

Alumno/a: José Manuel García Valverde

[Examen Refactorización]

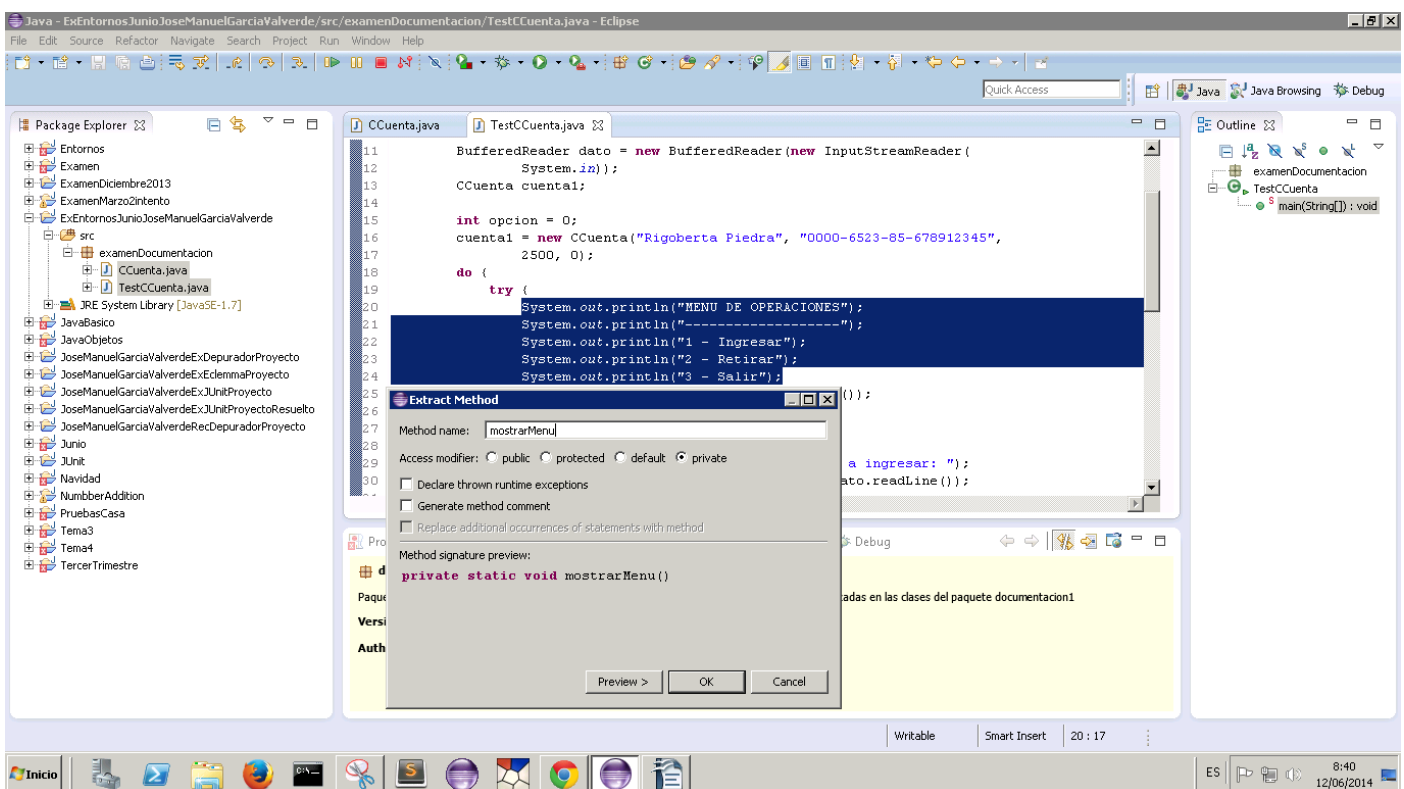
Refactorización:

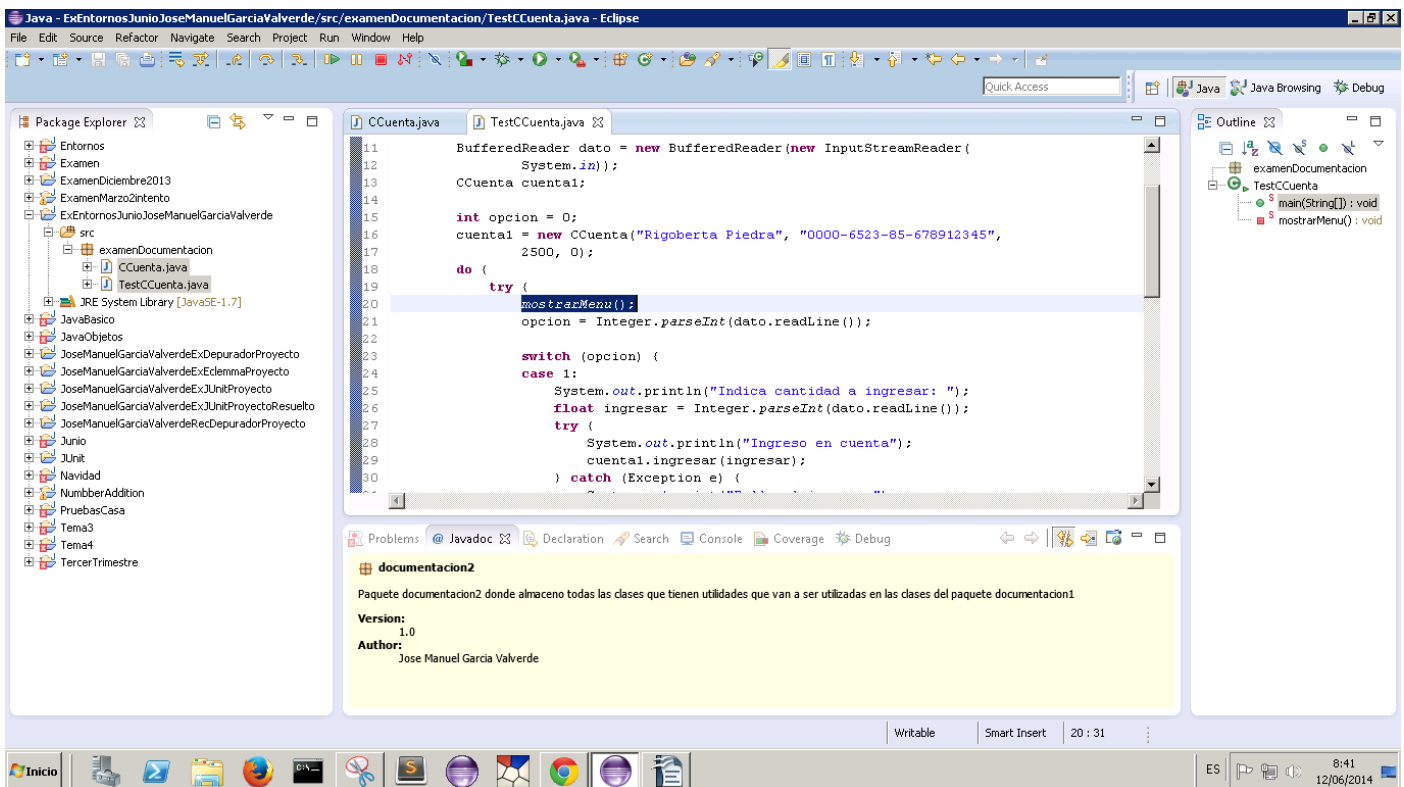
Deberás refactorizar ambas clases y entregar, aparte del resultado, un documento "ApellidosNombreRefactorizacion.pdf" con pantallazos del antes y después de cada uno de los pasos debidamente identificados, incluyendo el menú utilizado para ello.

