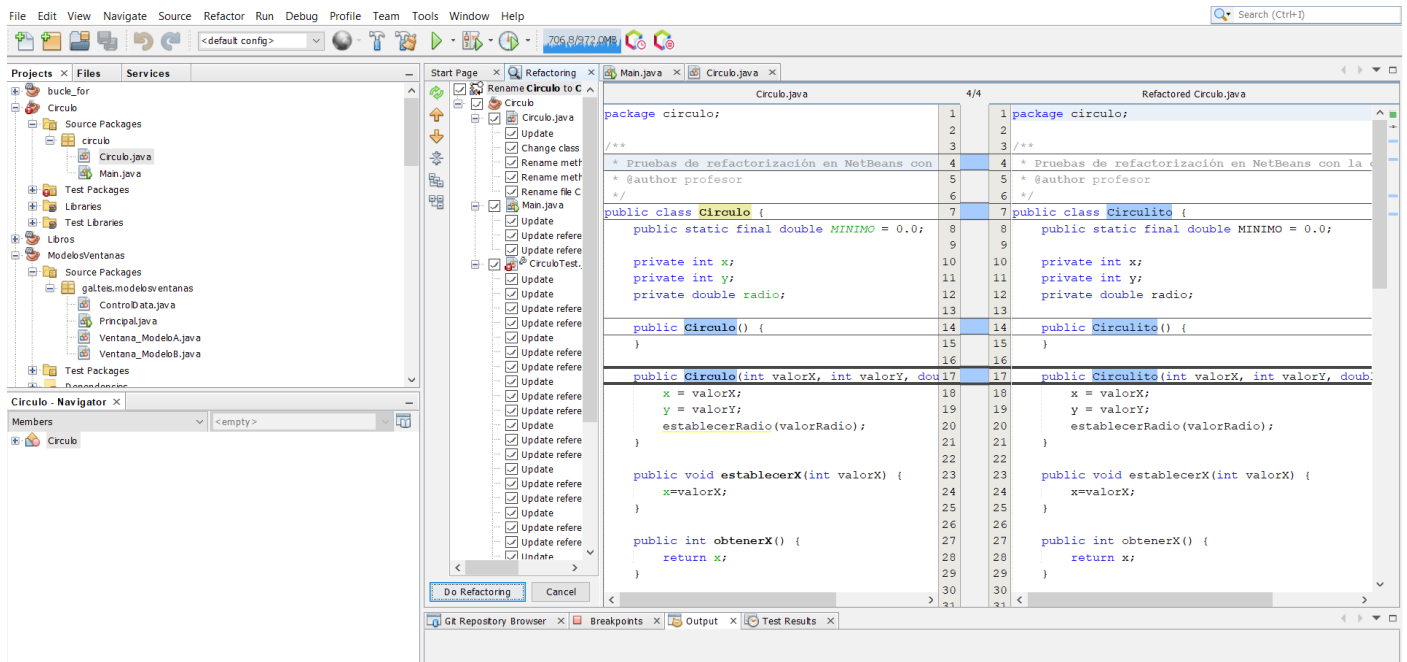


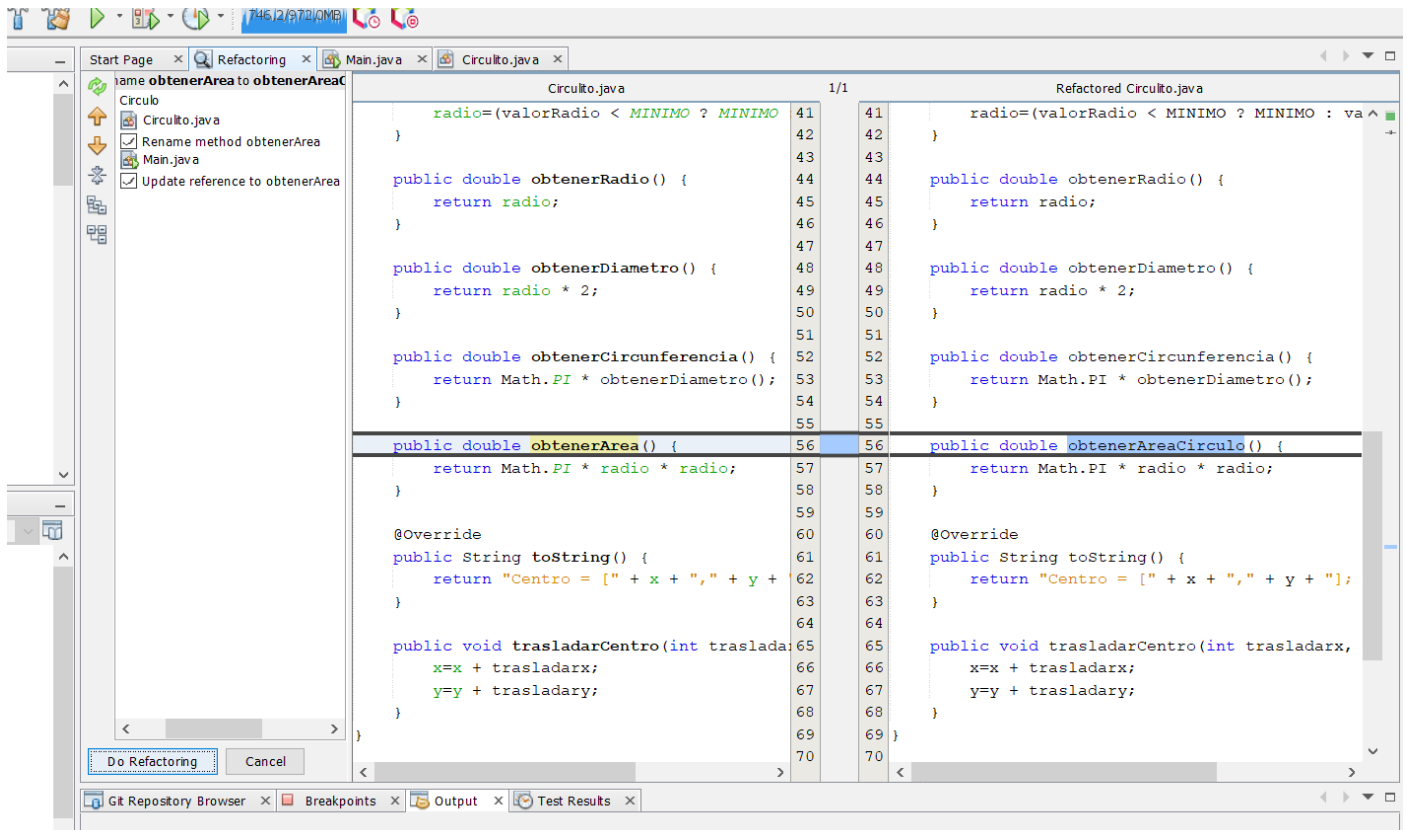
# A1 - Tarefa 1. 1. Exercício para practicar coa reestruturación de código Java en NetBeans

Realizar as seguintes reestruturacións sobre o proxecto Circulo que dispón dunha clase de probas unitarias JUnit. Recorda que despois de realizar a reestruturación de cada exercicio, débese comprobar coas probas unitarias que a clase segue pasando as probas.

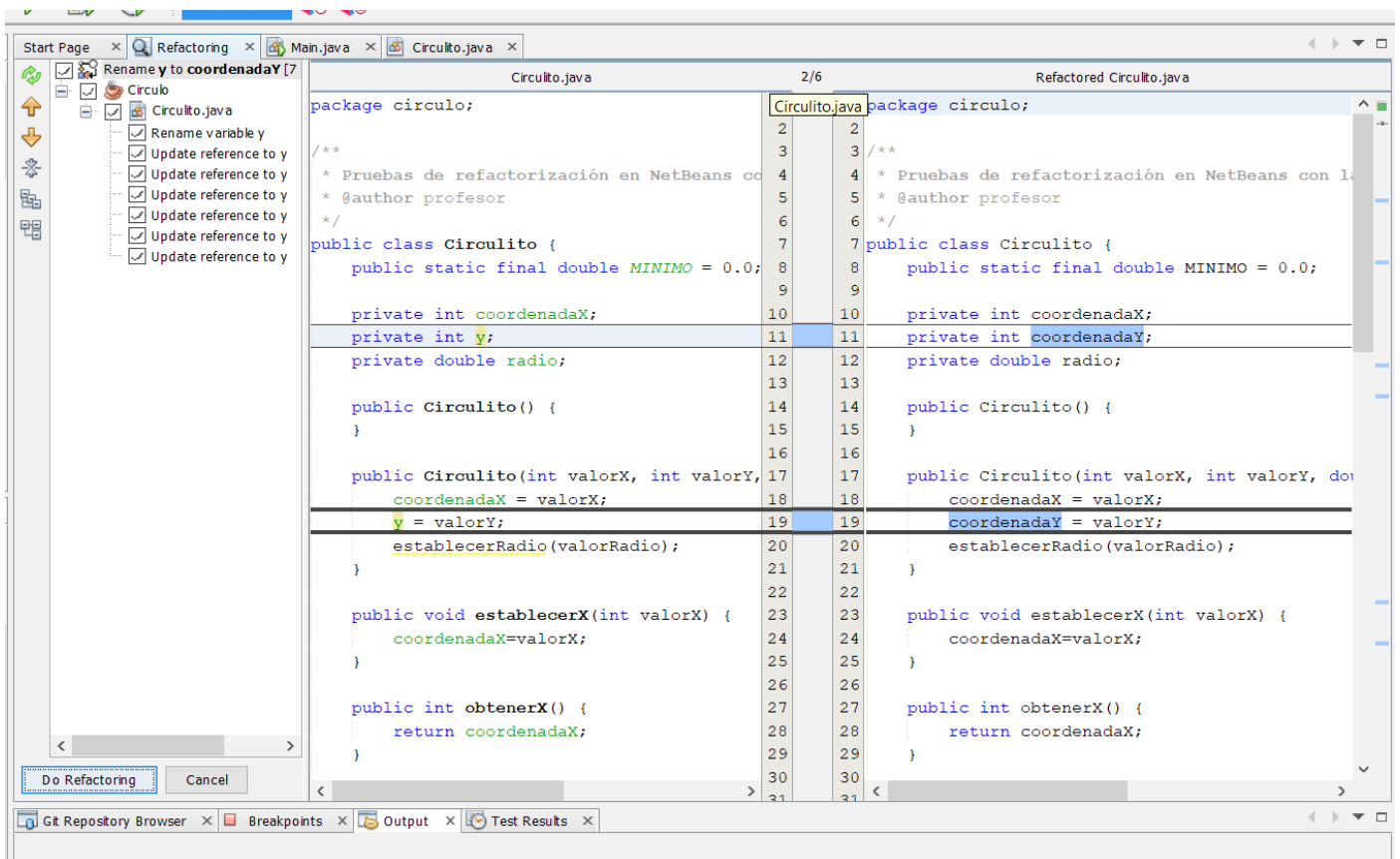
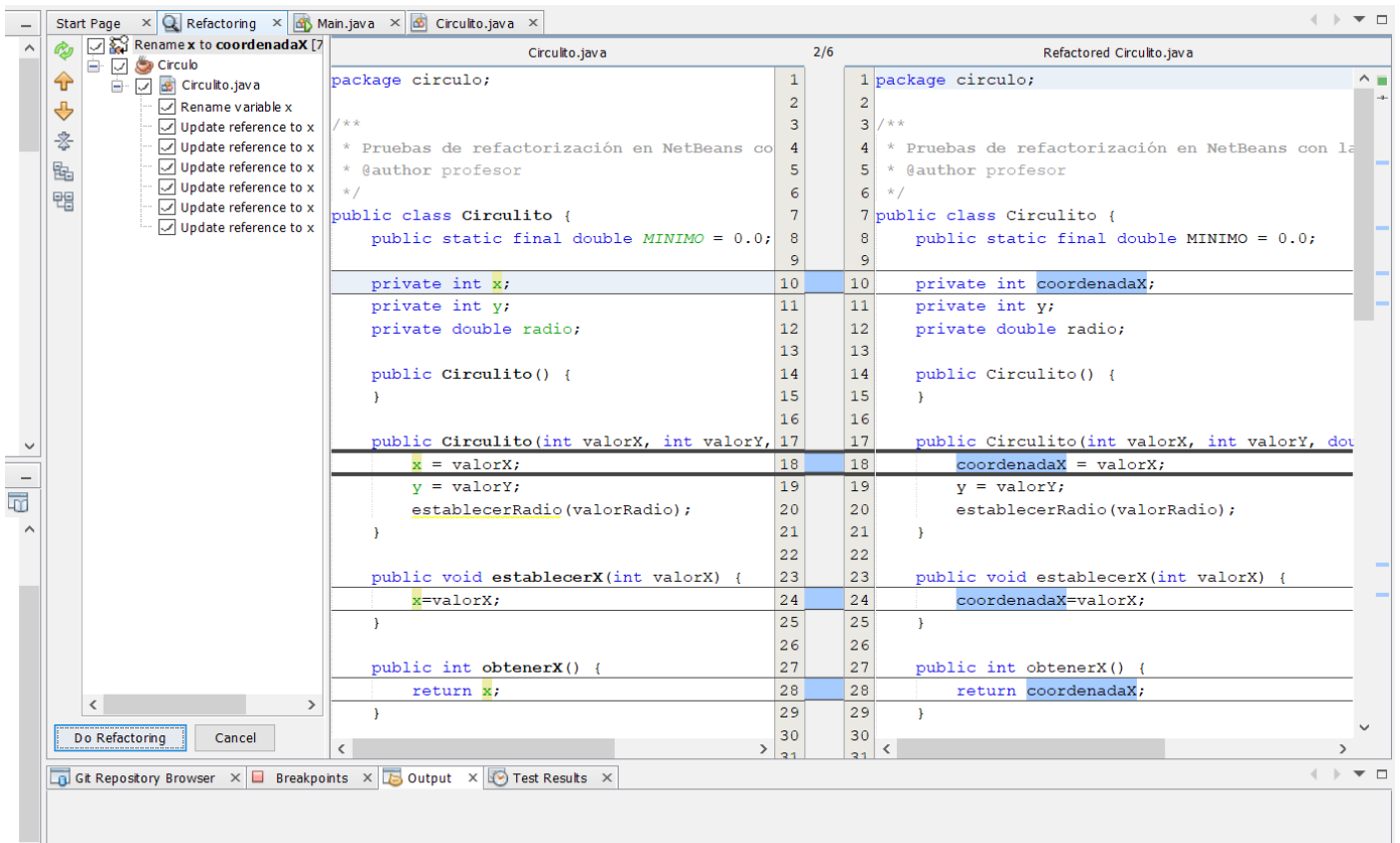
- Renomear a clase Circulo por Circulito.



