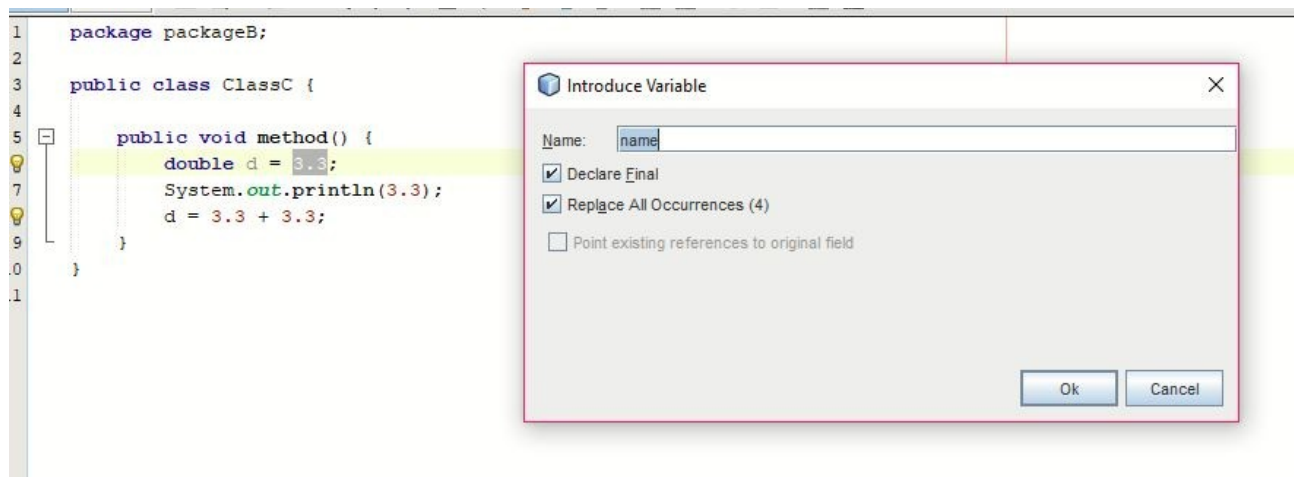


Refactorización

NetBeans.

1-

Con la opción variable se ha conseguido que un valor pase a ser una variable, esto nos es útil sobretodo si este valor coincide en muchos puntos y queremos poder cambiarlo desde un único punto una vez ya hecho el programa.



```
1 package packageB;
2
3 public class ClassC {
4
5     public void method() {
6         final double nuevo_nombre = 3.3;
7         double d = nuevo_nombre;
8         System.out.println(nuevo_nombre);
9         d = nuevo_nombre + nuevo_nombre;
10    }
11 }
12
```

2-

Esta opción nos sirve para convertir la selección en una constante y allí donde se use su valor sustituirla automáticamente.

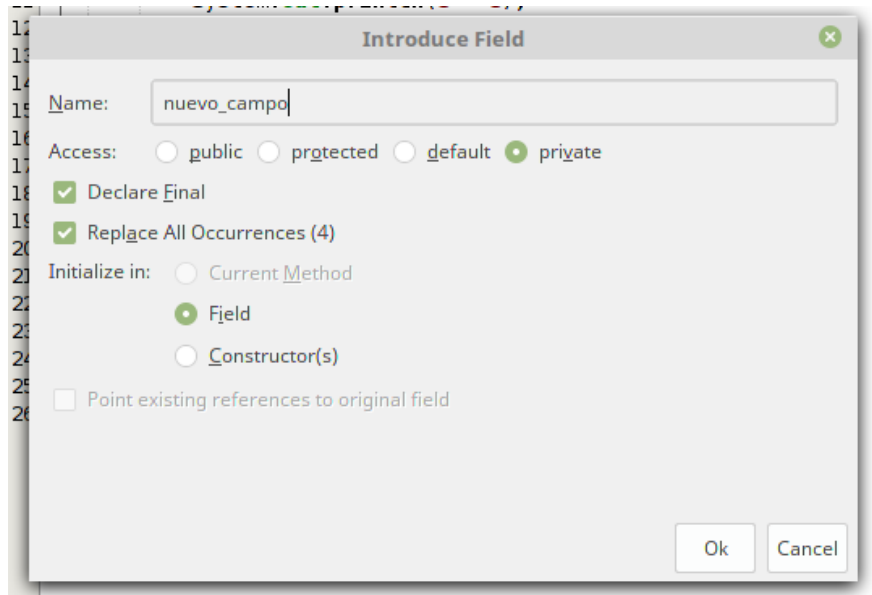
```
1  package packageB;
2
3  import java.util.LinkedList;
4  import java.util.List;
5
6  public class ClassB {
7      List l = new LinkedList();
8
9      String s;
10
11     public ClassB() {
12         this.s = MI_TEXTO;
13     }
14     private static final String MI_TEXTO = "text";
15
16     public void method() {
17         String local = MI_TEXTO;
18     }
19
20
21     public static int statMethod() {
22         int c = 3;
23         System.out.println(c);
24         return 3;
25     }
26 }
27
28
```

