**Ejemplo: Agencia bancaria**

Se requiere realizar un programa orientado a objetos para manejar los clientes y cuentas de una agencia bancaria.

Cada cliente de la agencia es identificado por su DNI, nombre y apellido. Los clientes pueden abrir una o varias cuentas bancarias, además pueden solicitar un estado de sus cuentas. Estas cuentas tienen un número de cuentas pueden ser en soles o dólares y tienen un saldo. Las operaciones que el cliente puede realizar con las cuentas son: consultar el saldo (retorna el saldo actual), depositar un monto (agrega un monto y retorna el nuevo saldo) y retirar un monto (disminuye un monto y retornas el nuevo saldo) y transferir un monto a otra cuenta (que puede ser de otro cliente, retira el monto y retorna el nuevo saldo).

En el programa principal cree dos clientes: Juan Perez y Manuel Fernandez con las siguientes cuentas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cliente** | **Cuenta** | **Tipo** | **Saldo Inicial** | **Saldo final** |
| Juan Perez | S0001 | Soles | 500 soles | 400 |
|  | D0001 | Dólares | 200 dólares | 150 |
| Manuel Fernandez | S0002 | Soles | 100 soles | 300 |

Imprima por pantalla los estados de cuentas para ambos clientes.

Para Juan Perez:

Realizar un retiro de 50 dólares en la cuenta D0001.

Realizar un depósito de 100 soles en la cuenta S0001.

Realizar una transferencia de la S0001 a la cuenta S0002 de 200 soles.

Imprima por pantalla nuevamente los estados de cuentas para ambos clientes (ver columna Saldo final).

Aplicar prueba unitaria a la operación de transferir un monto a otra cuenta.

**Puntaje:**

Clases correctamente definidas: 4 puntos

Relación entre clases: 2 puntos

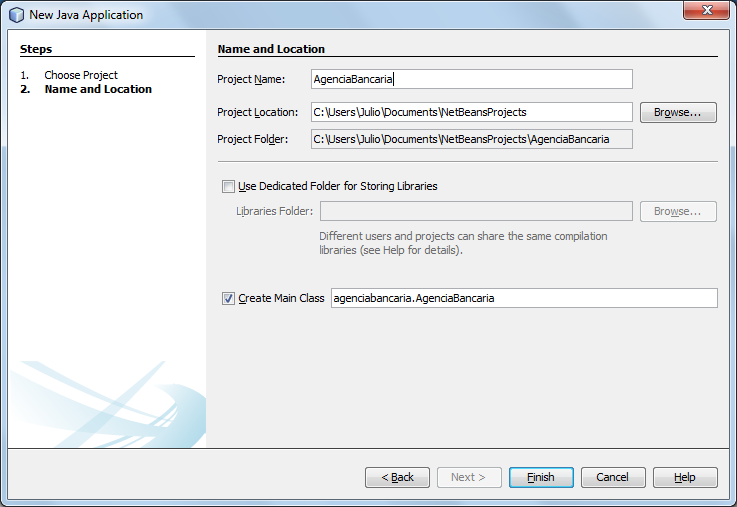
Programa de Agencia correctamente elaborado: 3 puntos

Prueba unitaria a método transferir monto: 1 punto

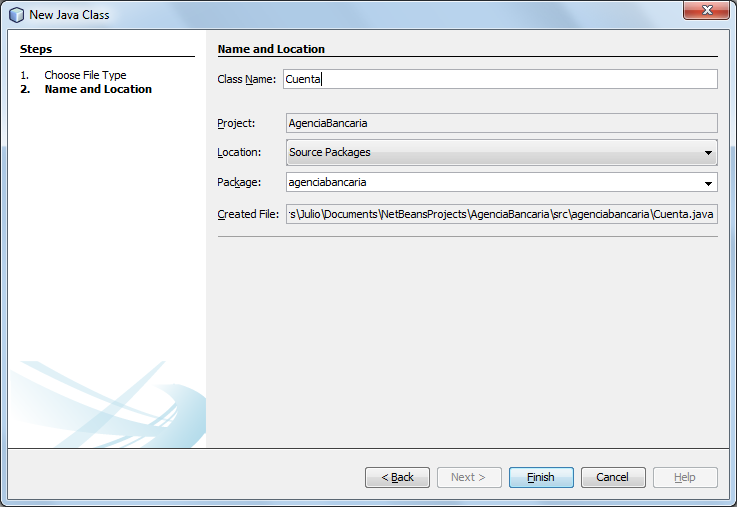
**Desarrollo**

Lo primero es identificar las clases: Estas son AgenciaBancaria (donde estará el programa principal, método main), Cliente y Cuenta.