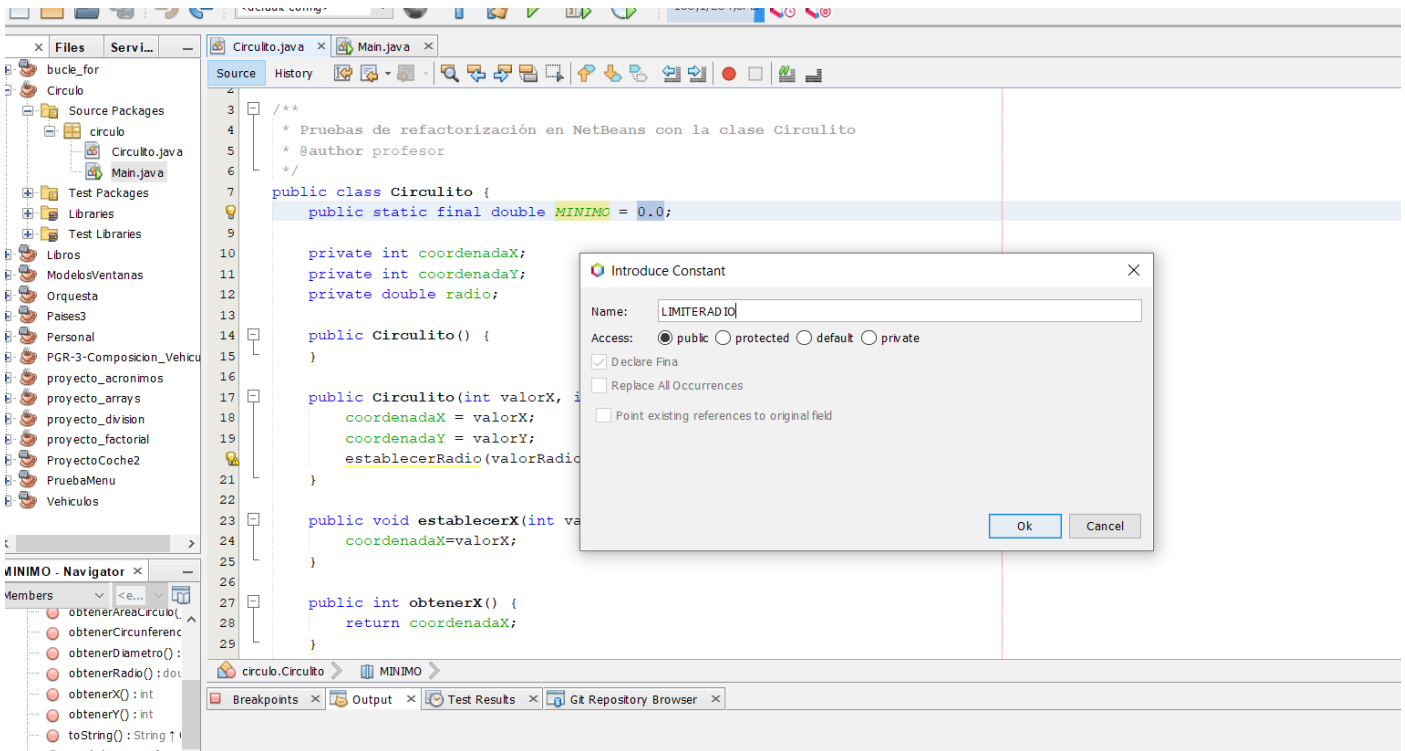
- Renomear o método ObtenerArea por ObtenerAreaCirculo.



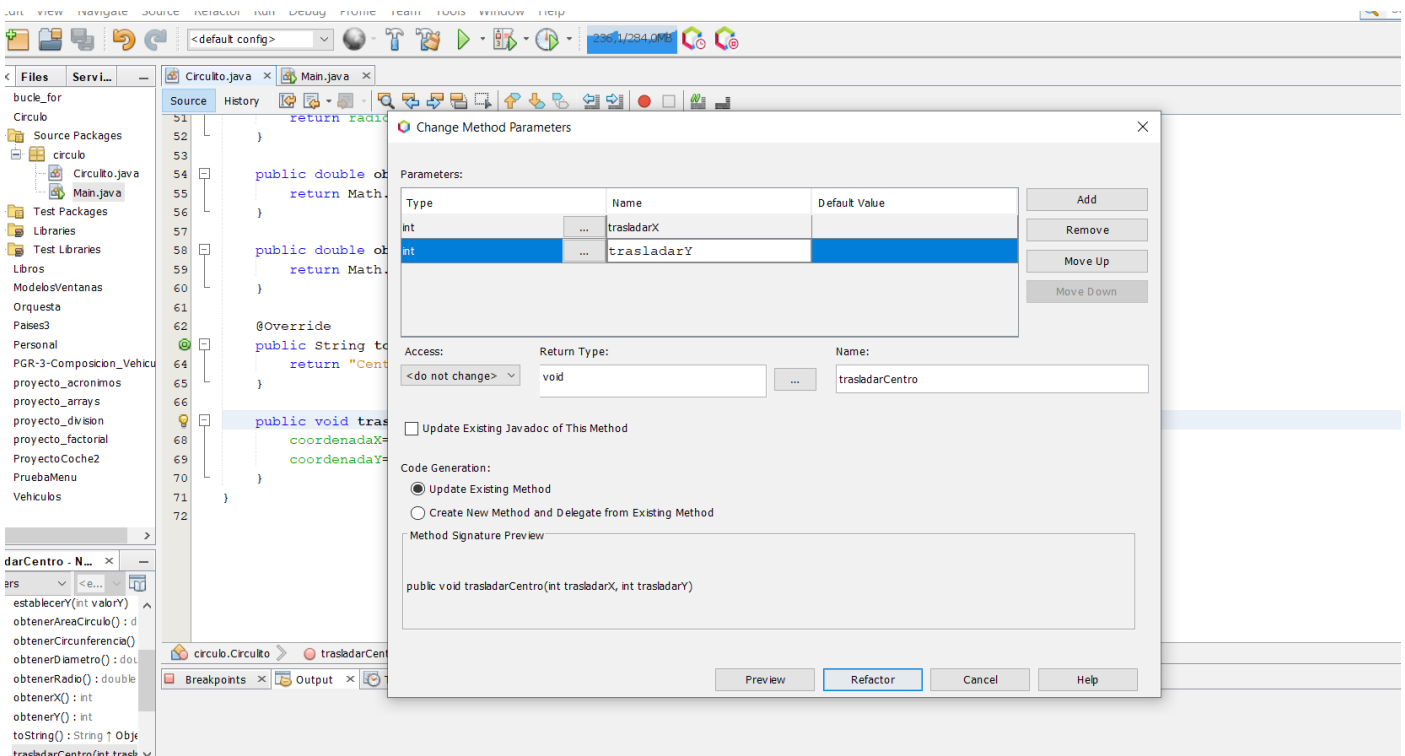
- Renomear os campo x e y por coordenadaX e coordenadaY.



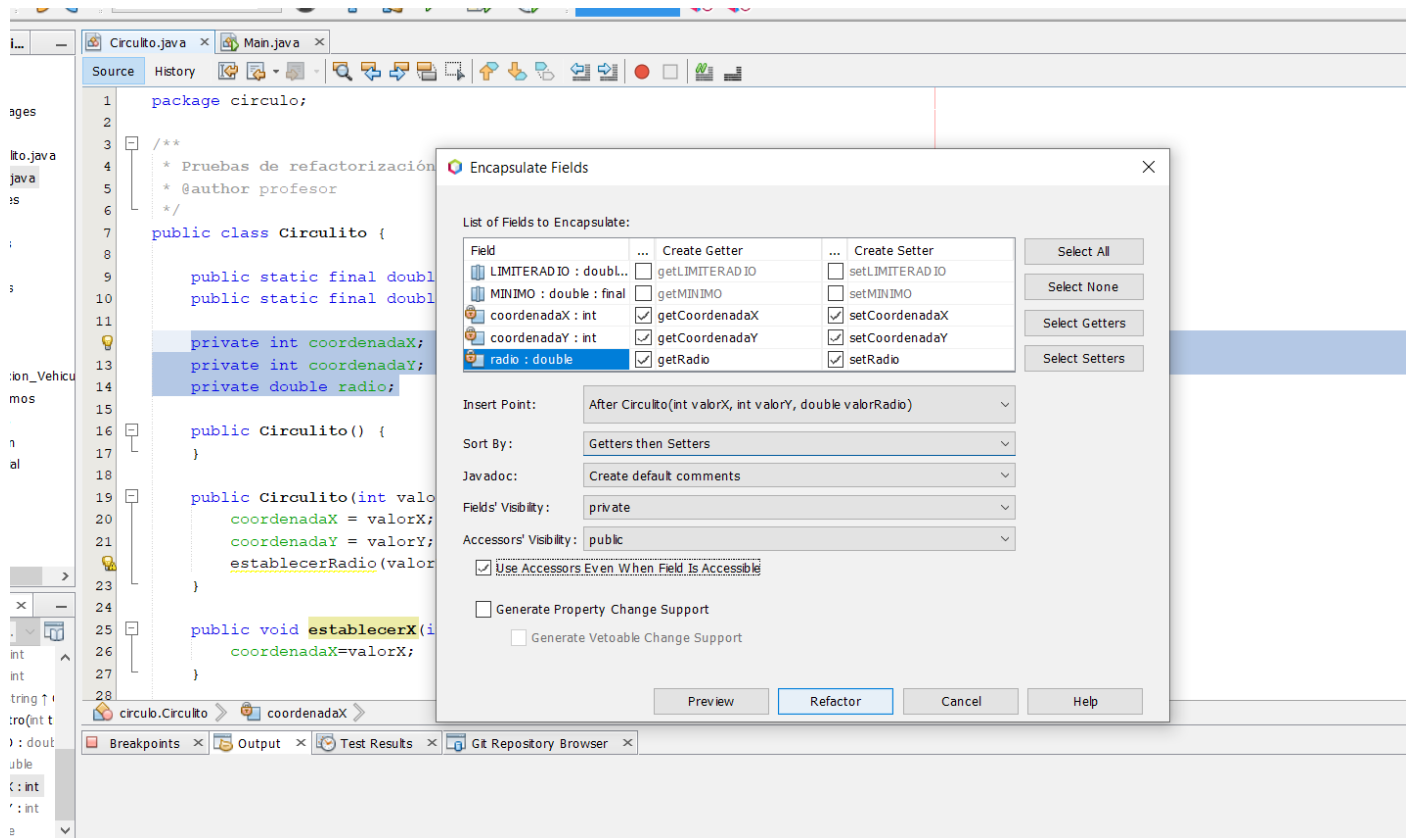
- Introducir constante LIMITERADIO de tipo double co valor 0.0



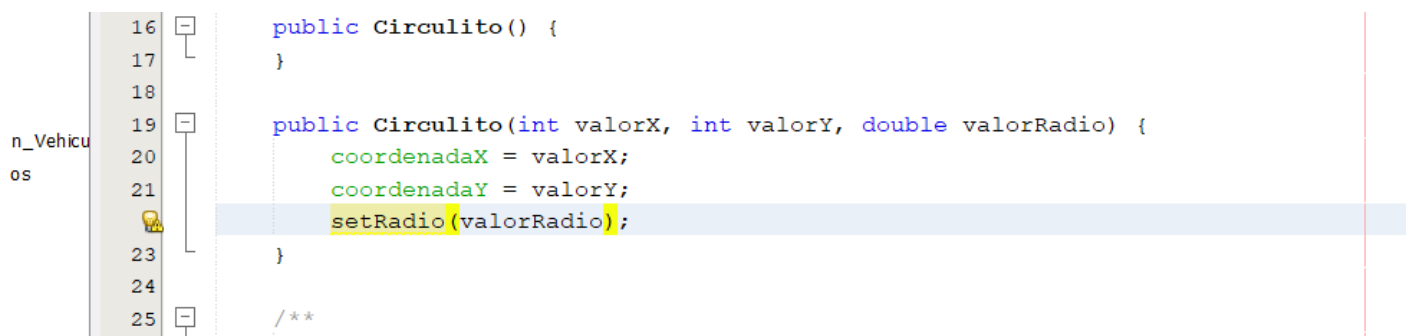
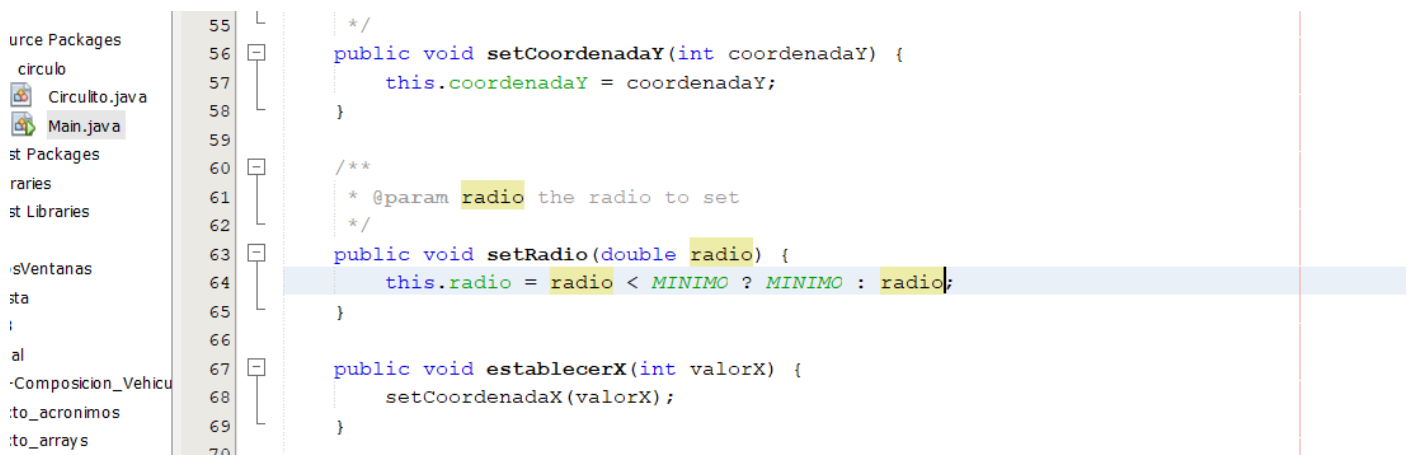
- Cambiar parámetros do método trasladarCentro para que teña dous parámetros trasladarX e trasladarY de tipo int. Facer os cambios necesarios para que o código do método permita engadir á coordenada x o valor de trasladarX e o engadir á coordenada y o valor de trasladarY.

