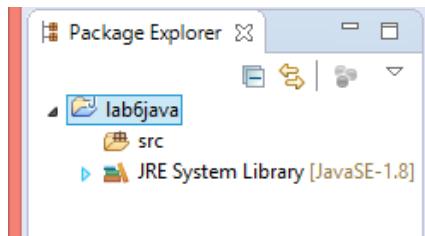


# LUCRARE PRACTICĂ NR. 6

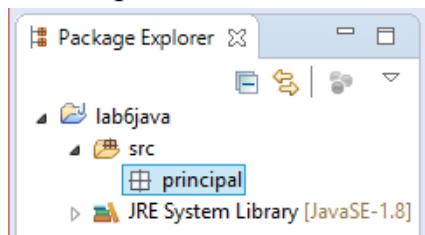
## Programarea Java

- Îndepliniți următoarele sarcini într-un IDE. În continuare se vor ilustra pașii în IDE Eclipse.

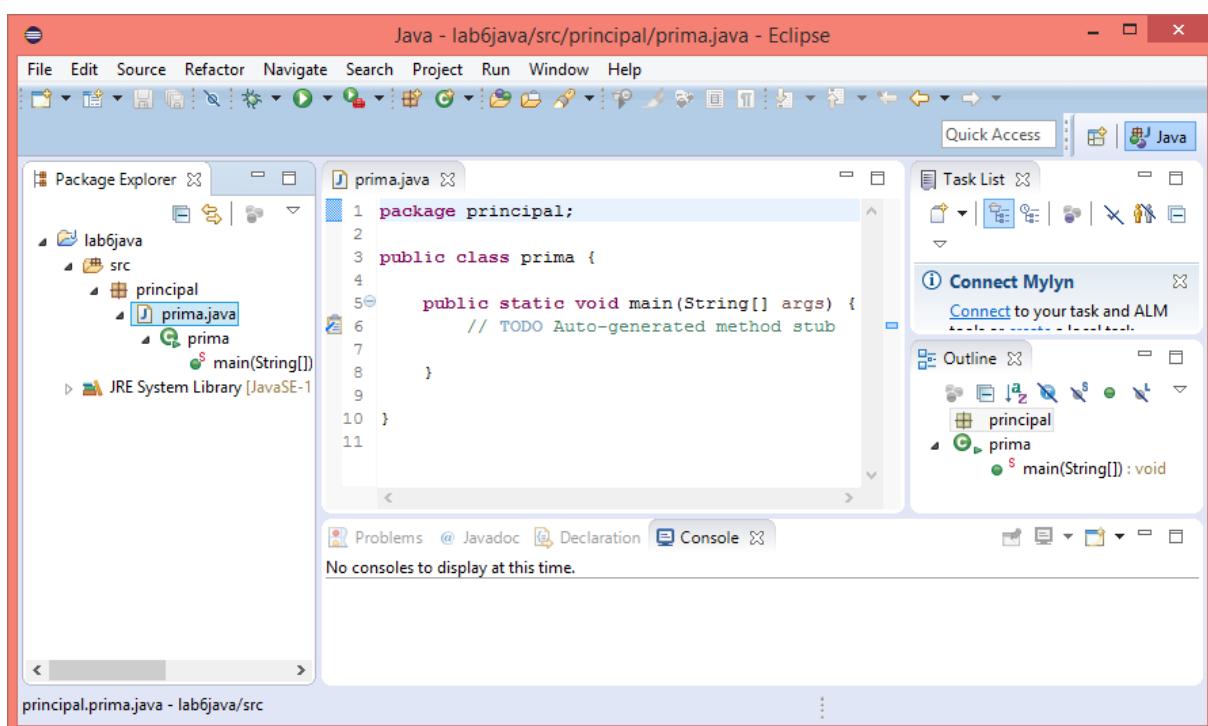
1. Creați un nou proiect, cu numele **lab6java**.



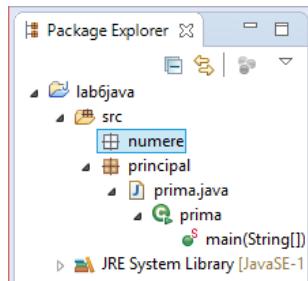
2. În acest proiect creați un nou pachet cu numele **principal**.



