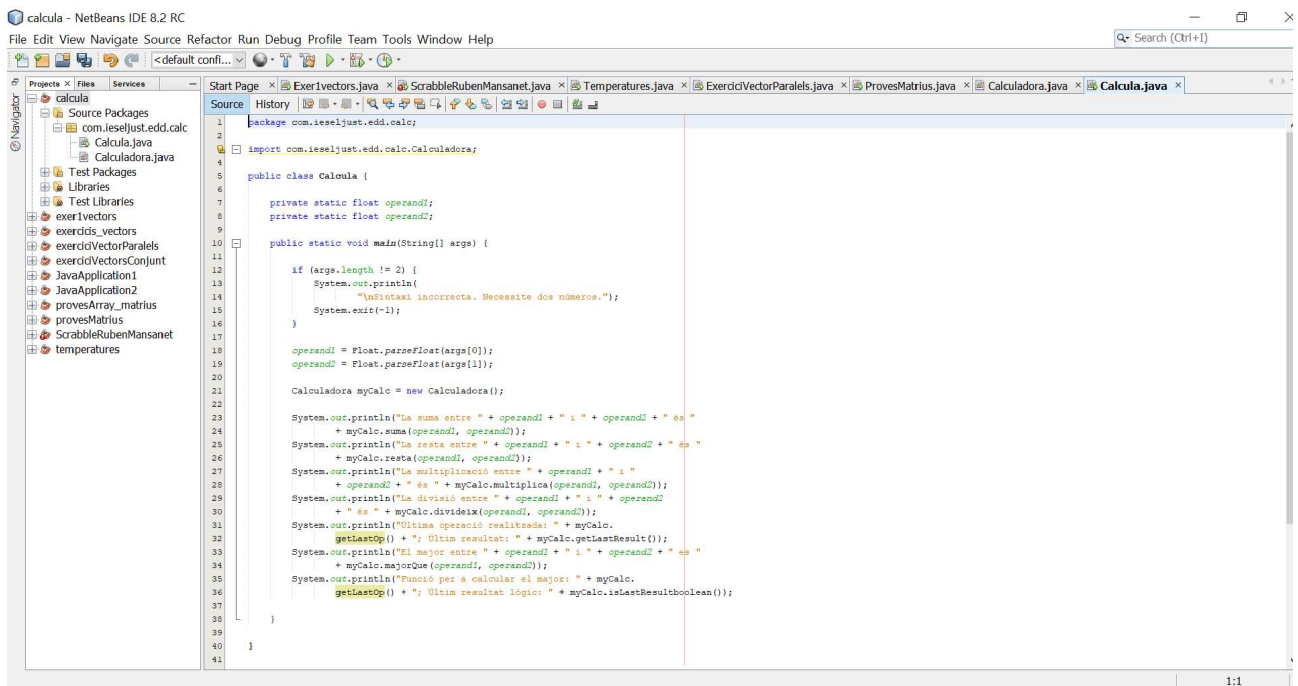


## ACTIVITATS ANT

### TO-DO

A partir del codi proporcionat, implementeu la nova funcionalitat a la calculadora MajorQue, que indique si el primer argument que li proporcionem és major que el segon. El tipus de valor de retorn haurà de ser lògic.

