

SOLUZIONI

Scrivere le risposte (a, b, c oppure V, F) nelle rispettive caselle del file di testo allegato al messaggio inviato dal docente. I candidati devono consegnare entro un'ora dall'inizio della prova, inviando al docente il file delle risposte, usando la funzione "rispondi" del cliente di posta elettronica. Chi si ritira dalla prova lo deve comunicare al docente per posta elettronica.

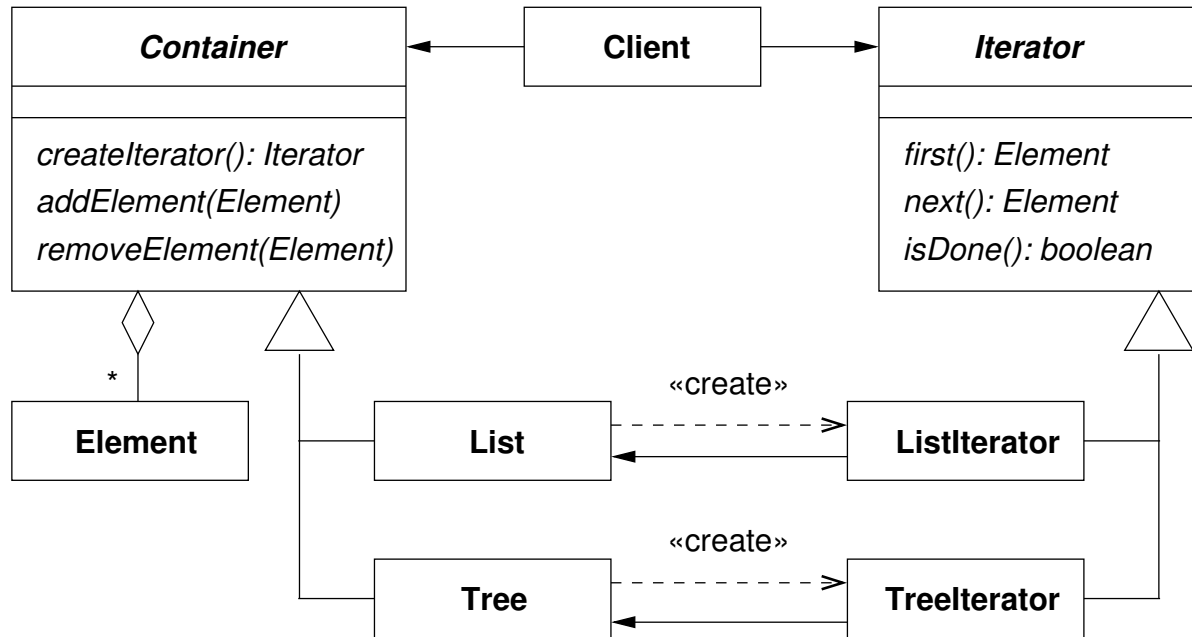


Fig. 1

- A1 In Fig. 1, ListIterator** ha operazioni che
- (a) restituiscono oggetti di tipo **List**. ☐
 - (b) restituiscono oggetti di tipo **Element**. ☒
 - (c) restituiscono oggetti di tipo **Container**. ☐

- A2 In Fig. 1, Tree**
- (a) implementa **Container**. ☒
 - (b) implementa **TreeIterator**. ☐
 - (c) usa **Container**. ☐

- A3 In Fig. 1, Client**
- (a) usa puntatori a **List**. ☐
 - (b) usa puntatori a **ListIterator**. ☐
 - (c) usa puntatori a **Container**. ☒

- A4 In Fig. 1, Client**
- (a) usa puntatori a **TreeIterator**. ☐
 - (b) usa puntatori a **ListIterator**. ☐

(c) usa puntatori a **Iterator**.

☒

A5 In Fig. 1, createiterator()

(a) può essere implementata da **Container**.

☐

(b) deve essere implementata da **List**.

☒

(c) deve essere implementata da **Iterator**.

