

- Scrivere immediatamente, su ogni foglio che vi è stato consegnato, cognome, nome, numero di matricola.
- Non è consentito consultare appunti, libri, colleghi, né qualunque dispositivo elettronico, pena l'immediato annullamento della prova.
- Tempo a disposizione: 3 ore.

### Esercizio 1

Una casa automobilistica vuole realizzare un sistema per supportare la personalizzazione delle automobili presso i propri concessionari. Il sistema permette agli addetti della casa automobilistica di specificare, per ciascun modello di automobile in commercio, tutte le opzioni disponibili. In particolare, un certo modello di automobile può prevedere diverse possibili motorizzazioni (sia elettriche, che diesel, che benzina), diversi colori, e diversi pacchetti di optional. Ciascuna opzione è caratterizzata da un prezzo (in €) e da una descrizione testuale. Inoltre, le opzioni di motorizzazione elettriche sono caratterizzate da un'autonomia media, mentre quelle termiche da un consumo medio. Il venditore, che lavora presso un concessionario, previa autenticazione, può configurare l'auto di un cliente selezionando un modello, una motorizzazione, un colore, e uno o più pacchetti di optional tra quelli disponibili. Alla conferma di queste informazioni, il sistema calcola il prezzo totale del veicolo sommando al prezzo base del modello il prezzo delle opzioni selezionate. Inoltre, utilizzando le API del servizio esterno FastFinance, il sistema mostra anche il prezzo stimato di una rata, qualora il cliente voglia acquistare a rate mensili. Infine, nel caso il cliente accetti il prezzo proposto, l'addetto può inviare l'ordine alla casa madre per la produzione.

- Dettagliare il caso d'uso relativo alla funzionalità di configurazione di un mezzo e invio dell'ordine per mezzo di mock-up e descrizioni testuali strutturate secondo il formalismo di Cockburn. Usare la propria conoscenza del dominio per derivare dettagli non definiti nei requisiti.
- Definire un class diagram di analisi del sistema, inteso come modello di dominio, relativo al caso d'uso della configurazione di un veicolo e invio dell'ordine. È possibile rifarsi alle euristiche *EBC*, e di *Abbott*.
- Fornire un sequence diagram di analisi per il caso d'uso relativo alla configurazione di un veicolo e invio dell'ordine.

### Esercizio 2

```
public class Ex1 {  
    public static void main(String[] args) {  
        Spectator a = new Spectator(), b = new Spectator();  
        Subject o = new Subject();  
        o.register(a); o.register(b);  
        o.update();  
    }  
}
```

```
public class Spectator {  
    public void update() {  
        System.out.println("Update detected");  
    }  
}
```

```
import java.util.HashSet;  
import java.util.Set;  
  
public class Subject {  
    Set<Spectator> s = new HashSet<Spectator>();  
    public void register(Spectator o) {  
        s.add(o);  
    }  
    public void update() {  
        for(Spectator o:s) {  
            o.update();  
        }  
    }  
}
```

Si considerino le classi Java riportate sopra.

- Si descriva con un sequence diagram un'invocazione del metodo `main` della classe `Ex1`.
- Nel codice d'esempio è presente un design pattern? Se sì, dire quale e descrivere uno scenario concreto, diverso da quello del codice dell'esercizio, in cui tale pattern può essere utilizzato.

### **Esercizio 3**

Il metodo `calcolaImposta` della classe `TaxUtils` calcola le imposte dovute in un sistema a tassazione progressiva con due scaglioni. Il metodo prende in input quattro parametri:

- `imponibile`: un float non negativo indicante la base imponibile, in Euro.
- `soglia`: un float non negativo indicante la soglia di imponibile che determina il passaggio dal primo al secondo scaglione.
- `aliquota1`: un float compreso tra 0 e 1, estremi esclusi. Indica l'aliquota di tassazione per il primo scaglione.
- `aliquota2`: un float compreso tra 0 e 1, estremi esclusi. Indica l'aliquota di tassazione per il secondo scaglione.

Per esempio, se l'imponibile è 1000€, la soglia è fissata a 700€, e le aliquote sono rispettivamente al 10% e al 30%, l'imposta dovuta sarà calcolata come segue. Per la parte di imponibile rientrante nel primo scaglione, ovvero non superiore alla soglia, si applica l'aliquota del 10%, corrispondente a 70€. Per la parte di imponibile che rientra nel secondo scaglione, ovvero i restanti 300€, si applica la seconda aliquota al 30%, corrispondente a 100€ di imposta. In totale, le imposte nell'esempio ammontano a  $70€ + 100€ = 170€$ .

Se uno dei parametri non è valido, il metodo solleva una `TaxException`.

- a) Indicare, per ciascuno dei parametri del metodo `calcolaImposta`, le classi di equivalenza individuate.
- b) Scrivere quattro test JUnit con strategia Black Box per il metodo `calcolaImposta`, indicando per ciascuno di essi quali classi di equivalenza copre. Si richiede inoltre che un test corrisponda a scenari in cui i parametri non sono validi, e che i restanti tre corrispondano a scenari in cui i parametri sono validi.
- c) Quanti test sono necessari per testare il metodo con strategia SECT?

### **Esercizio 4**

Fornire 5 requisiti funzionali e 5 requisiti non funzionali per una app mobile per la somministrazione di esami a risposta multipla/aperta per corsi universitari.