

Lezione 12 – Esercitazione

prof. Marcello Sette

<mailto://marcello.sette@gmail.com>

<http://sette.dnsalias.org>

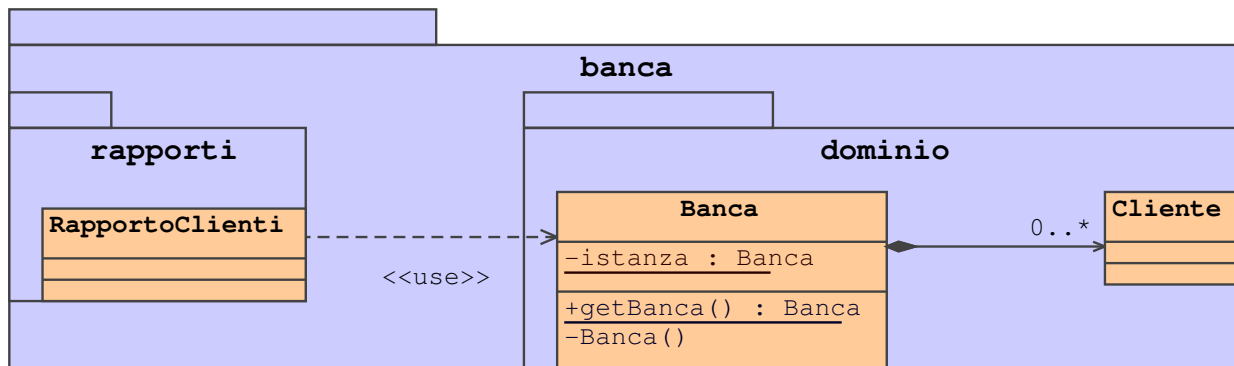
Esercizio 1

Obiettivo

In questo esercizio verrà modificata la classe `Banca` in modo che realizzi un Singoletto.

Nota: a questo punto il progetto ha bisogno di una gerarchia di pacchetti più articolata, poiché dovremo creare una classe `RapportoClienti` che appartiene al pacchetto `banca.rapporti`; inoltre, le classi del dominio precedente dovranno essere poste nel pacchetto `banca.dominio` e si dovranno modificare le dichiarazioni di pacchetto in ognuno dei file precedenti.

Traccia



1. Nei precedenti esercizi, i rapporti di stampa erano immersi nel metodo `main` della classe `TestBanca`. In questo esercizio il codice è stato estratto in una classe `RapportoClienti` posta nel pacchetto `banca.rapporti`.

Posizionarsi nel pacchetto di default fornito dal docente. In esso è presente una nuova versione della classe `TestBanca` ed una cartella `banca`.

Nella cartella `banca` è presente una sottocartella `rapporti`, nella quale è data una nuova classe `RapportoClienti`.

2. Rinominare la propria cartella `banca` (così come si trovava alla fine dell'esercizio precedente) in `dominio` e ricopiarla nella cartella `banca` fornita dal docente.

3. Correggere in tutte le classi della cartella `dominio` il nome del pacchetto a cui esse appartengono (ora è `banca.dominio`).
4. Modificare la classe `Banca` in modo da realizzare un Singoletto come rappresentato nel precedente diagramma UML.

Test del codice

Compilare ed eseguire `TestBanca`. L'output dovrebbe essere:

```
RAPPORTO CLIENTI
=====
```

```
Cliente: Carla Rossi
```

```
Saldo del libretto di risparmio: 500.0
```

```
Saldo del conto corrente: 200.0
```

```
Cliente: Anna Bruni
```

```
Saldo del conto corrente: 200.0
```

```
Cliente: Raul Falchi
```

```
Saldo del conto corrente: 200.0
```

```
Saldo del libretto di risparmio: 1500.0
```

```
Cliente: Vale Bova
```

```
Saldo del conto corrente: 200.0
```

```
Saldo del libretto di risparmio: 150.0
```

Esercizio 2

È possibile modificare la visibilità `protected` di `saldo` nella classe `Conto` in modo che sia nuovamente `private` e ripristinare così l'incapsulazione perfetta?

La risposta è sì. Ma questo ha conseguenze nel codice della classe `ContoCorrente`.

In particolare:

1. poiché in un `ContoCorrente` il saldo può essere negativo, mentre nella superclasse esso è imposto non-negativo, occorre aggiungere in `ContoCorrente` un attributo che tenga conto della parte di saldo in rosso;
2. non è sufficiente sovrapporre (e riscrivere) solo il metodo `preleva`, ma è necessario sovrapporre anche `getSaldo` e `deposita` in modo da tenere conto sia della parte positiva (quella invisibile della superclasse), sia della parte negativa (propria della classe).

Esibire le necessarie (ed eleganti) modifiche nella classe `ContoCorrente` che rendano nuovamente il programma funzionante.