☐

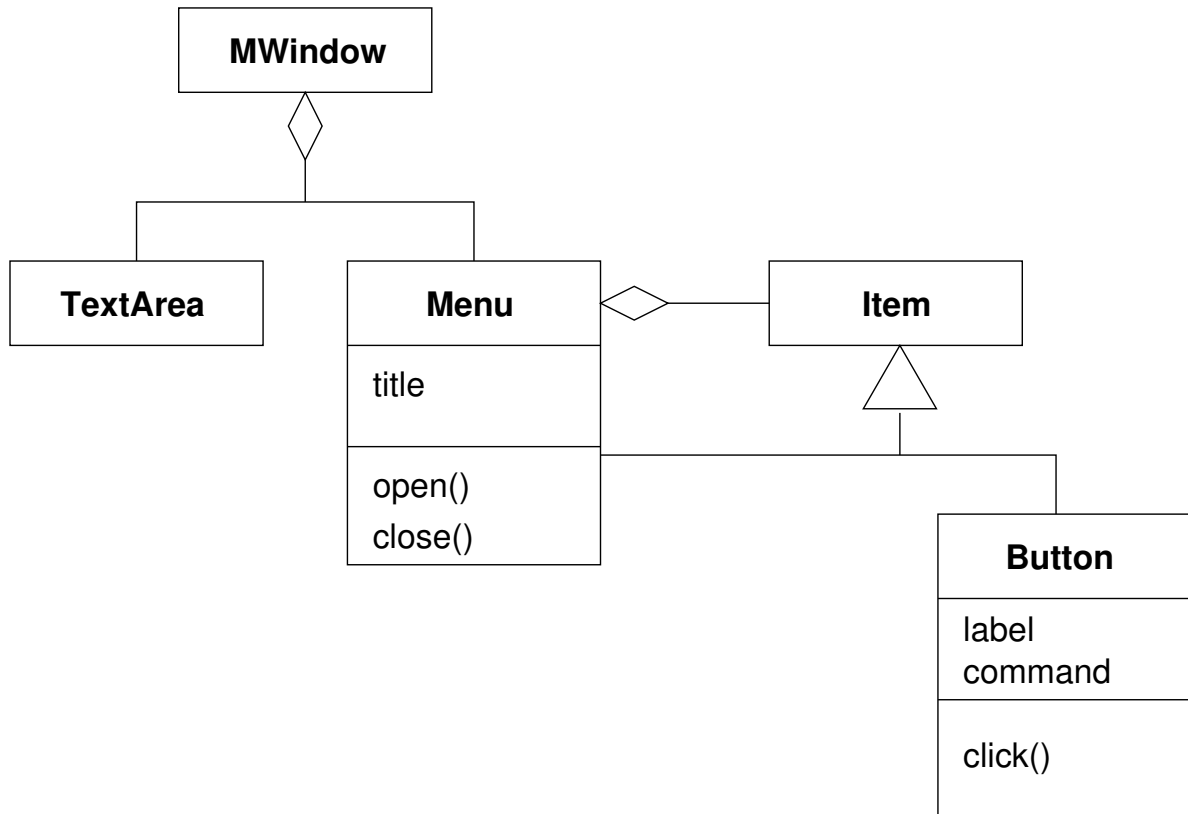


Fig. 2

- B1 In Fig. 2,**
- (a) Un oggetto **Menu** può contenere oggetti **Button** ☒
 - (b) La classe **Menu** deriva dalla classe **Button** ☐
 - (c) La classe **Menu** contiene la classe **Button** ☐
- B2 In Fig. 2,**
- (a) Un oggetto **Button** può contenere oggetti **Menu** ☐
 - (b) La classe **Button** deriva dalla classe **Item** ☒
 - (c) La classe **Button** è base della classe **Item** ☐
- B3 In Fig. 2,**
- (a) La classe **Menu** deriva dalla classe **Mwindow** ☐
 - (b) Un oggetto **Mwindow** può contenere oggetti **Menu** ☒
 - (c) Un oggetto **Menu** può contenere oggetti **Mwindow** ☐
- B4 In Fig. 2,**
- (a) La classe **Item** è base della classe **Button** ☒
 - (b) La classe **Item** contiene la classe **Button** ☐
 - (c) Un oggetto **Button** può contenere oggetti **Item** ☐
- B5 In Fig. 2,**
- (a) **Menu** eredita l'operazione click ☐
 - (b) **Menu** eredita l'operazione open ☐
 - (c) **Menu** implementa l'operazione open ☒