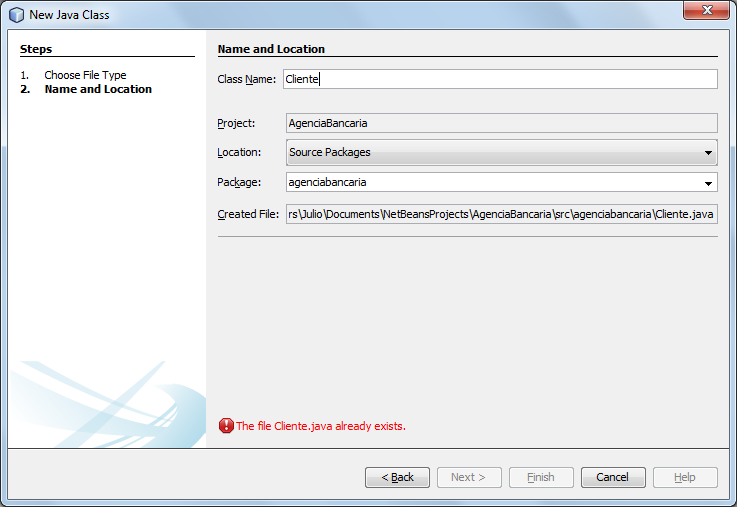
A continuación en NetBeans creamos el proyecto AgenciaBancaria.



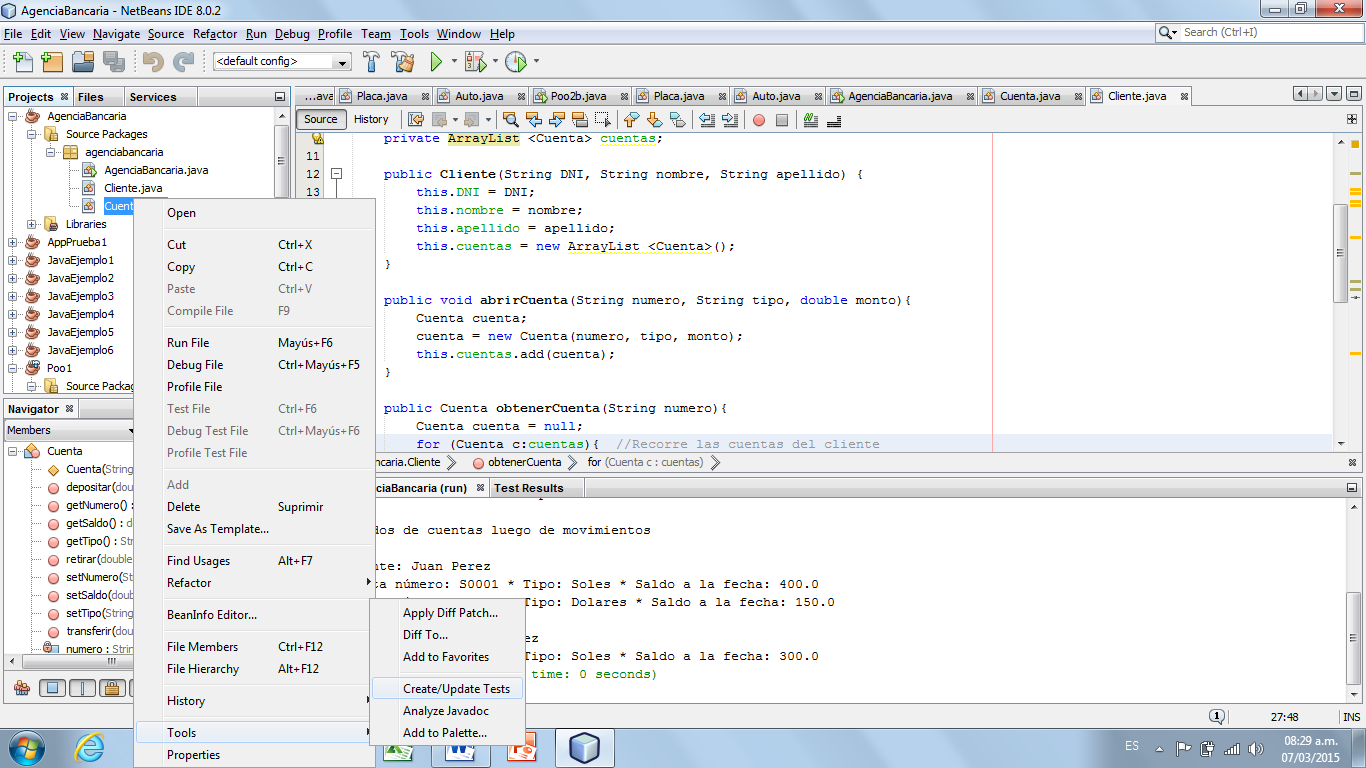
Ahora iniciamos creando la clase Cuenta, con los atributos numero, tipo (soles o dólares), saldo. Y definimos las operaciones Cuenta(constructor), depositar(monto), retirar(monto) y transferir(monto, cuentadestino).

