ESAME DI STATO PER L'ABILITAZIONE ALLA PROFESSIONE DI INGEGNERE
SECONDA SESSIONE 2016 – SEZIONE A
SETTORE INFORMAZIONE
Prova Pratica di Progettazione
TEMA N. 4: INFORMATICA

Un modello di razzi inerenti un programma spaziale è fornito di un software per il direzionamento della propulsione, eseguito da un computer di bordo. Questo software riceve dati da un sensore che rileva la velocità, con una frequenza di cento letture al secondo. Per aumentare la robustezza ai guasti il sensore è stato replicato e il software, quando rileva un problema con il primo sensore, utilizza in alternativa il secondo. Sulla base della velocità rilevata, il software invia comandi al sistema di propulsione per gestire la rotta.

Si consideri per semplicità che il linguaggio di programmazione offra solo i seguenti 4 tipi numerici (il candidato assuma di avere a disposizione anche tipi non numerici a piacere).

int: interi con segno a 32 bit.

long: interi con segno a 64 bit.

float: numeri in virgola mobile con segno a 32 bit, secondo la specifica IEEE 754, con mantissa a 23 bit e esponente a 8 bit.

double: numeri in virgola mobile con segno a 64 bit, secondo la specifica IEEE 754, con mantissa a 52 bit e esponente a 11 bit.

Segue parte del codice per il controllo del razzo.

```
void inertialSensor1() {
   float verticalVelocSensor = sensor1GetVertical();
   float horizontalVelocSensor = sensor1GetHorizontal();
   if (inIntRange(verticalVelocSensor) && inIntRange(horizontalVelocSensor)) {
      int verticalVelocBias = (int)verticalVelocSensor;
      int horizontalVelocBias = (int)horizontalVelocSensor;
      adjustDirection(verticalVelocBias,horizontalVelocBias);
   } else {
      inertialSensor2();
   }
}

void inertialSensor2() {
```

```
    float verticalVelocSensor = sensor2GetVertical();
    float horizontalVelocSensor = sensor2GetHorizontal();
    if (inIntRange(verticalVelocSensor) && inIntRange(horizontalVelocSensor)) {
        int verticalVelocBias = (int)verticalVelocSensor;
        int horizontalVelocBias = (int)horizontalVelocSensor;
        adjustDirection(verticalVelocBias,horizontalVelocBias);
    } else {
        adjustDirection(-1,-1);
    }
}

boolean inIntRange(float x) {
    return (x <= Integer.MAX_VALUE) && (x >= Integer.MIN_VALUE);
}
```

Il metodo inertialSensor1 legge il sensore di velocità numero 1 e se non rileva problemi segnala la velocità (dopo una conversione da tipo float a tipo int) al gestore della propulsione affinché, se necessario, corregga la rotta. Se invece rileva un problema, chiama il metodo inertialSensor2. Si noti che la velocità può avere valori sia positivi che negativi su entrambi gli assi considerati.

Il metodo inertialSensor2 legge il sensore di velocità numero 2 e se non rileva problemi segnala la velocità (dopo una conversione da tipo float a tipo int) al gestore della propulsione affinché, se necessario, corregga la rotta. Se invece rileva un problema, chiama il gestore della propulsione con parametri -1, -1.

Il metodo inIntRange controlla se il parametro di tipo float passato è contenuto nell'intervallo di valori rappresentabile con il tipo int. In caso affermativo ritorna true, altrimenti ritorna false.

1) Il codice precedentemente descritto presenta problemi di robustezza. Il candidato descriva i problemi del codice, in particolare si consideri cosa succede quando entrambi i sensori di un asse (verticale o orizzontale) rilevano una velocità che non rientra nell'intervallo rappresentabile con una variabile di tipo int.

2) Il candidato scriva, in pseudocodice o in un linguaggio di programmazione a piacere (tenendo conto delle limitazioni esposte sopra sui tipi numerici) un codice alternativo a quello qui presentato, che risolva i problemi individuati al punto 1, e che inoltre spedisca, per ogni ciclo di esecuzione, le informazioni lette da entrambi i sensori di velocità 1 e 2 ad un centro di controllo a terra. Si assuma che il razzo sia dotato di un dispositivo per la comunicazione wireless e di un opportuno modulo software per la sua gestione.

3) Il candidato progetti un sistema software per il centro di controllo, il quale registri uno storico con tutti i dati delle velocità ricevuti da tutti i razzi che sono assegnati a quel centro di controllo. Ogni razzo ha un suo modello e un suo identificatore univoco. Ogni informazione sulle velocità deve essere corredata dall'istante temporale in cui essa è stata rilevata dai sensori. Si tenga in considerazione che la frequenza con cui si ricevono i dati varia in base al modello di razzo, e che è possibile che vi siano interruzioni nella comunicazione che precludano la ricezione di una o più letture dei sensori. Le informazioni devono essere salvate in maniera stabile in un database relazionale, che risiede presso il centro di controllo. Il personale del centro di controllo può chiedere al software l'elenco di tutti i lanci eseguiti, e selezionare un lancio per visualizzare tutti i dati su di esso. Al candidato si richiede di
   a) Eseguire un'analisi dei requisiti, distinguendoli tra funzionali e non funzionali, preferibilmente usando diagrammi di casi d'uso UML o altri diagrammi similari.
   b) Progettare l'architettura del sistema. Includere preferibilmente diagrammi UML delle classi e di deployment o altri diagrammi similari. Il candidato descriva i principali design pattern utilizzati.
   c) Produrre in linguaggio SQL i comandi necessari per soddisfare i casi d'uso.

Si scelgano le tecnologie ritenute più adatte e si forniscano giustificazioni per le scelte fatte.

Il candidato, qualora lo ritenga necessario, può aggiungere assunzioni ragionevoli che integrino le specifiche qui descritte.