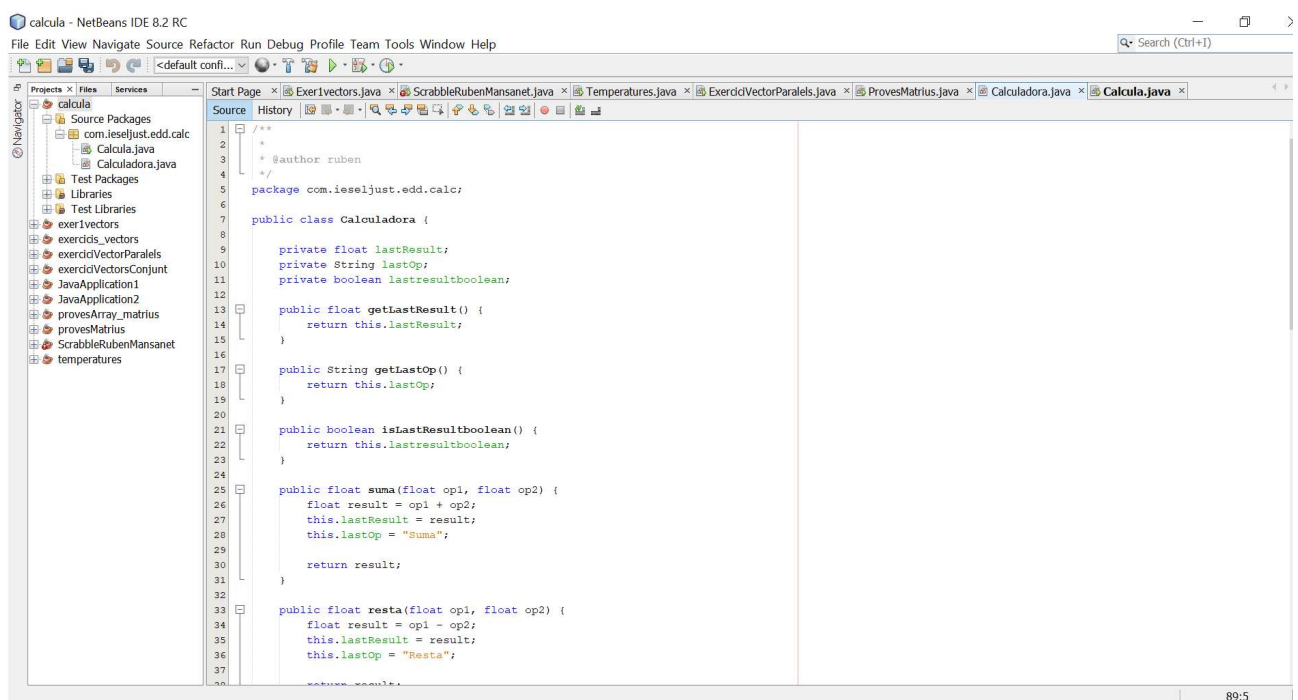
Feu ús d'aquest mètode en l'aplicació mostrant, immediatament després d'escriure el resultat de la divisió, el resultat d'aquesta comparació.



```

1 package com.ieseljust.edd.calc;
2
3 import com.ieseljust.edd.calc.Calculadora;
4
5 public class Calcula {
6
7     private static float operand1;
8     private static float operand2;
9
10    public static void main(String[] args) {
11
12        if (args.length != 2) {
13            System.out.println(
14                "\nSintaxi incorrecta. Necessite dos números.");
15            System.exit(-1);
16        }
17
18        operand1 = Float.parseFloat(args[0]);
19        operand2 = Float.parseFloat(args[1]);
20
21        Calculadora myCalc = new Calculadora();
22
23        System.out.println("La suma entre " + operand1 + " i " + operand2 + " és "
24            + myCalc.suma(operand1, operand2));
25        System.out.println("La resta entre " + operand1 + " i " + operand2 + " és "
26            + myCalc.resta(operand1, operand2));
27        System.out.println("La multiplicació entre " + operand1 + " i "
28            + operand2 + " és " + myCalc.multiplika(operand1, operand2));
29        System.out.println("La divisió entre " + operand1 + " i " + operand2
30            + " és " + myCalc.divideix(operand1, operand2));
31        System.out.println("Última operació realitzada: " + myCalc.
32            getLaastOp() + "; Últim resultat: " + myCalc.getLaastResult());
33        System.out.println("El major entre " + operand1 + " i " + operand2 + " és "
34            + myCalc.majorQue(operand1, operand2));
35        System.out.println("Funció per a calcular el major: " + myCalc.
36            getLaastOp() + "; Últim resultat lògic: " + myCalc.isLaastResultboolean());
37    }
38 }

