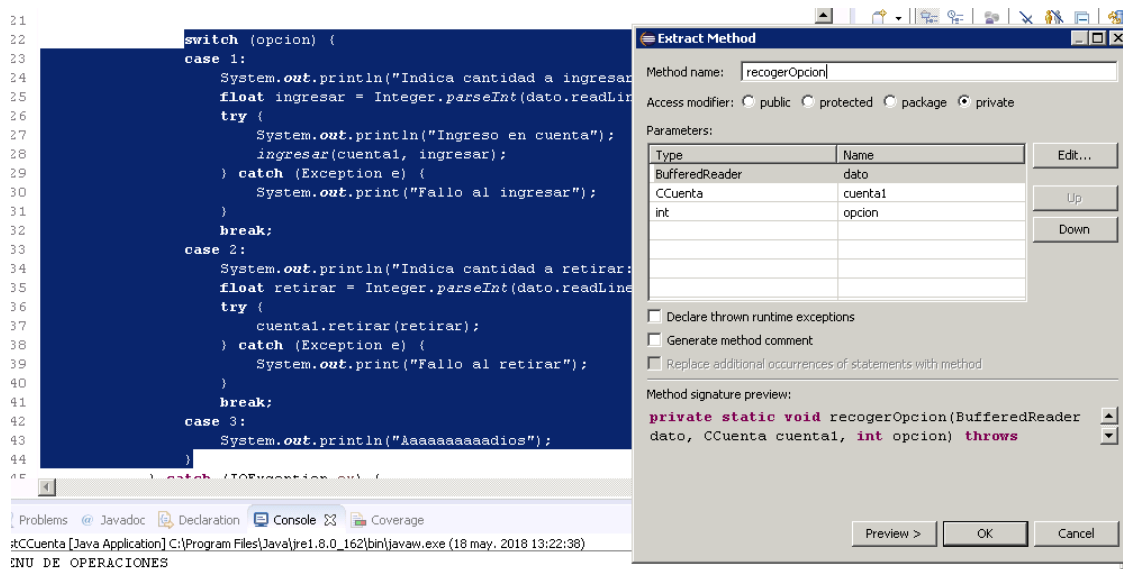


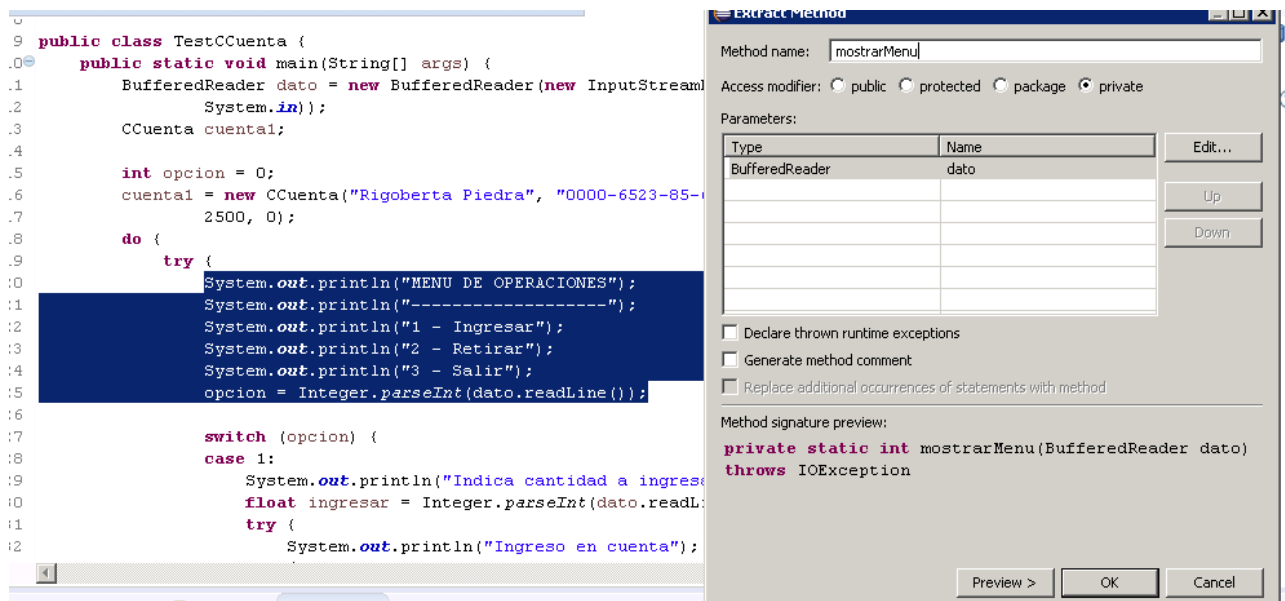
EXAMEN MAYO ENDES REFACTORIZACIÓN

1. TestCCuenta.java: Extrae los métodos recogerOpcion, mostrarMenu, ingresar, retirar.