3. În pachetul **principal** creați clasa publică **prima**, care să conțină metoda **main**.



4. În proiectul **lab2java** mai creați un pachet cu numele **numere**.



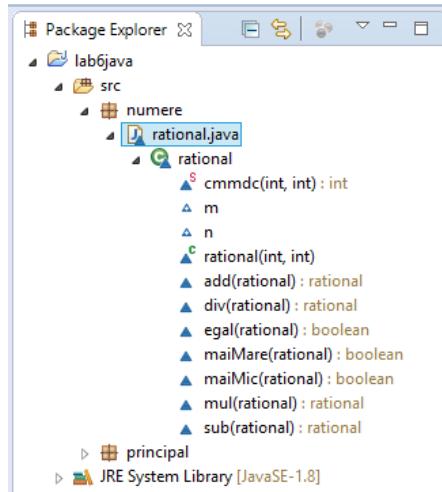
5. În pachetul **numere** creați o clasă **rational**, cu următorul conținut:

```
class rational {
    int m,n;
    rational (int m, int n)
    {
        if (n!=0)
        {
            int d=cmmdc(m,n);
            this.m=m/d;
            this.n=n/d;
        }
        else System.out.println("Numitorul este nul!");
    }
    rational mul(rational r)
    {
        return new rational (m*r.m,n*r.n);
    }
    static int cmmdc(int m, int n)
    {
        if (n!=0) return cmmdc(n,m%n);
        else return m;
    }
    rational add(rational r)
    {
        return new rational (m*r.n+r.m*n,n*r.n);
    }
    rational sub(rational r)
    {
        return new rational (m*r.n-r.m*n,n*r.n);
    }
    rational div(rational r)
    {
        if (r.m!=0) {
            return new rational (m*r.n,n*r.m);
        }
        else return new rational(0,1);
    }
    boolean maiMare(rational r)
    {
        return (double) m/n > (double) r.m/r.n;
    }
    boolean maiMic(rational r)
    {
        return (double) m/n < (double) r.m/r.n;
    }
    boolean egal(rational r)
    {
```

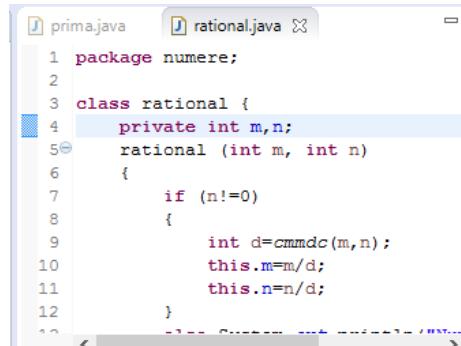
```

        return (double) m/n == (double) r.m/r.n;
    }
}

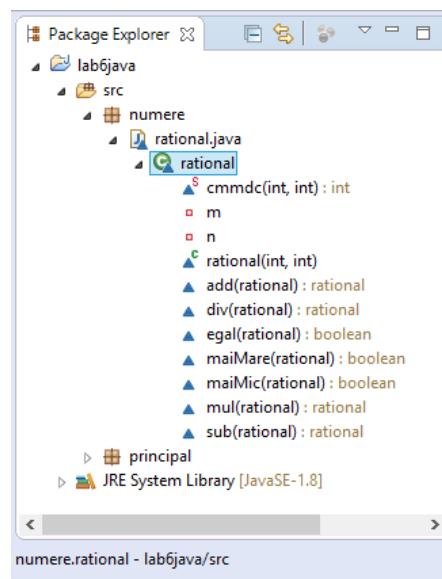
```

