

**Compito Scritto di Ingegneria del Software**

**9 aprile 2010**

**Parte teorica, punti 14**

**Tempo a disposizione: 1 ora**

**Esercizio 4**

Si descrivano i diversi stili di specifica.

PUNTI 7

**Esercizio 5**

Si descrivano i passi principali della procedura di conteggio dei Function Point.

PUNTI 7

## **Compito Scritto di Ingegneria del Software**

**9 aprile 2010**

**Parte pratica, punti 18**

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

### **Esercizio n. 1**

Modellare con una rete di Petri un sistema di controllo di un cancello comandato da un telecomando. Il telecomando ha un pulsante che può essere premuto dall'utente. Il cancello può essere in quattro stati: aperto, chiuso, in apertura e in chiusura. Se il pulsante sul telecomando viene premuto quando il cancello è chiuso, esso comincia ad aprirsi. Se, durante l'apertura, il pulsante viene nuovamente premuto, il cancello interrompe l'apertura e comincia a chiudersi. Se il pulsante viene premuto quando il cancello è aperto, esso comincia a chiudersi. Se, durante la chiusura, il pulsante viene premuto, il cancello interrompe la chiusura e comincia ad aprirsi.

Il cancello è dotato di un lampeggiante che è attivo solo quando il cancello è in movimento.

Inizialmente il cancello è chiuso e il lampeggiante è spento.

PUNTI 5

### **Esercizio n. 2**

Si dia una specifica in  $Z$  di un supermercato e, in particolare, della gestione della cassa e del database degli acquisti. Il sistema deve registrare le transazioni di vendita effettuate: ogni transazione è identificata da un codice intero univoco e deve memorizzare tutti gli articoli acquistati insieme alla loro quantità. Si supponga che sia disponibile un database dei prezzi degli articoli come funzione definita dall'utente. Il supermercato distribuisce inoltre carte di pagamento ai clienti identificate da un numero intero. Nel caso in cui la transazione di acquisto sia effettuata da un cliente in possesso di una carta di pagamento, la transazione va a buon fine solo se sulla carta c'è denaro a sufficienza e alla transazione deve essere associato l'identificativo della carta. Si modellino in  $Z$  le seguenti operazioni

1) assegnazione a un cliente di una carta di pagamento: dato il nome e il cognome del cliente, si restituisca il codice identificativo della carta e si assegni alla carta un saldo iniziale di 10000 centesimi di euro.

2) vendita ad un cliente senza carta di pagamento: dato un bag di articoli, si aggiorni il database delle vendite aggiungendo una nuova transazione e si restituisca il suo identificativo

3) vendita ad un cliente con carta di pagamento: dato un bag di articoli e un cliente, si verifica se la carta del cliente ha credito sufficiente. Se questo è vero, si completa la vendita aggiornando il saldo della tessera. Si aggiorni il database delle vendite aggiungendo una nuova transazione e si restituisca il suo identificativo.