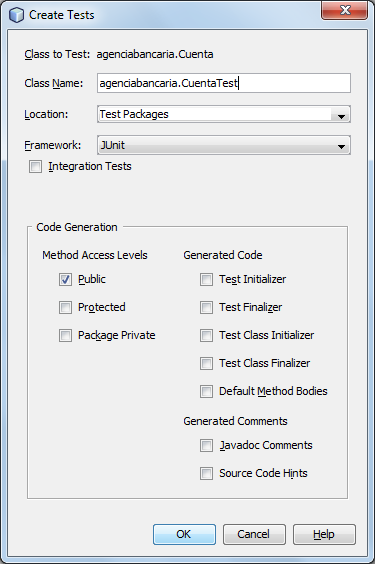


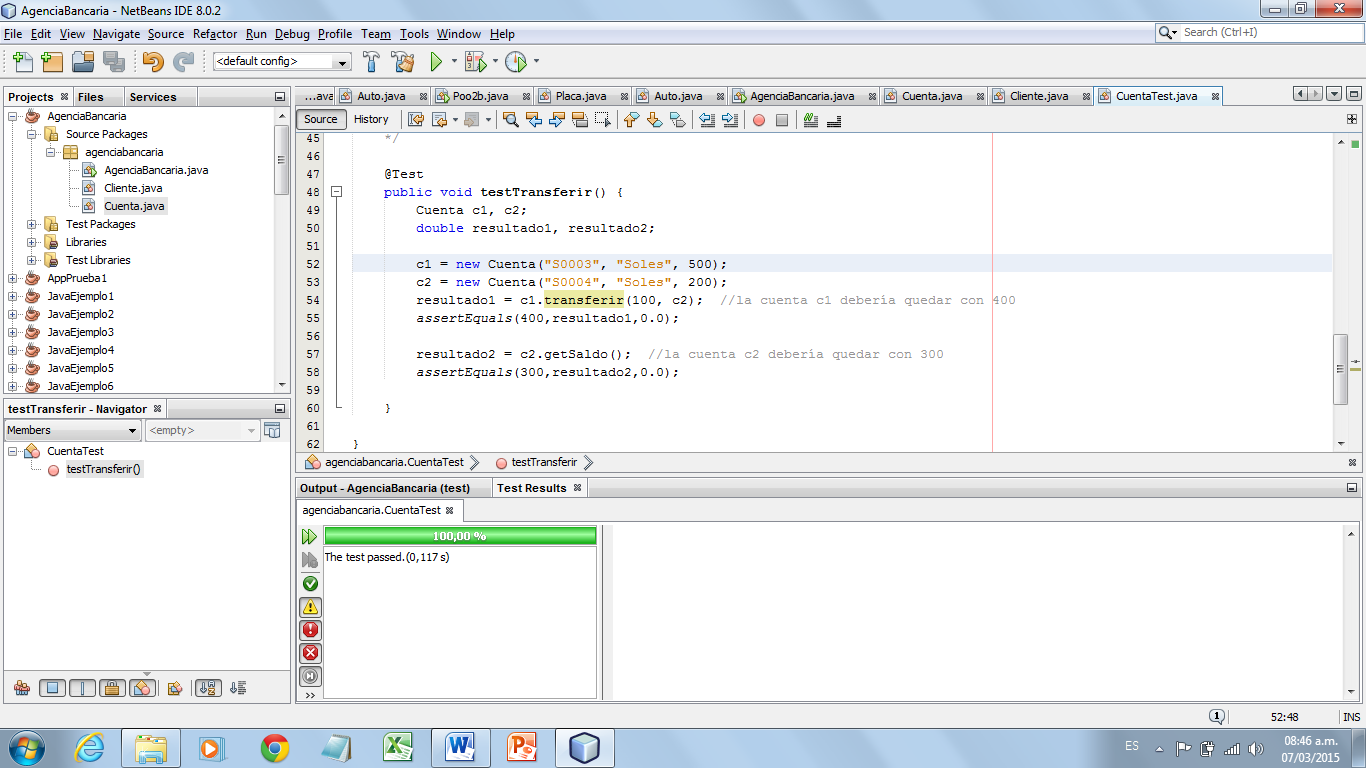
Luego creamos la clase Cliente, con los atributos que puede tener un ArrayList de cuentas. Los atributos de la clase Cliente serían DNI de tipo String, nombre de tipo String, apellido de tipo String y un ArrayList de cuentas de tipo Cuenta. Las operaciones de esta clase serían abrirCuenta, emitirEstadoCuentas y y obtenerCuenta que devolverá una referencia a la cuenta de un cliente..



Prueba unitaria a método transferir:







Ver fuentes del proyecto: AgenciaBancaria.zip