Realiza las siguientes actividades:

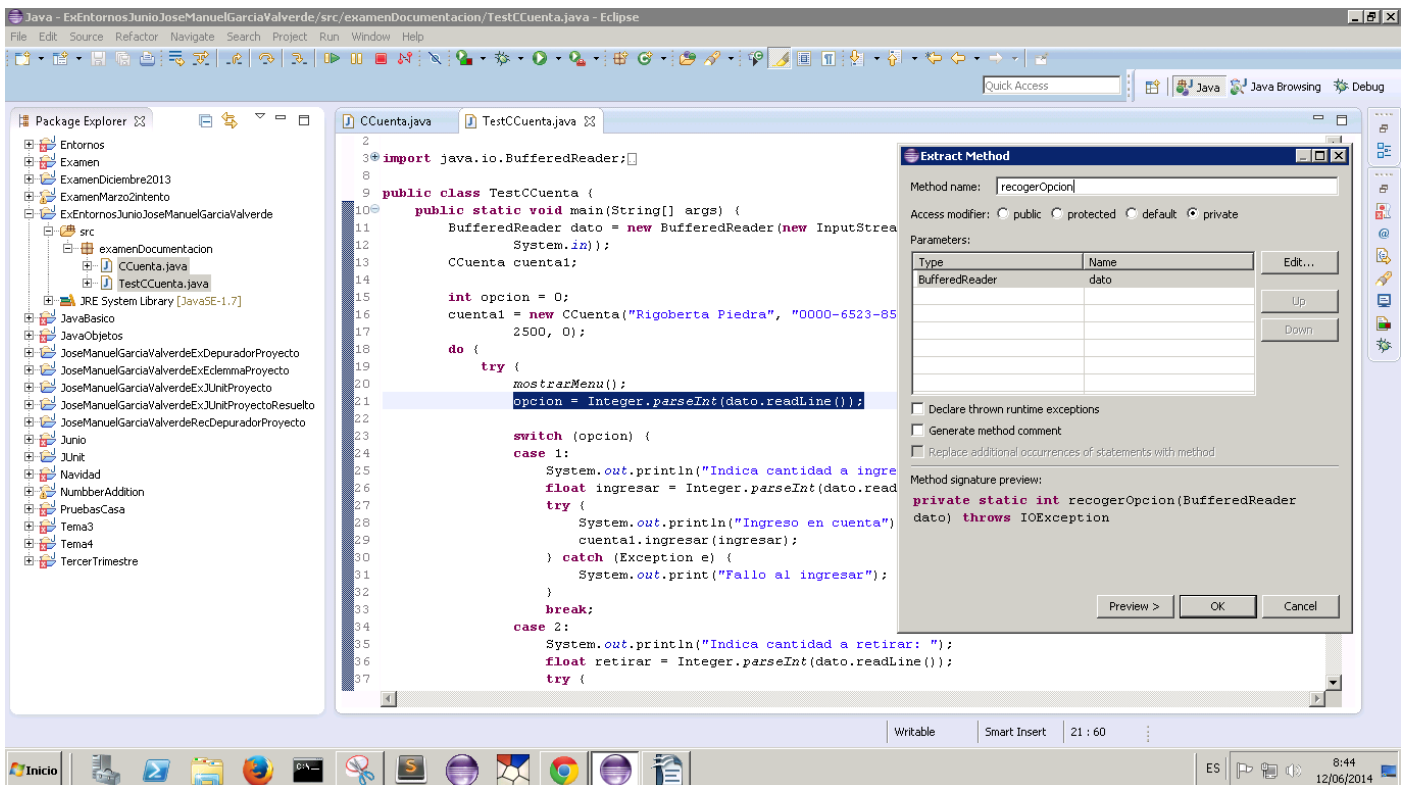
- En la clase TestCCuenta
 - TestCCuenta.java: Extrae los métodos recogerOpcion, mostrarMenu, ingresar, retirar

