

# **Compito Scritto di Ingegneria del Software**

**10 gennaio 2010**

**Parte teorica, punti 14**

**Tempo a disposizione: 1 ora**

## **Esercizio 1**

Si descrivano le macchine a stati finiti e i loro impieghi e limiti.

PUNTI 7

## **Esercizio 2**

Si descrivano peculiarità e caratteristiche dell'UML e dell'Object-Oriented Design.

PUNTI 7

# **Compito Scritto di Ingegneria del Software**

**10 gennaio 2010**

**Parte pratica, punti 18**

**Tempo a disposizione: 1 ora e mezza**

## **Esercizio 3**

Una linea di produzione è costituita da due macchinari, identici, che producono un pezzo alla volta e lo depositano su una posizione libera su un nastro trasportatore. I due macchinari producono un pezzo solo se c'è un posto libero sul nastro. Il nastro porta i pezzi ad un terzo macchinario che preleva un pezzo alla volta ed esegue una seconda lavorazione. Dopo che il pezzo è stato prelevato, il nastro avanza lasciando un posto vuoto all'inizio per il deposito di un nuovo pezzo. Il nastro ha lunghezza due e inizialmente ha il primo posto vuoto e il secondo pieno. Specificare il sistema con una rete di Petri.

**PUNTI 5**

## **Esercizio 4**

Si specifichi in Z la procedura di gestione del portafoglio azioni dei clienti di una filiale bancaria. Ogni cliente della filiale ha un portafoglio azioni che memorizza, per ogni azione posseduta, la quantità. Ogni cliente ha un inoltre un conto corrente presso la filiale. Il prezzo di ciascuna azione si suppone noto alla filiale e fisso. Si modellino con tipi definiti dall'utente i numeri di conto, i nomi delle aziende quotate in borsa e la funzione prezzo.

Si modellino in particolare le seguenti operazioni:

1. Acquisto di azioni: fornendo il numero di conto del cliente, nome dell'azienda e quantità, viene acquistato il numero di azioni della società richiesto. Le azioni vengono aggiunte al portafoglio del cliente e l'importo dedotto dal saldo del suo conto corrente. L'operazione fallisce se il cliente non ha sufficiente credito nel conto corrente.
2. Vendita di azioni: fornendo il numero di conto del cliente e il nome dell'azienda, la filiale vende tutte le azioni dell'azienda nel portafoglio del cliente. Il valore delle azioni viene aggiunto al saldo del conto corrente del cliente. L'operazione fallisce se l'azienda non è nel portafoglio del cliente.

**PUNTI 7**

## Esercizio 5

Si disegni il flusso di esecuzione e si esegua l'analisi di flusso, per mezzo di espressioni regolari, per le variabili del seguente programma.

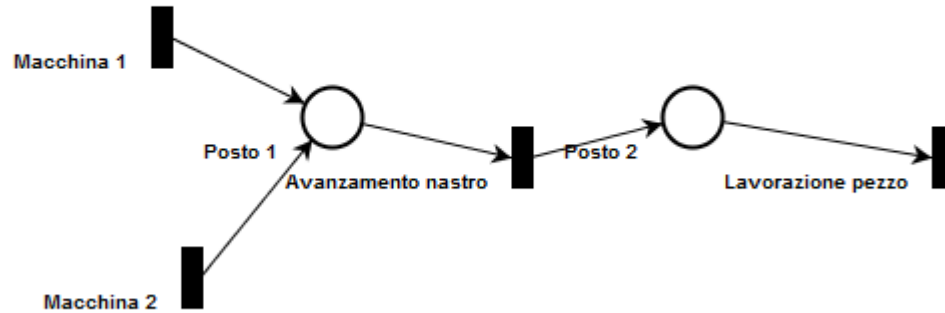
Si individuino inoltre eventuali sequenze non corrette di operazioni e, per ognuna, almeno un caso di test che porti alla sua esecuzione.

```
1. #include<stdio.h>
2. int main (void) {
3.     int a, b = 1, c;
4.     scanf("%d", &a);
5.     if (a > 0) {
6.         a = a - b;
7.         b = b + 1;
8.         c = a / b;
9.     }
10.    else {
11.        b = a / c;
12.    }
13.    b = a - 1;
14.    printf("%d\n", c );
15.    return 0;
16. }
```

