

# Fondamenti di Ingegneria del Software A.A. 2019-20

## Laboratorio su JUnit

Scaricare da Aulaweb il progetto Java **ATM 1.0\_original** e importarlo in Eclipse (Import -> Existing Projects into Workspace). Il progetto simula un bancomat ATM (Automated Teller Machine) e tratta le seguenti transazioni: **prelievo di contanti** (classe *Withdrawal*), **deposito di contanti** (classe *Deposit*) e **visualizzazione del saldo di un conto corrente** (classe *BalanceInquiry*). Il progetto utilizza ulteriori classi che simulano la tastiera per l'inserimento dei dati da parte dell'utente (classe *Keypad*), lo schermo per la visualizzazione dei messaggi inviati all'utente (classe *Screen*), il database della banca (classe *BankDatabase*), i conti correnti (classe *Account*), il distributore di contanti per i prelievi (classe *CashDispenser*) e la cassetta in cui depositare i contanti (classe *DepositSlot*).

Avviando il *main()* della classe *ATMCaseStudy*, viene richiesto l'inserimento del **numero di conto** e del **PIN**. Per semplicità, si possono usare le seguenti coppie:

- Numero di conto = 12345, PIN = 54321
- Numero di conto = 98765, PIN = 56789
- 

Scopo dell'esercitazione è sostituire la rappresentazione della **valuta** nelle varie transazioni, da tipo primitivo *double* a classe *Euro*. È pertanto richiesto di applicare la procedura seguente:

1. Dopo aver preso confidenza con il codice fornito ed averlo eseguito
  - a. Creare due package chiamati **code** e **test**;
  - b. Applicare il refactoring 'move' e spostare tutte le classi del progetto ATM nel package **code**;
  - c. Aggiungere al package **code** la classe *Euro*, presente a **pagina 2 del testo**;
2. Implementare alcuni casi di test JUnit (usare JUnit 5) per testare **tutti** i metodi della classe *Euro*
  - a. È richiesto di creare una classe di test *TestEuro* contenente i metodi di test organizzati e annotati in modo opportuno (*@Test*, *@ParameterizedTest*, *@BeforeEach*, etc);
  - b. Posizionare la classe *TestEuro* nel package test, separato da quello del codice che si intende testare, e utilizzare le asserzioni *assertTrue* e/o *assertEquals* (quelle base fornite dal framework JUnit 5)
  - c. Se volete sperimentare i test parametrici (non è richiesto comunque nell'esercizio) potere fare riferimento a <https://www.baeldung.com/parameterized-tests-junit-5>
3. Sostituire in tutto il codice del progetto ATM il tipo primitivo *double* con la classe *Euro* introdotta e fare in modo che la classe *Screen* visualizzi la valuta correttamente in euro
  - a. Si suggerisce di iniziare la modifica a partire dalla classe *Account* e poi propagarla al resto del codice, prestando particolare attenzione ai riferimenti al saldo di un conto corrente (*availableBalance* e *totalBalance*) e agli importi di prelievo/deposito (*amount*). Verificare, eseguendo l'applicazione, che la modifica sia stata effettuata in modo corretto.
4. Implementare alcuni casi di test JUnit per testare i metodi delle classi *Account* e *BankDatabase* così modificate, creando opportune classi di test come indicato al punto 2 per testare la classe *Euro*.
  - a. Posizionare i nuovi casi di test nel package **test**

Consegnare uno zip chiamato **Labo4\_COGNOME1[\_COGNOMEn]** contenente il progetto ATM modificato e corredato dai casi di test implementati (in Eclipse, Export → Archive File). Verificare che il codice sia funzionante, ben organizzato e limitato alle modifiche richieste nel testo.

```
public class Euro {  
    private long valore;  
  
    public Euro(long euro, long cent) {  
        if (euro >= 0) {  
            valore = euro*100 + cent;  
        } else {  
            valore = euro*100 - cent;  
        }  
    }  
  
    public Euro(double d) {  
        valore = (long)(d*100);  
    }  
  
    public long getValore() {  
        return valore;  
    }  
  
    public Euro somma(Euro e) {  
        this.valore = this.valore + e.getValore();  
        return this;  
    }  
  
    public Euro sottrai(Euro e) {  
        this.valore = this.valore - e.getValore();  
        return this;  
    }  
  
    public boolean ugualeA(Euro e){  
        if (valore == e.getValore())  
            return true;  
        else return false;  
    }  
  
    public boolean minoreDi(Euro e){  
        if (valore <= e.getValore())  
            return true;  
        else return false;  
    }  
  
    public String stampa(){  
        return (double)valore/100 +" euro";  
    }  
}
```