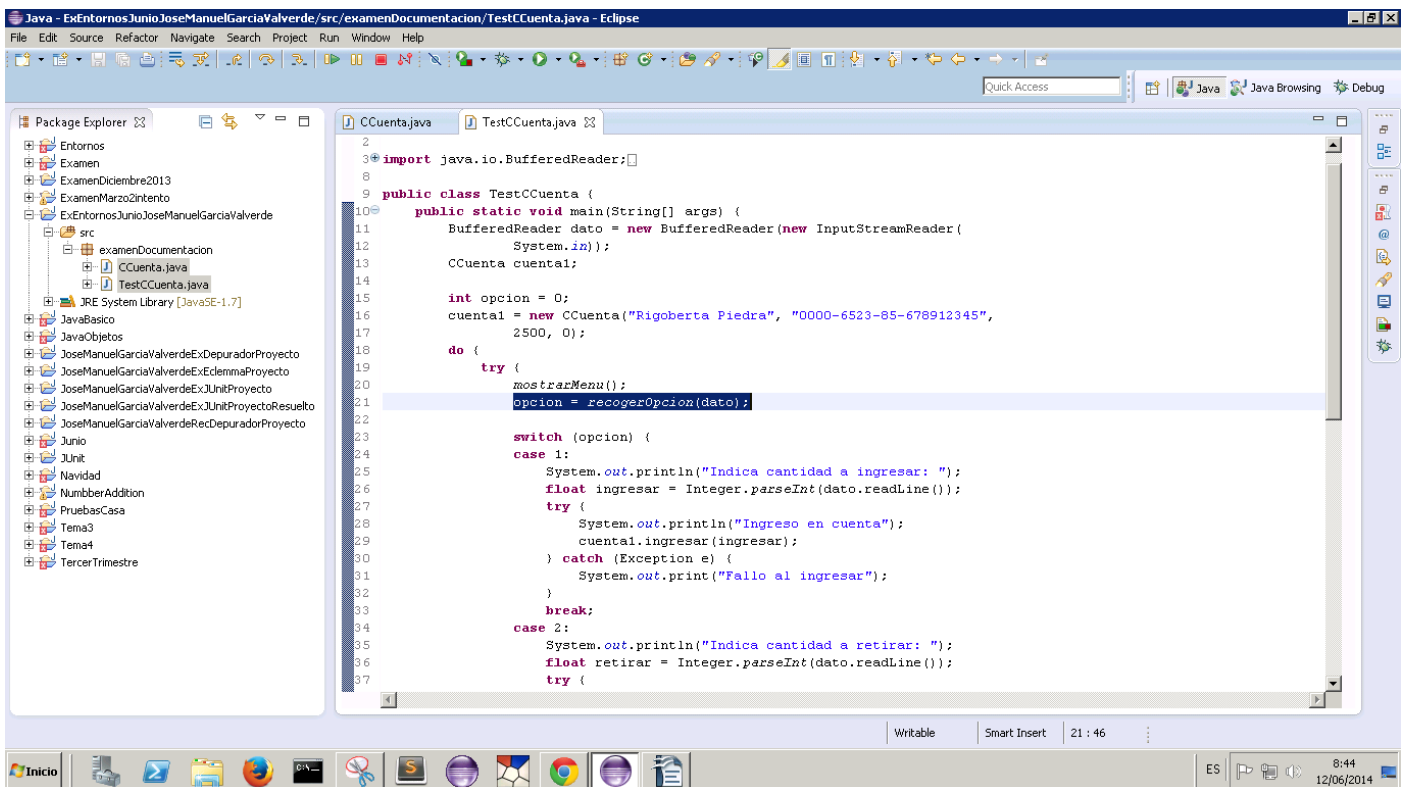
mostrarMenu



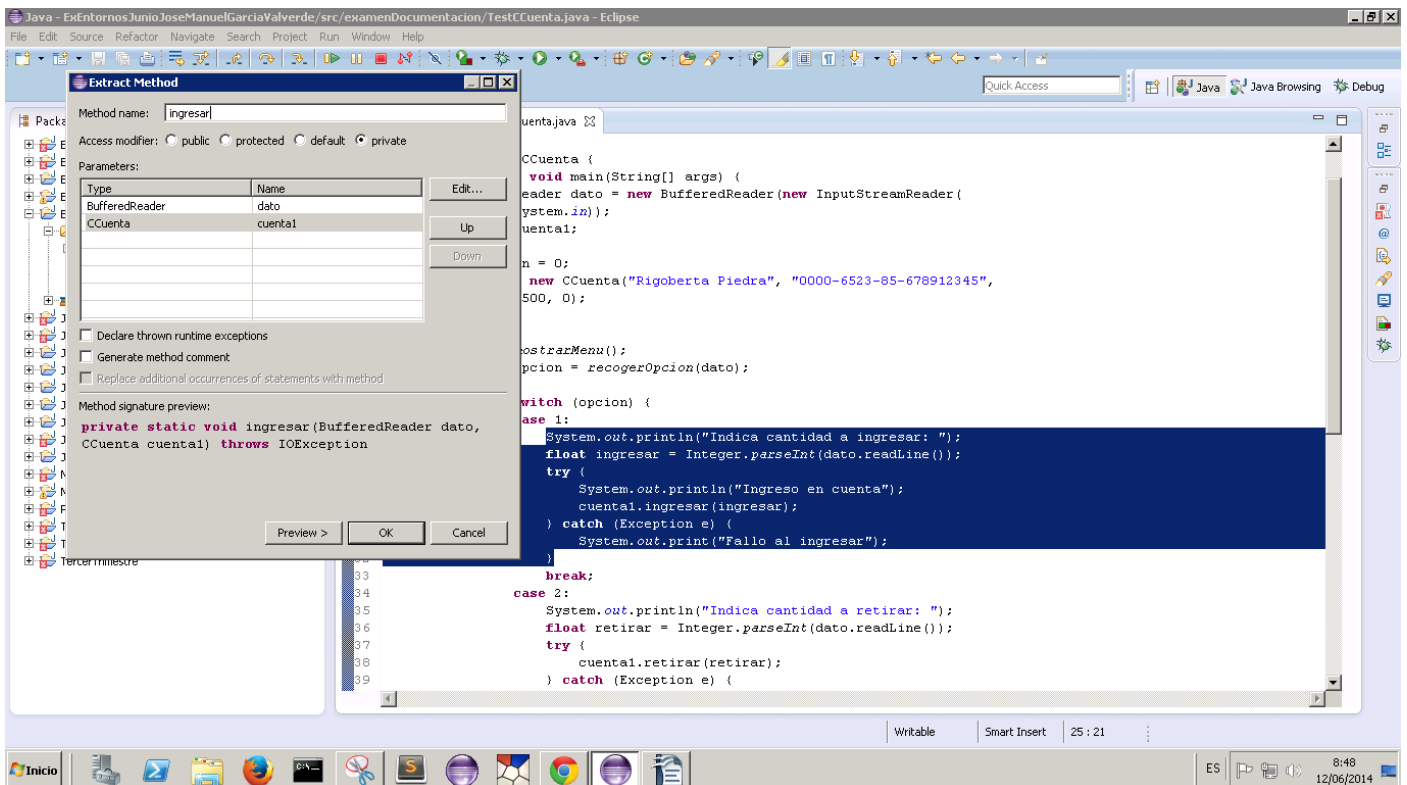


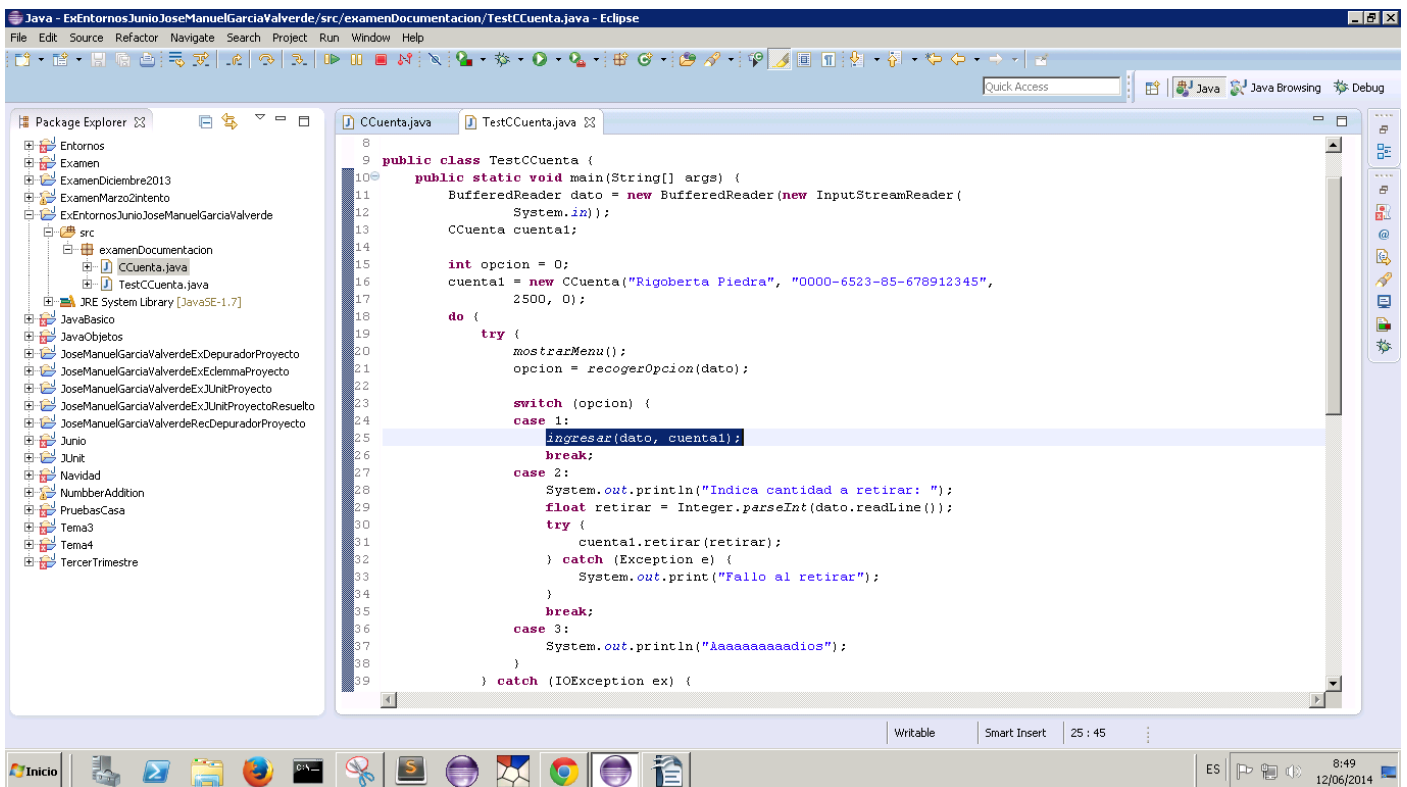
recogerOpcion:



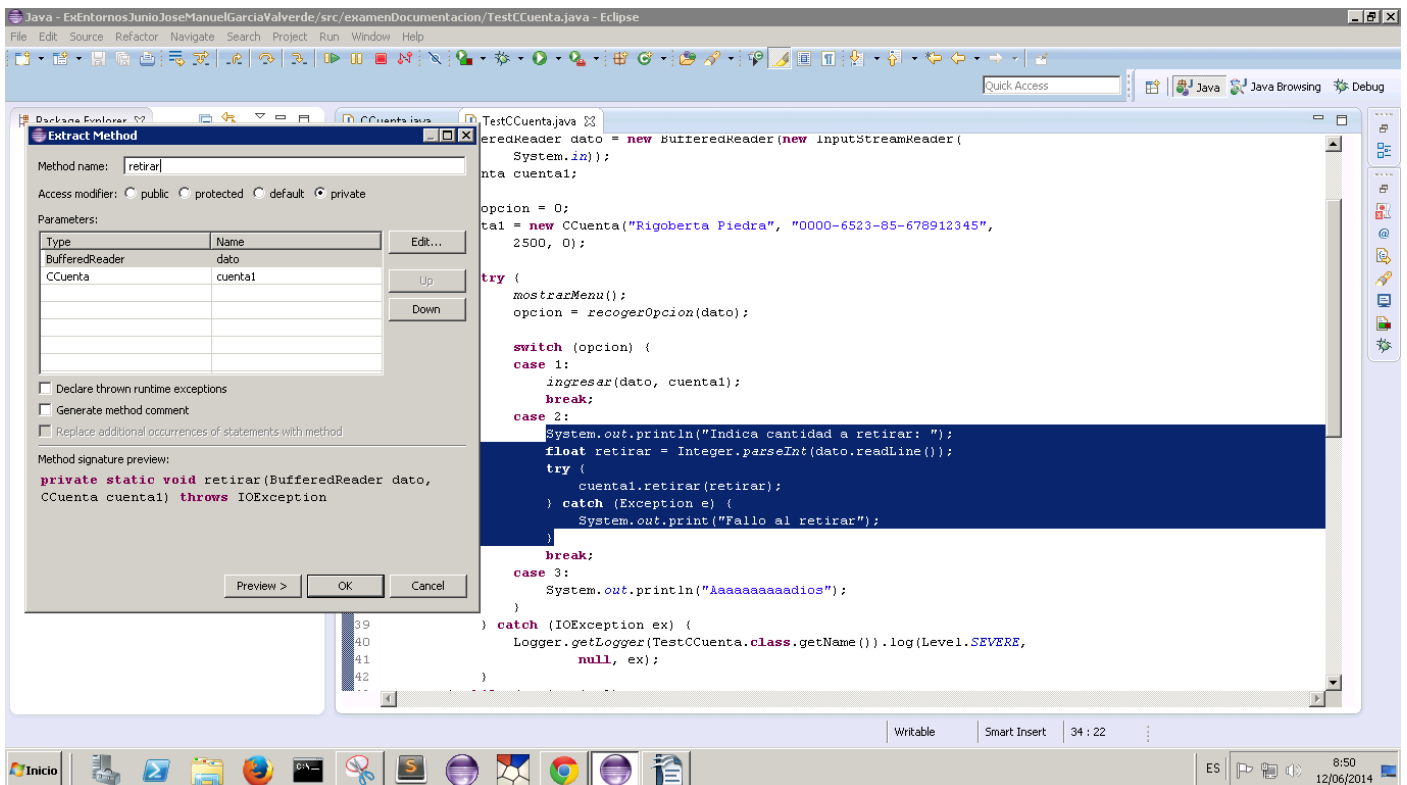


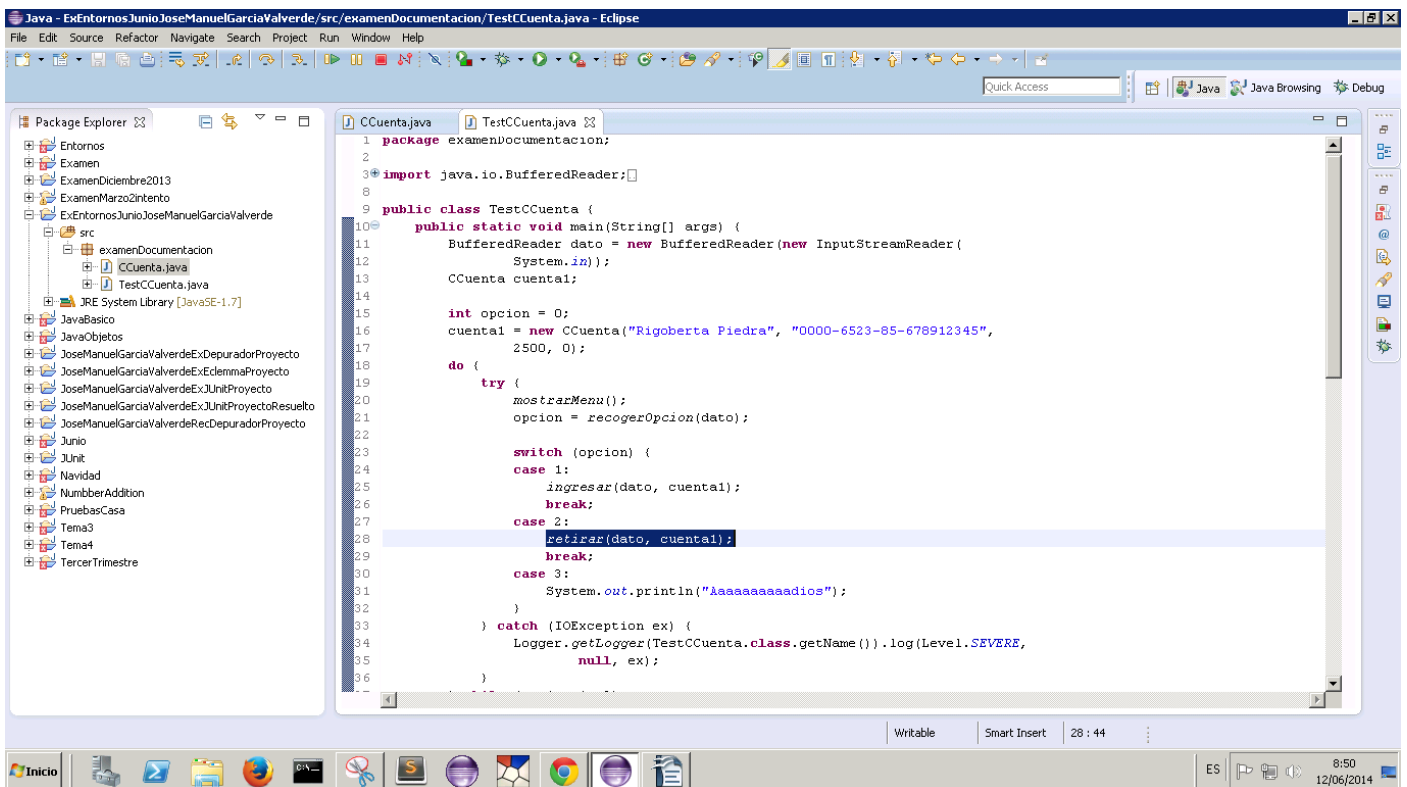
Ingresar:



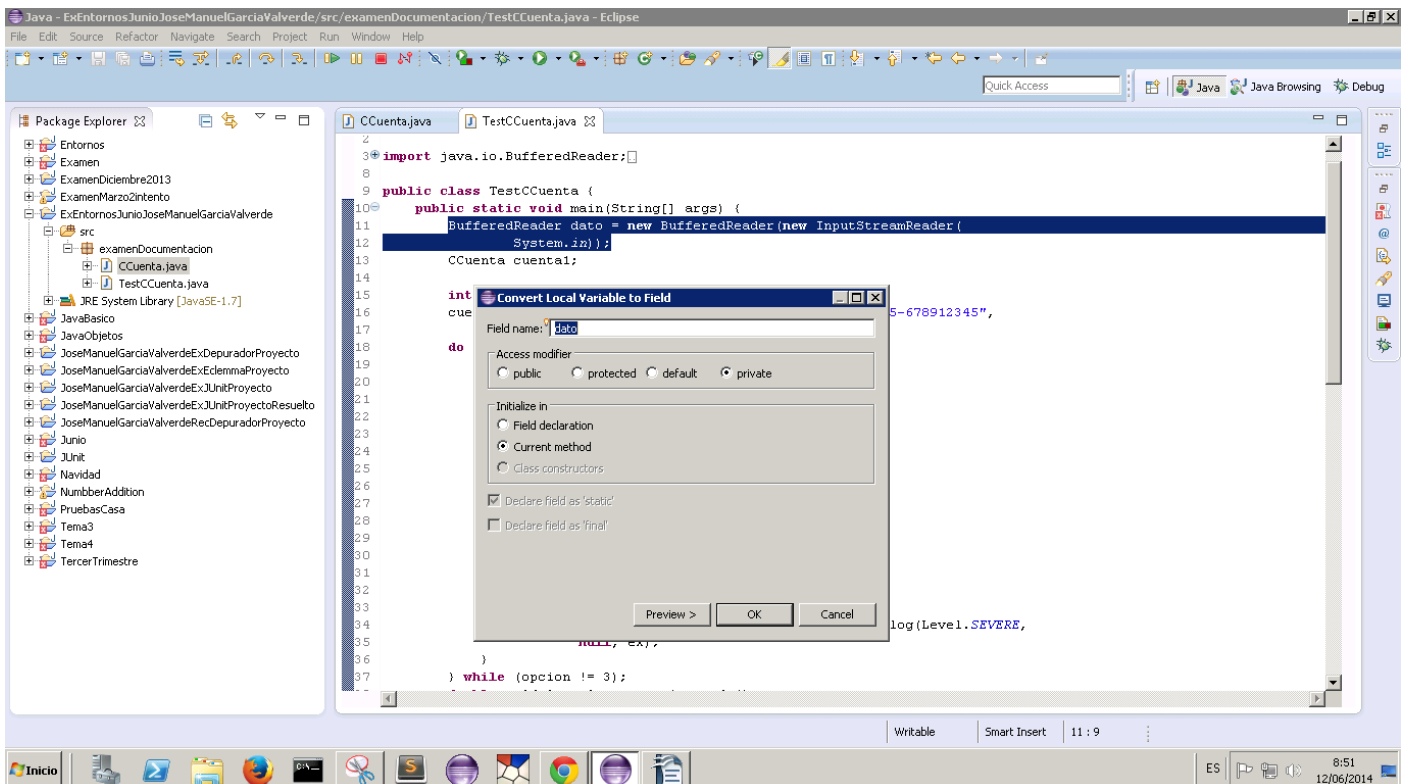


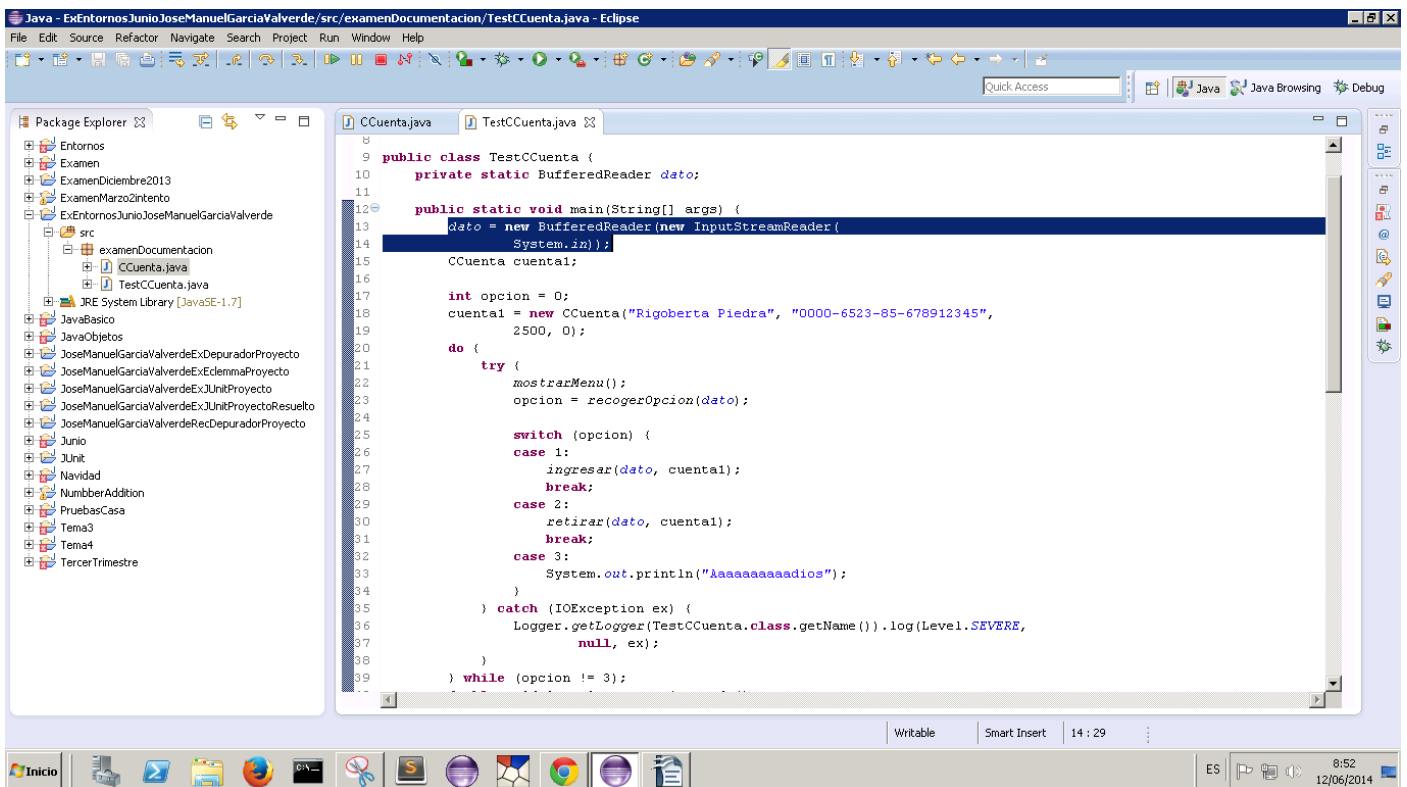
Retirar:





- TestCCuenta.java: Convierte dato en campo



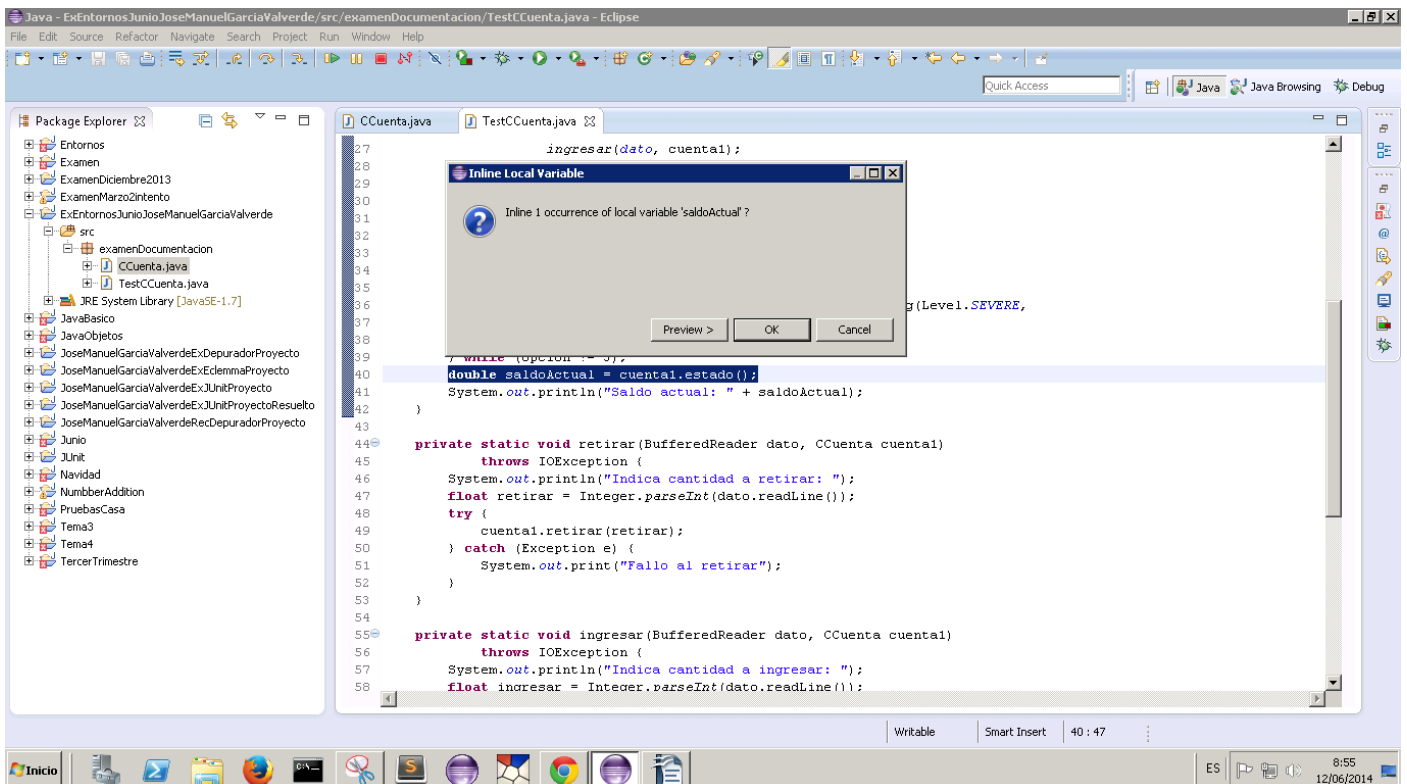


```

8
9 public class TestCCuenta {
10     private static BufferedReader dato;
11
12     public static void main(String[] args) {
13         dato = new BufferedReader(new InputStreamReader(
14             System.in));
15         CCuenta cuenta1;
16
17         int opcion = 0;
18         cuenta1 = new CCuenta("Rigoberta Piedra", "0000-6523-85-678912345",
19             2500, 0);
20         do {
21             try {
22                 mostrarMenu();
23                 opcion = recogerOpcion(dato);
24
25                 switch (opcion) {
26                     case 1:
27                         ingresar(dato, cuenta1);
28                         break;
29                     case 2:
30                         retirar(dato, cuenta1);
31                         break;
32                     case 3:
33                         System.out.println("aaaaaaaaaadios");
34                 }
35             } catch (IOException ex) {
36                 Logger.getLogger(TestCCuenta.class.getName()).log(Level.SEVERE,
37                     null, ex);
38             }
39         } while (opcion != 3);