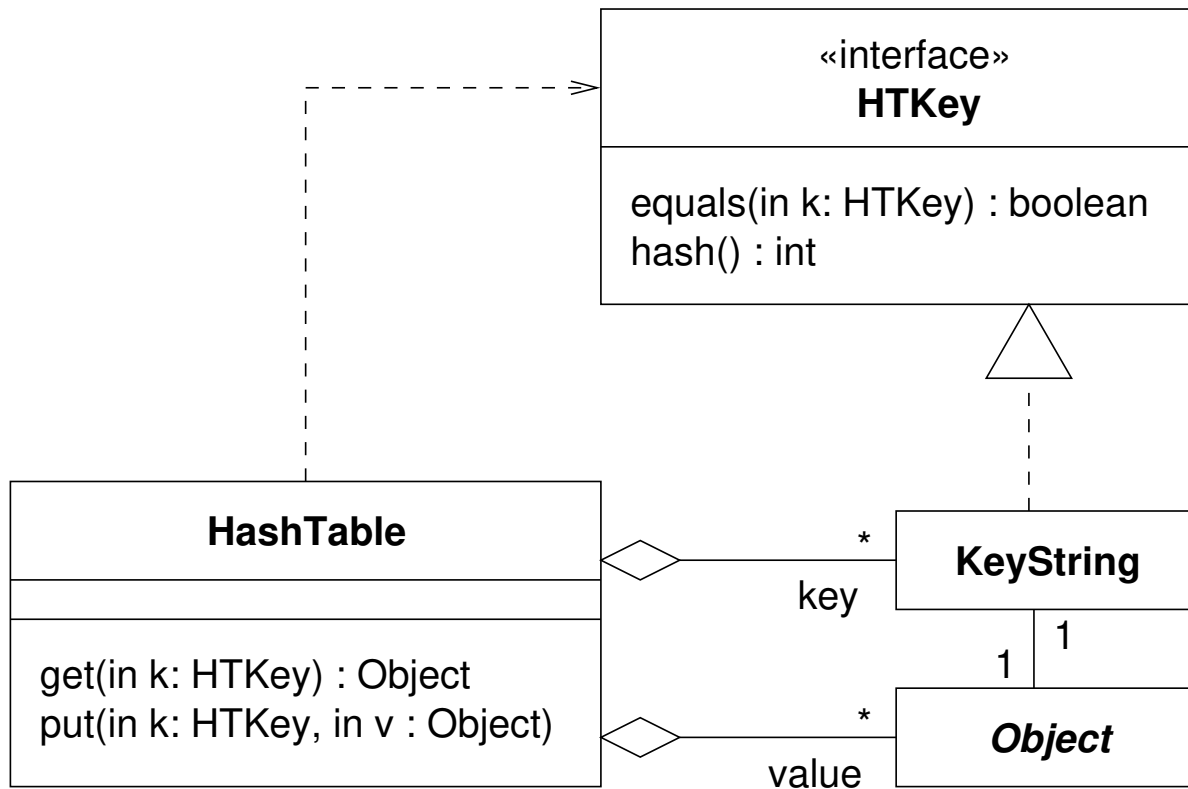


Fig. 3

- C1 In Fig. 3, HashTable**
- (a) implementa **HTKey**. ☐
 - (b) richiede **HTKey**. ☒
 - (c) offre **HTKey**. ☐
- C2 In Fig. 3, lasciando HashTable immutata si può sostituire KeyString con un'altra classe?**
- (a) no, **HashTable** può usare solo chiavi **KeyString**. ☐
 - (b) sí, **HashTable** può usare chiavi di altro tipo. ☒
 - (c) sí, **HashTable** può usare chiavi di qualsiasi tipo. ☐
- C3 In Fig. 3, KeyString**
- (a) realizza **HTKey**. ☒
 - (b) dipende da **HTKey**. ☐
 - (c) appartiene a **HTKey**. ☐
- C4 In Fig. 3, Object**
- (a) implementa **HashTable**. ☐
 - (b) deriva da **HashTable**. ☐
 - (c) appartiene a **HashTable**. ☒
- C5 In Fig. 3, put()**
- (a) è polimorfica. ☒
 - (b) è astratta. ☐
 - (c) è protetta. ☐

