

ZADAĆA 4: PREOPTEREĆIVANJE OPERATORA

Svaki zadatak nosi određeni broj bodova po podzadatku koji su naznačeni u zagradi. Svi studenti koji predaju zadaću trebaju pristupiti na ScriptRunner5 sustav (<https://mathos.scriptrunner.carnet.hr/>) i tamo objavljivati svoje zadaće u mapi ZADACE->ZADACA_4.

Svi oni studenti koji predaju zadaće moraju biti nazočni na vježbama.

ZADATAK 1 (5+10+10+25):

Sljedećim kodom vam je dan primjer (prototip) klase **Vektor**.

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

class Vektor {
public:
    int *P;
    int len;
    int curr_len;
public:
    // konstruktori
    Vektor() {P = NULL;}
    Vektor(int s);
    Vektor(const Vektor &V); // konstruktor kopiranja

    // ostale metode (interface klase Vektor)
    bool append(int); // dodaje int na kraj
    bool search(int); // vrati 'true' ako je pronašao traženi int
    void ispisi(); // ispisuje Vektor na ekran (standardni output)
    void sort(); // sortira elemente Vektora
    void Zbroji(const Vektor& V1, const Vektor& V2); // V1 + V2
    void Mnozi(const Vektor& V1); // V1 * V2 skalarno

    // operatori
    void operator=(const Vektor&);
    Vektor operator+(const Vektor&) const;
    Vektor operator-(const Vektor&) const;
    int operator*(const Vektor&) const;
    void operator+=(const Vektor&);
    void operator-=(const Vektor&);
    bool operator==(const Vektor&) const;
    int& operator[](int) const;

    // destruktork
    ~Vektor() {delete [] P;}
};
```

Implementirajte:

- 1) ponovo SAMI napišite konstruktore **Vektor(int)** i **Vektor(const Vektor&)**. Primijetite da je pretpostavljeni (defaultni) konstruktor **Vektor()** u gornjoj klasi **Vektor** već implementiran.

NAPOMENA: Sve ovo već imate implementirano u materijalima s predavanja. Od vas se očekuje da ponovite gradivo s predavanja, ZATVORITE materijale i onda sami implementirate ova dva konstruktora. Ako to uspijete, tek tada ste sigurni da ste razumjeli svog predavača.

- 2) ponovo SAMI napišite metode **append(int)**, **search(int)**, **ispisi()**, **sort()**, **Zbroji(const Vektor&, const Vektor)**, **Mnozi(const Vektor&, const Vektor&)**.

NAPOMENA: Sve ovo, izuzev metode **Mnozi(const Vektor&, const Vektor&)**, već imate implementirano u materijalima s predavanja. Od vas se očekuje da ponovite gradivo s predavanja, ZATVORITE materijale i onda sami implementirate sve ove metode. Ako to uspijete, tek tada ste sigurni da ste razumjeli svog predavača.

- 3) ponovo SAMI napišite operatore **operator=(const Vektor&)**, **operator+(const Vektor&)**, **operator*(const Vektor&)**, **operator-(const Vektor&)** i **operator[] (int)**

NAPOMENA: Ova četiri operatora već imate implementirana u materijalima s predavanja. Od vas se očekuje da ponovite gradivo s predavanja, ZATVORITE materijale i onda sami implementirate sve ove operatore. Ako to uspijete, tek tada ste sigurni da ste razumjeli svog predavača.

- 4) operatore **operator+=(const Vektor&)**, **operator-=(const Vektor&)**, **operator*=(const Vektor&)** i **operator/=(const Vektor&)**

ZADATAK 2 (5+10+10+10+5+10):

Racionalni brojevi u C++ programskom jeziku su reprezentirani `float` tipom podataka. Ponekad je poželjno omogućiti računanje s racionalnim brojevima na klasičan matematički način, dakle svaki broj iz $x \in \mathbb{Q}$ predstaviti kao par (a, b) za koje je $x = a/b$.

Neka je zadan je sljedeći prototip klase **Razlomak**:

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

class Razlomak
{
public:
    int brojnik, nazivnik;
public:
    Razlomak();
    Razlomak(int a, int b);
    Razlomak(const Razlomak q);

    // operatori pridruzivanja
    void operator = (const Razlomak&);
    void operator += (const Razlomak&);
    void operator -= (const Razlomak&);
    void operator *= (const Razlomak&);
    void operator /= (const Razlomak&);
```

```

// aritmeticki operatori
Razlomak operator + (const Razlomak&) const;
Razlomak operator - (const Razlomak&) const;
Razlomak operator * (const Razlomak&) const;
Razlomak operator / (const Razlomak&) const;

// logicki operatori
bool operator == (const Razlomak&) const;
bool operator <= (const Razlomak&) const;
bool operator >= (const Razlomak&) const;
bool operator < (const Razlomak&) const;
bool operator > (const Razlomak&) const;

// pretvaranje razlomka u odgovarajuci float
float decimal() const; // Razlomak u float
float round() const;   // Najbliže cijelo

// ukoliko razlomak nije maksimalno skracen, skрати ga
// i vraca false, u protivnom vraca true
bool skрати();

// ispisi razlomak na ekran
void ispisi();

// ispisi razlomak na ekran kao mješoviti broj
void ispisi_mjesoviti();

};

```

Potrebno je implementirati sljedeće:

1. pretpostavljeni konstruktor, konstruktor koji prima vrijednosti brojnika i nazivnika, te konstruktor kopiranja;
2. operatore pridruživanja =, +=, -=, *=, /=
3. aritmetičke operatore +, -, *, / koji vraćaju novi racionalni broj;
4. logičke operatore ==, <=, >=, <, > koji odgovaraju operatorima usporedbe nad racionalnim brojevima.
5. Metode
 - a. **decimal()** - pretvara razlomak u float (npr. $\frac{1}{2} \rightarrow 0.5$),
 - b. **round()** - zaokružuje razlomak na najbliže cijelo. Ukoliko se broj nalazi točno između dva cijela broja proizvoljno odlučite na koji od ta dva ćete ga zaokružiti.
 - c. **ispisi()** i **ispisi_mjesoviti()** - ispisuju razlomak na ekran kao klasičan razlomak (brojnik kroz nazivnik) ili kao mješoviti broj.
 - d. **skрати()** - vraća `false` ukoliko razlomak nije skraćen, a zatim ga skрати, odnosno `true` ukoliko je već skraćen (npr. $\frac{1}{2} \rightarrow true$, $\frac{2}{8} \rightarrow false$)

NAPOMENA: Testirajte svaku svoju metodu u **main()** funkciji.