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

Esame di Ingegneria dei Sistemi Software

Appello del 7 febbraio 2006

Nome e cognome:
Matricola:

Il punteggio in trentesimi associato a ciascuna domanda è indicato fra parentesi.

Scegliere una risposta per ciascuna domanda:

1	**L'implementazione di un oggetto CORBA si chiama**	(1)
	ORB.	☐
	servant.	☒
	stub.	☐
2	**Il piano di test di sistema**	(1)
	definisce i test di convalida.	☒
	pianifica i test di integrazione.	☐
	definisce i test strutturali.	☐
3	**Una specifica si dice *formale* se**	(1)
	non è sostanziale.	☐
	è rigorosa.	☒
	è in forma grafica.	☐
4	**Un'espressione regolare**	(1)
	è associativa e commutativa.	☐
	descrive un insieme di sequenze di simboli.	☒
	è sintatticamente corretta.	☐
5	**Un modulo si dice *coeso* se**	(1)
	è di piccole dimensioni.	☐
	dipende da pochi moduli fornitori.	☐
	offre servizi omogenei.	☒

Rispondere alle domande, usando solo lo spazio disponibile:

6 Che cos'è una *classe concreta*? (3)

Una classe istanziabile, perché interamente implementata.

7 Qual è il modo piú utile di usare l'eredità multipla? (3)

Definire un'interfaccia come composizione di altre interfacce.

8 A cosa serve il servizio di *naming* nelle architetture CORBA? (3)

Serve ad ottenere un riferimento a un oggetto remoto identificato da un nome simbolico.

9 Scrivere la tabella di verità della formula proposizionale $(A \vee B) \Rightarrow A$. È valida? (3)

A	B	$A \vee B$	$(A \vee B) \Rightarrow A$
T	**T**	T	$T \Rightarrow T = T$
T	**F**	T	$T \Rightarrow T = T$
F	**T**	T	$T \Rightarrow F = F$
F	**F**	F	$F \Rightarrow F = T$

Non è valida.

10 Negli Statechart, quali informazioni si possono associare ad una transizione? (3)

Nome ed attributi dell'evento, condizione di guardia ed azione.

11 Disegnare un diagramma UML corrispondente al codice di Fig. 1. (4)
(È un'applicazione del design pattern *Decorator*).

12 Con riferimento all'esercizio precedente: (6)
implementare `CryptedConnection::send()`, che deve cifrare una stringa e inviarla;
implementare `CryptedConnection::receive()`, che deve ricevere una stringa e decifrarla;
scrivere una funzione (globale) `ssend(ConnectionBase* c, string s)` che deve inviare un messaggio su una connessione;
supponendo già implementate tutte le altre operazioni, scrivere un programma principale che mandi il messaggio "ciao" allo host `ing.unipi.it:13`, in chiaro, e il messaggio "zdravstvuj" allo host `kremvax.kgb.su:17`, in cifra, usando la `ssend()` in tutti e due i casi (usare una chiave a scelta).

```
class ConnectionBase {                          |  void
public:                                         |  SecureConnection::
    virtual void send(string s) = 0;            |  send(string s)
    virtual string receive() = 0;               |  {
};                                              |      conn_->send(s);
class Connection : public ConnectionBase {      |  }
    string host_;                               |  string
public:                                         |  SecureConnection::
    Connection(string h) : host_(h) {};         |  receive()
    void send(string s);                        |  {
    string receive();                           |      return conn_->receive();
};                                              |  }
      class SecureConnection : public ConnectionBase {
          ConnectionBase* conn_;
      public:
          SecureConnection(Connection* c) : conn_(c) {};
          void send(string s);
          string receive();
      };
      class CryptedConnection : public SecureConnection {
          string key_;
          string encrypt(string s);
          string decrypt(string s);
      public:
          CryptedConnection(Connection* c) : SecureConnection(c) {};
          void send(string s);
          string receive();
          void set_key(string s);
      };
```

Figura 1: Domanda 11.

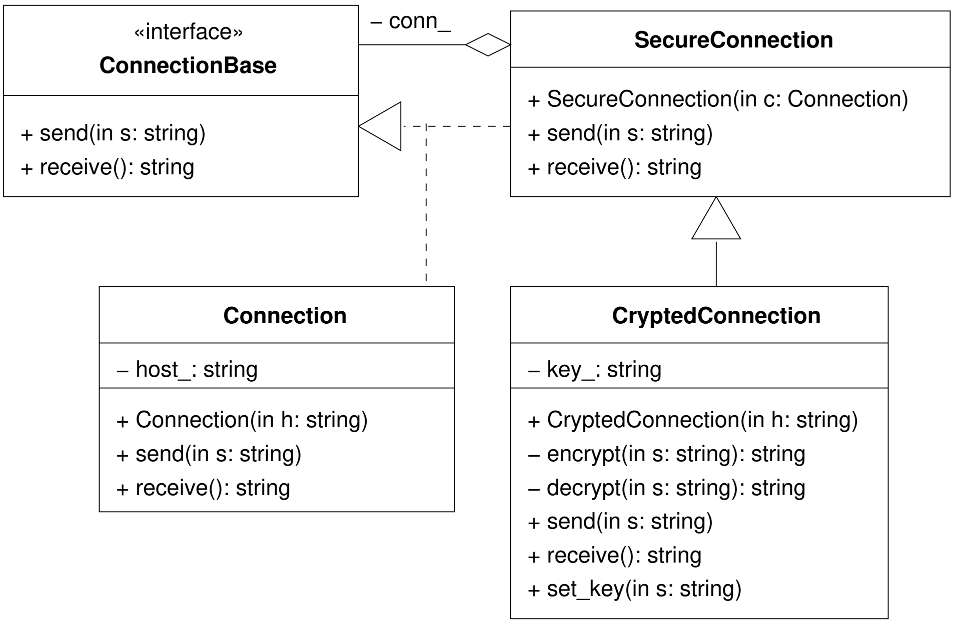

Figura 2: Domanda 11, soluzione.

```
void
CryptedConnection::
send(string s)
{
    SecureConnection::send(encrypt(s));
}

string
CryptedConnection::
receive()
{
    return decrypt(SecureConnection::receive());
}

void
ssend(ConnectionBase* c, string s)
{
    c->send(s);
}

int
main()
{
    Connection ing("ing.unipi.it:13");
    ssend(&ing, "ciao");
    Connection kremvax("kremvax.kgb.su:17");
    CryptedConnection sc(&kremvax);
    sc.set_key("abracadabra");
    ssend(&sc, "zdravstvuj");
}
```

Figura 3: Domanda 12, soluzione.