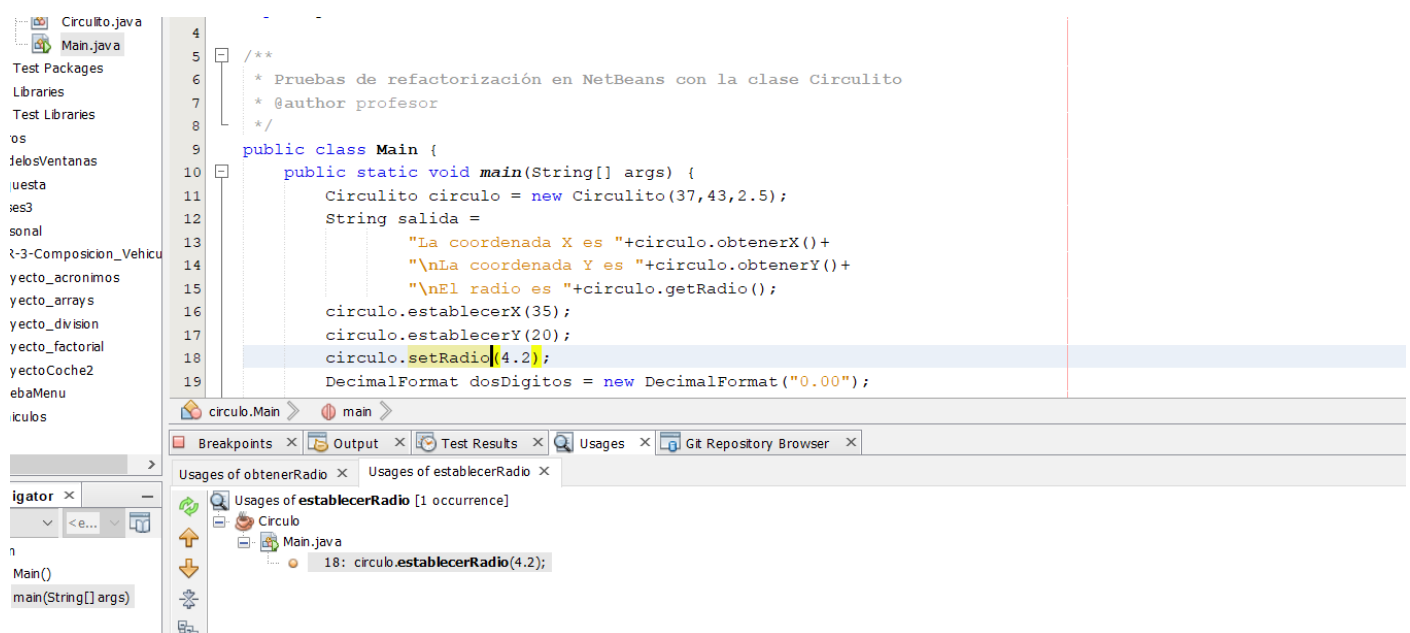
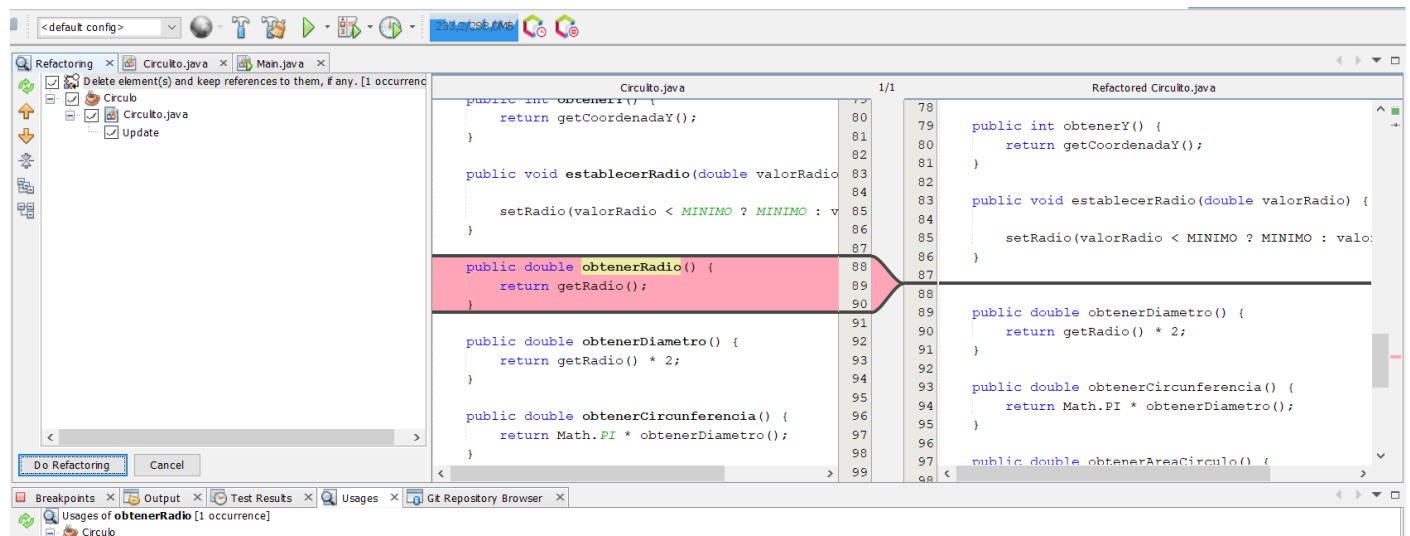
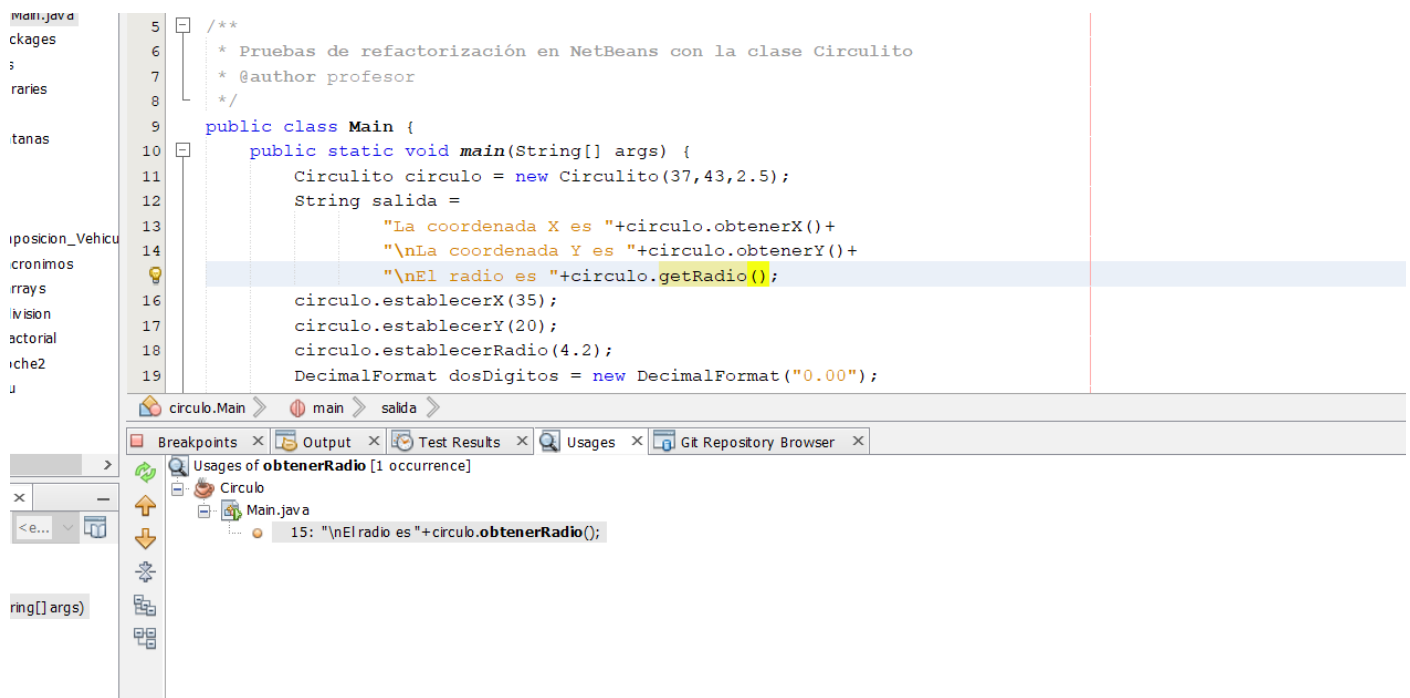


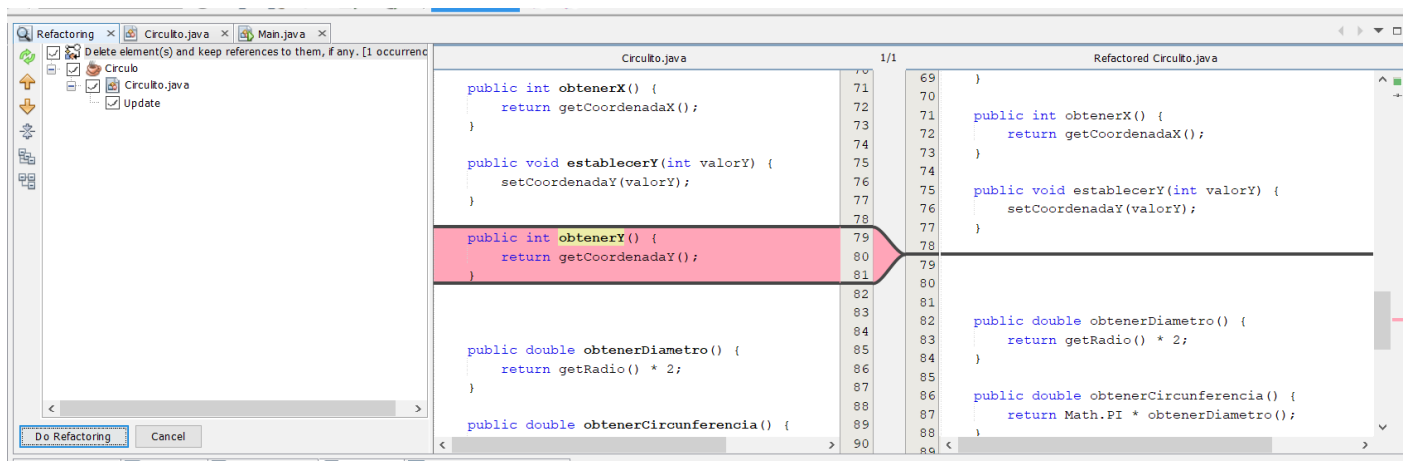
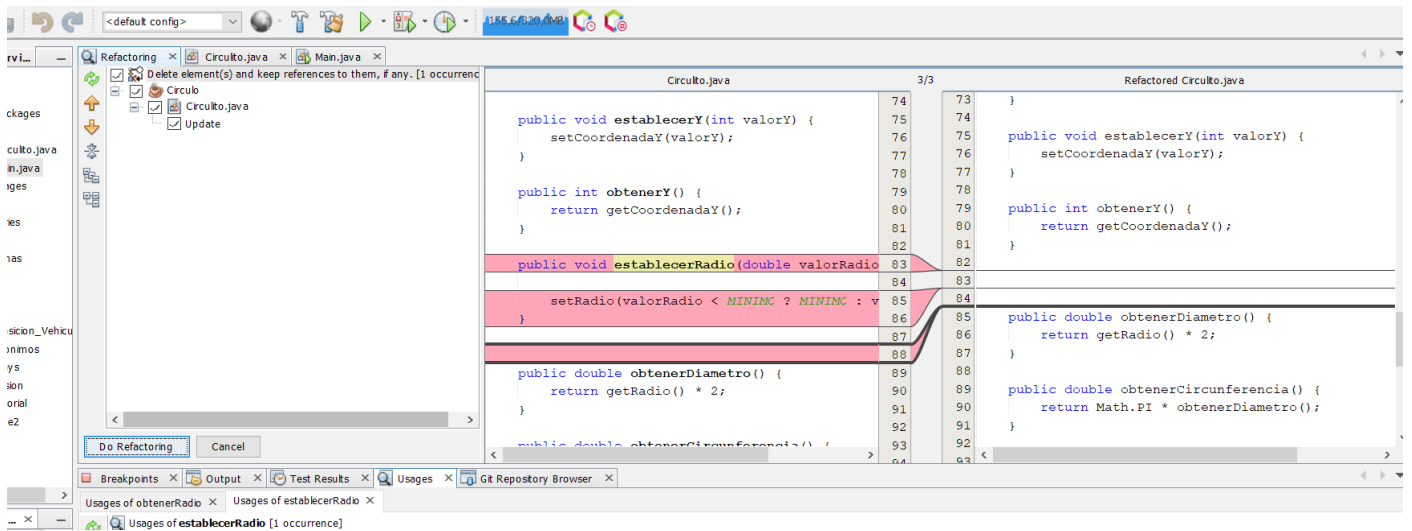
- Encapsular os tres campos do método (coordenadaX coordenadaY, radio).