```

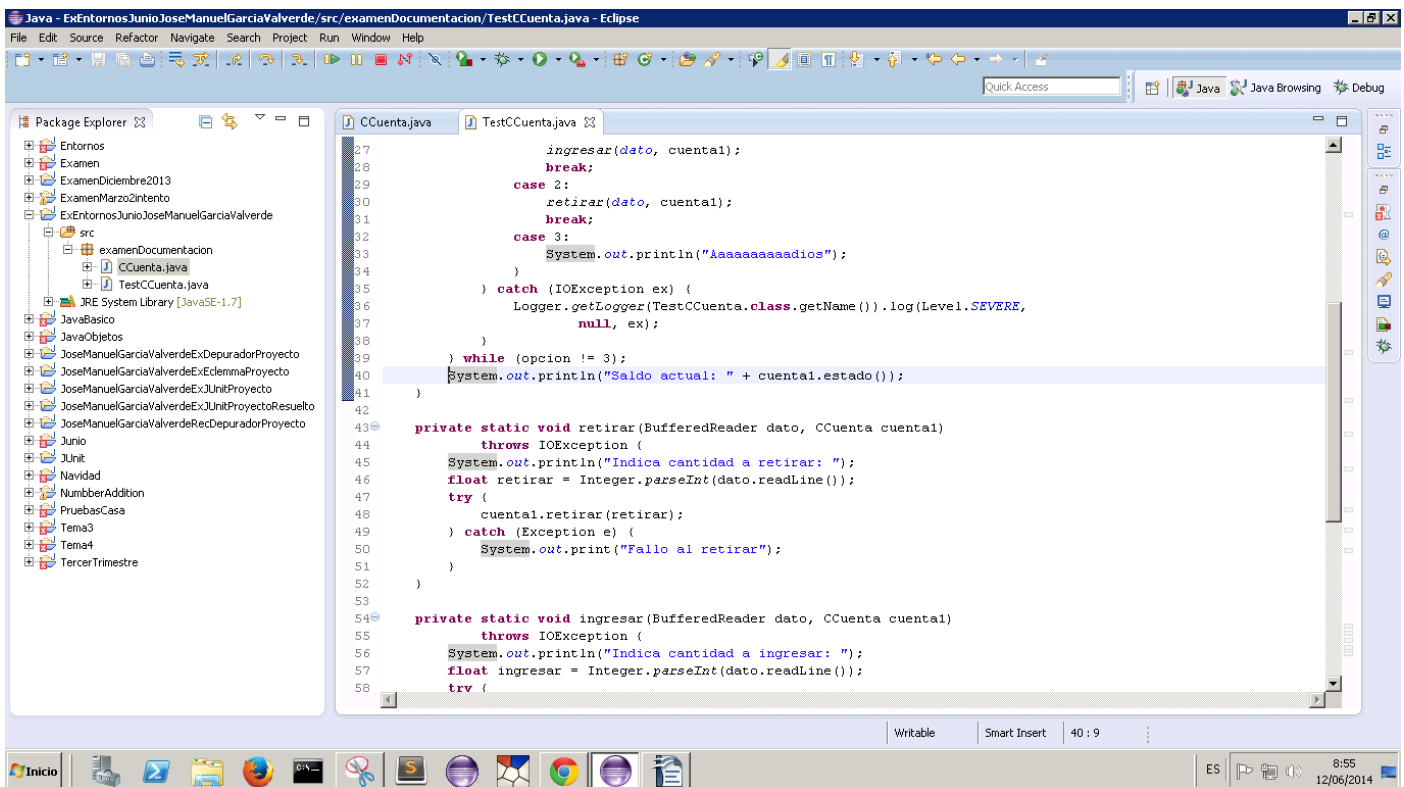
- TestCCuenta.java: Incorpora saldoActual en la misma línea.



```

27     ingresar(dato, cuenta1);
28
29
30
31
32
33
34
35
36
37
38
39
40     double saldoActual = cuenta1.estado();
41     System.out.println("Saldo actual: " + saldoActual);
42 }
43
44 private static void retirar(BufferedReader dato, CCuenta cuenta1)
45     throws IOException {
46     System.out.println("Indica cantidad a retirar: ");
47     float retirar = Integer.parseInt(dato.readLine());
48     try {
49         cuenta1.retirar(retirar);
50     } catch (Exception e) {
51         System.out.print("Fallo al retirar");
52     }
53 }
54
55 private static void ingresar(BufferedReader dato, CCuenta cuenta1)
56     throws IOException {
57     System.out.println("Indica cantidad a ingresar: ");
58     float ingresar = Integer.parseInt(dato.readLine());

```

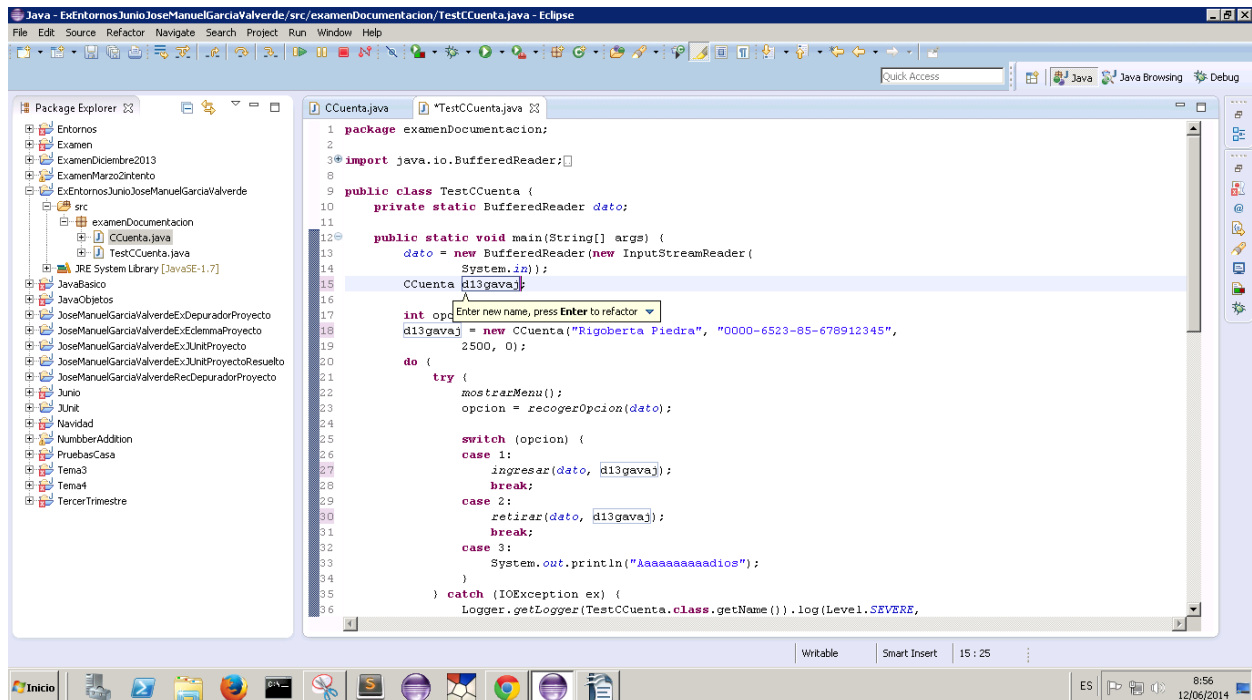



```

27         ingresar(dato, cuenta1);
28         break;
29     case 2:
30         retirar(dato, cuenta1);
31         break;
32     case 3:
33         System.out.println("Aaaaaaaaadios");
34     }
35     } catch (IOException ex) {
36         Logger.getLogger(TestCCuenta.class.getName()).log(Level.SEVERE,
37             null, ex);
38     }
39     } while (opcion != 3);
40     System.out.println("Saldo actual: " + cuenta1.estado());
41 }
42
43 private static void retirar(BufferedReader dato, CCuenta cuenta1)
44     throws IOException {
45     System.out.println("Indica cantidad a retirar: ");
46     float retirar = Integer.parseInt(dato.readLine());
47     try {
48         cuenta1.retirar(retirar);
49     } catch (Exception e) {
50         System.out.print("Fallo al retirar");
51     }
52 }
53
54 private static void ingresar(BufferedReader dato, CCuenta cuenta1)
55     throws IOException {
56     System.out.println("Indica cantidad a ingresar: ");
57     float ingresar = Integer.parseInt(dato.readLine());
58     try {