PUNTI 6

# Soluzione

## Esercizio 3



## Esercizio 2

Tipi definiti dall'utente:

[Conti, Aziende, Prezzi]

Conti={insieme dei numeri di conto di tutti i clienti}

Aziende={insieme delle aziende quotate in borsa}

Prezzo=funzione che fornisce il prezzo di un'azione, Prezzi: Aziende  $\rightarrow \mathbb{R}$

Variabili che descrivono lo stato del sistema:

- 1) saldo: è una funzione totale che associa ad un conto corrente un saldo in euro;
- 2) portafoglio: è una funzione parziale che associa ad un conto corrente e ad una azienda il numero di azioni possedute

Banca

saldo: Conti  $\rightarrow \mathbb{R}$

portafoglio: Conti  $\times$  Aziende  $\rightarrow \mathbb{N}$

dom saldo  $\supseteq \{c \mid \exists a \cdot (c,a) \in \text{dom portafoglio}\}$

InitBanca

$\Delta$ Banca

saldo'={}

portafoglio'={}

Success

rep!: Report

rep! = 'Okay'

1) acquisto di azioni:

AcquistoOK

$\Delta$ Banca

conto?: Conti

azienda?: Aziende

quantità?:  $\mathbb{N}$

saldo(conto?)  $\geq$  Prezzo(azienda?)\*quantità

portafoglio'= $\text{portafoglio} \cup \{(conto?,azienda?) \mapsto \text{quantità?}\}$

saldo'=saldo  $\oplus$  {conto?  $\mapsto$  saldo(conto?)-Prezzo(azienda?)\*quantità}

SaldoInsufficiente

$\exists$ Banca

conto?: Conti

azienda?: Aziende

quantità?:  $\mathbb{N}$

rep!: Report

Prezzo(azienda?)\*quantità > saldo(conto?)

rep! = 'Saldo insufficiente'

Acquisto  $\cong$  AcquistoOK  $\wedge$  Success

$\vee$

SaldoInsufficiente

2) vendita di azioni

VenditaOK

$\Delta$ banca

conto?: Conti

azienda?: Aziende

(conto?,azienda?)  $\in$  dom portafoglio

portafoglio'= $\text{portafoglio} \setminus \{(conto?,azienda?) \mapsto \text{portafoglio}((conto?,azienda?))\}$

conto'=conto  $\oplus$  {conto?  $\mapsto$  saldo(conto?)+Prezzo(azienda?)\*quantità}

AziendaNonPresente

$\exists$ Banca

conto?: Conti

azienda?: Aziende

$(\text{conto?}, \text{azienda?}) \notin \text{dom portafoglio}$

rep! = 'Azione non in portafoglio'

$\text{Vendita} \equiv \text{VenditaOK} \wedge \text{Success}$

$\vee$

AziendaNonPresente

## Esercizio 5

<b>riga</b>	<b>a</b>	<b>b</b>	<b>c</b>
	3	a	ad a
	4	d	
	5	u	
	6	ud	u
	7		ud
	8	u	u d
	11	u	d u
	13	u	d
	14		u

Espressioni regolari:

per a: **adu (udu+u) u**  
per b: **ad (uudu+d) d**  
per c: **a (d+u) u**

Osservazioni:

- Nel caso “else” la variabile c viene usata senza essere definita. Tale anomalia può essere evidenziata da qualsiasi test case con  $a < 0$ .
- Inoltre, sempre nel caso “else” abbiamo una sequenza anomala per la variabile b, che viene definita tre volte di seguito, senza mai essere utilizzata.