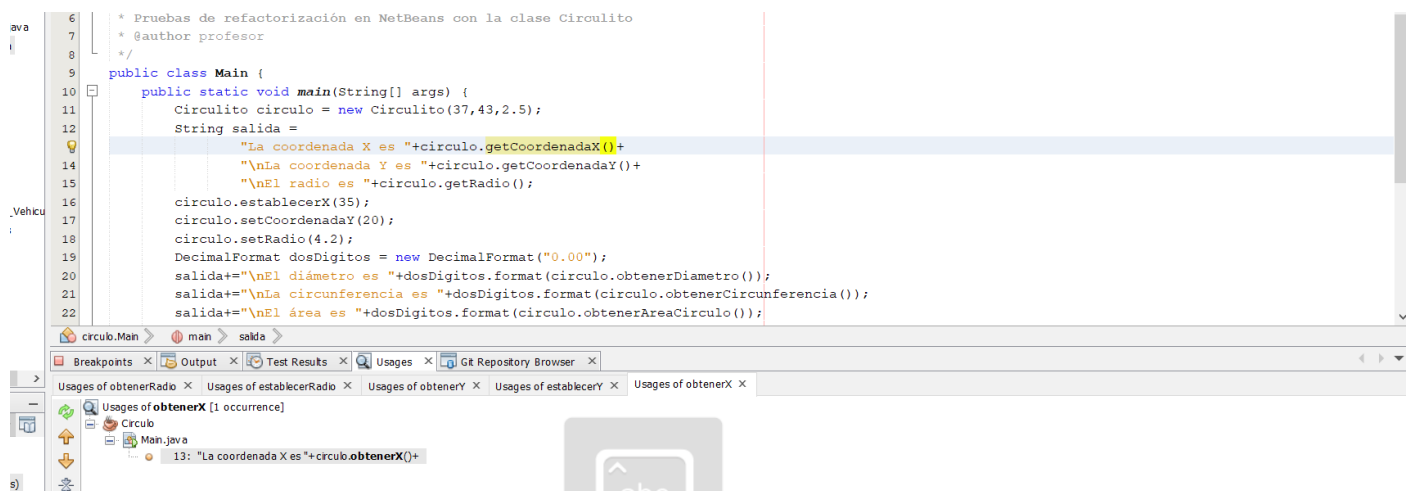
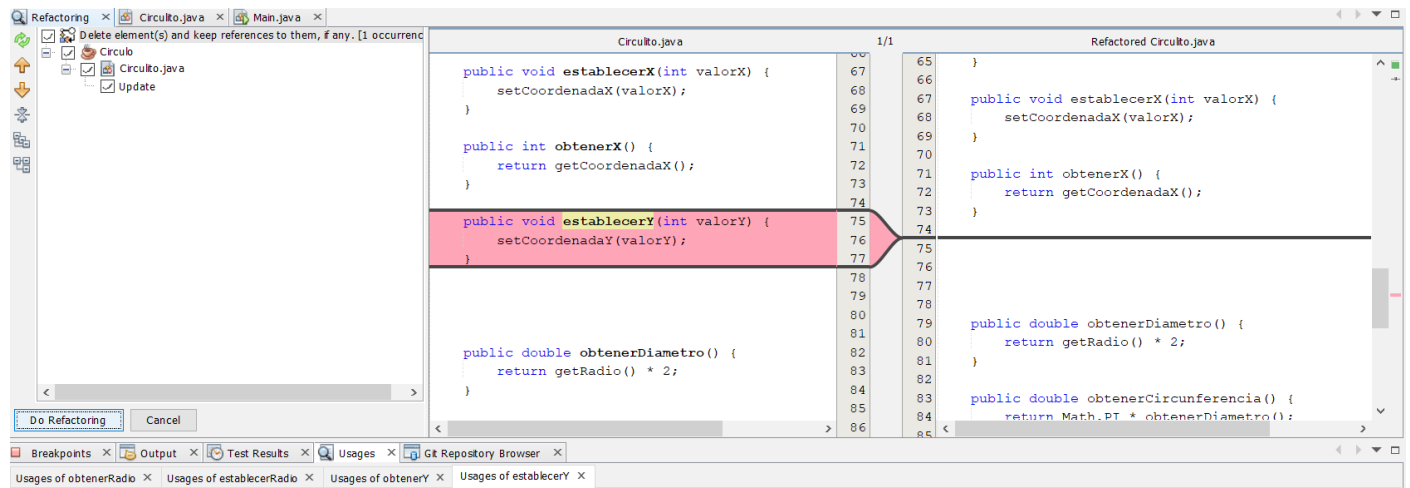
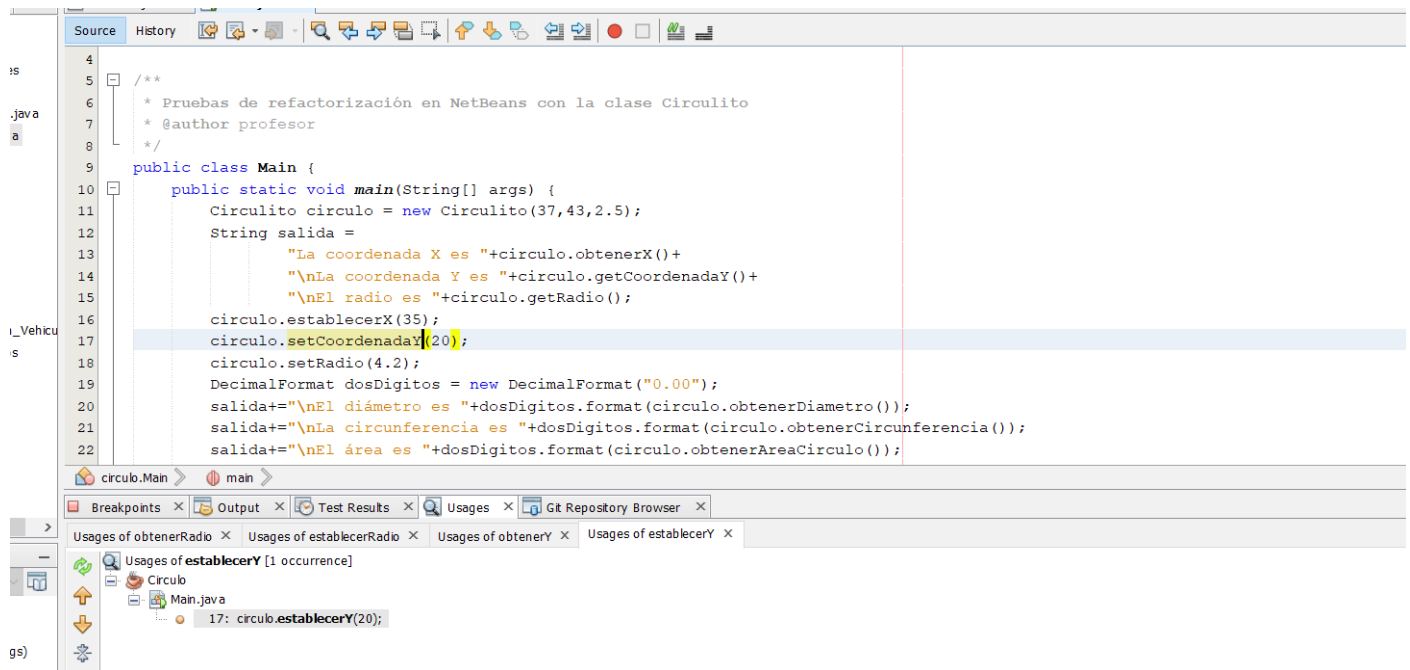


- Eliminar de forma segura os métodos obtenerX, obtenerY, obtenerRadio, establecerX, establecerY e establecerRadio que ahora son innecesarios haciendo os cambios necesarios no código para que sexan substituídos polos correspondentes métodos tipo get e set creados.

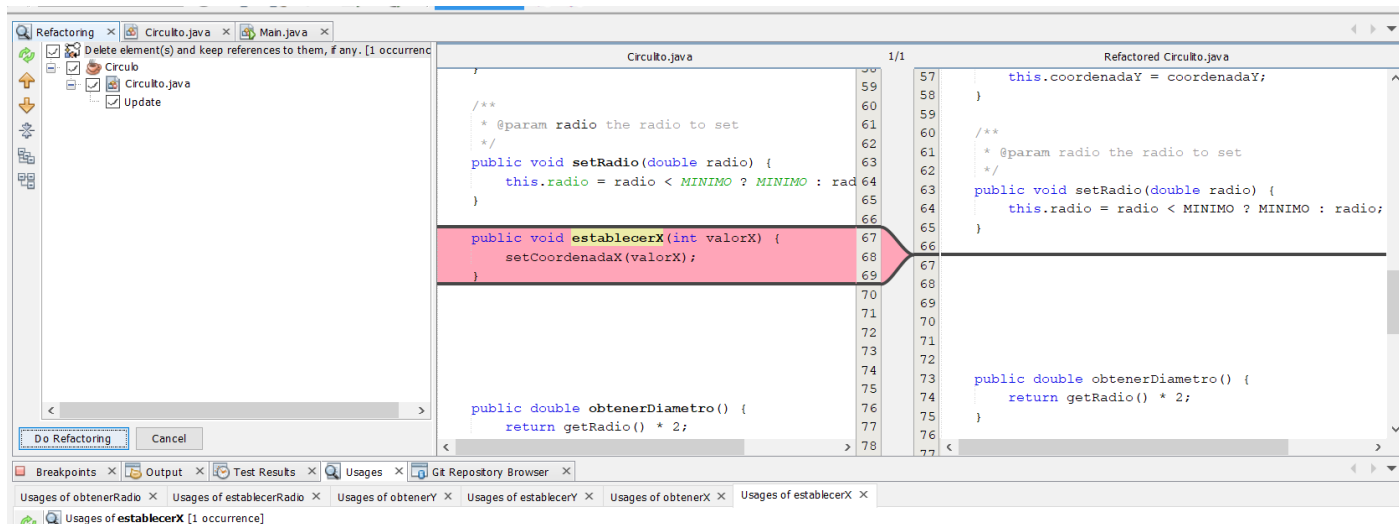
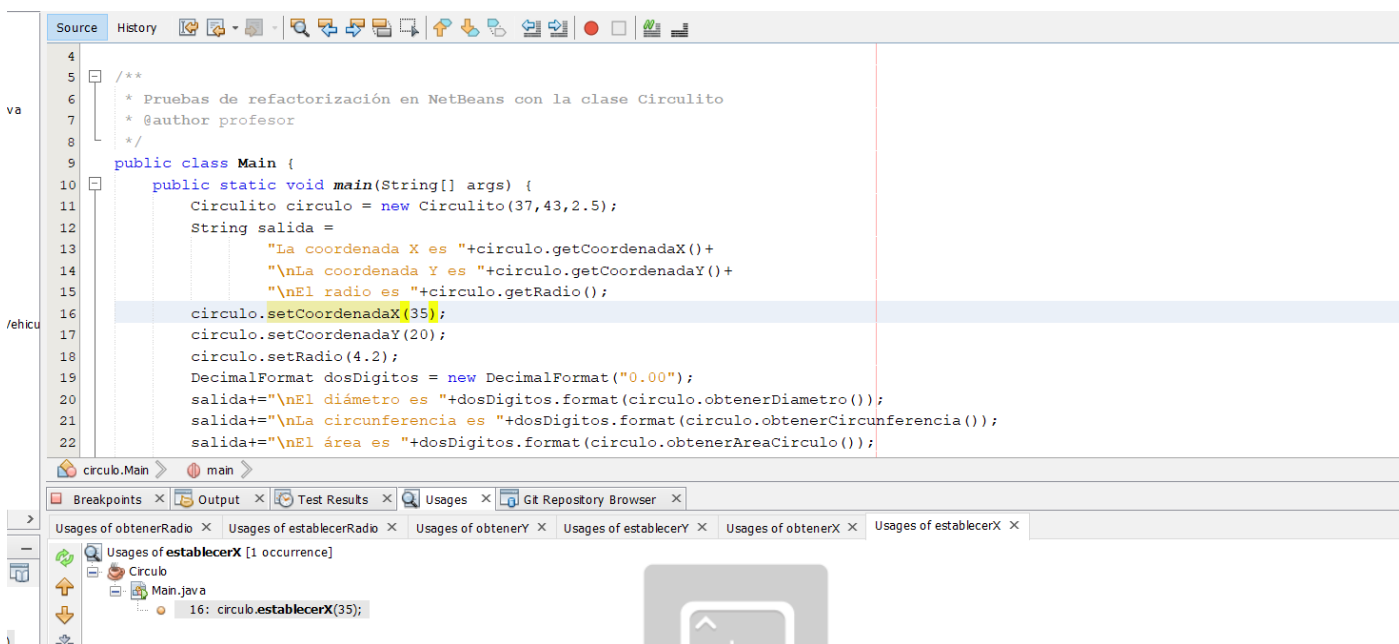
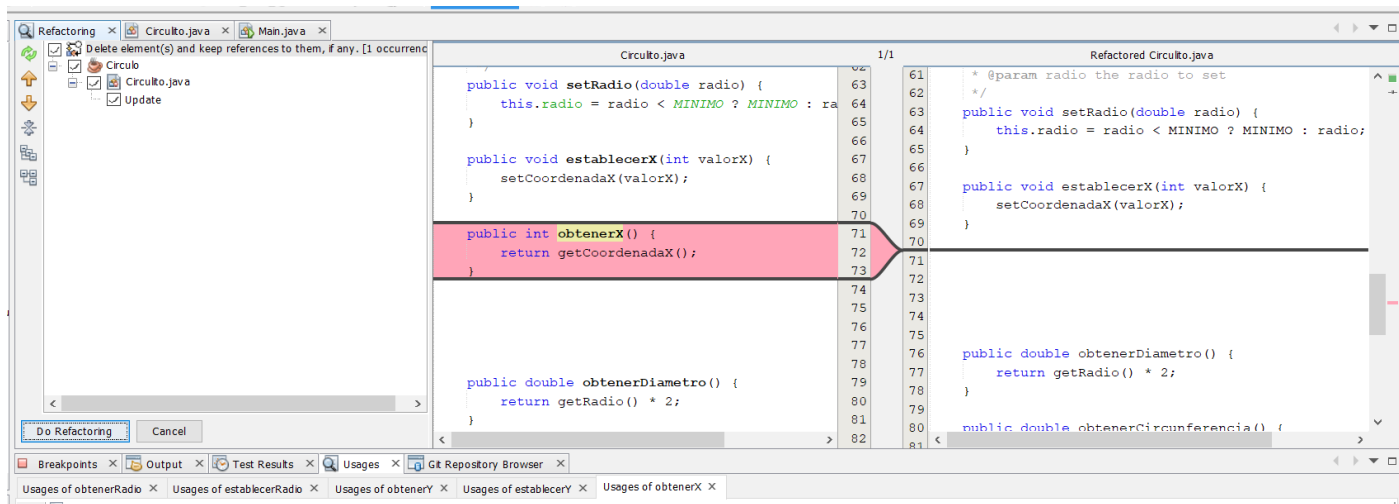










