, samo je izvršen default konstruktor
    else if(i<min_ops || i>max_ops) throw INDEKS;
    else return niz[i-min_ops];
}

double operator*(const vektor &a, const vektor &b)
{
    if(!a.niz || !b.niz) throw vektor::PRAZAN;
    else if((a.max_ops-a.min_ops) != (b.max_ops-b.min_ops)) throw vektor::DUZINA;
    else
    {
        double s=0;
        for(int i=0; i<a.max_ops-a.min_ops+1; i++)
            s+=a.niz[i]*b.niz[i];
        return s;
    }
}

int main()
{
    while(1)
    {
        try
        {
            int maks, minm;
            cout<<"Unesi opseg indeksa prvog niza"<<endl;
            cin>>minm>>maks;
            if(cin.eof()) throw 1;
            vektor v1(minm, maks);
            cout<<"Elementi prvog niza"<<endl;
            for(int i=minm; i<=maks; i++) cin>>v1[i];

            cout<<"Unesi opseg indeksa drugog niza"<<endl;
            cin>>minm>>maks;
            vektor v2(minm, maks);
            cout<<"Elementi drugog niza"<<endl;
            for(int i=minm; i<=maks; i++) cin>>v2[i];

            cout<<"Skalarni proizvod dva zadata niza je "<<v1*v2<<endl;
        }
        catch(vektor::greska g)
        {
            char *poruke[]={ " ", "Neispravan opseg indeksa!", \
                "Neuspjelo dodjeljivanje memorije!", "vektor je prazan!", \
                "Indeks je izvan opsega!", "Neusaglasene duzine vektora"};
            cout<<poruke[g]<<endl;
        }
        catch(...)
        {
            cout<<"Kraj unosa"<<endl;
            break;
        }
    }
}

```