

<p align="center">Corso di Ingegneria del Software Corso di Laurea in INGEGNERIA INFORMATICA Prova di Esame del 04/07/2013</p>	<p align="center">Allievo Cognome e Nome:</p> <p align="center">Matricola:</p>
---	--

Descrizione del Problema

- *Si vuole realizzare un sistema software per realizzare una gestione automatizzata di un'area di parcheggio per autovetture.*

Il parcheggio è composto da più settori (identificati con una lettera dell'alfabeto, es. A, B, C...), ognuno con una diversa capienza (ossia numero massimo di autovetture contenibili, espressa come un intero positivo) e destinato ad un tipo specifico di autovetture (ossia motocicli, automobili, o autocarri). Alcuni tipi di autovetture hanno più settori disponibili per il parcheggio rispetto ad altre. I posti di uno specifico settore hanno tutti la stessa tariffa oraria per la sosta, mentre tale tariffa può variare al variare del settore.

L'area di parcheggio è dotata di un unico varco di ingresso ed uno di uscita, presidiati da un responsabile del parcheggio che ha le responsabilità di controllare l'accesso, ricevere il pagamento per la sosta avvenuta, abilitare l'uscita dell'autovettura a seguito del pagamento.

Il parcheggio è anche dotato di un display che riporta per ogni categoria di vetture il numero di posti disponibili. Il display si aggiorna automaticamente dopo l'ingresso e l'uscita delle vetture.

Per accedere al parcheggio, qualora esso abbia ancora posti disponibili per quella tipologia di vettura, il conducente della vettura si presenta al varco di ingresso ed il responsabile registra l'inizio di una nuova sosta, assegnando uno dei posti disponibili in uno dei settori riservati alla tipologia di vettura che sta entrando. Contestualmente registra la targa della vettura, il settore ed il posto assegnato, la data e l'ora di inizio della sosta ed il numero di un ticket cartaceo rilasciato (il numero del ticket è un intero su 8 cifre).

Per effettuare l'uscita dal parcheggio, l'utente si presenta al varco di uscita e effettua il pagamento al responsabile presentando il ticket ricevuto all'ingresso. Il sistema permette al responsabile del parcheggio di calcolare l'importo dovuto (in base all'ora di inizio e alla tariffa del settore, rilevati attraverso il numero di ticket) e registrare l'avvenuto pagamento. Ai fini del pagamento, ogni frazione di ora di sosta viene comunque tariffata con l'intera tariffa oraria (ad es. una sosta di 1 ora e 30 minuti viene tariffata come una sosta di 2 ore). Contestualmente il sistema aggiorna lo spazio disponibile nel parcheggio ed il display.

Il sistema permette al responsabile del parcheggio di generare report periodici sugli incassi giornalieri e mensili di tutto il parcheggio, oppure per settori, ed inoltre i report sul numero di soste giornaliere e mensili per ciascun settore.

QUESITI

1. Si disegni il diagramma dei **casi d'uso** e quello delle **classi concettuali** (System Domain Model) per tale sistema, tenendo conto anche delle responsabilità da attribuire alle classi.
2. Si progetti una soluzione basata sul modello architetturale a più livelli (Multi-Layer) per la realizzazione di questo sistema. Al fine di descrivere la soluzione proposta, il candidato dovrà fornire:
 - 2.1) il Package diagram descrittivo dell'architettura del sistema contenente anche le classi di dettaglio necessarie per l'implementazione;
3. Si progettino inoltre i Sequence diagram di dettaglio per la realizzazione delle seguenti funzionalità:
 - a. **“Ingresso nel Parcheggio”** che, data la tipologia di vettura, ricerca il primo settore con un posto disponibile per essa, lo assegna alla vettura registrando la targa della vettura, il settore ed il posto assegnato, la data e l'ora di inizio della sosta ed il numero di un ticket cartaceo rilasciato.
 - b. **“calcolo incasso massimo mensile di un dato settore del parcheggio”** che dato il settore e un mese pregresso calcola l'incasso giornaliero massimo ottenuto per quel settore.