Ejercicio 3)

Este ejercicio se encuentra junto al código adjunto.

3-

Esta opción cambia todas las concurrencias por una variable.



```
1 package packageB;
2
3 import java.util.LinkedList;
4 import java.util.List;
5
6 public class ClassA {
7
8     public void method() {
9         int x = nuevo_campo;
10        int y = nuevo_campo + 3;
11        System.out.println(nuevo_campo);
12    }
13
14    private final int nuevo_campo = 3 + 3;
15
16    public void method2() {
17        int x = nuevo_campo;
18    }
19
20    String constExpr = "CONST";
21
22    public static void method3() {
23        String s = "Text";
24    }
25
26    String konst = "a";
27 }
```

4-

En este caso sustituye una variable dentro de la función y la convierte en un parámetro de esta.

```

70
71 public static void statMethod(int a) {
72     int b=2;
73
74     System.out.println(a+b);
75
76 }

```

5-//NO ENCONTRADO//

6-

Esta opción nos permite separar en dos clases distintas los métodos de forma automática.

The screenshot shows an IDE with the following components:

- Source Packages:** A tree view on the left showing a project structure with packages A, B, and C, and various classes including ClassA.java, SubClass.java, ClassB.java, ClassC.java, Prueba1.java, and Person.java.
- ClassA.java Code:**

```

10
11 double field;
12
13 public ClassA() {
14 }
15
16 public class SubClass {
17     double subfield;
18     public void subMethod() {
19     }
20 }
21
22 private packageA.SubClass g_variable;
23
24 public void noReturn() {
25
26     int a = 0;
27     int b = 0;
28
29     System.out.println(a+b);
30 }
31
32
33 public void singleOutput() {
34     int a = 0;
35     int b = 0;
36
37     int c = 0;
38 }