PUNTI 7

**Esercizio n.3**

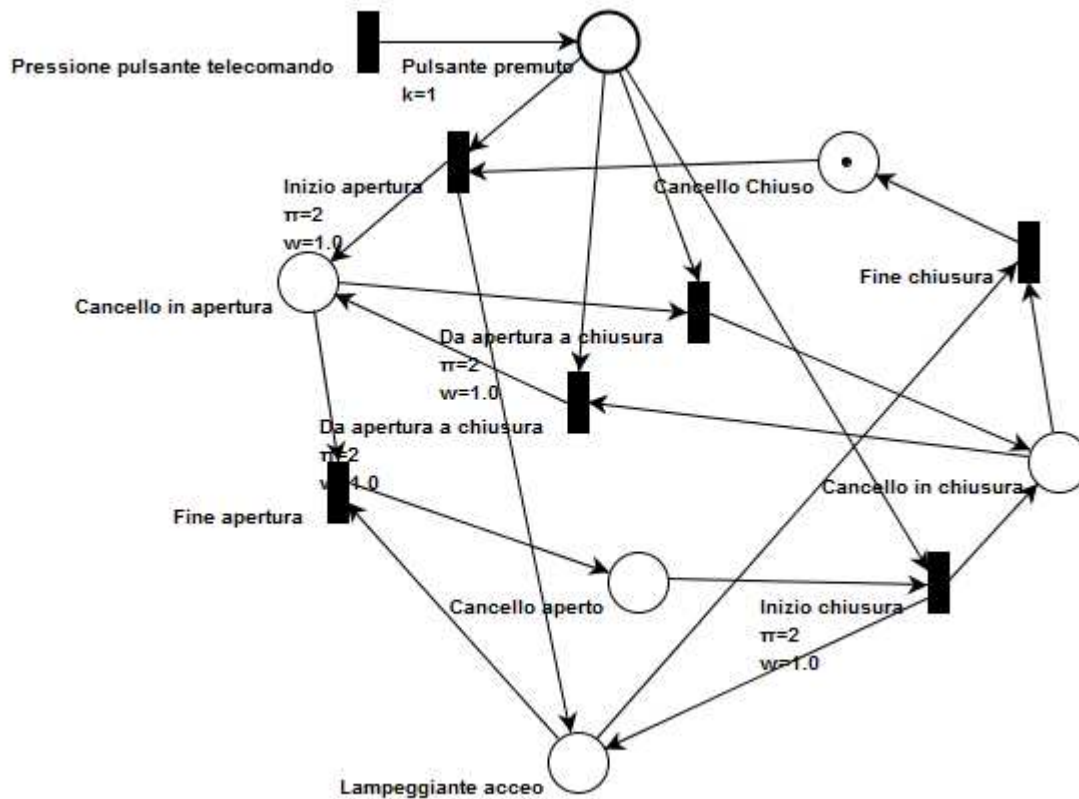
```
function MCD(a, b)
if a>0 and b>0
    while a ≠ b
        if a > b
            a := a - b
        else
            b := b - a
    return a
else
    return 0
```

Si descriva il risultato di una esecuzione simbolica della seguente funzione indicando i valori dello stato del programma e della path condition dopo ogni istruzione, nell'ipotesi di voler eseguire ogni comando del programma (ed il ciclo while almeno due volte). Si individui almeno un test case che consenta tale esecuzione.

PUNTI 6

## Soluzione

### Esercizio 1



### Esercizio 2

Tipi definiti dall'utente:

[Clienti,Articoli]

Clienti = {(Nome,Cognome) | Nome e Cognome indicano una persona}

Articoli={insieme degli articoli venduti dal supermercato}

Costanti

prezzi: Articoli  $\rightarrow \mathbb{N}$  : funzione che associa a ciascun articolo il suo prezzo

Variabili che descrivono lo stato del sistema:

1) carte è una funzione che associa un cliente al suo numero di carta;

2) saldi è una funzione che associa una carta al suo saldo

3) transazioni è una funzione che associa ad ogni transazione un bag di articoli

4) transazioni\_carte è una funzione che associa ad una transazione un identificativo di una carta

Supermercato

carte: Clienti  $\rightarrow \mathbb{N}$

saldi:  $\mathbb{N} \rightarrow \mathbb{N}$

transazioni:  $\mathbb{N} \rightarrow \text{bag Articoli}$

transazioni\_carte  $\mathbb{N} \rightarrow \mathbb{N}$

dom saldi = ran carte

ran transazioni\_carte  $\subseteq$  ran carte

dom transazioni\_carte  $\subseteq$  dom transazioni

InitSupermercato

$\Delta$ Supermercato

carte' =  $\emptyset$

saldi' =  $\emptyset$

transazioni' =  $\emptyset$

transazioni\_carte' =  $\emptyset$

Success

rep!: Report

rep! = 'Okay'

