

- 
- 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.
  - Gli esercizi relativi al modulo A devono essere svolti su fogli differenti rispetto a quelli del modulo B.
  - Tempo a disposizione: 3 ore.
- 

#### **Modulo A - Esercizio 1**

In scenari di Cloud Computing, descrivere, in al più una pagina, le differenze tra i modelli di servizio IaaS, PaaS e SaaS, evidenziandone vantaggi e svantaggi.

#### **Modulo A - Esercizio 2**

Il metodo `classify` della classe `IrisClassification` viene utilizzato per classificare esemplari di piante del genere *Iris* nelle specie *versicolor*, *setosa* e *virginica*. Il metodo prende in input due parametri:

- `float petal_len` è numero reale positivo rappresentante la lunghezza del petalo del fiore, in millimetri;
- `float petal_width` è un numero reale positivo rappresentante la larghezza del petalo, in millimetri.

In accordo con gli studi di Ronald Fisher, il metodo è implementato come da frammento di codice riportato di seguito.

```
public class IrisClassification {  
    static String classify(float petal_len, float petal_width) {  
        1 String prediction = "unknown";  
        2 if(petal_len <= 2) {  
            3     prediction = "setosa";  
        }  
        4 else {  
            if(petal_width > 2) {  
                prediction = "virginica";  
            }  
            else if(petal_width - petal_len <= 0.5) {  
                prediction = "versicolor";  
            }  
        }  
        return prediction;  
    }  
}
```

- i. Rappresentare il CFG del metodo;
- ii. Scrivere quattro test JUnit con strategia White Box per il metodo `classify`, indicando per ciascuno di essi quale cammino copre nel CFG. Ove possibile, si richiede che i test JUnit coprano cammini distinti nel CFG.

#### **Modulo A - Esercizio 3**

Fornire 5 requisiti funzionali e 5 requisiti non funzionali per una app di gestione programmi fedeltà (e.g.: raccolta punti, offerte speciali riservate ad alcuni clienti, coupon, etc..) di una catena di supermercati.

#### **Modulo B - Esercizio 4**

Si vuole realizzare un'applicazione che consente, a partire da uno o più ingredienti, di ricercare delle ricette. Per ogni ricetta si deve visualizzare la lista degli ingredienti, il tempo di cottura, il livello di difficoltà, numero di persone a cui si riferiscono le quantità espresse dalla ricetta e spesa media per persona. Una ricetta può essere segnalata come preferita e, in tal caso, questa potrà essere visualizzata in una schermata apposita. Si preveda inoltre di poter perfezionare la ricerca utilizzando appositi filtri per livello di difficoltà, tempo di preparazione e spesa media. Infine, l'utente, solo se autenticato, potrà lasciare un commento alle ricette.

Si realizzino i mockup per l'applicazione di cui sopra.

#### **Modulo B - Esercizio 5**

A partire dai mockup, realizzare gli statechart di ricerca e commento delle ricette e di visualizzazione dei preferiti con rimozione di una ricetta.

#### **Modulo B - Esercizio 6**

In relazione all'applicazione progettata dei punti precedenti Definire un piano per valutare l'usabilità del prodotto sopra indicato in due fasi della progettazione, la prima, astratta, basata su prototipi e la seconda in betatesting sul campo, simulando la presenza in app di un sistema di monitoraggio in background.