

Nome e cognome:

Matricola:

Il punteggio relativo a ciascuna domanda, indicato fra parentesi, è in trentesimi. I candidati devono consegnare entro un'ora dall'inizio della prova.

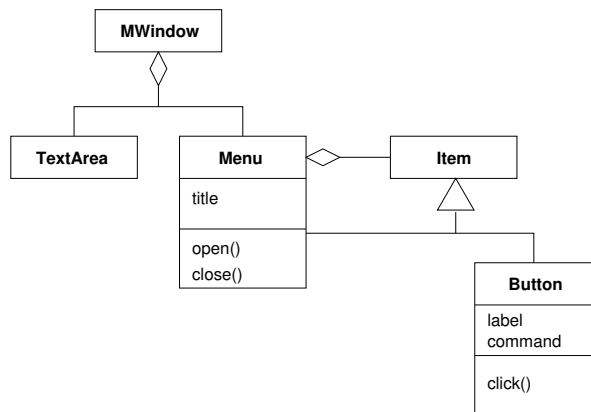


Figura 1: Domande 1–5.

- 1 In Fig. 1, (1)
 - Un oggetto **Menu** può contenere oggetti **Button**
 - La classe **Menu** deriva dalla classe **Button**
 - La classe **Menu** contiene la classe **Button**
- 2 In Fig. 1, (1)
 - La classe **Menu** deriva dalla classe **Mwindow**
 - Un oggetto **Mwindow** può contenere oggetti **Menu**
 - Un oggetto **Menu** può contenere oggetti **Mwindow**
- 3 In Fig. 1, (1)
 - Un oggetto **Button** può contenere oggetti **Menu**
 - La classe **Button** deriva dalla classe **Item**
 - La classe **Button** è base della classe **Item**
- 4 In Fig. 1, (1)
 - La classe **Item** è base della classe **Button**
 - La classe **Item** contiene la classe **Button**
 - Un oggetto **Button** può contenere oggetti **Item**
- 5 In Fig. 1, (1)
 - Menu** eredita l'operazione **click**
 - Menu** eredita l'operazione **open**

- Menu** implementa l'operazione **open**
- 6 Disegnare una macchina a stati** che specifichi quanto segue: un motore (5)
 può girare in due versi, ma non può passare direttamente da un verso all'altro,
 dovendo essere fermato prima di invertire il movimento. Il suo controllore
 accetta i segnali **stop**, **forward** e **reverse**.
- 7 Scrivere le dichiarazioni corrispondenti allo schema di Fig. 3.** (5)
- 8 In Fig. 4, HashTable** (1)
 implementa **HTKey**.
 richiede **HTKey**.
 offre **HTKey**.
- 9 In Fig. 4, KeyString** (1)
 realizza **HTKey**.
 dipende da **HTKey**.
 appartiene a **HTKey**.
- 10 In Fig. 4, lasciando HashTable immutata si può sostituire KeyString** (1)
con un'altra classe?
 no, **HashTable** può usare solo chiavi **KeyString**.
 sí, **HashTable** può usare chiavi di altro tipo.
 sí, **HashTable** può usare chiavi di qualsiasi tipo.
- 11 In Fig. 4, Object** (1)
 implementa **HashTable**.
 deriva da **HashTable**.
 appartiene a **HashTable**.
- 12 In Fig. 4, put()** (1)
 è polimorfica.
 è astratta.
 è protetta.
- 13 Il modello a cascata è** (1)
 un metodo di progetto orientato agli oggetti
 un processo di sviluppo del SW con fasi sequenziali separate
 un linguaggio formale di specifica
- 14 I modelli evolutivi** (1)
 sviluppano il sistema in passi incrementali
 si basano sempre su metodi formali
 sono adatti soprattutto ad applicazioni ben conosciute
- 15 Le applicazioni che mantengono grandi quantità di informazioni si** (1)
dicono
 orientate ai dati
 in tempo reale
 orientate agli oggetti
- 16 Le applicazioni che reagiscono a stimoli esterni si dicono** (1)
 orientate alle funzioni
 concorrenti
 orientate al controllo

- 17 **L'analisi dei requisiti è** (1)
 la definizione dei sottosistemi
 la definizione delle proprietà e dei comportamenti richiesti
 la documentazione del processo di sviluppo
- 18 **Cosa significa che il SW è "non lineare"?** (1)
 I sistemi complessi hanno un'architettura a strati.
 Piccole modifiche nel codice causano grandi cambiamenti di comportamento.
 Il grafo di controllo può contenere dei cicli.
- 19 **Cosa s'intende per *information hiding*?** (1)
 Impedire l'accesso a dati personali.
 Impedire l'accesso a dettagli implementativi.
 Impedire l'accesso al codice sorgente.
- 20 **Il test di unità** (1)
 Avviene di solito nella fase di codifica.
 Viene pianificato in fase di analisi e specifica dei requisiti.
 Fa parte della manutenzione del SW.
- 21 **Un modello di processo è** (1)
 una procedura standardizzata
 una generalizzazione di una famiglia di processi di sviluppo
 una metodologia di specifica dei requisiti
- 22 **I sistemi in tempo reale sono caratterizzati da** (1)
 condivisione di risorse.
 vincoli sui tempi di risposta.
 prestazioni elevate.