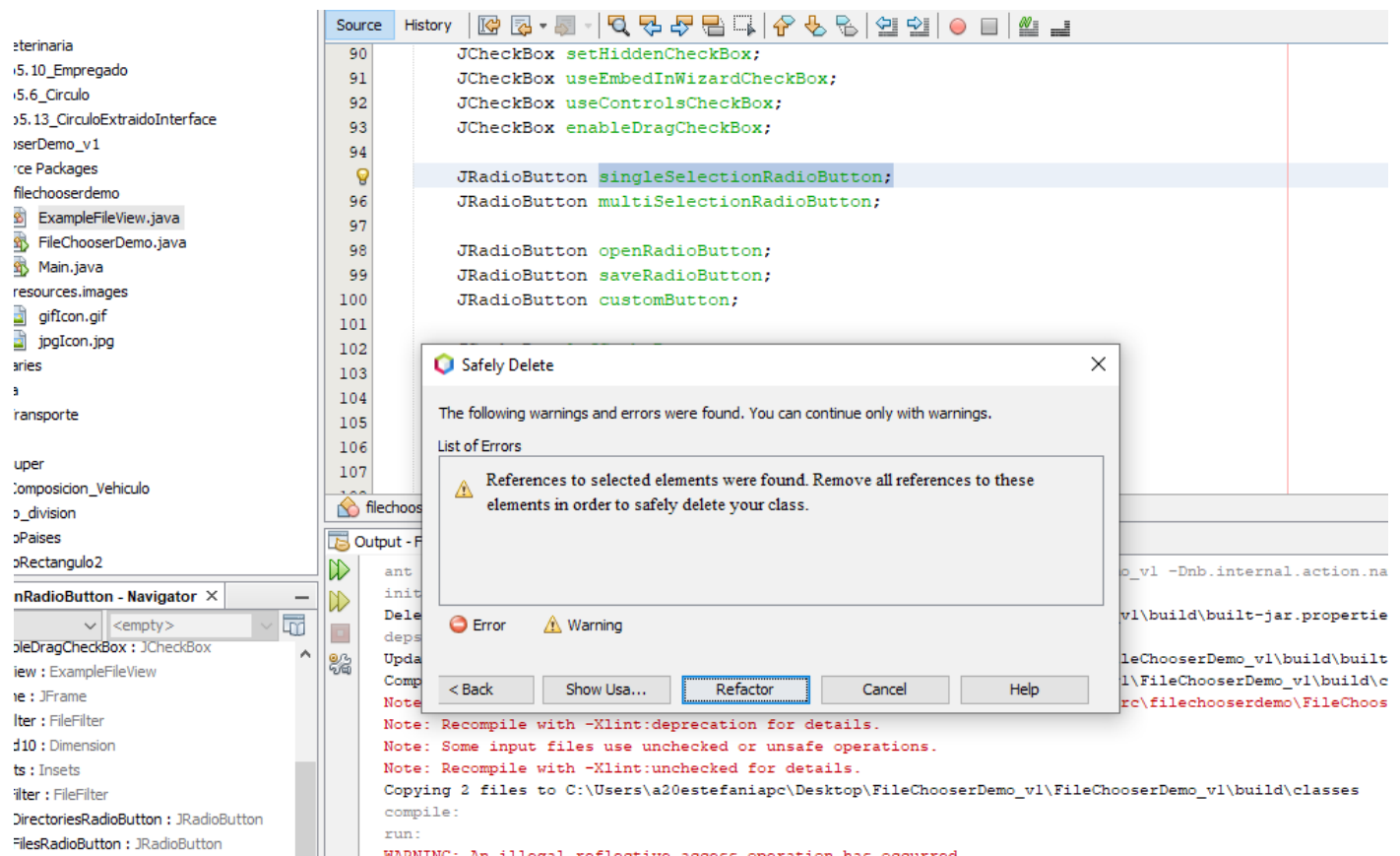




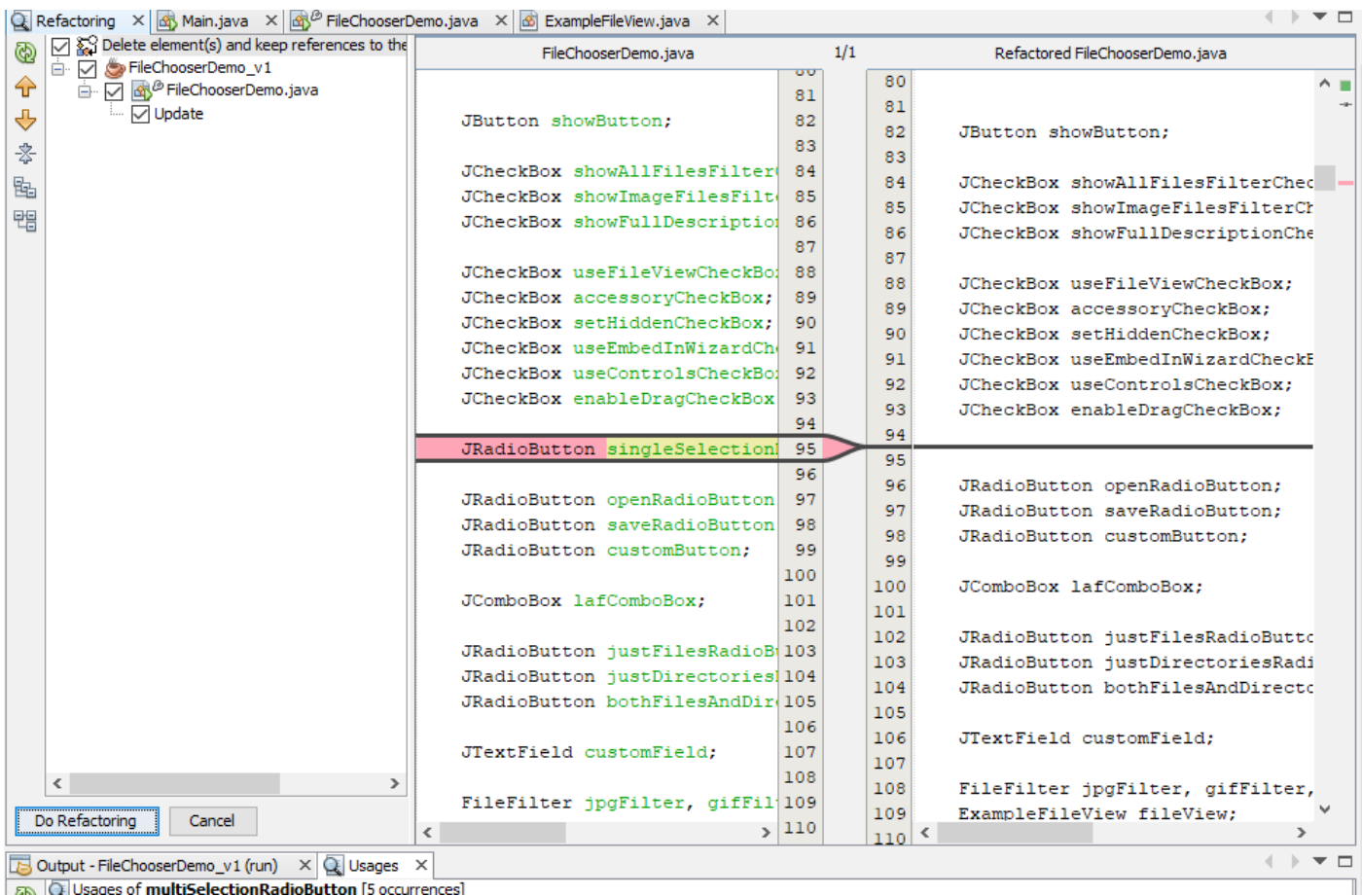
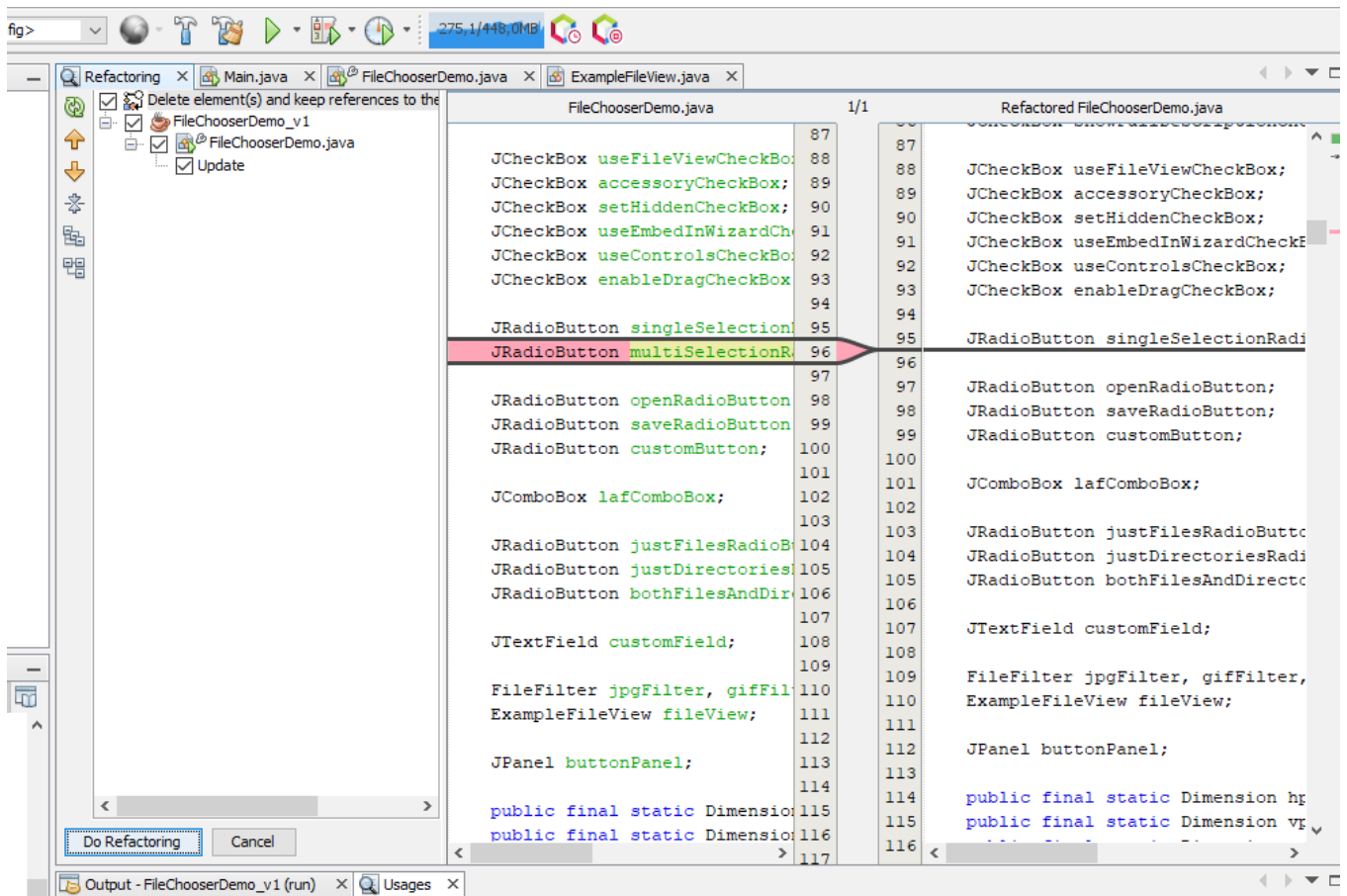
- Realizar as seguintes reestruturacións sobre o proxecto FileChooserDemo.
- No arquivo fonte FileChooserDemo.java, eliminar con Reestructurar/Eliminación segura... os compoñentes

