UNIVERSITÀ 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 24 febbraio 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 Quale di queste ER riconosce la stringa `duudu`**?** (1)
`u((d|u)d)*` ☐
`d((u|d)u)*` ☒
`d(u(u|d))*` ☐

2 La formula $A \Rightarrow (B \Rightarrow A)$ **è** (1)
vera. ☐
soddisfacibile. ☐
valida. ☒

3 Se, in una rete di Petri, una transizione u **è abilitata solo dopo lo scatto di una transizione** t**,** (1)
le due transizioni sono in sequenza. ☒
le due transizioni sono in concorrenza. ☐
le due transizioni sono in conflitto. ☐

4 Il piano di test di sistema viene prodotto (1)
nella fase di analisi e specifica. ☒
nella fase di progetto. ☐
nella fase di codifica. ☐

5 La marcatura di una rete di Petri (1)
Associa un intero non negativo ad ogni place di ingresso. ☐
Associa un intero non negativo ad ogni place. ☒
Associa un intero non negativo ad ogni arco. ☐

Rispondere alle domande, usando solo lo spazio disponibile:

6 Che cosa significa la formula $M_1\,[s\rangle \Rightarrow M_2\,[t\rangle$? (3)

Se la transizione s è abilitata nella marcatura M_1, allora la transizione t è abilitata nella marcatura M_2.
(**N.B.**: l'implicazione non rappresenta né la causalità né la successione temporale; la formula non vuol dire che s e t sono in sequenza).

7 Qual è la principale differenza fra Statechart e Automi a Stati Finiti? (3)

Gli Statechart permettono una descrizione strutturata e gerarchica degli stati di un automa, che possono essere decomposti in sottostati.

8 Che cos'è lo studio di fattibilità? (3)

Un'attività volta a stabilire se un prodotto è tecnicamente realizzabile ed economicamente conveniente, a proporre alcune strategie per la sua realizzazione, e a stimarne i costi.

9 A cosa servono i diagrammi di deployment? (3)

Servono a descrivere l'architettura hardware, mostrando i nodi computazionali, le loro interconnessioni, ed eventualmente i componenti software associati ai nodi.

10 [V] Che cosa sono i modelli di processo evolutivi? (3)

Modelli di processi in cui il software viene sviluppato incrementalmente in passi successivi, ognuno dei quali produce una parte o una versione del sistema.

10 [N] Che cosa sono gli oggetti transienti nelle architetture CORBA? (3)

Oggetti che esistono solo finché esiste il POA che li ha creati.

11 Con riferimento alla Fig. 1, rispondere alle domande. (5)

	V	F
un **ConcreteScrollWdw** contiene un **Window**	☒	☐
un **ConcreteScrollWdw** è un **PlainWdw**	☐	☒
ConcreteScrollWdw implementa **PlainWdw**	☐	☒
ConcreteScrollWdw estende il comportamento di **PlainWdw**	☒	☐
un **Window** contiene un **PlainWdw**	☐	☒

12 Con riferimento al codice di Fig. 2, disegnare un diagramma di classi che lo descriva. (5)

Scrivere inoltre un programma che stampi sull'uscita standard il risultato delle due operazioni della classe **Proxy_Greeting**, supponendo che ci si connetta al port 1492 della macchina `issw.unipi.it`.

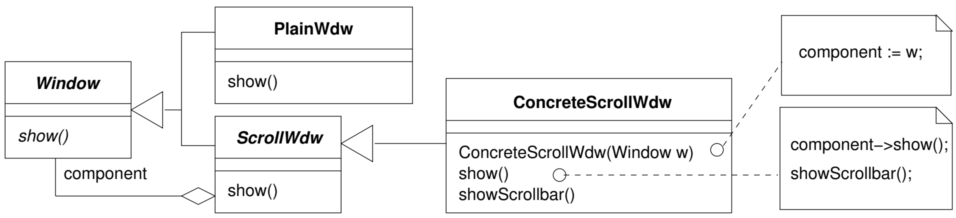

Figura 1: Domanda 11.

```
class Connection {
    string host;
    int port;
    string buffer;
public:
    Connection(string h, int p)
        : host(h), port(p) {};
    void send(string s);
    string receive();
};

class Greeting {
public:
    virtual string
        say_hello() = 0;
    virtual string
        sag_gutenTag() = 0;
};
```

```
class Proxy_Greeting : public Greeting {
    Connection* conn;
public:
    Proxy_Greeting(Connection* c)
        : conn(c) {};
    string say_hello();
    string sag_gutenTag();
};

string
Proxy_Greeting::
say_hello()
{
    conn->send("say_hello");
    return conn->receive();
}
```

```
string
Proxy_Greeting::
sag_gutenTag()
{
    conn->send("sag_gutenTag");
    return conn->receive();
}
```

Figura 2: Domanda 12.

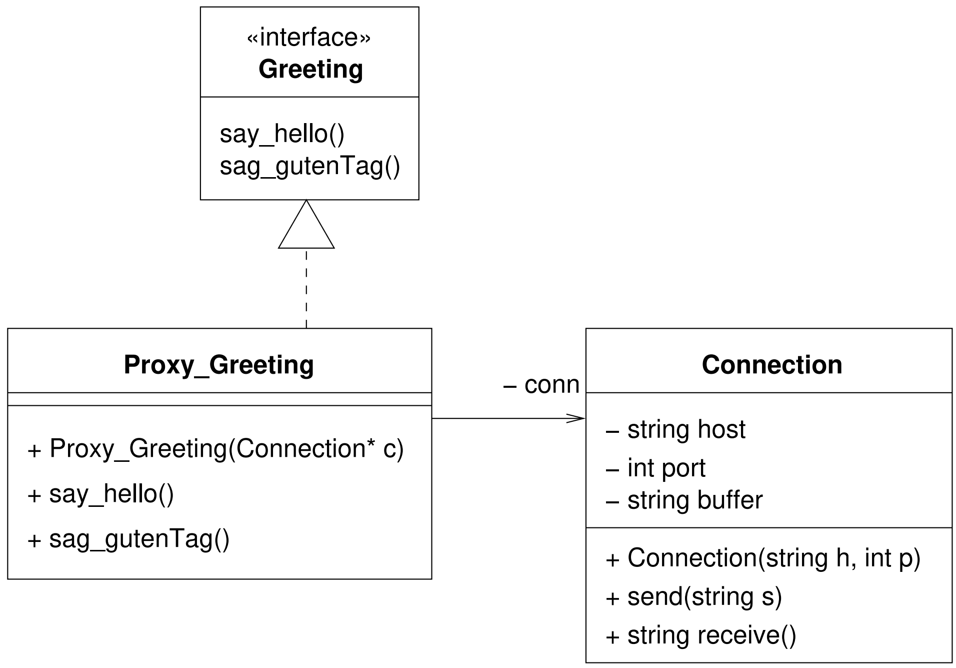

Figura 3: Domanda 11, soluzione (1).

```
int
main()
{
    Connection c("issw.unipi.it", 1492);
    Proxy_Greeting g(&c);
    cout << g.say_hello() << endl;
    cout << g.sag_gutenTag() << endl;
    return (0);
}
```

Figura 4: Domanda 11, soluzione (2).