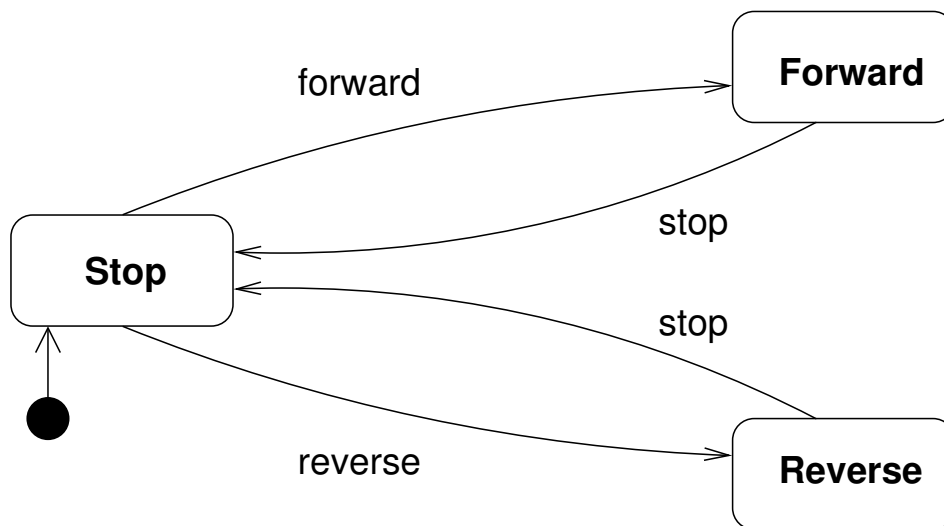


Figura 2: Domanda 6, soluzione.

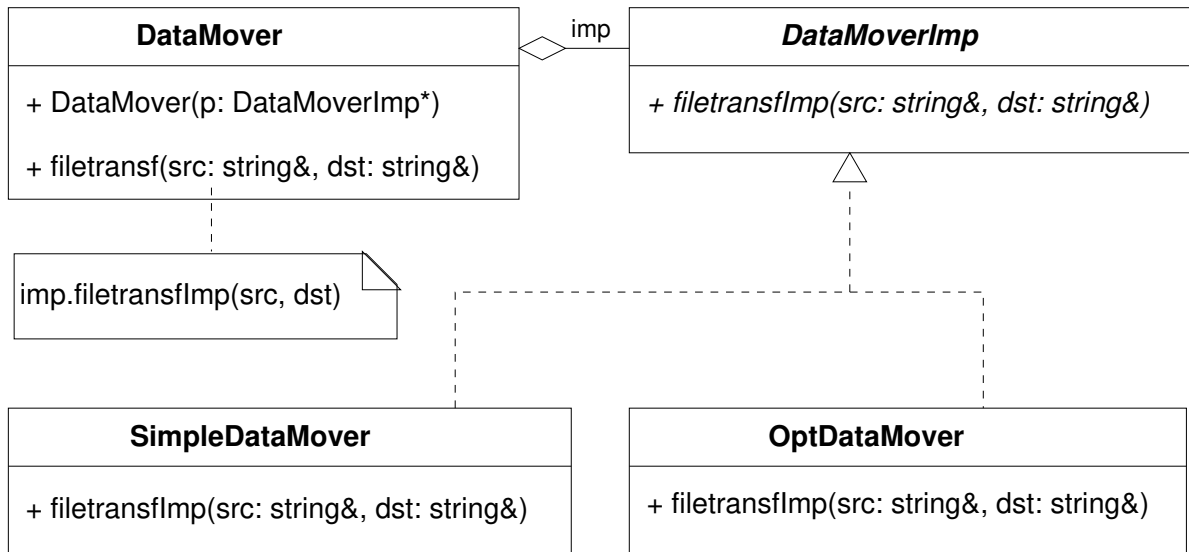


Figura 3: Domanda 7.

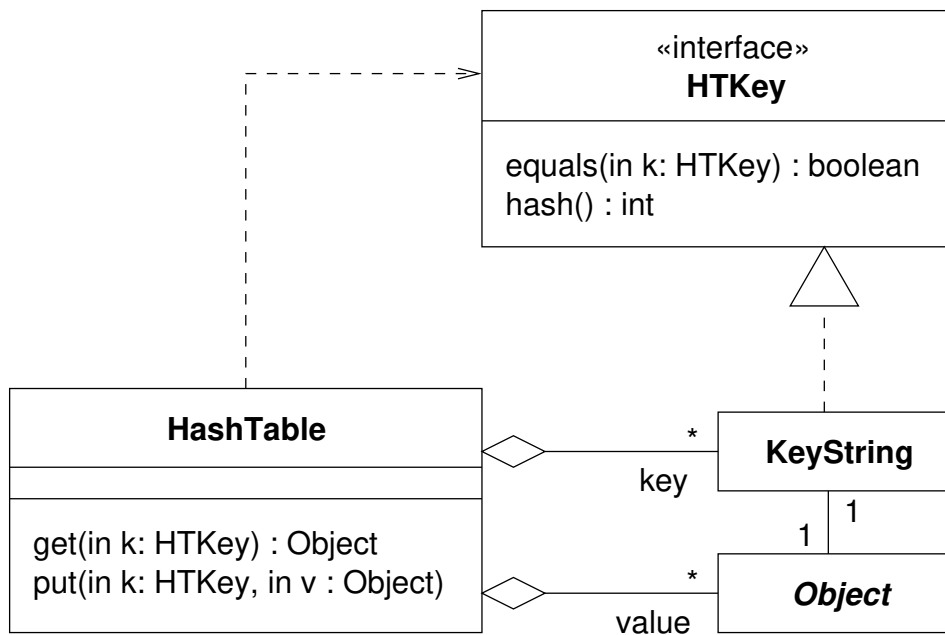


Figura 4: Domande 8-12.

```

#include <iostream>
#include <string>

using namespace std;

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);
};

```

Figura 5: Domanda 7, soluzione.