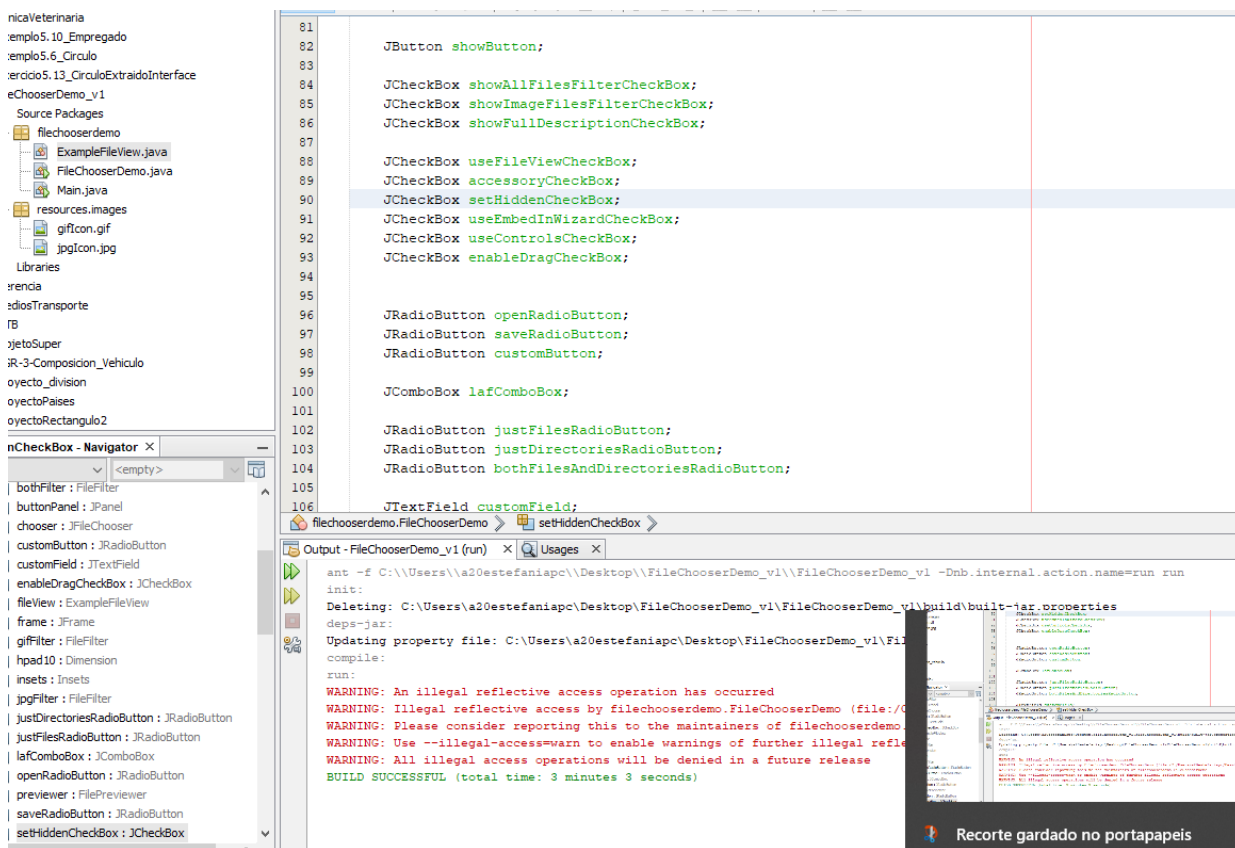
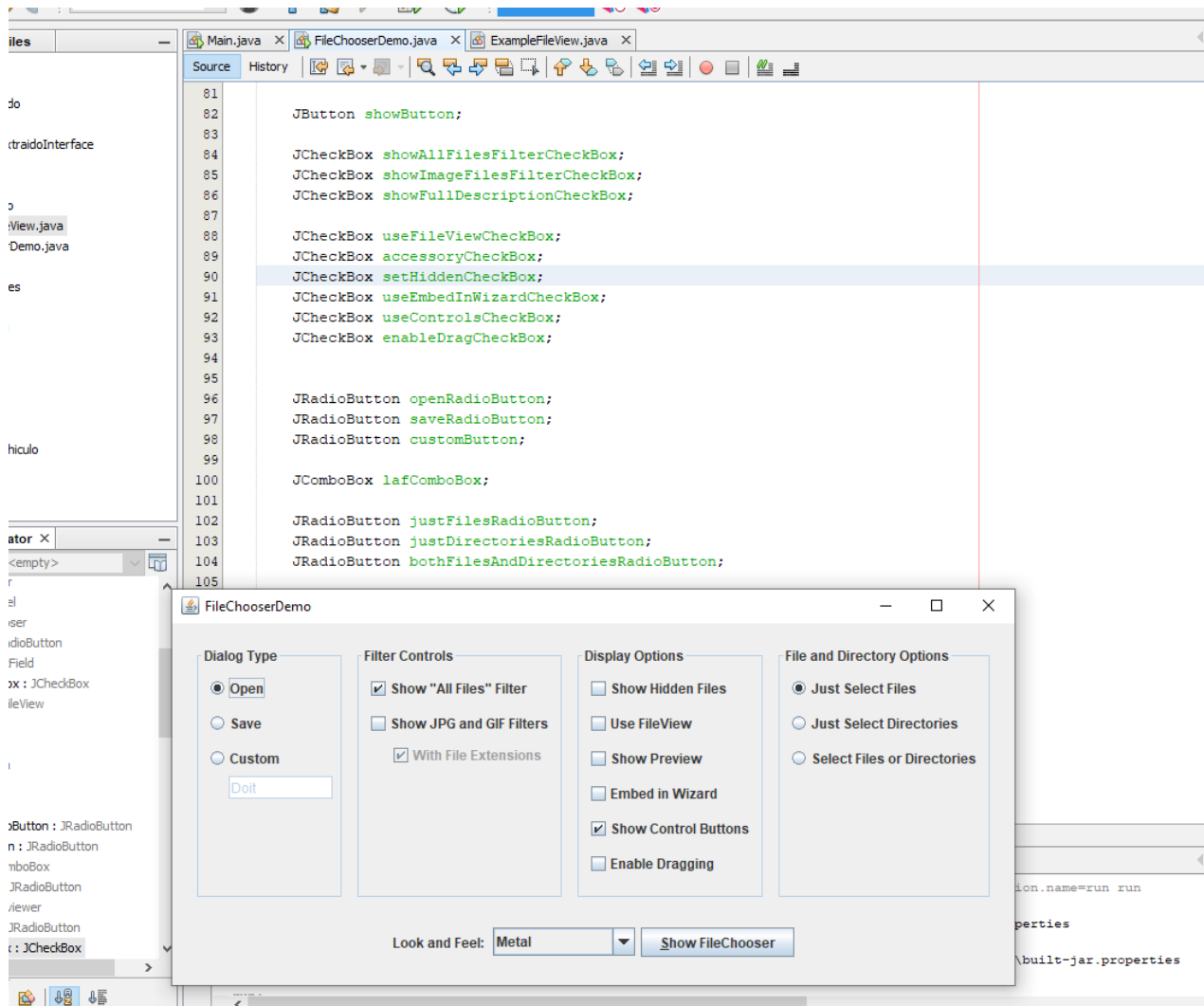
```
JRadioButton singleSelectionRadioButton;
JRadioButton multiSelectionRadioButton;
```

Antes de eliminalos, ir a "Vista previa" e "Mostrar usos" modificar o código fonte para eliminar todos os usos destes compoñentes no programa, e cando xa non que de ningún uso reestructurar con "Reestructurar/Eliminación segura". Comproba que o proxecto segue funcionando.

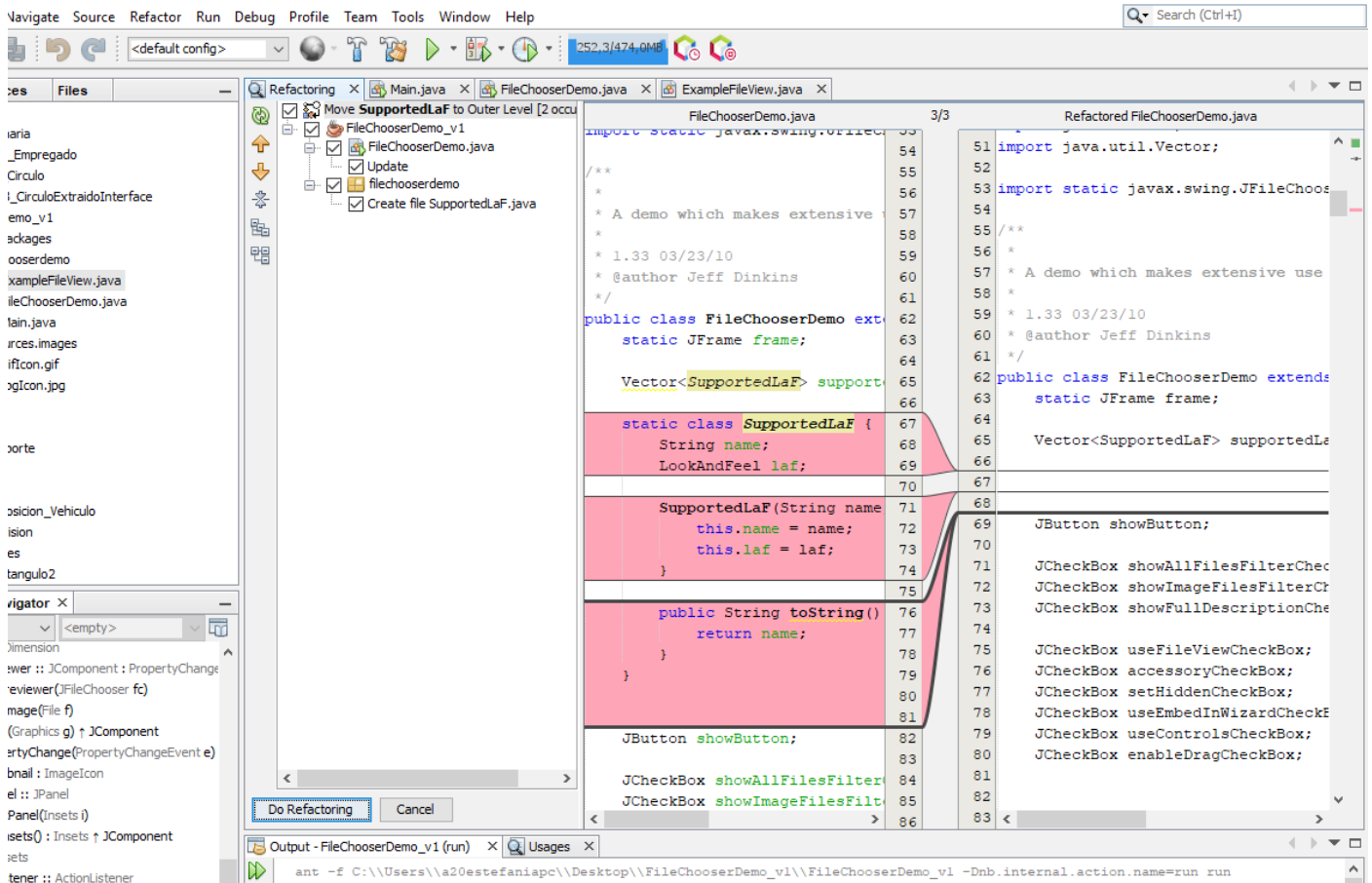
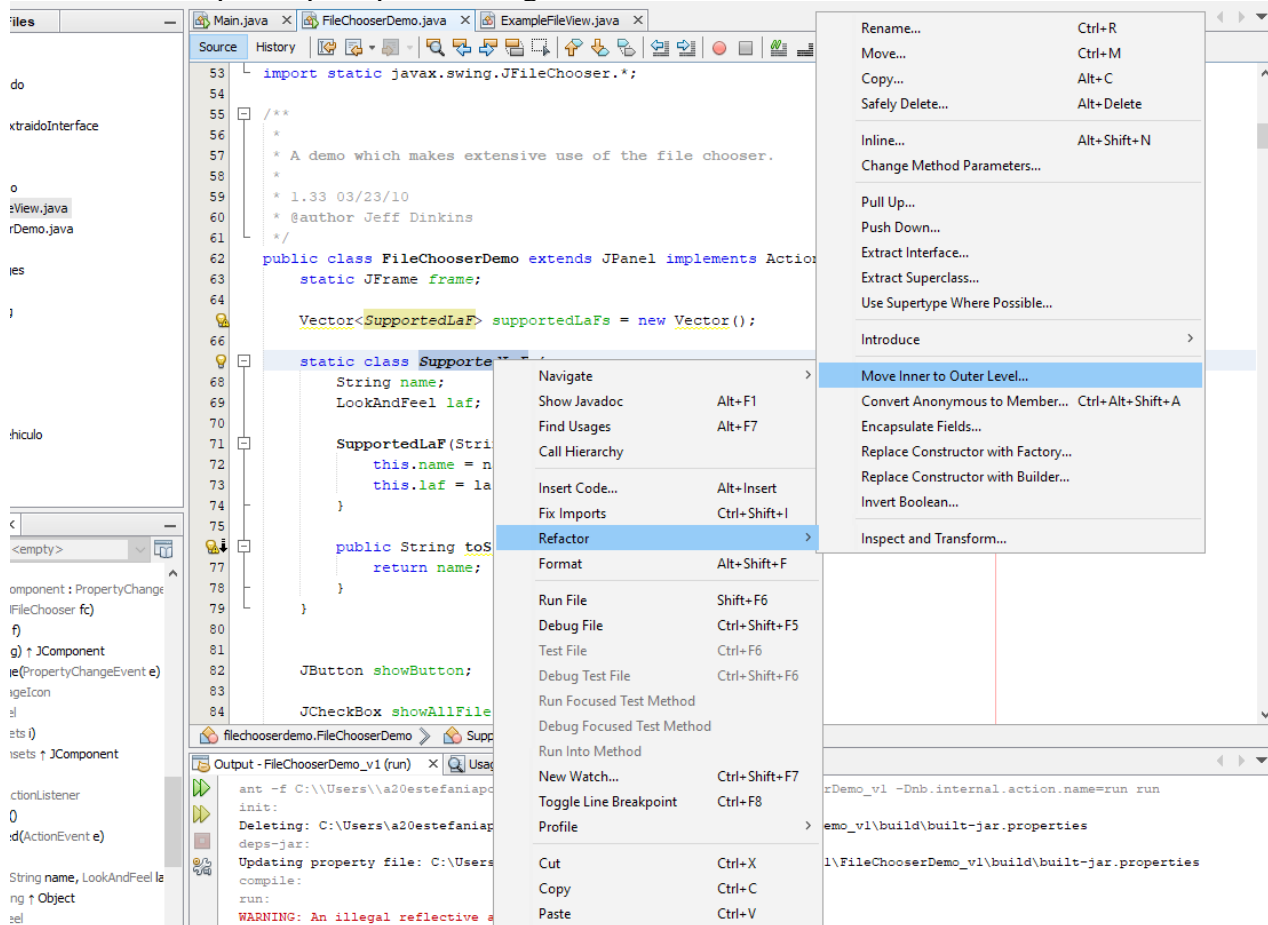








- No arquivo fonte FileChooserDemo.java, mover a classe static class SupportedLaF de nível interior a exterior. Comproba que o proxecto segue funcionando correctamente.



302.6/474.0MB

Files

Main.java x FileChooserDemo.java x ExampleFileView.java x SupportedLaF.java x

Source History

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package filechooserdemo;
7
8   /**
9    *
10   * @author luisalib
11   */
12   public class Main {
13
14       /**
15        * @param args the command line arguments
```

Output - FileChooserDemo\_v1 (run) x Usages x

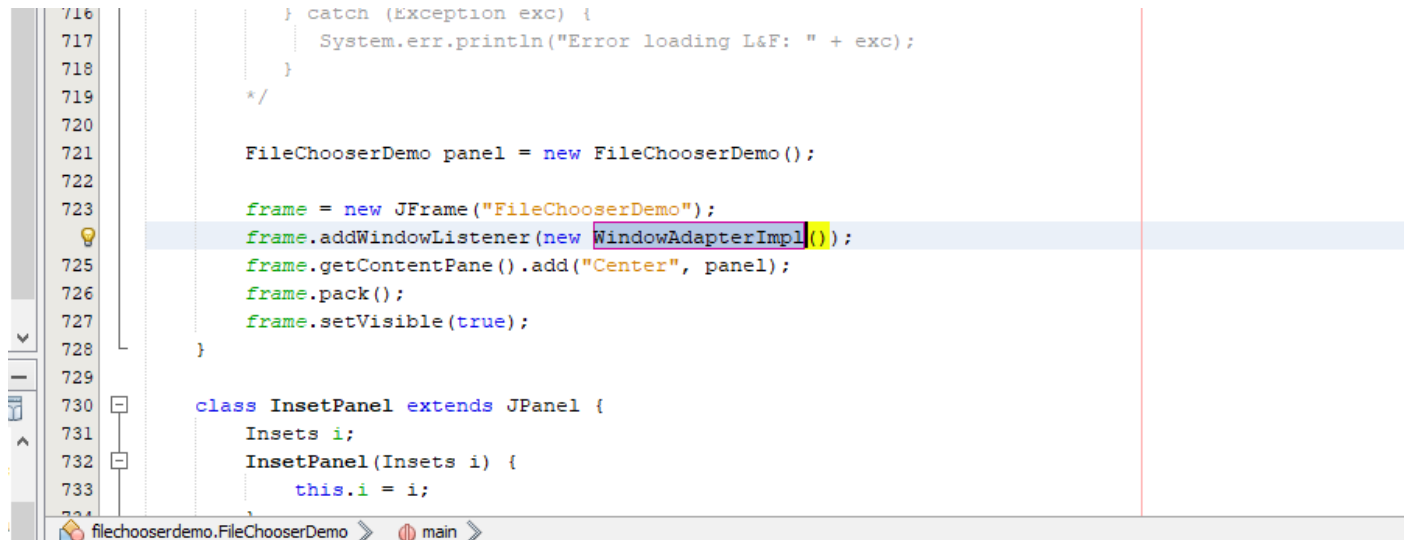
```
ant -f C:\\Users\\a20estefaniapc\\Desktop\\FileChooserDemo_v1\\FileChooserDemo_v1 -Dnb.internal.action.name=run run
init:
Deleting: C:\\Users\\a20estefaniapc\\Desktop\\FileChooserDemo_v1\\FileChooserDemo_v1\\build\\build-jar.properties
deps-jar:
Updating property file: C:\\Users\\a20estefaniapc\\Desktop\\FileChooserDemo_v1\\FileChooserDemo_v1\\build\\build-jar.properties
Compiling 2 source files to C:\\Users\\a20estefaniapc\\Desktop\\FileChooserDemo_v1\\FileChooserDemo_v1\\build\\classes
Note: C:\\Users\\a20estefaniapc\\Desktop\\FileChooserDemo_v1\\FileChooserDemo_v1\\src\\filechooserdemo\\FileChooserDemo.java uses or
Note: Recompile with -Xlint:deprecation for details.
Note: C:\\Users\\a20estefaniapc\\Desktop\\FileChooserDemo_v1\\FileChooserDemo_v1\\src\\filechooserdemo\\FileChooserDemo.java uses un
Note: Recompile with -Xlint:unchecked for details.
compile:
run:
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by filechooserdemo.FileChooserDemo (file:/C:/Users/a20estefaniapc/Desktop/FileChooserDemo
WARNING: Please consider reporting this to the maintainers of filechooserdemo.FileChooserDemo
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
BUILD SUCCESSFUL (total time: 3 seconds)
```



- No archivo fuente FileChooserDemo.java, busca o fragmento de código