```

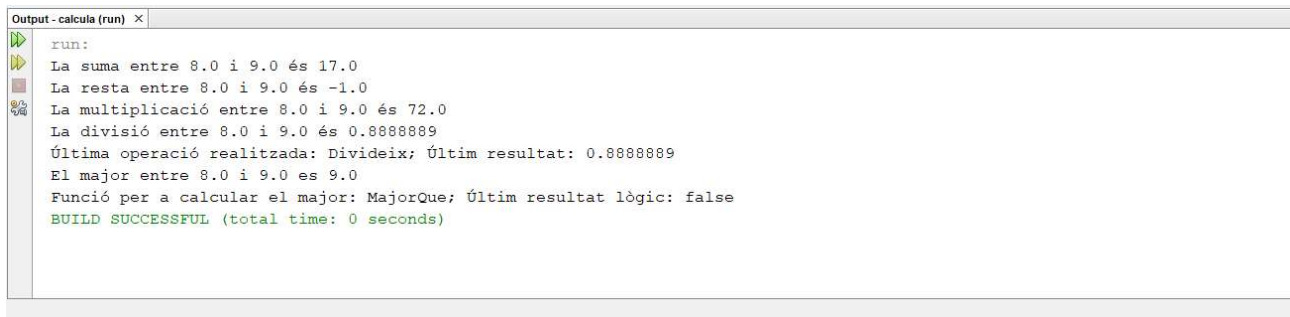
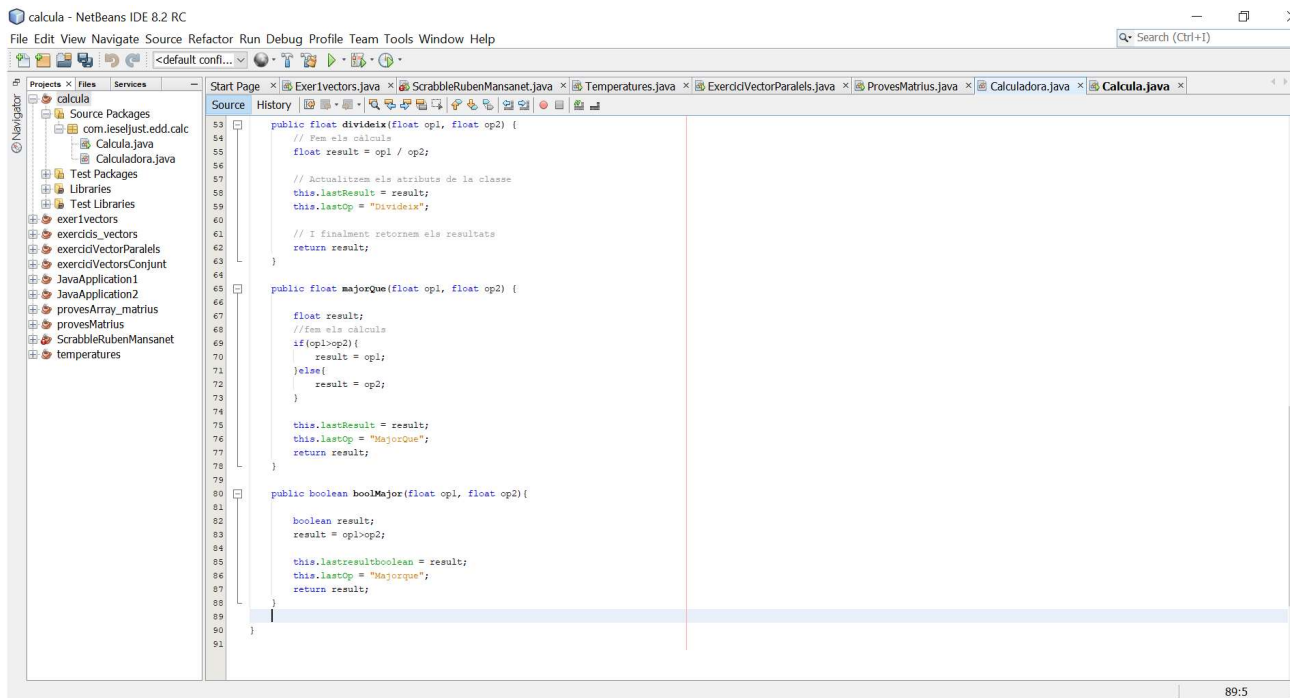


```

1 /**
2  *
3  * @author ruben
4  */
5 package com.ieseljust.edd.calc;
6
7 public class Calculadora {
8
9     private float lastResult;
10    private String lastOp;
11    private boolean lastResultboolean;
12
13    public float getLaastResult() {
14        return this.lastResult;
15    }
16
17    public String getLaastOp() {
18        return this.lastOp;
19    }
20
21    public boolean isLaastResultboolean() {
22        return this.lastResultboolean;
23    }
24
25    public float suma(float op1, float op2) {
26        float result = op1 + op2;
27        this.lastResult = result;
28        this.lastOp = "Suma";
29
30        return result;
31    }
32
33    public float resta(float op1, float op2) {
34        float result = op1 - op2;
35        this.lastResult = result;
36        this.lastOp = "Resta";
37
38        return result;
39    }
40 }