```
- SubClass.java Code:**

```

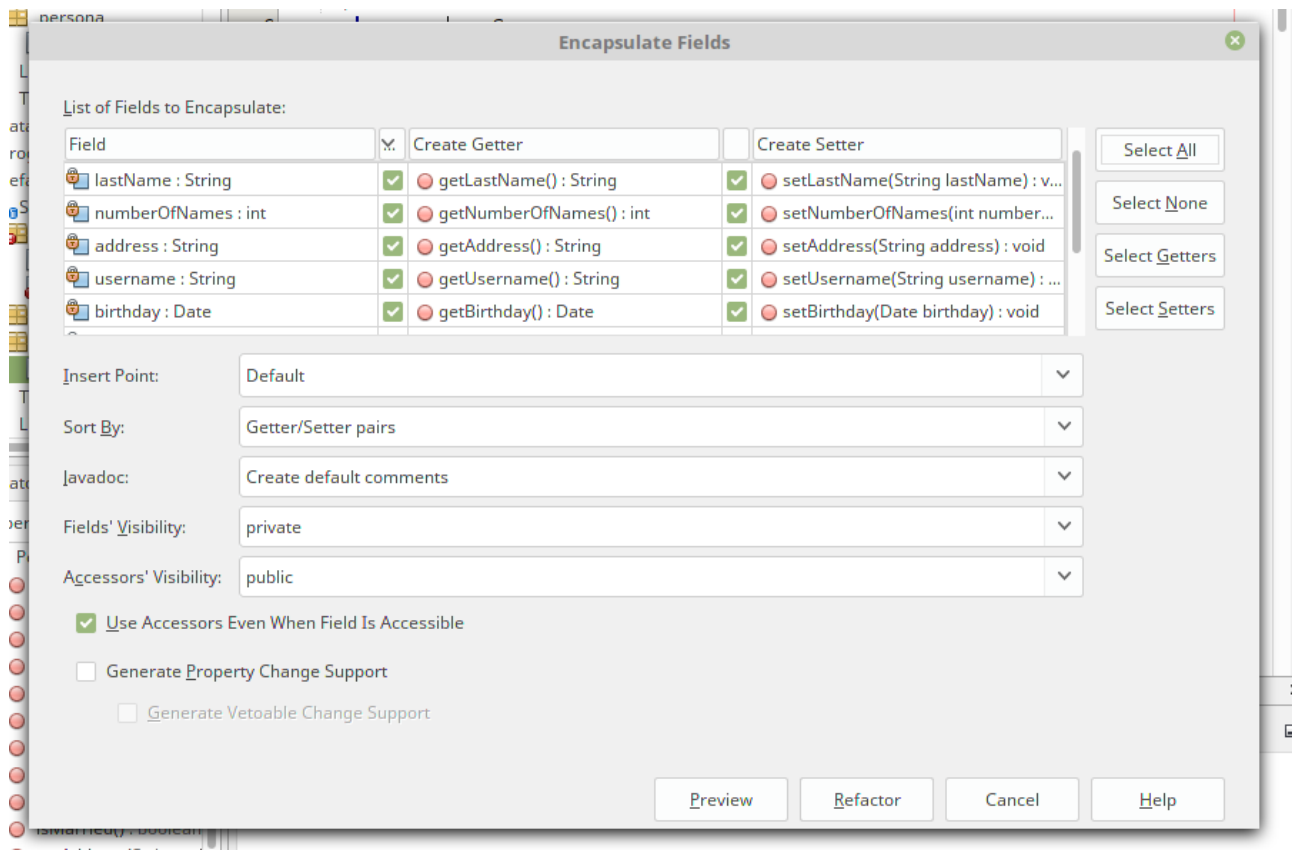
10
11 double field;
12
13 public ClassA() {
14 }
15
16 public class SubClass {
17     double subfield;
18     public void subMethod() {
19     }
20 }
21
22 private packageA.SubClass g_variable;
23
24 public void noReturn() {
25
26     int a = 0;
27     int b = 0;
28
29     System.out.println(a+b);
30 }
31
32
33 public void singleOutput() {
34     int a = 0;
35     int b = 0;
36
37     int c = 0;
38 }

```
- Output Window:** A table at the bottom showing error messages:

Description	File	Location
an enclosing instance that contains pack...	SubClass.java	...actoring/src/packageA/SubClass.java
cannot find symbol symbol: variable A...	SubClass.java	...actoring/src/packageA/SubClass.java
incompatible types: packageA.SubClass ...	SubClass.java	...actoring/src/packageA/SubClass.java

7-

Esta opción nos sirve para encapsular ciertos campos con su correspondientes metodos de acceso.