```
frame.addWindowListener(new WindowAdapter() {  
public void windowClosing(WindowEvent e) {System.exit(0);}  
});
```

Modifica a clase anónima para convertela en miembro ca opción de menú Reestructurar/Convertir anónimo en miembro.



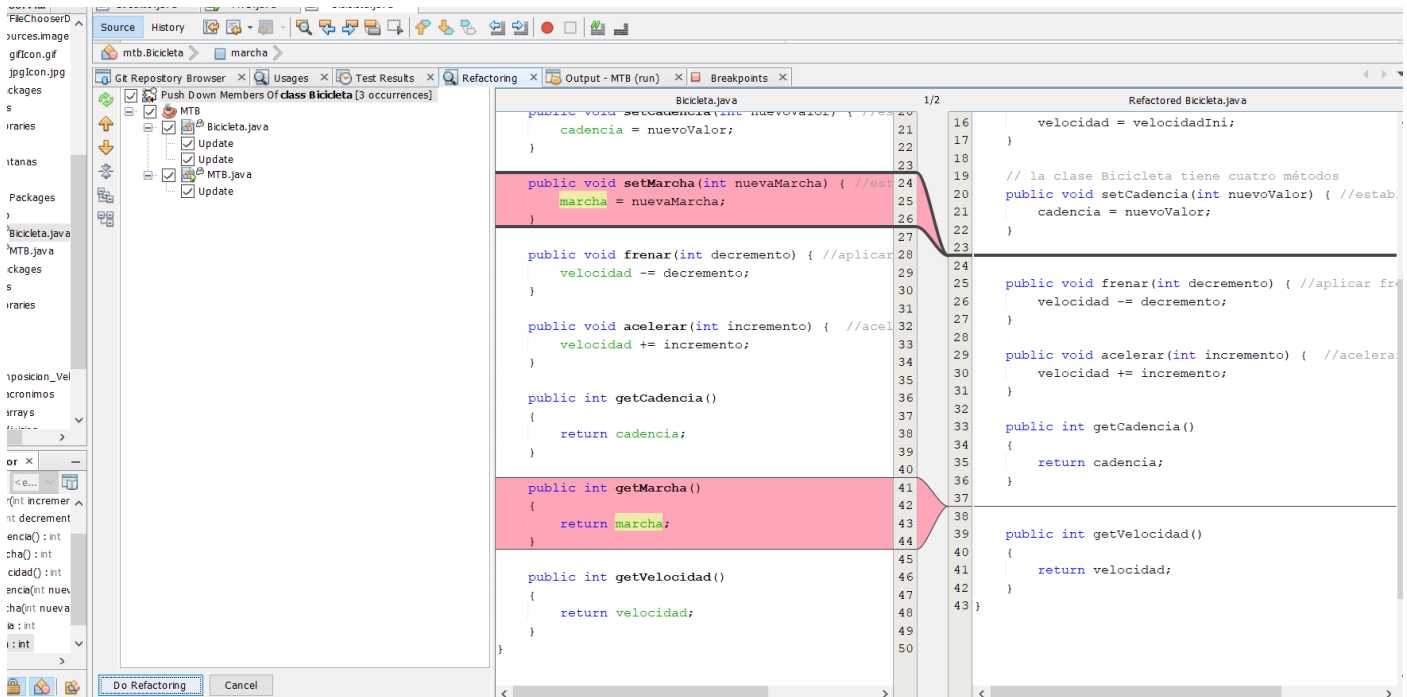
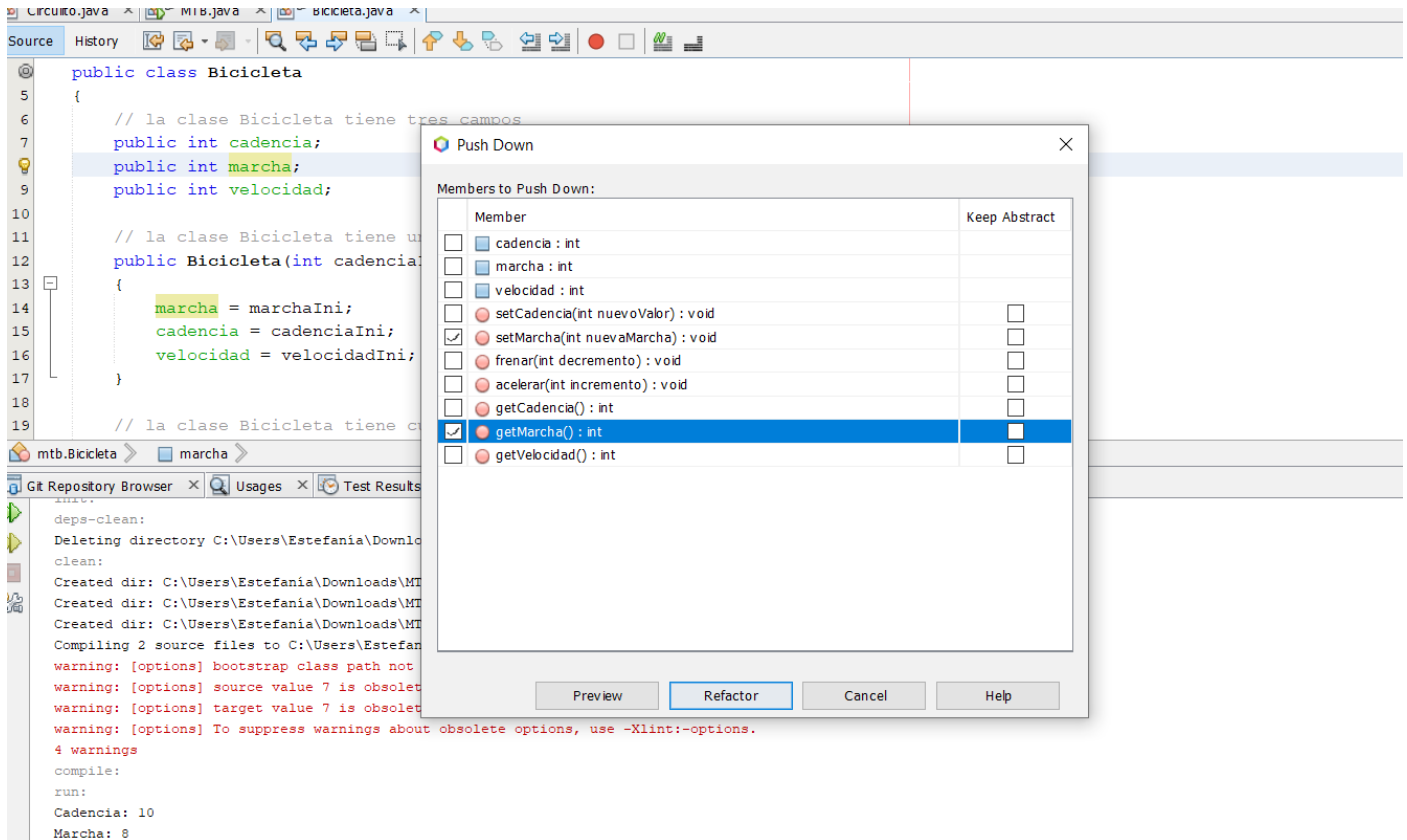
```
716         } catch (Exception exc) {  
717             System.err.println("Error loading L&F: " + exc);  
718         }  
719     }  
720  
721     FileChooserDemo panel = new FileChooserDemo();  
722  
723     JFrame frame = new JFrame("FileChooserDemo");  
724     frame.addWindowListener(new WindowAdapterImpl());  
725     frame.getContentPane().add("Center", panel);  
726     frame.pack();  
727     frame.setVisible(true);  
728 }  
729  
730 class InsetPanel extends JPanel {  
731     Insets i;  
732     InsetPanel(Insets i) {  
733         this.i = i;  
734     }  
735 }
```

filechooserdemo.FileChooserDemo > main >



Realizar as seguintes reestruturacións sobre o proxecto MTB.

- Move o atributo marcha, o método getMarcha e o método setMarcha dende a superclase Bicicleta a subclase MTB ca opción de menú Reestructurar/Descender probablemente quede algún problema que solucionar e non se faga todo automaticamente soluciona e explica dito problema.





```

3
4 public class MTB extends Bicicleta
5 {
6
7     // la subclase MTB agrega un campo
8     public int altoAsiento;
9     public int marcha;
10
11     // la subclase MTB tiene un constructor
12     public MTB(int cadenciaIni, int velocidadIni, int marchaIni, int altoIni)
13     {
14         super(cadenciaIni, velocidadIni, marchaIni);
15         altoAsiento = altoIni;
16         marcha = marcha;
17     }
18
19     // la subclase MTB agrega un método
20     public void setAltoAsiento(int nuevoValor)
21     {
22         altoAsiento = nuevoValor;
23     }
24

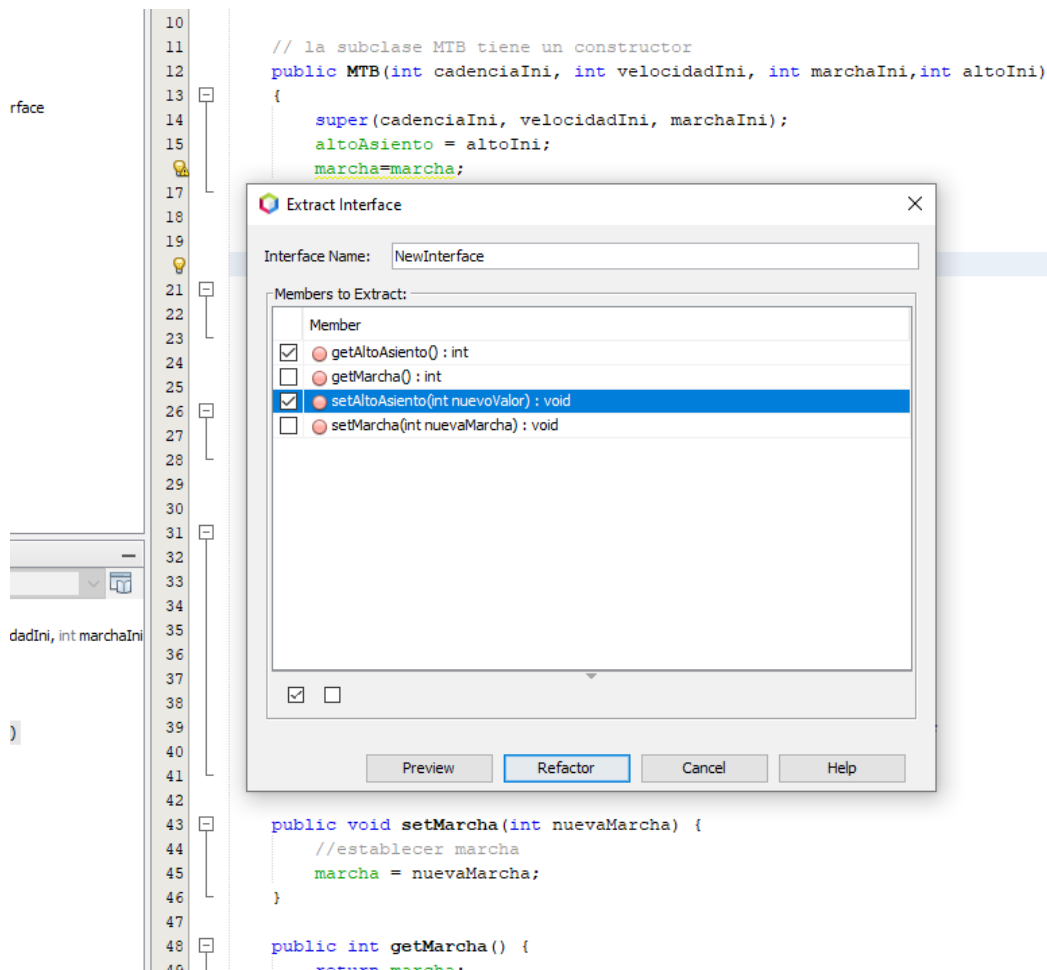
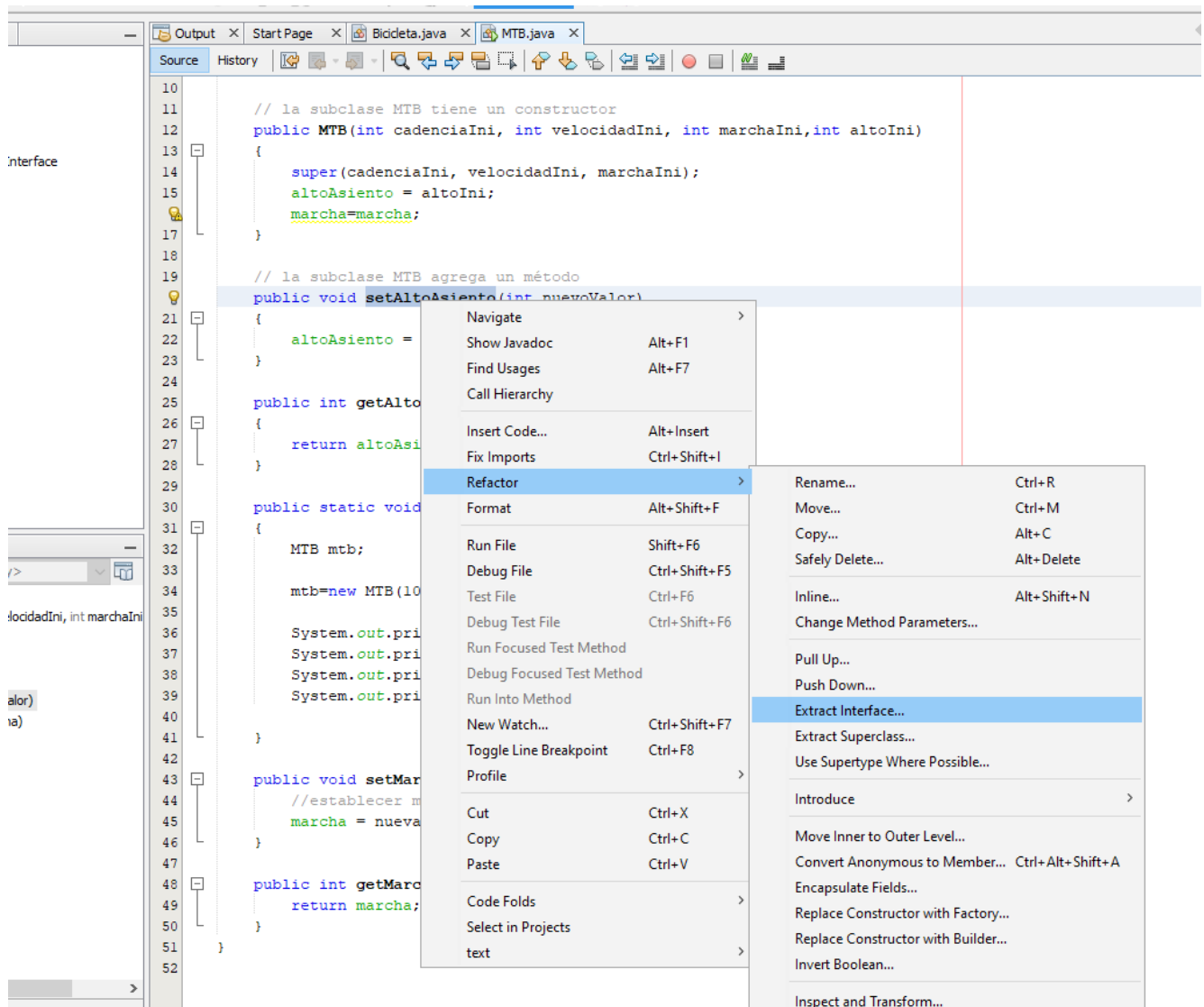
```

