

**Istruzioni:** Scrivere immediatamente, su ogni foglio che vi è stato consegnato, cognome, nome, numero di matricola. Durante la prova, non è consentito consultare materiale didattico, né interagire in alcun modo con dispositivi elettronici o colleghi, pena l'immediato annullamento della prova. Il tempo a disposizione per completare la prova è di **3 ore**.

### Esercizio 1

Si vuole progettare un software di supporto alla gestione di un centro culturale comunale non-profit e self-service. Il sistema permette a un amministratore di gestire i libri presenti in inventario, e ai cittadini di prendere in prestito libri, interagendo con i totem presenti in sede. In particolare, previa autenticazione con SPID, un cittadino può visualizzare i libri disponibili, effettuando ricerche in base a titolo, genere e autore. Una volta trovato un libro di interesse, un cittadino lo può prendere in prestito. Dopo aver selezionato l'oggetto di interesse, l'utente indica una data di termine del prestito (che non può eccedere i 30 giorni), entro la quale si impegna a rendere il libro, e conferma la richiesta di prestito. Un utente non può prendere in prestito più di cinque oggetti alla volta. In seguito alla conferma, il sistema utilizza API messe a disposizione dal componente robotizzato “InventoryRetriever”, sviluppato da una terza parte, per recuperare l'oggetto di interesse dal magazzino e consegnarlo automaticamente in un apposito vano predisposto in sede, dove l'utente potrà ritirarlo. Un utente può, inoltre, restituire un libro preso in prestito. A tal fine, l'utente seleziona, tra gli al più cinque prestiti correntemente attivi, il libro che vuole restituire. Dopodiché, inserisce il libro nell'apposito vano per le restituzioni, e conferma l'operazione. Il sistema “InventoryRetriever” verificherà che il libro restituito sia quello giusto, e lo riporrà in magazzino.

- Si modellino tutti i requisiti del sistema descritto sopra utilizzando uno Use Case Diagram;
- Realizzare i mock-up dell'applicazione descritta, relativamente alla funzionalità di prestito libri.
- Dettagliare il caso d'uso relativo alla funzionalità di prestito libri, per mezzo descrizioni testuali strutturate secondo un formalismo tabellare come quello proposto da A. Cockburn. Usare la propria conoscenza del dominio per derivare dettagli non definiti nei requisiti.

### Esercizio 2

Una stampante, una volta accesa, rimane in attesa dell'invio di documenti da stampare. In presenza di un documento da stampare, la stampante rifiuta tutte le successive richieste di stampa finché non ha ultimato la stampa corrente. In ogni momento, in presenza di un apposito segnale inviato dal produttore della stampante via Internet, la stampante può interrompere le sue operazioni correnti per scaricare e poi installare un aggiornamento. Al termine dell'installazione dell'aggiornamento, le attività eventualmente interrotte riprendono. Durante le operazioni di aggiornamento, inoltre, la stampante può essere spenta. Una operazione di stampa, invece, può essere annullata e interrotta bruscamente premendo un apposito pulsante. Una stampa può anche essere spenta

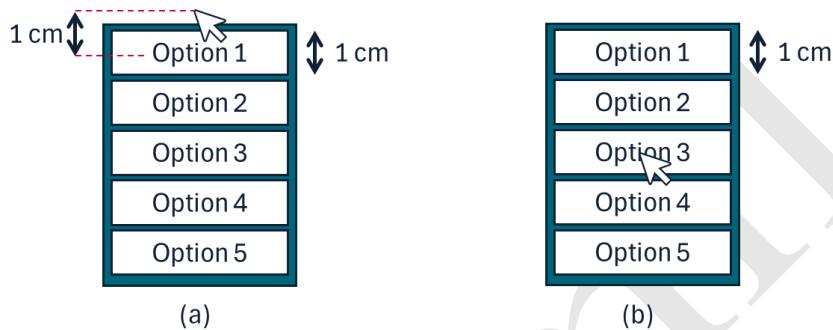
Si rappresenti il comportamento della stampante sopra descritta utilizzando il formalismo degli StateChart. Si richiede esplicitamente che la modellazione sia gerarchica e che siano utilizzati history state.

### Esercizio 3

Spiegare, nell'ambito del Software Design, che cosa si intende per “coesione”, descrivendo in che modo questa proprietà influenza i processi di sviluppo software. Fornire un esempio concreto di una classe che presenta bassa coesione, nel dominio di una clinica veterinaria.

### Esercizio 4

Nell'azienda dove lavorate come Ingegneri del Software, due vostri colleghi hanno proposto due design alternativi per il menù contestuale da visualizzare quando viene effettuato il click destro del mouse all'interno di un prodotto software che sarà utilizzato in contesti safety-critical, in cui i tempi di esecuzione dei task hanno importanza cruciale. I due design alternativi sono raffigurati nella figura seguente, e differiscono soltanto per il posizionamento del menù contestuale rispetto al puntatore del mouse. Nella variante (a), il menù è posizionato al di sotto del puntatore, mentre nella variante (b) il menù viene visualizzato in modo tale che il puntatore si trovi esattamente al centro, sull'Opzione 3. In qualità di esperti di HCI, vi viene chiesto di valutare quale dei due design permetterebbe di minimizzare i tempi di selezione dell'opzione desiderata, motivando adeguatamente la risposta. Si assuma che tutte le opzioni siano equiprobabili.



### Esercizio 5

Si consideri il seguente frammento di codice.

```

1. public class Customer {
2.
3.     private String name;
4.     private Address address;
5.
6.     /* constructors, getter and setters omitted for brevity */
7. }
8.
9. public class Address {
10.
11.    private String street;
12.    private String city;
13.    private String postalCode;
14.
15.    /* constructors, getter and setters omitted for brevity */
16. }
17.
18. public class Order {
19.
20.     public void printShippingLabel(Customer customer) {
21.         Address address = customer.getAddress();
22.         System.out.println("Shipping to: " + customer.getName());
23.         System.out.println("Street: " + address.getStreet());
24.         System.out.println("City: " + address.getCity());
25.         System.out.println("Postal Code: " + address.getPostalCode());
26.     }
27. }
```

Sono presenti code smell? Se sì, descrivere i code smell presenti, il loro impatto sulla qualità del codice, e indicare un possibile refactoring per risolverli.