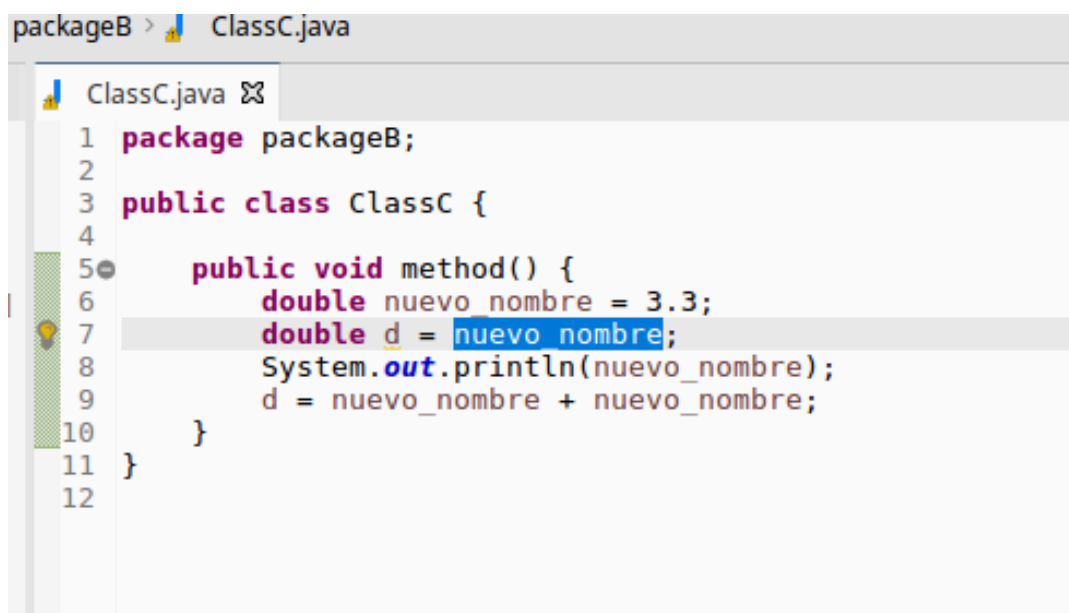
Ejercicios)

1) tabla comparativa

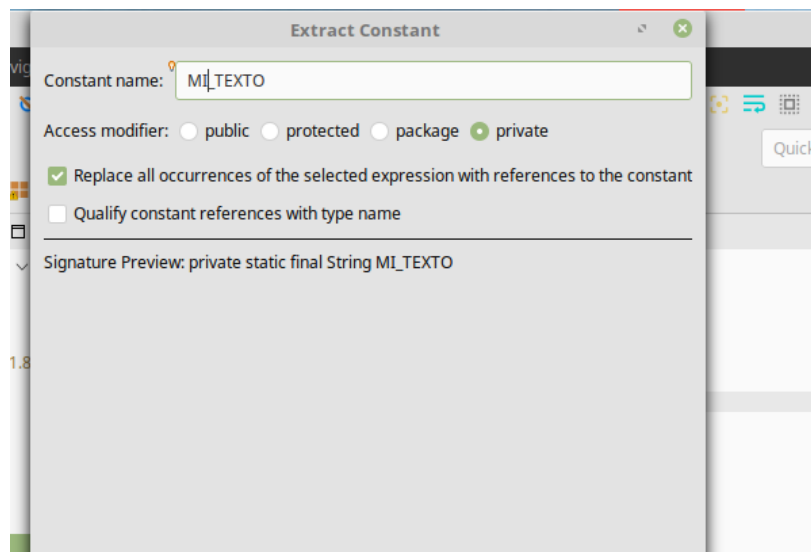
Funcion	NetBeans	Eclipse
Introducir variable	Si	Si
Introducir Constante	Si	Si
Introducir field	Si	No
Introducir parametro	Si	Si
Introducir Metodo	Si	Si
Introducir local extension	Si	No
Extraer interfaz	Si	Si
Extraer Súper Clase	Si	Si

Apartados en eclipse)

