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 del 21 gennaio 2005

**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 Se i dati di test si scelgono in base ai requisiti** (1)
si fa un test di sistema. ☐
si fa un test strutturale. ☐
si fa un test funzionale. ☒

**2 "Polimorfismo" significa** (1)
avere piú implementazioni di una stessa interfaccia. ☒
derivare piú classi da una stessa base. ☐
derivare una classe da piú basi. ☐

**3 Una formula deducibile usando solo gli assiomi è** (1)
un teorema. ☒
una dimostrazione. ☐
una regola d'inferenza. ☐

**4 Negli Statechart i segnali sono** (1)
computazioni interrompibili. ☐
computazioni non interrompibili. ☐
un tipo di eventi. ☒

**5 Nelle reti di Petri la marcatura è** (1)
una funzione che associa posti a transizioni ($P \rightarrow T$). ☐
una funzione che associa posti a numeri naturali ($P \rightarrow \mathbb{N}$). ☒
una funzione che associa archi a numeri naturali ($F \rightarrow \mathbb{N}$). ☐

Rispondere alle domande, usando solo lo spazio disponibile:

**6 Dimostrare che la formula proposizionale $\neg(A \wedge B)$ equivale a $\neg A \vee \neg B$.** (3)

| $A$ | $B$ | $\neg(A \wedge B)$ | $\neg A \vee \neg B$ |
|---|---|---|---|
| **T** | **T** | $F$ | $F \vee F = F$ |
| **T** | **F** | $T$ | $F \vee T = T$ |
| **F** | **T** | $T$ | $T \vee F = T$ |
| **F** | **F** | $T$ | $T \vee T = T$ |

**7 Qual'è la differenza fra una libreria e un framework?** (3)

Una libreria offre moduli semplici da comporre per ottenere funzionalità piú complesse, mentre un framework offre dei componenti complessi interagenti secondo meccanismi predefiniti che vengono specializzati per adattarli ad applicazioni specifiche.

**8 Qual'è la differenza fra un'anomalia e un guasto?** (3)

Una anomalia è un difetto del codice sorgente, un guasto è un comportamento non corretto rispetto alle specifiche.

**9 Che cos'è una partizione, nel progetto di sistema?** (3)

Un sottosistema che realizza una funzione del sistema complessivo.

**10 [V] Quali diagrammi UML descrivono l'architettura fisica?** (3)

I diagrammi di deployment e i diagrammi dei componenti.

**10 [N] Che cos'è il servizio di naming?** (3)

Un servizio CORBA che permette di ottenere un riferimento a un oggetto remoto identificato da un nome simbolico.

**11 Descrivere a parole la struttura rappresentata in Fig. 1.** (5)
(L'operazione *action()* viene invocata implicitamente quando l'utente "clicca" un bottone. L'op. getText() restituisce il testo scritto nel campo).

Un Widget può essere di tipo Container o Simple. I Widget di tipo Container possono contenere altri Widget, ed hanno l'operazione add() per aggiungere un Widget al contenitore. Quest'operazione prende come argomento un puntatore al Widget che viene aggiunto. Un Widget di tipo Container è Window, che ha l'operazione setTitle() per impostare il titolo della finestra. Quest'operazione prende come argomento una stringa di caratteri. Due Widget di tipo Simple sono Button e TextField, con le operazioni descritte nel testo. L'operazione action() non è implementata: l'implementazione viene fornita da una classe derivata e prende come argomento un puntatore a una struttura dati di tipo generico.

**12 Usando il framework dell'esercizio precedente, scrivere un programma C++ che:** (5)
(1) crea una finestra col titolo "Main Window"; (2) vi inserisce un bottone e un campo di testo; (3) scrive sull'uscita standard il contenuto del campo testo, quando viene premuto il bottone. Includere il file `widgets.h`.

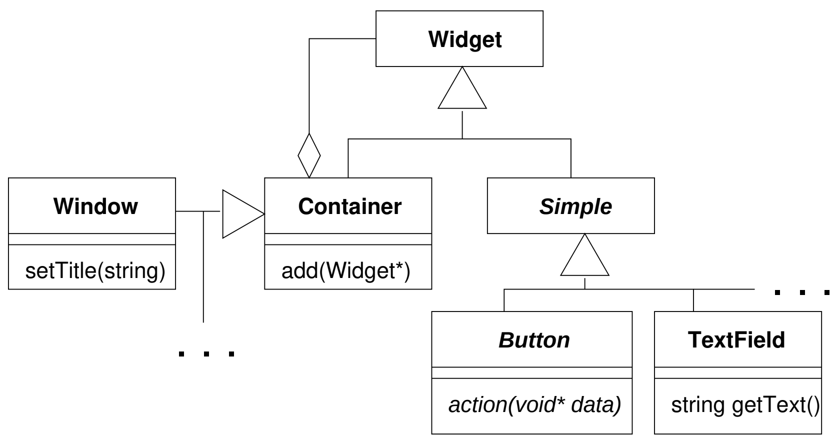


Figura 1: Domanda 11.

```
#include <iostream>
#include "widgets.h"

using namespace std;

class MyButton : public Button {
    TextField* tf;
public:
    MyButton(TextField* t) : tf(t) {};
    void action(void* data);
};

void
MyButton::
action(void* data)
{
    cout << tf->getText() << endl;
}

int
main()
{
    Window* w = new Window;
    TextField* t = new TextField;
    MyButton* b = new MyButton;

    w->setTitle("Main Window");
    w->add(t);
    w->add(b);
}
```

Figura 2: Domanda 12 (soluzione).