```

6 // la clase Bicicleta tiene tres campos
7 public int cadencia;
8 public int velocidad;
9
10 // la clase Bicicleta tiene un constructor
11 public Bicicleta(int cadenciaIni, int velocidadIni, int marchaIni)
12 {
13     cadencia = cadenciaIni;
14     velocidad = velocidadIni;
15 }
16
17 // la clase Bicicleta tiene cuatro métodos
18 public void setCadencia(int nuevoValor) { //establecer cadencia
19     cadencia = nuevoValor;
20 }
21

```

- Extrae un interface para os métodos getAltoAsiento e setAltoAsiento, usando a opción de menú Reestructurar/Extraer Interface.



Output x Start Page x Refactoring x BiciBeta.java x MTB.java x

Extract Interface class MTB [4 occurrences]

- ☒ MTB
- ☒ MTB.java
- ☒ Update
- ☒ Update
- ☒ Update
- ☒ mtb
- ☒ Create file NewInterface.java

11 marchaIni

MTB.java 3/3

```
package mtb;

public class MTB extends BiciBeta
{
    // la subclase MTB agrega un car
    public int altoAsiento;
    public int marcha;

    // la subclase MTB tiene un cons
    public MTB(int cadenciaIni, int
    {
        super(cadenciaIni, velocidad
        altoAsiento = altoIni;
        marcha=marcha;
    }

    // la subclase MTB agrega un mé
    public void setAltoAsiento(int r
    {
        altoAsiento = nuevoValor;
    }

    public int getAltoAsiento()
    {
        return altoAsiento;
    }

    public static void main (String
    {
        MTB mtb;

        mtb=new MTB(10, 30, 8, 40);

        System.out.println("Cadencia:
        System.out.println("Marcha:
        System.out.println("Velocida
        System.out.println("Alto as

    }

    public void setMarcha(int nueva
    //establecer marcha
    marcha = nuevaMarcha;
}
```

Refactored MTB.java

```
package mtb;

public class MTB extends BiciBeta impl
{
    // la subclase MTB agrega un campo
    public int altoAsiento;
    public int marcha;

    // la subclase MTB tiene un constru
    public MTB(int cadenciaIni, int vel
    {
        super(cadenciaIni, velocidadIni
        altoAsiento = altoIni;
        marcha=marcha;
    }

    // la subclase MTB agrega un método
    @Override
    public void setAltoAsiento(int nuev
    {
        altoAsiento = nuevoValor;
    }

    @Override
    public int getAltoAsiento()
    {
        return altoAsiento;
    }

    public static void main (String []
    {
        MTB mtb;

        mtb=new MTB(10, 30, 8, 40);

        System.out.println("Cadencia: "
        System.out.println("Marcha: "+m
        System.out.println("Velocidad:
        System.out.println("Alto asient

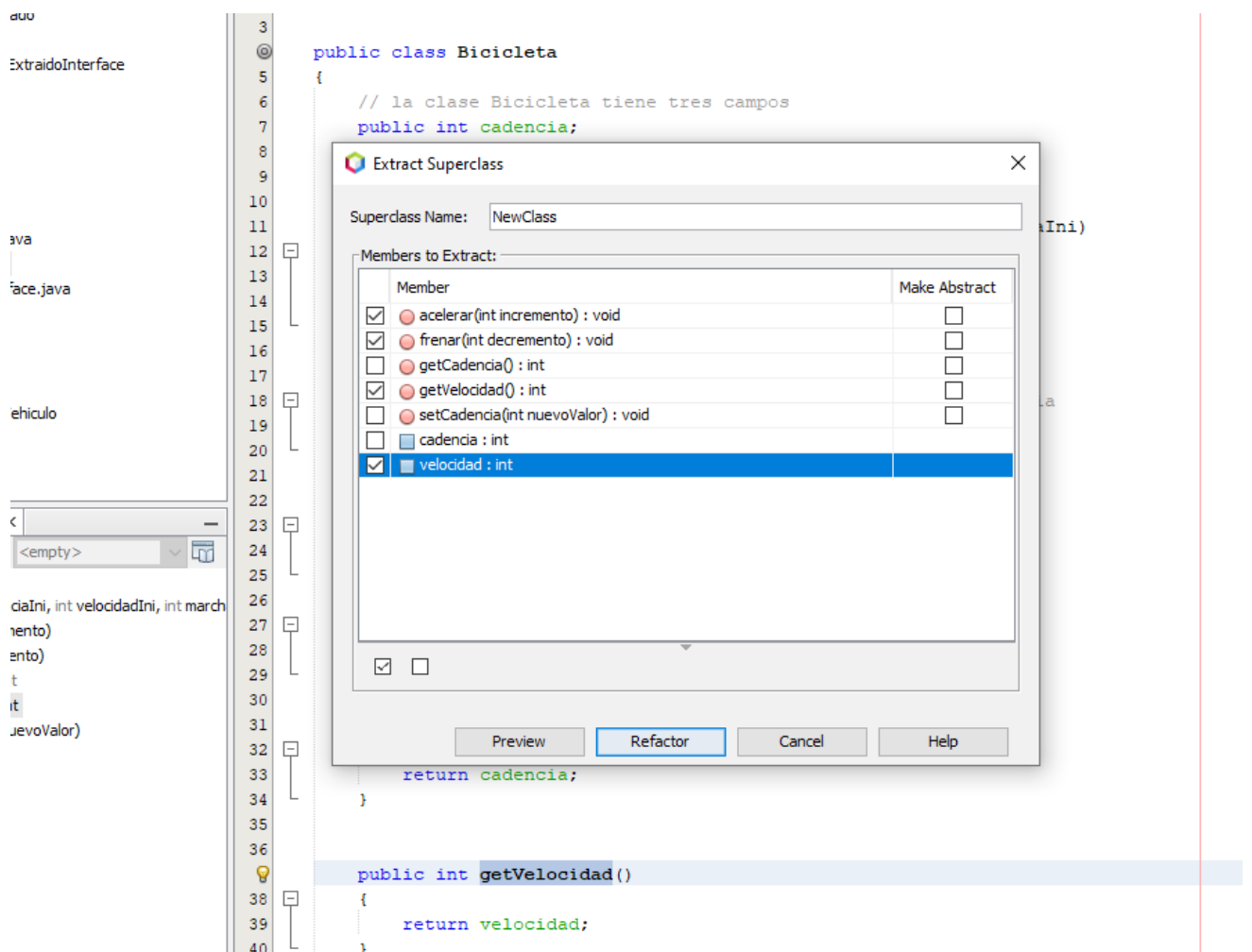
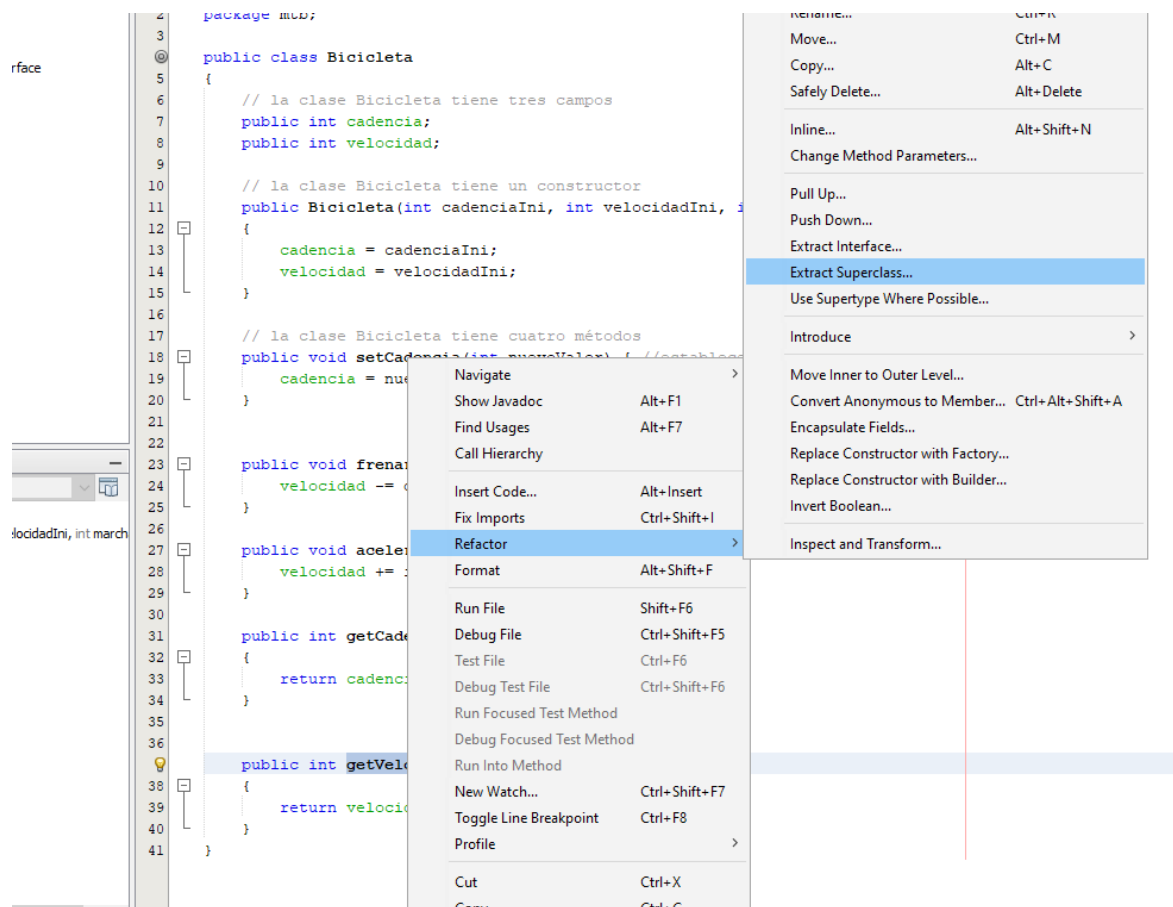
    }

    public void setMarcha(int nuevaMarc
```

Do Refactoring Cancel

20:17/1:14 INS

- Extrae una superclase co campo velocidad e os métodos getVelocidad, acelerar e frenar usando a opción de menú Reestructurar/Extraer Superclase.



Start Page

Refactoring

Extract Superclass class Bicicleta [6 occurrences]

MTB

Bicideta.java

Update

Update

Update

Update

Update

mtb

Create file NewClass.java

Bicideta.java

MTB.java

NewInterface.java

6/6

Refactored Bicideta.java

package mtb;

public class Bicicleta

{

// la clase Bicicleta tiene

public int cadencia;

public int velocidad;

// la clase Bicicleta tiene

public Bicicleta(int cadencia

{

cadencia = cadenciaIni;

velocidad = velocidadIni

}

// la clase Bicicleta tiene

public void setCadencia(int

{

cadencia = nuevoValor;

}

public void frenar(int decre

velocidad -= decremento

}

public void acelerar(int in

velocidad += incremento

}

public int getCadencia()

{

return cadencia;

}

public int getVelocidad()

{

return velocidad;

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

package mtb;

public class Bicicleta extends NewClass

{

// la clase Bicicleta tiene tres ca

public int cadencia;

// la clase Bicicleta tiene un cons

public Bicicleta(int cadenciaIni, i

{

cadencia = cadenciaIni;

velocidad = velocidadIni;

}

// la clase Bicicleta tiene cuatro

public void setCadencia(int nuevoVa

cadencia = nuevoValor;

}

public int getCadencia()

{

return cadencia;

}

}

Refactoring

Cancel