1-

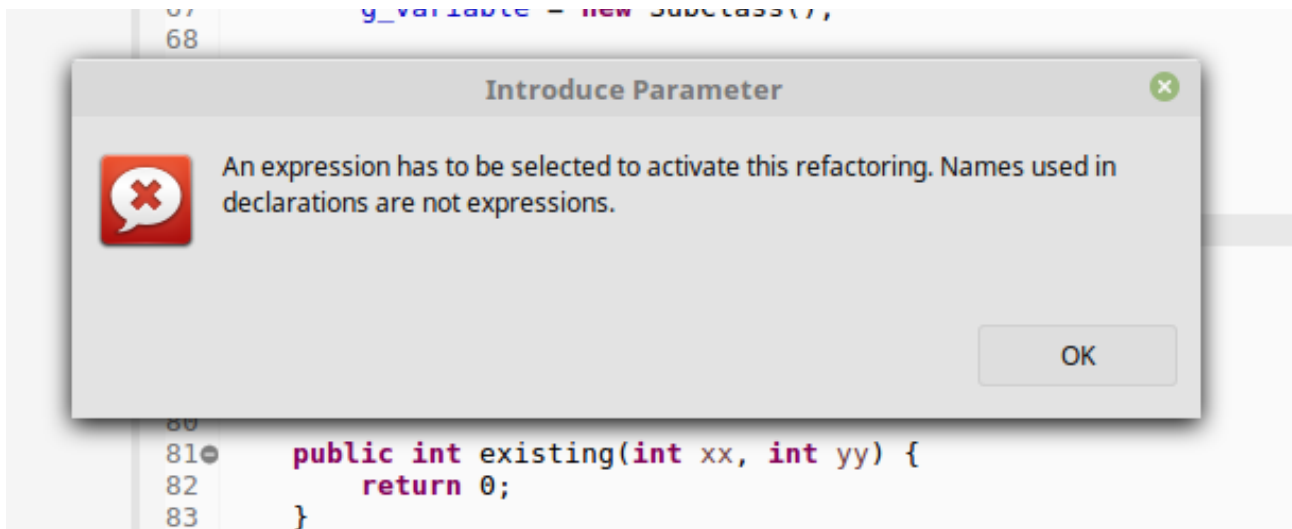


2-

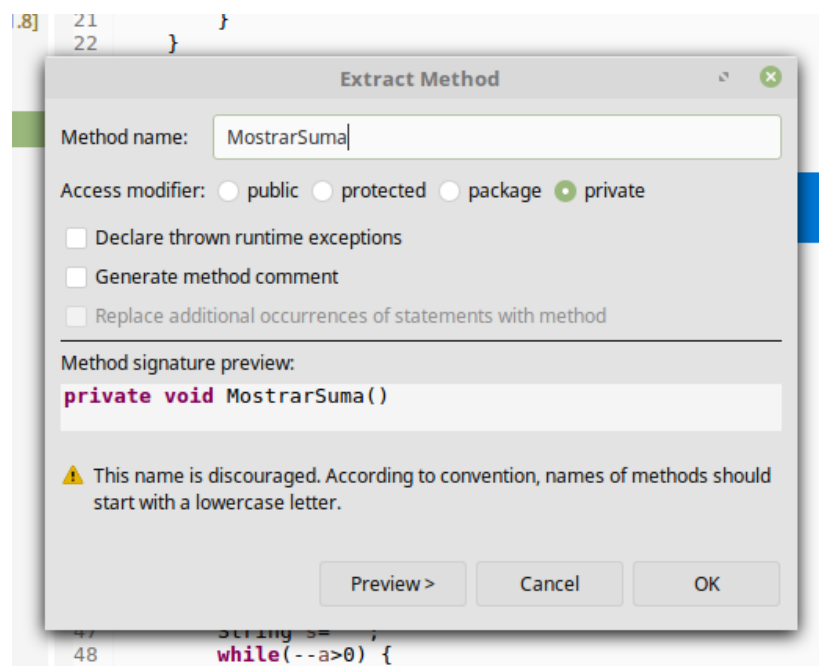


```
ClassB.java
1 package packageB;
2
3 import java.util.LinkedList;
4
5
6 public class ClassB {
7     private static final String MI_TEXTO = "text";
8
9     List l = new LinkedList();
10
11     String s = MI_TEXTO;
12
13     public void method() {
14         String local = MI_TEXTO;
15     }
16
17     public static int statMethod() {
18         int c = 3;
19         System.out.println(c);
20         return 3;
21     }
22 }
23
24
```

4-



5-




```
    }  
    private SubClass g_variable;  
    public void noReturn() {  
        MostrarSuma();  
    }  
    private void MostrarSuma() {  
        int a = 0;  
        int b = 0;
```