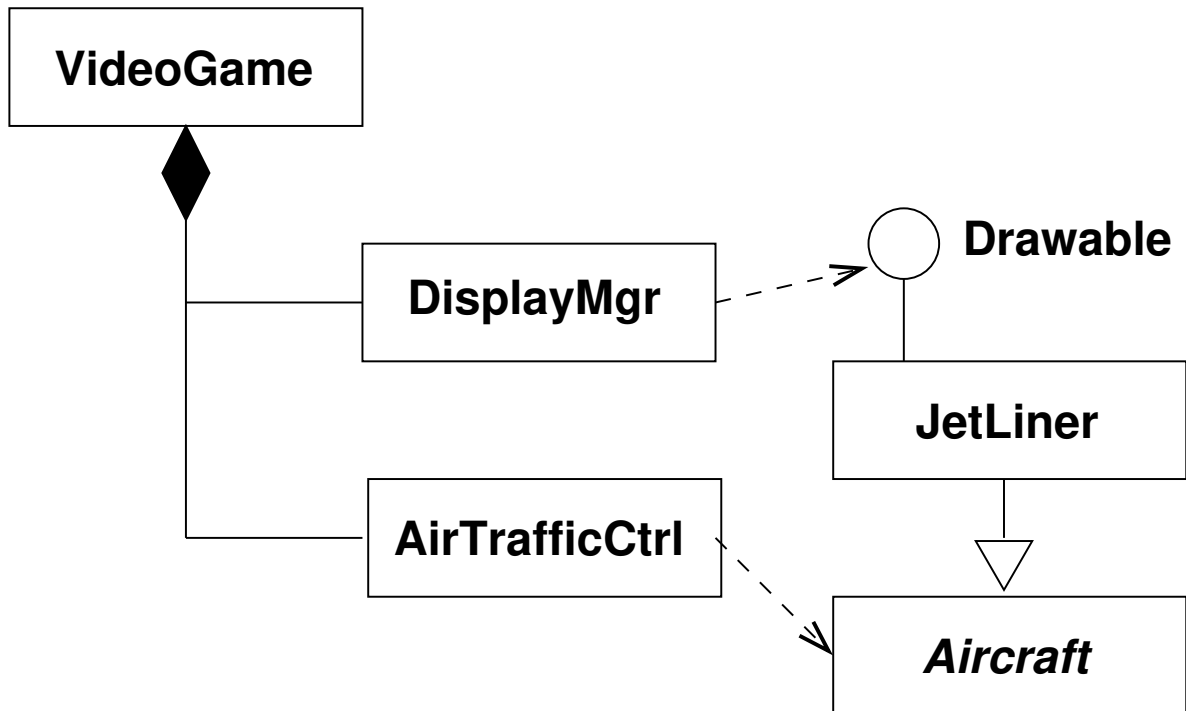


Fig. 4

- D1 In Fig. 4,
- (a) **Drawable** è un'interfaccia richiesta da **Jetliner** ☐
 - (b) **Drawable** è un'interfaccia implementata da **Jetliner** ☒
 - (c) **Drawable** è un'interfaccia implementata da **Aircraft** ☐
- D2 In Fig. 4,
- (a) **DisplayManager** offre l'interfaccia di **Drawable** ☐
 - (b) **Drawable** è un'interfaccia richiesta da **DisplayManager** ☒
 - (c) **DisplayManager** deriva da **Drawable** ☐
- D3 In Fig. 4,
- (a) **JetLiner** offre l'interfaccia di **Aircraft** ☒
 - (b) **Aircraft** deriva da **JetLiner** ☐
 - (c) **JetLiner** usa l'interfaccia di **Aircraft** ☐
- D4 In Fig. 4,
- (a) **AirTrafficCtrl** fa parte di **VideoGame** ☒
 - (b) **AirTrafficCtrl** fa parte di **DisplayManager** ☐
 - (c) **VideoGame** fa parte di **AirTrafficCtrl** ☐
- D5 In Fig. 4,
- (a) **AirTrafficCtrl** implementa **Aircraft** ☐
 - (b) **Aircraft** implementa **AirTrafficCtrl** ☐
 - (c) **AirTrafficCtrl** usa **Aircraft** ☒