1) assegnazione a un cliente di una carta di pagamento

RichiediNuovaCarta

$\Delta$ Supermercato

cliente?: Clienti

carta!:  $\mathbb{N}$

cliente?  $\notin$  dom carte

carta!  $\notin$  ran carte

carte' = carte  $\cup$  {cliente?  $\mapsto$  carta!}

saldi' = saldi  $\cup$  {carta!  $\mapsto$  10000}

transazioni' = transazioni

transazioni\_carte' = transazioni\_carte

ClienteGiàConCarta

$\exists$ Supermercato

cliente?: Clienti

rep!: Report

cliente?  $\in$  dom carte

rep! = Cliente già con carta'

AssegnazioneCarta  $\equiv$  RichiediNuovaCarta  $\wedge$  Success

$\vee$

ClienteGiàConCarta

2) vendita ad un cliente senza carta di pagamento

VenditaClienteSenzaCarta

$\Delta$ Supermercato

cliente?: Clienti

carrello?: bag Articoli

id!:  $\mathbb{N}$

cliente?  $\notin$  dom carte

carte' = carte

saldi' = saldi

id!  $\notin$  dom transazioni

transazioni' = transazioni  $\cup$  {id!  $\mapsto$  carrello?}

transazioni\_carte' = transazioni\_carte

ClienteConCarta

$\exists$ Supermercato

cliente?: Clienti

rep!: Report

cliente?  $\in$  dom carte

rep! = Cliente con carta'

$VenditaSenzaCarta \cong VenditaClienteSenzaCarta \wedge Success$

$\vee$

ClienteConCarta

### 3) vendita ad un cliente con carta di pagamento

#### Descrizione assiomatica

$\text{prezzo\_carrello}: \text{bag Articoli} \rightarrow \mathbb{N}$

$\forall a: \text{Articolo}, q: \mathbb{N}, c: \text{bag Articoli} \cdot$

$\text{prezzo\_carrello}[\ ] = 0$

$\text{prezzo\_carrello}(\{a \mapsto q\} \cup c) = q * \text{prezzi}(a) + \text{prezzo\_carrello}(c)$

#### VenditaClienteConCarta

$\Delta \text{Supermercato}$

$\text{cliente?}: \text{Clienti}$

$\text{carrello?}: \text{bag Articoli}$

$\text{id!}: \mathbb{N}$

$\text{cliente?} \in \text{dom carte}$

$\text{prezzo\_carrello}(\text{carrello?}) \leq \text{saldi}(\text{carte}(\text{cliente?}))$

$\text{carte}' = \text{carte}$

$\text{saldi}' = \text{saldi} \oplus \{ \text{carte}(\text{cliente?}) \mapsto \text{saldi}(\text{carte}(\text{cliente?})) - \text{prezzo\_carrello}(\text{carrello?}) \}$

$\text{id!} \notin \text{dom transazioni}$

$\text{transazioni}' = \text{transazioni} \cup \{ \text{id!} \mapsto \text{carrello?} \}$

$\text{transazioni\_carte}' = \text{transazioni\_carte} \cup \{ \text{id!} \mapsto \text{carte}(\text{cliente?}) \}$

#### ClienteSenzaCarta

$\exists \text{Supermercato}$

$\text{cliente?}: \text{Clienti}$

$\text{rep!}: \text{Report}$

$\text{cliente?} \notin \text{dom carte}$

$\text{rep!} = \text{Cliente senza carta}'$

#### SaldoInsufficiente

$\exists \text{Supermercato}$

$\text{cliente?}: \text{Clienti}$

$\text{carrello?}: \text{bag Articoli}$

$\text{cliente?} \in \text{dom carte}$

$\text{prezzo\_carrello}(\text{carrello?}) > \text{saldi}(\text{carte}(\text{cliente?}))$

$\text{VenditaConCarta} \cong \text{VenditaClienteConCarta} \wedge \text{Success}$

$\vee$

$\text{ClienteSenzaCarta}$

$\vee$

$\text{SaldoInsufficiente}$