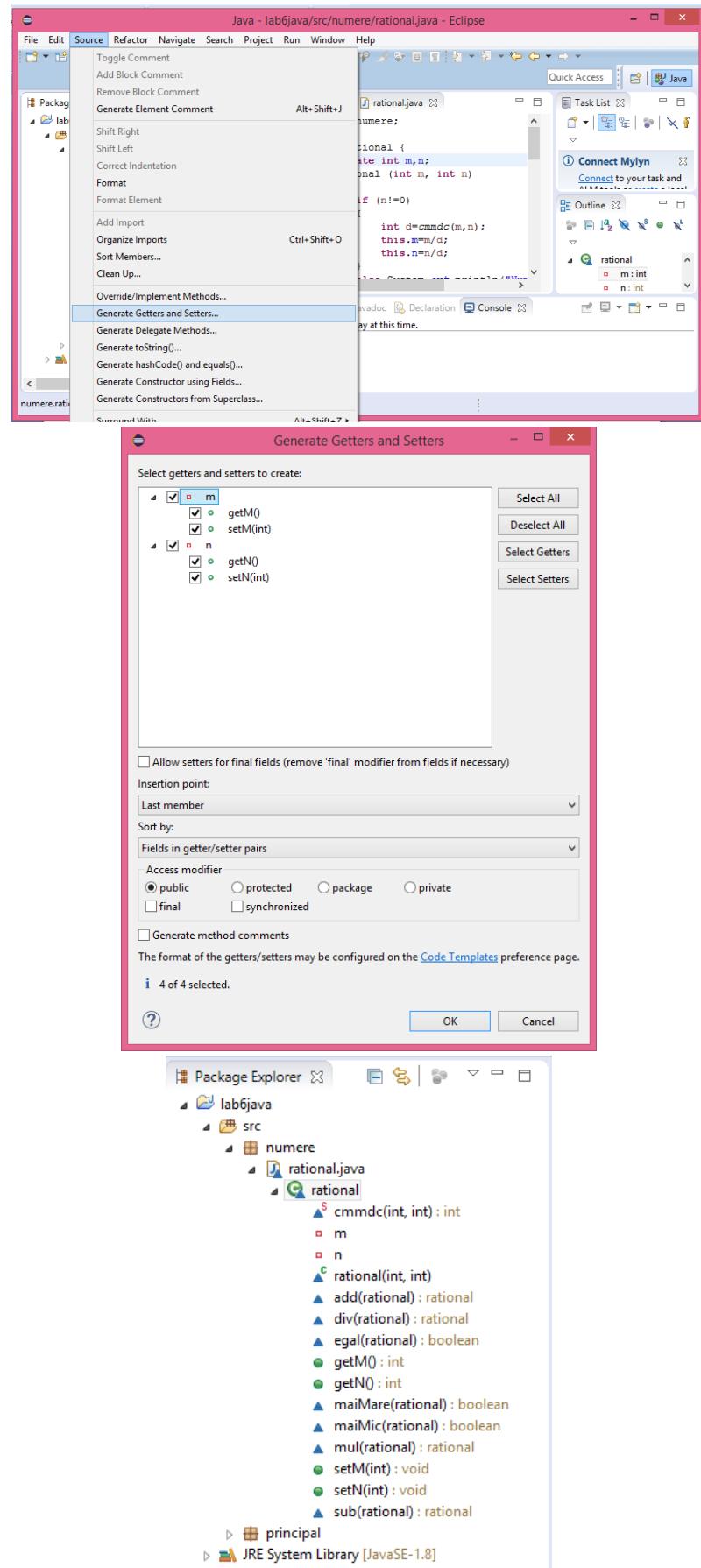


6. La variabilele **m** și **n** ale clasei **rational** puneti modificatorul **private** și creați acestor variabile într-un mod automat metodele **set** și **get**.



În momentul în care clasa **rational** va fi activă în fereastra **Package Explorer**, în meniul **Source** se va selecta **Generate Getters and Setters**.





7. În clasa **rational** creați o altă clasă, care să fie o clasă internă cu numele **Rvector**, care să conțină două metode:

- Pentru calculul sumei unui vector de numere raționale, rezultatul să fie returnat un număr **rational**.
- Pentru sortarea crescător a vectorului de numere rationale, rezultatul să fie un vector de numere de tip **rational**.

```
class Rvector {  
  
    public String suma(Rational arr[]) {  
        Rational sum = new Rational(0, 1);  
  
        for(int i=0; i<arr.length; i++) {  
            sum = sum.add(arr[i]);  
        }  
  
        return sum.toString();  
    }  
  
    public Rational[] sort(Rational arr[]) {  
        for(int i=0; i<arr.length; i++) {  
            for(int j=0; j<arr.length - 1; j++) {  
                if (arr[j].maiMare(arr[j+1])) {  
                    Rational r = arr[j];  
                    arr[j] = arr[j+1];  
                    arr[j+1] = r;  
                }  
            }  
        }  
        return arr;  
    }  
}
```

