

Corso di Ingegneria del Software – Seconda Verifica Intermedia 1 Giugno 2018

V. Gervasi, L. Semini - Dipartimento di Informatica, Università di Pisa, a.a. 2017/18

Nome _____		<table border="1"><thead><tr><th colspan="7">cattedra</th></tr><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>...</th></tr></thead><tbody><tr><th>1</th><td>A1</td><td>B1</td><td>C1</td><td>D1</td><td>E1</td><td>...</td></tr><tr><th>2</th><td>A2</td><td>B2</td><td>C2</td><td>D2</td><td>E2</td><td>...</td></tr><tr><th>3</th><td>A3</td><td>B3</td><td>C3</td><td>D3</td><td>E3</td><td>...</td></tr><tr><th>...</th><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr></tbody></table>	cattedra								A	B	C	D	E	...	1	A1	B1	C1	D1	E1	...	2	A2	B2	C2	D2	E2	...	3	A3	B3	C3	D3	E3
cattedra																																												
	A		B	C	D	E	...																																					
1	A1		B1	C1	D1	E1	...																																					
2	A2		B2	C2	D2	E2	...																																					
3	A3	B3	C3	D3	E3	...																																						
...																																						
Cognome _____																																												
Matricola _____	Corso _____																																											
Aula _____	Posizione nell'aula _____ → (senza contare i banchi vuoti)																																											

Domanda 1. Con riferimento al sistema VASA, si dia un diagramma di sequenza a livello di analisi che descriva lo scambio di messaggi fra gli attori e i componenti hardware del sistema presenti nel seggio, corrispondenti all'espressione del voto da parte di un singolo elettore, dall'ingresso nei locali del seggio fino all'uscita. Si assuma, per questo esercizio, che tutti i controlli di legittimità (identificazione dell'elettore, integrità della scheda, ecc.) abbiano esito positivo.

Domanda 2. Il PC in dotazione al seggio, collegato in locale al lettore ottico (LO) e tramite rete sicura al server centrale (SC) del Ministero, ospita il software di gestione delle operazioni di voto. Tali operazioni includono l'uso del LO per riconoscere i QR code e controllare la validità delle schede votate, la trasmissione dei dati di affluenza al SC a orari predefiniti, l'uso del LO per leggere i voti sulle schede in fase di spoglio (con eventuale assegnazione manuale per le schede "dubbe"), e la trasmissione in tempo reale dei risultati al SC. Si dia un diagramma architetturale, del tipo più opportuno, che mostri come si intenderebbe strutturare tale software, eventualmente indicando gli stili standard adottati.

Domanda 3. Si consideri la funzionalità di fornitura in tempo reale dei risultati parziali durante lo spoglio. La funzionalità andrà fornita dal SC a tutti gli utenti (autorizzati) che la richiedano. Ciascun utente avrà il suo software per la lettura dei risultati e la ri-pubblicazione sui rispettivi siti o simili; l'interfaccia offerta dal SC dovrà quindi essere generica. Quale stile si potrebbe usare per fornire questa interfaccia? Si documenti compiutamente, in UML, l'interfaccia offerta.

Domanda 4. Si fornisca un diagramma di deployment del sistema VASA, in cui siano visibili sia i nodi di ciascun seggio, sia il SC, sia gli utenti generici dei flussi forniti dal Ministero.

Domanda 5. Una parte del software del PC, quella che si occupa di verificare che le schede restituite dall'elettore dopo l'espressione del voto siano le stesse consegnate dal Presidente, potrebbe essere realizzata con l'implementazione seguente:

```
public class QRCheck {

    private Set<QR> fuori = new HashSet<QR>();
    private int count;

    public void apriSeggio() { fuori.clear(); count=0; }

    public void consegna(QR scheda) { fuori.add(scheda); }

    public void ritira(QR scheda) {
        if (!fuori.contains(scheda))
            throw new SchedaNonValidaException("Brogli!");
        else
            count++;
    }

    public void chiudiSeggio() {
        if (fuori.size()>0)
            throw new SchedeNonRestituiteException("Brogli!");
    }

    public int affluenza() { return count; }
}
```

Si assuma che il chiamante rispetti sempre il protocollo della classe, ovvero che **apriSeggio()** e **chiudiSeggio()** vengano chiamate, rispettivamente, all'inizio e alla fine della giornata di votazioni; che ogni volta che il presidente consegna una scheda a un elettore venga chiamato **consegna(*q*)**, in cui *q* è il QR code di quella scheda letto dal LO, e che quando l'elettore restituisce la scheda dopo aver espresso il suo voto, venga chiamato **ritira(*q*)**, in cui di nuovo *q* è il QR code della scheda restituita, come letto dal LO. Se tutto va bene, la scheda a questo punto viene inserita nell'urna. In ogni momento, **affluenza()** deve restituire il numero di schede valide inserite nell'urna. Si noti che il numero di schede che possono essere state consegnate ma non ancora ritirate dipende, ovviamente, dal numero di cabine presenti nel seggio; le operazioni relative a elettori diversi non sono necessariamente in sequenza (per esempio, il Presidente può consegnare due schede, poi ritirare una di queste, consegnarne altre due, ritirare la seconda delle prime due, e poi ritirare le seconde due in ordine inverso rispetto alla consegna).

Si dia un test di unità per la classe QRCheck, esprimendolo sotto forma di una sequenza di chiamate di metodo e di asserzioni (con l'intesa che la sequenza parte con un oggetto QRCheck appena creato, e che ogni asserzione dovrà essere valida nel punto in cui compare nella sequenza). In particolare, si vuole che il test garantisca una copertura del 100% dei comandi e una ragionevole selezione di altri casi. Il test rivela difetti nella classe?