

SAPIENZA Università di Roma
Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corsi di Laurea in Ingegneria Informatica ed Automatica
Corso di Progettazione del Software
Esame del **7 aprile 2022**
Tempo a disposizione: 2 ore e mezza

Requisiti. L'applicazione da progettare riguarda la gestione di un pizzeria express. Sono naturalmente di interesse i tipi di pizza, ciascuna con un nome, una descrizione, ed il prezzo. Poi sono di interesse gli ordini, ciascuno con il proprio codice (una stringa), il cliente che l'ha effettuato e l'insieme dei diversi tipi di pizza (almeno 1) ciascuno con la relativa quantità (per esempio 2 pizze margherita, 1 capricciosa e 3 napoletana). Dei clienti interessa il nome, l'indirizzo (una stringa), e il numero di telefono (una stringa). Ogni cliente (per essere memorizzato nel sistema) deve aver effettuato almeno un ordine. Dato un ordine, deve essere possibile risalire al cliente che l'ha effettuato. Alcuni ordini sono prioritari e in tal caso è di interesse il sovrapprezzo richiesto. Un ordine può essere stato consegnato da un rider. Dei rider interessa il nome.

Siamo interessati al comportamento dell'ordine. Un ordine è inizialmente in *preparazione*. Quando riceve l'evento *partenza*, con payload un rider incaricato della consegna, passa allo stato *in consegna* e viene mandato una notifica al cliente dell'ordine stesso. Quando l'ordine viene *consegnato* passa nello stato *completato* inserendo, solo in quel momento, nel diagramma UML il rider che ha effettuato la consegna. Quando in completato non accetta più eventi.

Siamo interessati alla seguente attività che prende come parametro un insieme di ordini e un insieme di rider, verifica che il numero di rider sia pari al numero di ordini. In caso negativo, manda un segnale in output richiedendo l'aggiornamento e termina. In caso positivo procede contemporaneamente con le seguenti due sottoattività: *(i)* consegna degli ordini, e *(ii)* analisi. I dettagli di queste due sottoattività non interessano eccetto che la sottoattività di analisi restituisce un report (una stringa). Una volta che tali sottoattività sono completate, l'attività principale manda un segnale di output con il report calcolato dalla sottoattività di analisi e termina.

Domanda 1. Basandosi sui requisiti riportati sopra, effettuare la fase di analisi producendo lo schema concettuale in UML per l'applicazione, comprensivo del diagramma delle classi (inclusi vincoli non esprimibili in UML), diagramma stati e transizioni per la classe *Ordine*, diagramma delle attività, la segnatura delle attività complesse, delle attività atomiche e dei segnali di input/output. Motivare, qualora ce ne fosse bisogno, le scelte di progetto.

Domanda 2. Effettuare la fase di progetto, illustrando i prodotti rilevanti di tale fase e motivando, qualora ce ne fosse bisogno, le scelte effettuate. In particolare definire SOLO le responsabilità sulle associazioni del diagramma delle classi. Nella tabella, inserire anche il motivo di ognuna delle responsabilità usando le lettere M, O, R rispettivamente per molteplicità, operazioni e requisiti.

Domanda 3. Effettuare la fase di realizzazione, producendo un programma JAVA e motivando, qualora ce ne fosse bisogno, le scelte effettuate. È obbligatorio realizzare in JAVA solo i seguenti aspetti dello schema concettuale:

- La classe *Ordine* e le sue eventuali *sottoclassi* le classi che rappresentano *l'associazione* a responsabilità doppia fra *Ordine* e *Cliente* e la classe *OrdineFired*.
- *L'attività principale*.

Nota: Rispetto a quanto sopra specificato, gli eventuali studenti con DSA *(i)* NON devono produrre le segnature di cui alla Domanda 1, *(ii)* non devono realizzare eventuali sottoclassi della classe *Ordine*. Per evitare disguidi, è opportuno indicare esplicitamente DSA accanto al nome nel foglio consegnato.