Seconda Parte della Prova

Data di Consegna: 7 luglio 2013, entro le ore 14.00, per e-mail al docente: fasolino@unina.it (specificando come Oggetto della Mail: Prova Pratica IS- nome studente)

3. Implementare in linguaggio Java le classi del sistema necessarie a realizzare le due funzionalità delle quali è richiesta la definizione tramite sequence diagrams.

Dovrà inoltre essere definita la struttura del database (modello E/R e relazionale) per l'implementazione della persistenza e dovrà essere riempito con dati sufficienti ad eseguire i casi di prova.

Lo studente dovrà consegnare:

- Tutti i file del progetto Java (comprese le eventuali librerie – è consigliato l'uso di Eclipse per la gestione dei file del progetto). Come nome del progetto si indichi il proprio Cognome seguito dal numero di matricola completo (ad esempio: ROSSI_N46000000)
- Il database. In dettaglio dovranno essere prodotti (e nominati anch'essi con il Cognome ed il numero di matricola):
 - o Un file mwb prodotto da MySQL Workbench, da cui si vede il modello Entità-Relazione;
 - o script .sql ottenuto con la funzionalità Data Export di MySQL Workbench, contenente i dati di riempimento del database eventualmente utilizzati nei test.

Lo studente deve tenere particolarmente in conto la portabilità del sistema realizzato e la completezza del materiale inviato. L'applicazione dovrebbe essere funzionante su una macchina target che abbia installate Java SDK, MySQL Workbench ed Eclipse, in versioni analoghe a quelle viste durante il corso.

4. Progettare un insieme di casi di test black box (usando la tecnica delle Classi di Equivalenza) per provare la funzionalità **Ingresso nel Parcheggio** tenendo conto anche delle diverse pre-condizioni per tale funzionalità.

Si usi lo schema seguente per definire i casi di test

TC	Descrizione Test Case	Classi di Equivalenza coperte	Precondizioni	Input	Output Attesi	Post-Condizioni Attese	Output Ottenuti	Post-Condizioni Ottenute	Esito

Lo studente dovrà consegnare:

1. Un file di testo (formato .DOC , .RTF o .XLS) contenente le classi di equivalenza e la tabella dei casi di test progettati.

N.B.: Si consegnino sia il file di progetto VPP (Visual Paradigm) contenente tutti i diagrammi UML ritenuti utili a descrivere la soluzione proposta, sia un file di testo che spieghi e giustifichi le scelte progettuali fatte. Entrambi i file dovranno essere nominati con il Cognome del candidato seguito dal numero di matricola completo (ad esempio: ROSSI_M63000000).

Seconda Parte della Prova

Data di Consegna: 7 luglio 2013, entro le ore 14.00, per e-mail al docente: fasolino@unina.it
(specificando come Oggetto della Mail: Prova Pratica PSSS- nome studente)

2. Eseguire la progettazione di dettaglio e l'implementazione con tecnologie Java dei casi d'uso **"Ingresso nel Parcheggio"** e **"Uscita dal parcheggio"** e "Pagamento Sosta"

Lo studente dovrà consegnare:

3. La nuova versione del file di progetto VPP contenente tutti i diagrammi UML ritenuti necessari a descrivere la progettazione di dettaglio eseguita;
- Tutti i file del progetto Java dell'implementazione (comprese le eventuali librerie – è consigliato l'uso di Eclipse per la gestione dei file del progetto). Come nome del progetto si indichi il proprio Cognome seguito dal numero di matricola completo (ad esempio: ROSSI_M63000000)
- Il database. In dettaglio dovranno essere prodotti (e nominati anch'essi con il Cognome ed il numero di matricola):
 - o Un file mwb prodotto da MySQL Workbench, da cui si vede il modello Entità-Relazione;
 - o script .sql ottenuto con la funzionalità Data Export di MySQL Workbench, contenente i dati di riempimento del database eventualmente utilizzati nei test.

Lo studente deve tenere particolarmente in conto la portabilità del sistema realizzato e la completezza del materiale inviato. L'applicazione dovrebbe essere funzionante su una macchina target che abbia installate Java SDK, MySQL Workbench ed Eclipse, in versioni analoghe a quelle viste durante il corso.