8. În clasa principal elaborați metode care utilizează metodele clasei **rational** pentru a rezolva următoarele sarcini:

- Se citesc două numere raționale **q1** și **q2**, nenule. Se cere să se afișeze **q1+q2**, **q1-q2**, **q1\*q2**, **q1/q2**.

```
InputStreamReader stream = new InputStreamReader(System.in);  
BufferedReader bf = new BufferedReader(stream);  
  
Rational q1 = null;  
Rational q2 = null;  
Rational res = null;  
int n1, n2;  
  
System.out.println("Introduceti numarul Q1");  
n1 = Integer.parseInt(bf.readLine());  
System.out.println("--");  
n2 = Integer.parseInt(bf.readLine());  
q1 = new Rational(n1, n2);  
  
System.out.println("Introduceti numarul Q2");  
n1 = Integer.parseInt(bf.readLine());  
System.out.println("--");  
n2 = Integer.parseInt(bf.readLine());  
q2 = new Rational(n1, n2);  
  
res = q1.add(q2);  
System.out.println("Suma:");  
System.out.println(res.getM() + " / " + res.getN());  
  
res = q1.sub(q2);  
System.out.println("Diferenta:");  
System.out.println(res.getM() + " / " + res.getN());  
  
res = q1.mul(q2);  
System.out.println("Multirea:");  
System.out.println(res.getM() + " / " + res.getN());
```

- Se citesc  $k$  numere raționale. Se cere să se afișeze suma lor sub forma de număr rațional.

```

InputStreamReader stream = new InputStreamReader(System.in);
BufferedReader bf = new BufferedReader(stream);

System.out.println("N=");
int n = Integer.parseInt(bf.readLine());

Rational[] numbers = new Rational[n];
for(int i=0; i<n; i++) {
    System.out.println("Numarul:");
    int n1 = Integer.parseInt(bf.readLine());
    int n2 = Integer.parseInt(bf.readLine());
    Rational nr = new Rational(n1, n2);
    numbers[i] = nr;
}
Rational r = new Rational();
System.out.println("Suma: " + r.suma(numbers));

```

- Se citește un vector cu  $k$  componente numere raționale (obiecte ale clasei rational). Se cere să se afișeze vectorul sortat.

```

InputStreamReader stream = new InputStreamReader(System.in);
BufferedReader bf = new BufferedReader(stream);

System.out.println("N=");
int n = Integer.parseInt(bf.readLine());

Rational[] numbers = new Rational[n];
for(int i=0; i<n; i++) {
    System.out.println("Numarul:");
    int n1 = Integer.parseInt(bf.readLine());
    int n2 = Integer.parseInt(bf.readLine());
    Rational nr = new Rational(n1, n2);
    numbers[i] = nr;
}
Rational r = new Rational();
System.out.println("Sortarea");
System.out.print("Vectorul ne sortat: ");
for(int i=0; i<numbers.length; i++) {
    System.out.print(numbers[i].toString() + "; ");
}

System.out.print("\nVectorul sortat: ");
Rational[] vecSort = r.sort(numbers);
for(int i=0; i<vecSort.length; i++) {
    System.out.print(vecSort[i].toString() + "; ");
}

```

- În metoda main al clasei principal apelați metodele elaborate.

## PENTRU LUCRU INDIVIDUAL ȘI ANTRENAMENT!

1. Utilizând noțiunea de polimorfismul, scrieți un program care citește un număr natural  $n$  și dacă este prim, afișați succesorul său în mulțimea numerelor prime, dacă este impar (fără a fi prim) afișați succesorul său în mulțimea numerelor impare, iar dacă este par, afișați succesorul său în mulțimea numerelor pare.

Exemple:

n=7 se afișează 11

n=21 se afișeazză 23

n=8 se afișează 10

2. Fiind date interfața de mai jos și 3 implementări ale ei, se cere să se scrie un program care afișează valoarea celor 3 funcții conținute de implementări pentru o valoare citită de la tastatură:

```
interface functie {
    int f(int x);
}
class f1 implements functie
{
    public int f(int x) {return x*x-x; }
}
class f2 implements functie
{
    public int f(int x) {return 8*x-1; }
}
class f3 implements functie
{
    public int f(int x) {return 3*x*x-x+10; }
}
```