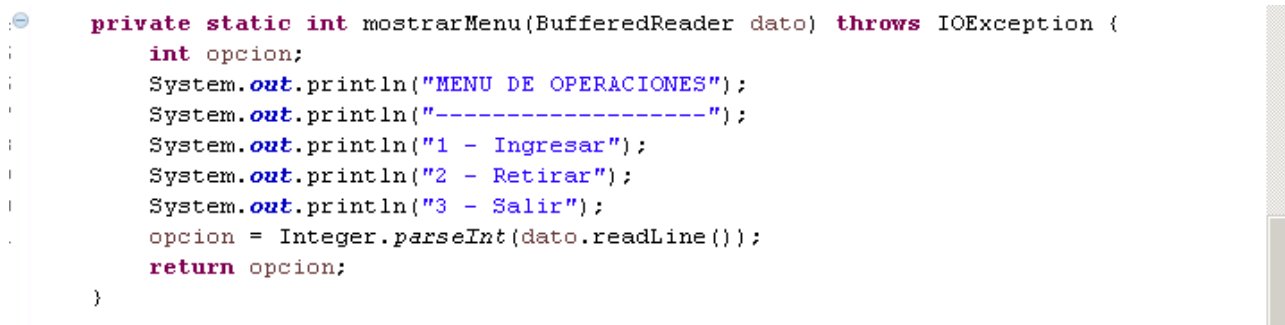
Método recogerOpcion:



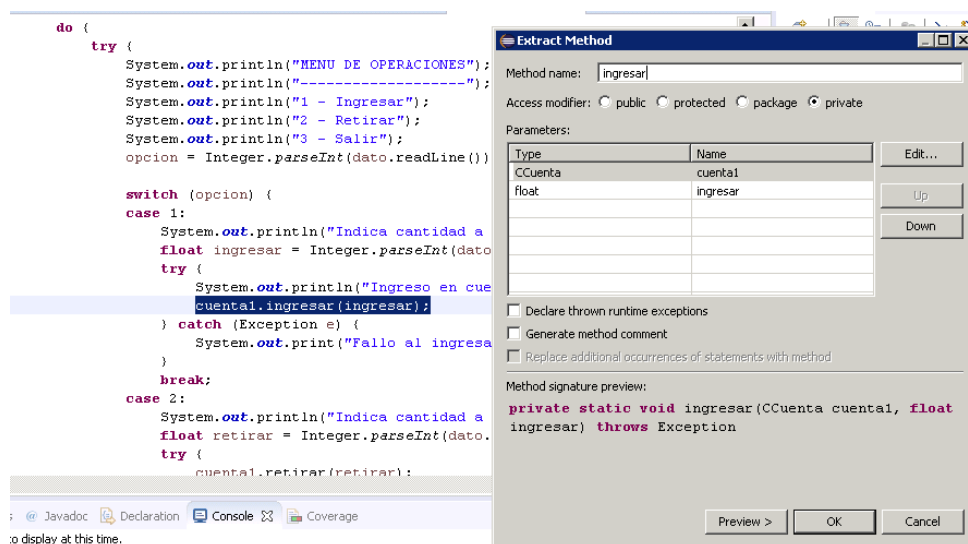
Método mostrarMenu:



Y este es el resultado:



Método ingresar:



Y este es el resultado:

```
18         do {
19             try {
20                 System.out.println("MENU DE OPERACIONES");
21                 System.out.println("-----");
22                 System.out.println("1 - Ingresar");
23                 System.out.println("2 - Retirar");
24                 System.out.println("3 - Salir");
25                 opcion = Integer.parseInt(dato.readLine());
26
27                 switch (opcion) {
28                     case 1:
29                         System.out.println("Indica cantidad a ingresar: ");
30                         float ingresar = Integer.parseInt(dato.readLine());
31                         try {
32                             System.out.println("Ingreso en cuenta");
33                             ingresar(cuenta1, ingresar);
34                         } catch (Exception e) {
35                             System.out.print("Fallo al ingresar");
36                         }
37                         break;
38                     case 2:
39                         System.out.println("Indica cantidad a retirar: ");
40                         float retirar = Integer.parseInt(dato.readLine());
41                         try {
42                             cuenta1.retirar(retirar);
```

Método retirar:

```
4         case 1:
5             System.out.println("Indica cantidad a ingresar");
6             float ingresar = Integer.parseInt(dato.readLine());
7             try {
8                 System.out.println("Ingreso en cuenta");
9                 ingresar(cuenta1, ingresar);
10            } catch (Exception e) {
11                System.out.print("Fallo al ingresar");
12            }
13            break;
14        case 2:
15            System.out.println("Indica cantidad a retirar:");
16            float retirar = Integer.parseInt(dato.readLine());
17            try {
18                cuenta1.retirar(retirar);
19            } catch (Exception e) {
20                System.out.print("Fallo al retirar");
21            }
22            break;
23        case 3:
24            System.out.println("Aaaaaaaaadios");
25        }
26    }
27
28    private static int mostrarMenu(BufferedReader dato) throws
```

Extract Method

Method name:

Access modifier: ☐ public ☐ protected ☐ package ☒ private

Parameters:

Type	Name	Edit...
CCuenta	cuenta1	
float	retirar	

☐ Declare thrown runtime exceptions
☐ Generate method comment
☐ Replace additional occurrences of statements with method

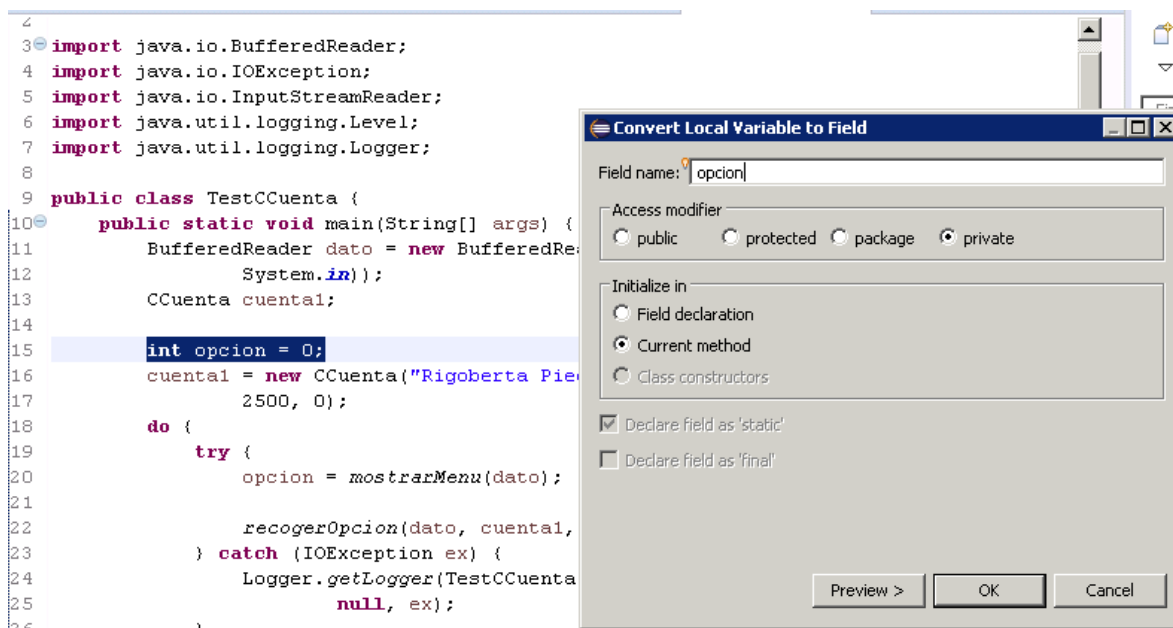
Method signature preview:
private static void extracted(CCuenta cuenta1, float retirar) **throws** Exception