```

## Ruben Mansanet 1DAMSP



## TO-DO

**Creeu el projecte Ant amb el vostre codi font i verifiqueu que aquest es neteja, es compila i s'executa correctament amb els arguments.**

```
eljust@ruben-VirtualBox:~/NetBeansProjects/calcula/src$ ant clean
Buildfile: /home/eljust/NetBeansProjects/calcula/src/build.xml

clean:
    [delete] Deleting directory /home/eljust/NetBeansProjects/calcula/src/classes

BUILD SUCCESSFUL
Total time: 0 seconds
eljust@ruben-VirtualBox:~/NetBeansProjects/calcula/src$ ant compile
Buildfile: /home/eljust/NetBeansProjects/calcula/src/build.xml

clean:

compile:
    [mkdir] Created dir: /home/eljust/NetBeansProjects/calcula/src/classes
    [javac] Compiling 2 source files to /home/eljust/NetBeansProjects/calcula/src/classes

BUILD SUCCESSFUL
Total time: 2 seconds
eljust@ruben-VirtualBox:~/NetBeansProjects/calcula/src$ ant run -Darg0=6 -Darg1=9
Buildfile: /home/eljust/NetBeansProjects/calcula/src/build.xml

clean:
    [delete] Deleting directory /home/eljust/NetBeansProjects/calcula/src/classes

compile:
    [mkdir] Created dir: /home/eljust/NetBeansProjects/calcula/src/classes
    [javac] Compiling 2 source files to /home/eljust/NetBeansProjects/calcula/src/classes

run:
    [java] La suma entre 6.0 i 9.0 és 15.0
    [java] La resta entre 6.0 i 9.0 és -3.0
    [java] La multiplicació entre 6.0 i 9.0 és 54.0
    [java] La divisió entre 6.0 i 9.0 és 0.6666667
    [java] Última operació realitzada: Divideix; Últim resultat: 0.6666667

BUILD SUCCESSFUL
Total time: 2 seconds
eljust@ruben-VirtualBox:~/NetBeansProjects/calcula/src$
```

## TO-DO

**Creeu ara el projecte de la calculadora amb Netbeans, amb el vostre codi font, i comproveu els resultats de l'execució. A mode de conclusió, reflexioneu (per escrit) sobre les següents qüestions:**

El codi es el mateix que el del primer TO-DO. Pase a contestar les preguntes directament.

**Quin dels dos mètodes (de forma manual o amb l'assistent de Netbeans) creieu que és millor?**

La meua opnió es que els dos mètodes estan bé sempre i quan siguem projectes xicotets com aquest. Preferisc l'assistent de Netbeans ja que al treballar en un entorn gràfic es més senzill i més pràctic a l'hora de treballar amb ell.

**Quin us resulta més còmode d'utilitzar?**

Netbeans per l'entorn gràfic, al igual que va proporcionant els errors que puguem apareixer durant el desenvolupament del programa. Si que he vist i he tingut algun problema amb Netbeans de que si no es crea el projecte bé desde un principi apareix un error de que no troba la classe.

### **Quin us ofereix més control sobre tot el què es crea?**

El de Netbeans, el poder tindre ben estructurat tot el projecte amb el main i la classe crec que es més ràpit treballar amb ell ja que crea el Xml i es més automàtic.