

**Corso di Ingegneria del Software – Secondo appello 17 luglio 2019**

*R. Gori, L. Semini - Dipartimento di Informatica, Università di Pisa, a.a. 2018/19*

<p>Nome _____</p> <p>Cognome _____</p> <p>Matricola _____ Corso _____</p> <p>Aula _____      Posizione nell'aula _____ →  <i>(senza contare i banchi vuoti)</i></p>	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td colspan="7">cattedra</td> </tr> <tr> <td></td> <td><b>A</b></td> <td><b>B</b></td> <td><b>C</b></td> <td><b>D</b></td> <td><b>E</b></td> <td>...</td> </tr> <tr> <td><b>1</b></td> <td>A1</td> <td>B1</td> <td>C1</td> <td>D1</td> <td>E1</td> <td>...</td> </tr> <tr> <td><b>2</b></td> <td>A2</td> <td>B2</td> <td>C2</td> <td>D2</td> <td>E2</td> <td>...</td> </tr> <tr> <td><b>3</b></td> <td>A3</td> <td>B3</td> <td>C3</td> <td>D3</td> <td>E3</td> <td>...</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> </tr> </table>	cattedra								<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	...	<b>1</b>	A1	B1	C1	D1	E1	...	<b>2</b>	A2	B2	C2	D2	E2	...	<b>3</b>	A3	B3	C3	D3	E3	...	...	...	...	...	...	...	...
cattedra																																											
	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	...																																					
<b>1</b>	A1	B1	C1	D1	E1	...																																					
<b>2</b>	A2	B2	C2	D2	E2	...																																					
<b>3</b>	A3	B3	C3	D3	E3	...																																					
...	...	...	...	...	...	...																																					

E' ora possibile, per l'utenza ordinaria, interrogare HoC da app su cellulare per sapere se ci sono posti liberi e, in caso, prenotarne uno. La prenotazione vale al massimo 20 minuti, dopo i quali scade. Al momento della prenotazione il sistema invia un codice, che l'utente inserirà in un tastierino numerico posto sulla macchinetta all'ingresso del parcheggio. L'inizio del parcheggio viene calcolato dal momento della prenotazione. Se l'utente non si presenta in tempo o se si dimentica di inserire il codice, gli vengono addebitati 2€. Al momento dell'installazione dell'app deve essere fornito un numero di carta di credito su cui eventualmente addebitare i 2€.

**Domanda 1.** Si dia un diagramma dei casi d'uso che specificano le nuove funzionalità. Per ognuno indicare: precondizione, post-condizione, attore principale, eventuali attori secondari.

**Domanda 2.** Si dia un diagramma d'attività che modelli il processo di prenotazione. Si abbia cura di descrivere il time out e di indicare partizioni e responsabilità.

— — —

Sono state individuate le seguenti componenti:

App Utente	Permette di interrogare il sistema per sapere se ci sono posti disponibili e permette di prenotare un posto
Interfaccia Utente	Realizza l'interfaccia utente presso la macchinetta di ingresso al parcheggio
Gestione Utenza Oraria	Realizza la business logic del sistema di assegnazione dei posti all'utenza oraria e gestisce il servizio di prenotazione del posto.
DB Posti	Mantiene le informazioni sui box del parcheggio.

**Domanda 3.** Dare la vista C&C del sottosistema sopra descritto e dare il diagramma di struttura composita che descrive la componente Gestione Utenza Oraria. A tal fine separare la parte che realizza la logica di assegnamento del posto dalla parte che realizza la logica delle prenotazioni.

**Domanda 4.** Descrivere, usando una vista strutturale, le relazioni (d'uso e di generalizzazione) tra le parti e le interfacce introdotte all'esercizio precedente per descrivere la struttura interna di Gestione Utenza Oraria.

**Domanda 5.** Gli abbonamenti mensili per una macchina costano:

di tipo A → 120 €

di tipo B → 180 €

di tipo C → 100 €

Gli abbonamenti trimestrali prevedono uno sconto del 10% rispetto a quelli mensili, quelli semestrali uno sconto del 18%. Inoltre, è previsto uno sconto del 5% per auto di peso inferiore ai 1000kg, e una maggiorazione del 12% per quelle di peso superiore ai 2000 kg..

a) Il metodo:

```
calcolaCosto(char tipo, int numeroMesi, int peso)
```

Restituisce una eccezione per dati non validi, altrimenti calcola il costo dell'abbonamento.

Definire un test combinatorio, riducendo il numero di casi usando la **tecnica del pairwise testing** e i **vincoli di errore**.

b) E' stata proposta la seguente implementazione di `calcolaCosto`:

```
1) int calcolaCosto(char tipo, int numeroMesi, int peso){
2) int costo=0;
3) switch (tipo) {
4) case "A": costo = 120; break;
5) case "B": costo = 180; break;
6) case "C": costo = 100; break;
7) default : throw new IllegalParameter();
8) }
9) switch (numeroMesi) {
10) case "1": break;
11) case "3": costo = costo * 3 * 0,9; break;
12) case "6": costo = costo * 6 * 0,82; break;
13) default : throw new IllegalParameter();
14) }
15) if (peso < 1000)
16)     costo = costo * 0,9;
17) if (peso > 2000)
18)     costo = costo * 1,2;
19) return costo;
20) }
```

Individuare un mutante che viene ucciso dal caso di test (<C,3,1000>, 486).

Individuare un mutante che viene ucciso dal caso di test (<A,6,1500>, 590).