Preview > OK Cancel

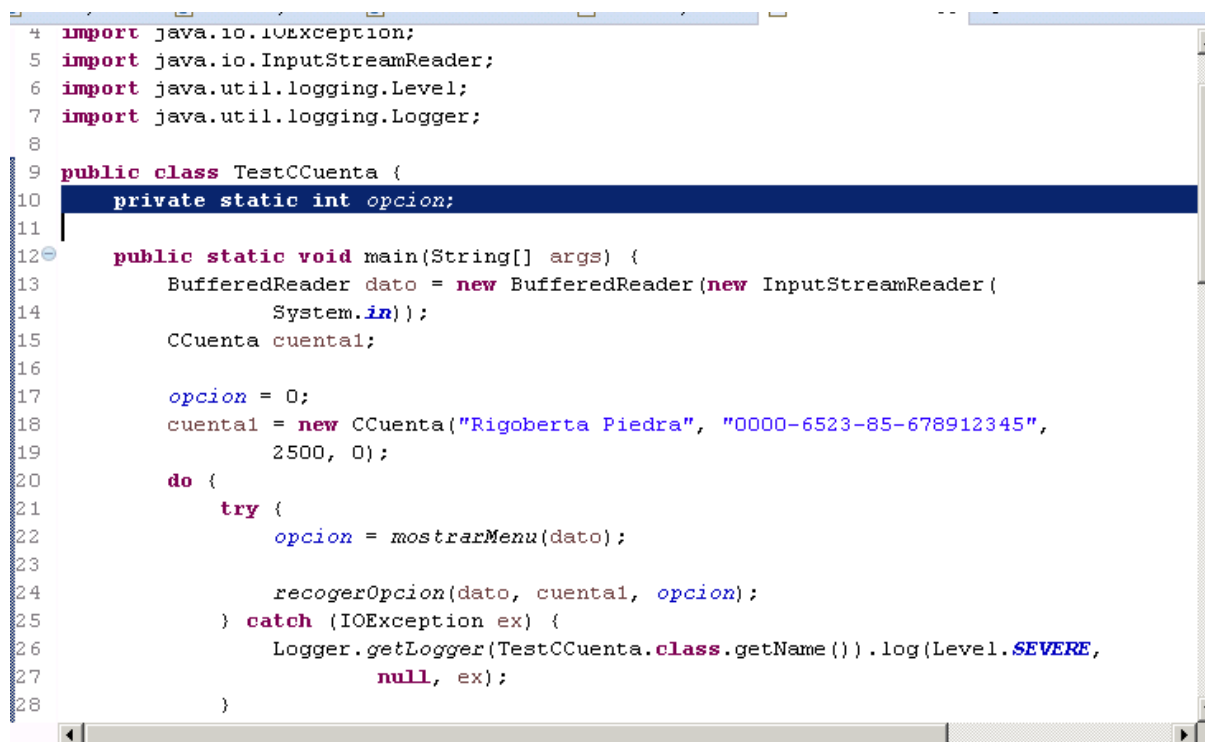
Y este es el resultado:

```
7
3 private static void retirar(CCuenta cuenta1, float retirar) throws Exception {
9     cuenta1.retirar(retirar);
1 }
1
```

2. TestCCuenta.java: Convierte dato en campo.

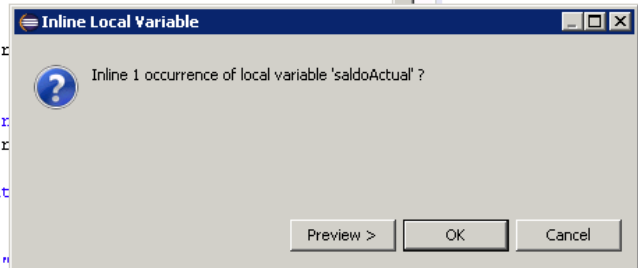


Y este es el resultado:



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

```
21     try {
22         opcion = mostrarMenu(dato);
23
24         recogerOpcion(dato, cuenta1, opcion);
25     } catch (IOException ex) {
26         Logger.getLogger(TestCCuenta.class.getName()).log(Level.SEVERE,
27             null, ex);
28     }
29 } while (opcion != 3);
30 double saldoActual = cuenta1.estado();
31 System.out.println("Saldo actual: " + saldoActual);
32 }
33
34 private static void recogerOpcion(BufferedReader dato, CCuenta cuenta1, int opcion) {
35     switch (opcion) {
36     case 1:
37         System.out.println("Indica cantidad a ingresar: ");
38         float ingresar = Integer.parseInt(dato.readLine());
39         try {
40             System.out.println("Ingreso en cuenta");
41             ingresar(cuenta1, ingresar);
42         } catch (Exception e) {
43             System.out.print("Fallo al ingresar");
44         }
45         break;
46     }
```



Y este es el resultado:

```
3     BufferedReader dato = new BufferedReader(new InputStreamReader(
4         System.in));
5     CCuenta cuenta1;
6
7     opcion = 0;
8     cuenta1 = new CCuenta("Rigoberta Piedra", "0000-6523-85-678912345",
9         2500, 0);
10    do {
11        try {
12            opcion = mostrarMenu(dato);
13
14            recogerOpcion(dato, cuenta1, opcion);
15        } catch (IOException ex) {
16            Logger.getLogger(TestCCuenta.class.getName()).log(Level.SEVERE,
17                null, ex);
18        }
19    } while (opcion != 3);
20    System.out.println("Saldo actual: " + cuenta1.estado());
21 }
22
23 private static void recogerOpcion(BufferedReader dato, CCuenta cuenta1, int opcion) {
24     switch (opcion) {
25     case 1:
26         System.out.println("Indica cantidad a ingresar: ");
27         float ingresar = Integer.parseInt(dato.readLine());
```

4. TestCCuenta.java: Renombra la variable local cuenta1 con tu nombre, por ejemplo.

```
0 private static int opcion;
1
2 public static void main(String[] args) {
3     BufferedReader dato = new BufferedReader(new InputStreamReader(
4         System.in));
5     CCuenta cuenta1;
6
7     opcion = 0;
8     cuenta1 = new CCuenta("Rigoberta Piedra", "0000-6523-85-678912345",
9         2500, 0);
10    do {
11        try {
12            opcion = mostrarMenu(dato);
13
14            recogerOpcion(dato, cuenta1, opcion);
15        } catch (IOException ex) {
16            Logger.getLogger(TestCCuenta.class.getName()).log(Level.SEVERE,
17                null, ex);
18        }
19    } while (opcion != 3);
20    System.out.println("Saldo actual: " + cuenta1.estado());
21 }
2
3 private static void recogerOpcion(BufferedReader dato, CCuenta cuenta1, int opcion
4     switch (opcion) {
```

Y este es el resultado:

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

5. Ccuenta.java: Cambia el nombre del método “ingresar” por “ingresando”.

```
13         tipoIngresos = tipo;
14     }
15
16     public double estado() {
17         return saldo;
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(saldo + 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         setSaldo(saldo - cantidad);
35     }
36
37     public void setSaldo(double saldo) {
```

Y este es el resultado:

```
CCuenta.java  TestCCuenta.java
13         tipoIngresos = tipo;
14     }
15
16     public double estado() {
17         return saldo;
18     }
19
20     public void ingresando(double cantidad) throws Exception {
21         if (cantidad < 0) {
22             throw new Exception("No se puede ingresar una cantidad negativa");
23         }
24         setSaldo(saldo + 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         setSaldo(saldo - cantidad);
35     }
36
37     public void setSaldo(double saldo) {
```