```

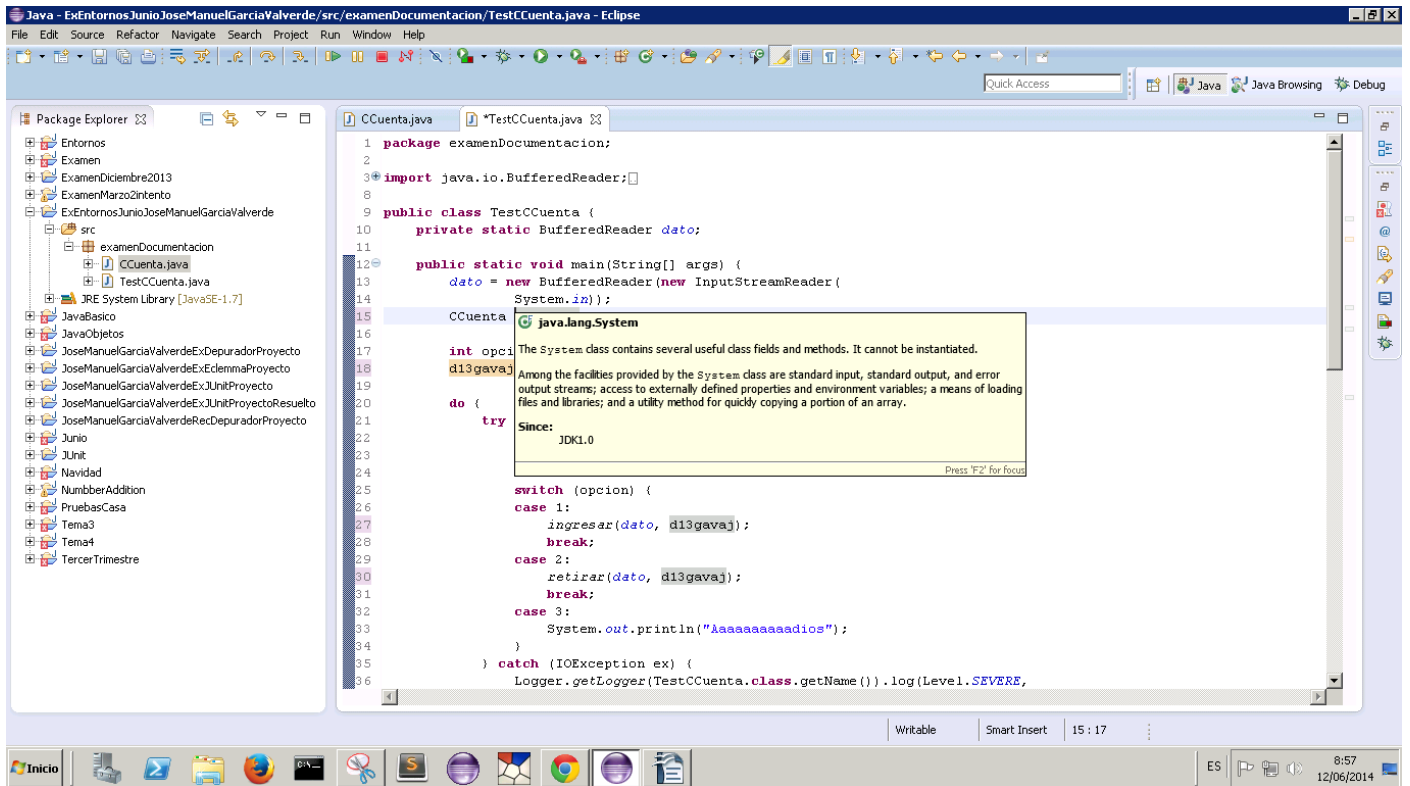
- TestCCuenta.java: Renombra la variable local cuenta1 con tu cuenta en el sistema (d13macom, por ejemplo)



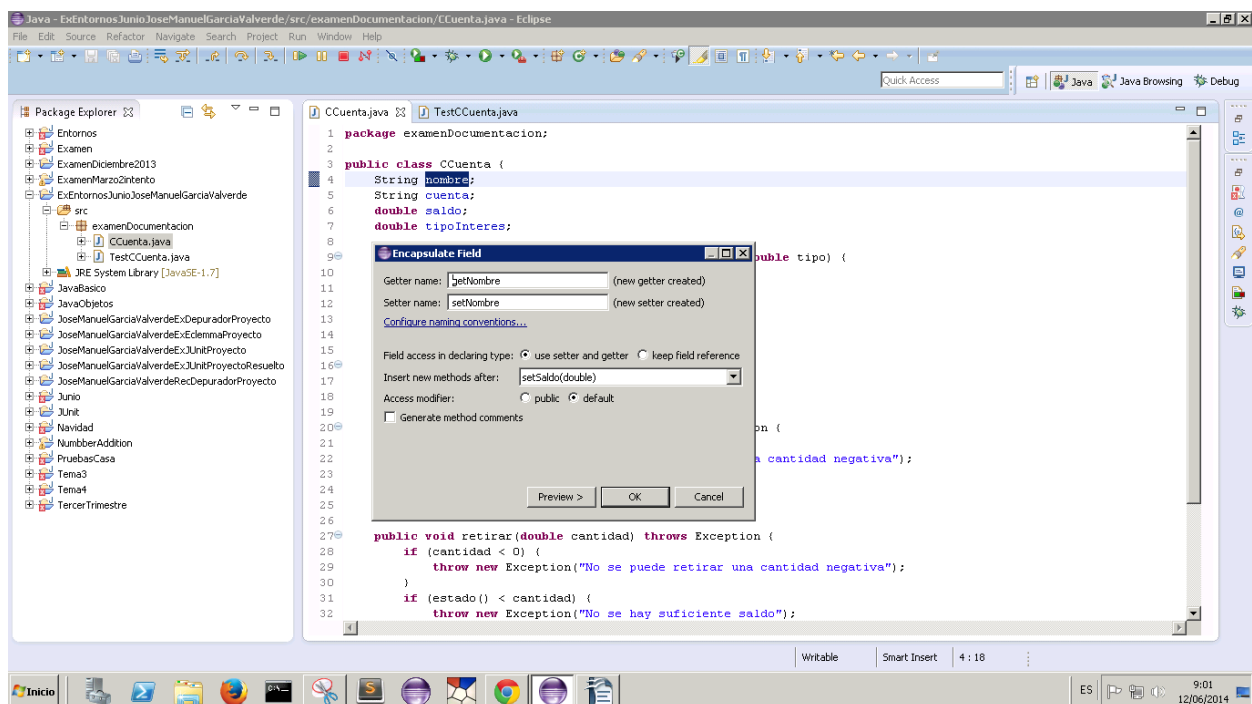
```

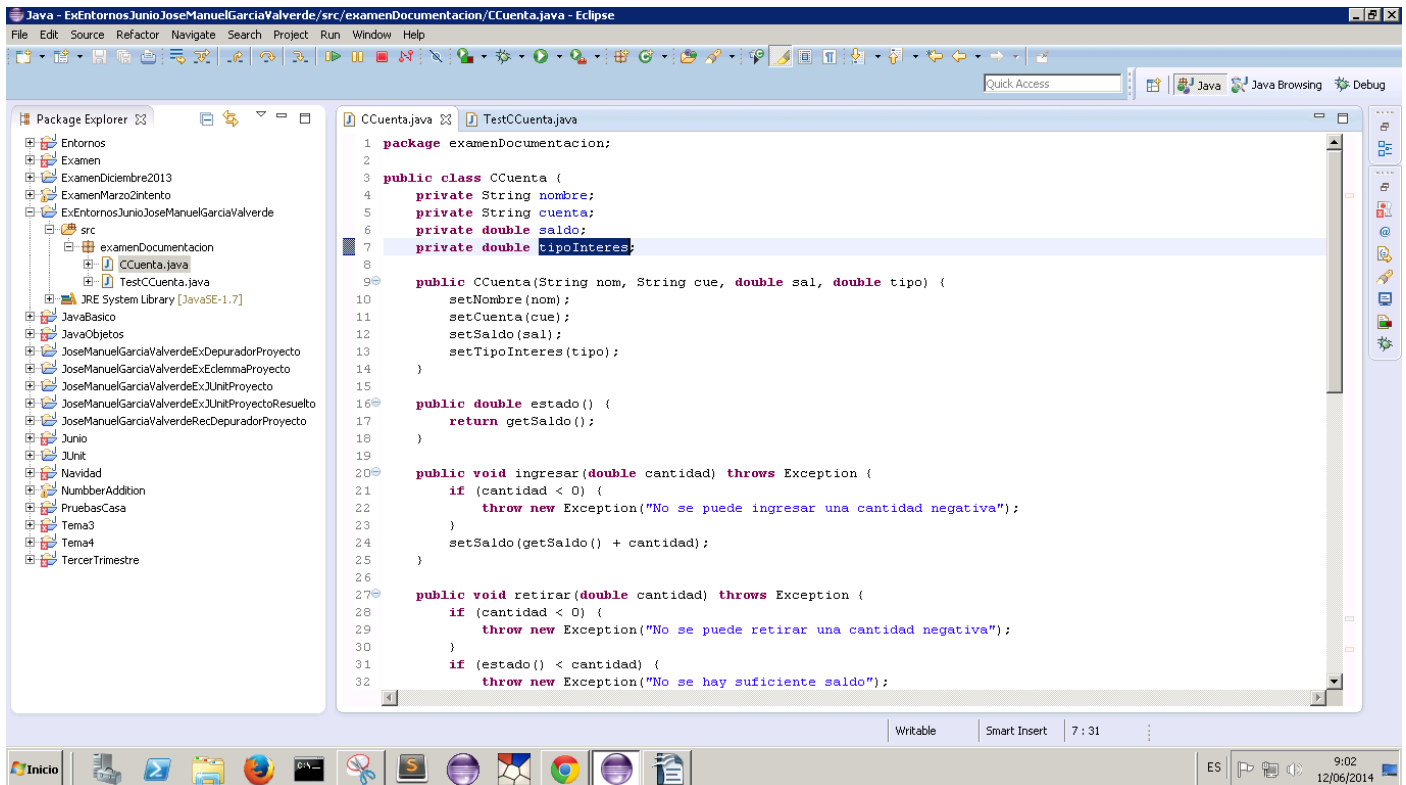
1 package examenDocumentacion;
2
3 import java.io.BufferedReader;
4
5 public class TestCCuenta {
6     private static BufferedReader dato;
7
8     public static void main(String[] args) {
9         dato = new BufferedReader(new InputStreamReader(
10             System.in));
11         CCuenta d13gava1;
12         int opcion;
13         d13gava1 = new CCuenta("Rigoberta Piedra", "0000-6523-85-678912345",
14             2500, 0);
15         do {
16             try {
17                 mostrarMenu();
18                 opcion = recogerOpcion(dato);
19                 switch (opcion) {
20                     case 1:
21                         ingresar(dato, d13gava1);
22                         break;
23                     case 2:
24                         retirar(dato, d13gava1);
25                         break;
26                     case 3:
27                         System.out.println("Aaaaaaaaadios");
28                     }
29                 } catch (IOException ex) {
30                     Logger.getLogger(TestCCuenta.class.getName()).log(Level.SEVERE,
31                         null, ex);
32                 }
33             } while (opcion != 3);
34         }
35     }
36 }

