



UNIVERSITÀ DEGLI STUDI DI PISA
Corso di Laurea in Ingegneria Informatica
c/o Dipartimento di Ingegneria dell'Informazione: Elettronica, Informatica,
Telecomunicazioni

Esami di Ingegneria del Software
e di Ingegneria dei Sistemi Software
Appello dell'11 giugno 2004

Nome e cognome:

Matricola:

Esame: **Ing. del Software** **Ing. dei Sistemi Software**

Il punteggio in trentesimi associato a ciascuna domanda è indicato fra parentesi. Ove si trovino coppie di domande con lo stesso numero, rispondere solo a quella corrispondente all'esame che si vuole sostenere: la domanda per l'esame di Ingegneria del Software (vecchio ordinamento) è identificata dalla sigla [V], quella per l'esame di Ingegneria dei Sistemi Software (nuovo ordinamento) dalla sigla [N].

Scegliere una risposta per ciascuna domanda:

- | | | |
|----------|---|-------------------------------------|
| 1 | In C++ le operazioni polimorfiche sono | (1) |
| | astratte. | <input type="checkbox"/> |
| | virtuali. | <input checked="" type="checkbox"/> |
| | statiche. | <input type="checkbox"/> |
| 2 | Il test funzionale è basato | (1) |
| | sulle specifiche. | <input checked="" type="checkbox"/> |
| | sul documento di progetto. | <input type="checkbox"/> |
| | sul codice sorgente. | <input type="checkbox"/> |
| 3 | Negli Statechart, i segnali sono | (1) |
| | computazioni interrompibili. | <input type="checkbox"/> |
| | computazioni non interrompibili. | <input type="checkbox"/> |
| | un tipo di eventi. | <input checked="" type="checkbox"/> |
| 4 | Un compilatore IDL | (1) |
| | produce codice eseguibile. | <input type="checkbox"/> |
| | produce codice sorgente. | <input checked="" type="checkbox"/> |
| | produce bytecode interpretabile. | <input type="checkbox"/> |
| 5 | Una <i>formula ben formata</i> | (1) |
| | è sintatticamente corretta. | <input checked="" type="checkbox"/> |
| | è valida. | <input type="checkbox"/> |
| | è dimostrabile. | <input type="checkbox"/> |

Rispondere alle domande, usando solo lo spazio disponibile:

6 Completare la seguente tabella (3)

x	y	$\neg x \vee y$
T	T	<i>T</i>
T	F	<i>F</i>
F	T	<i>T</i>
F	F	<i>T</i>

(la soluzione è in corsivo)

7 Che cosa significa la formula $M [t] \wedge \neg(M [u]) \wedge M [tu]$? (3)

Nella marcatura M la transizione t è abilitata, u non è abilitata, e la sequenza tu è abilitata, cioè t e u sono in sequenza.

8 Che cos'è un oggetto attivo? (3)

Un oggetto capace di attivare un flusso di controllo.

9 Qual'è la differenza fra convalida e verifica? (3)

Nella convalida un prodotto viene confrontato con i requisiti dell'utente, espressi in modo informale, mentre nella verifica viene confrontato con le specifiche, espresse in modo più formale.

10 Definire il criterio di copertura dei comandi. (3)

Ogni istruzione del modulo sotto test deve essere eseguita per almeno un dato di test.

11 Con riferimento alla Fig. 1, rispondere alle domande.

(5)

	V	F
Menu deriva da Mwindow	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Un oggetto Menu può contenere oggetti Button	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Un oggetto Button può contenere oggetti Menu	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Un oggetto Menu può contenere oggetti Menu	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Menu eredita l'operazione click	<input type="checkbox"/>	<input checked="" type="checkbox"/>

12 Disegnare un diagramma di classi corrispondente alle seguenti dichiarazioni: (5)

```
class DataMoverImp {
public:
    virtual void filetransfImp(const string& src, const string& dst) = 0;
};

class SimpleDataMover : public DataMoverImp {
public:
    void filetransfImp(const string& src, const string& dst);
};

class OptDataMover : public DataMoverImp {
public:
    void filetransfImp(const string& src, const string& dst);
};

class DataMover {
    DataMoverImp* imp;
public:
    DataMover(DataMoverImp* p) : imp(p) {};
    void filetransf(const string& src, const string& dst);
};

void SimpleDataMover::filetransfImp(const string& src, const string& dst)
{ /* ... */ }

void OptDataMover::filetransfImp(const string& src, const string& dst)
{ /* ... */ }

void DataMover::filetransf(const string& src, const string& dst)
{
    imp->filetransfImp(src, dst);
}
```

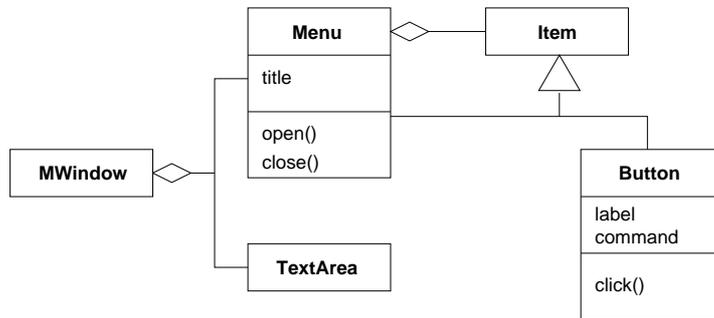


Figura 1: Domanda 11.

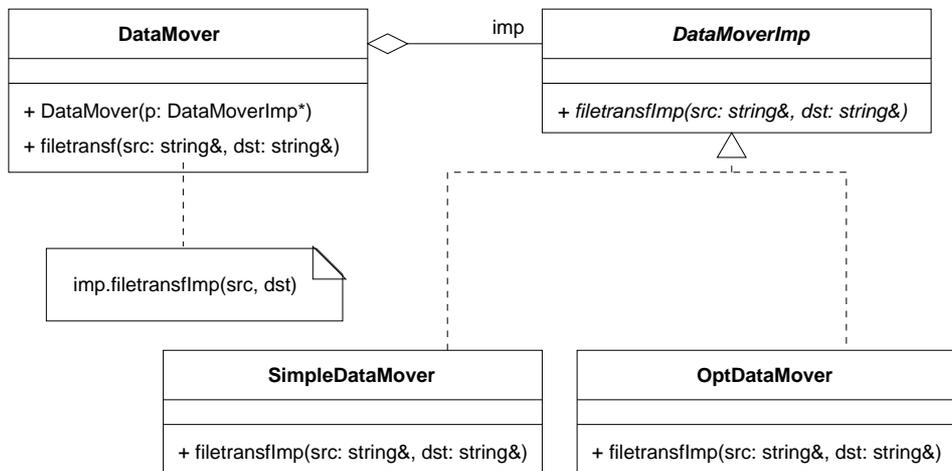


Figura 2: Domanda 12.

(Le operazioni sono specificate con una sintassi mista fra UML e C++).