```

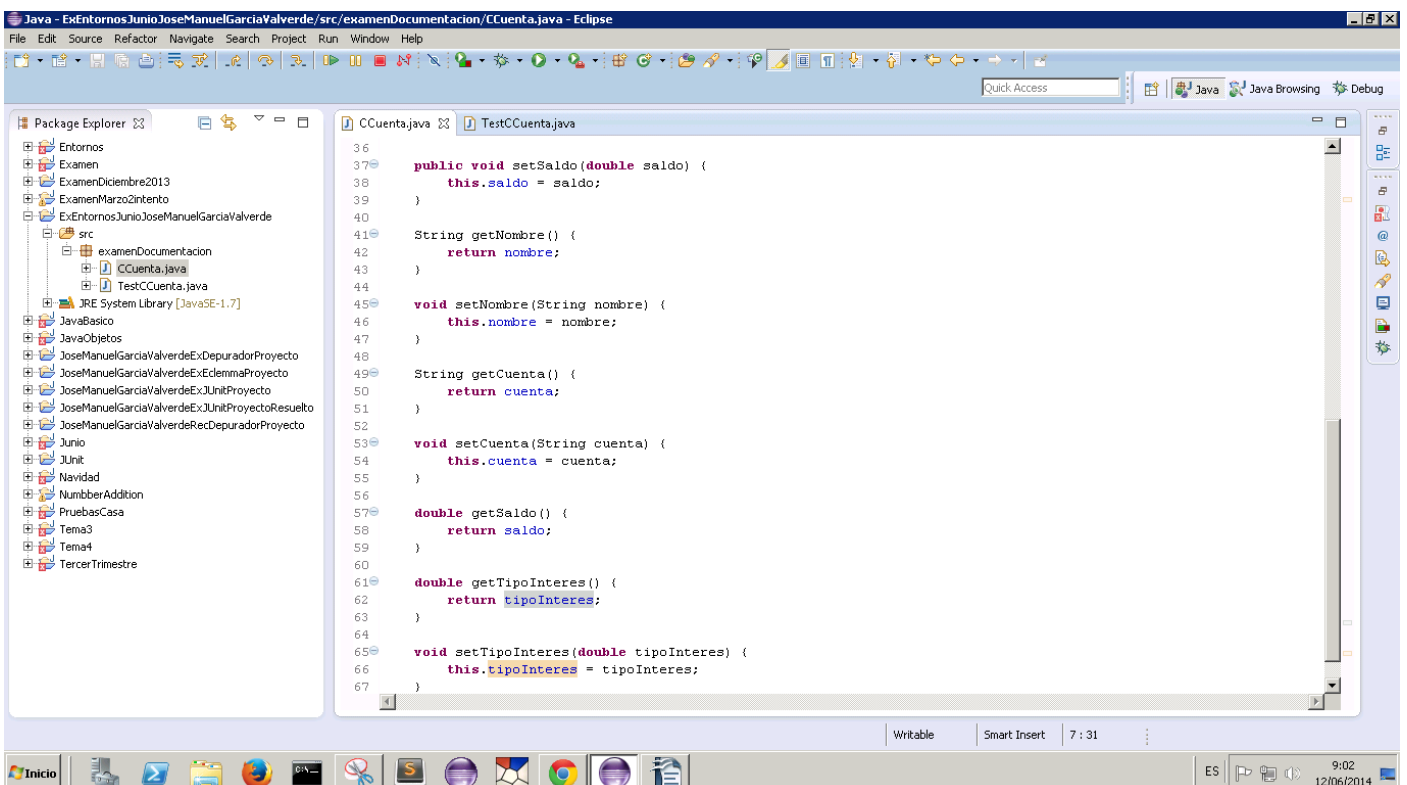


- En la clase CCuenta.java:
- CCuenta.java: Encapsula los campos y usa los métodos generados.





```
1 package examenDocumentacion;
2
3 public class CCuenta {
4     private String nombre;
5     private String cuenta;
6     private double saldo;
7     private double tipoInteres;
8
9     public CCuenta(String nom, String cue, double sal, double tipo) {
10         setNombre(nom);
11         setCuenta(cue);
12         setSaldo(sal);
13         setTipoInteres(tipo);
14     }
15
16     public double estado() {
17         return getSaldo();
18     }
19
20     public void ingresar(double cantidad) throws Exception {
21         if (cantidad < 0) {
22             throw new Exception("No se puede ingresar una cantidad negativa");
23         }
24         setSaldo(getSaldo() + cantidad);
25     }
26
27     public void retirar(double cantidad) throws Exception {
28         if (cantidad < 0) {
29             throw new Exception("No se puede retirar una cantidad negativa");
30         }
31         if (estado() < cantidad) {
32             throw new Exception("No se hay suficiente saldo");
33         }
34     }
35 }
```



```
36
37     public void setSaldo(double saldo) {
38         this.saldo = saldo;
39     }
40
41     String getNombre() {
42         return nombre;
43     }
44
45     void setNombre(String nombre) {
46         this.nombre = nombre;
47     }
48
49     String getCuenta() {
50         return cuenta;
51     }
52
53     void setCuenta(String cuenta) {
54         this.cuenta = cuenta;
55     }
56
57     double getSaldo() {
58         return saldo;
59     }
60
61     double getTipoInteres() {
62         return tipoInteres;
63     }
64
65     void setTipoInteres(double tipoInteres) {
66         this.tipoInteres = tipoInteres;
67     }
68 }
```