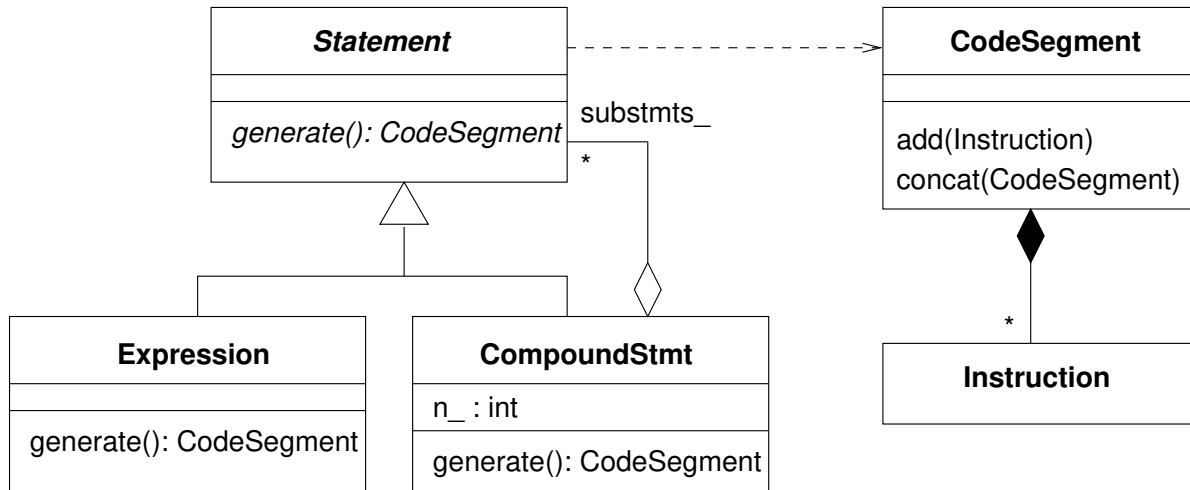


Fig. 5

- E1** In Fig. 5,
- (a) *Statement* implementa **CodeSegment**. ☐
 - (b) *Statement* dipende da **CodeSegment**. ☒
 - (c) **CodeSegment** implementa *Statement*. ☐
- E2** In Fig. 5,
- (a) un **CompoundStmt** può contenere delle **Instruction**. ☐
 - (b) un **CompoundStmt** può contenere dei **CodeSegment**. ☐
 - (c) una **Instruction** fa parte di un **CodeSegment**. ☒
- E3** In Fig. 5,
- (a) una **Expression** può contenere dei **CodeSegment**. ☐
 - (b) una **Expression** può contenere dei **CompoundStmt**. ☐
 - (c) un **CompoundStmt** può contenere delle **Expression**. ☒
- E4** In Fig. 5,
- (a) tutti gli *Statement* sono **Expression**. ☐
 - (b) tutte le **Expression** sono *Statement*. ☒
 - (c) tutti i **CompoundStmt** sono **Expression**. ☐
- E5** In Fig. 5,
- (a) `generate()` restituisce un oggetto di tipo **CodeSegment**. ☒
 - (b) `generate()` ha un argomento di tipo **CodeSegment**. ☐
 - (c) `generate()` ha un argomento di tipo *Statement*. ☐

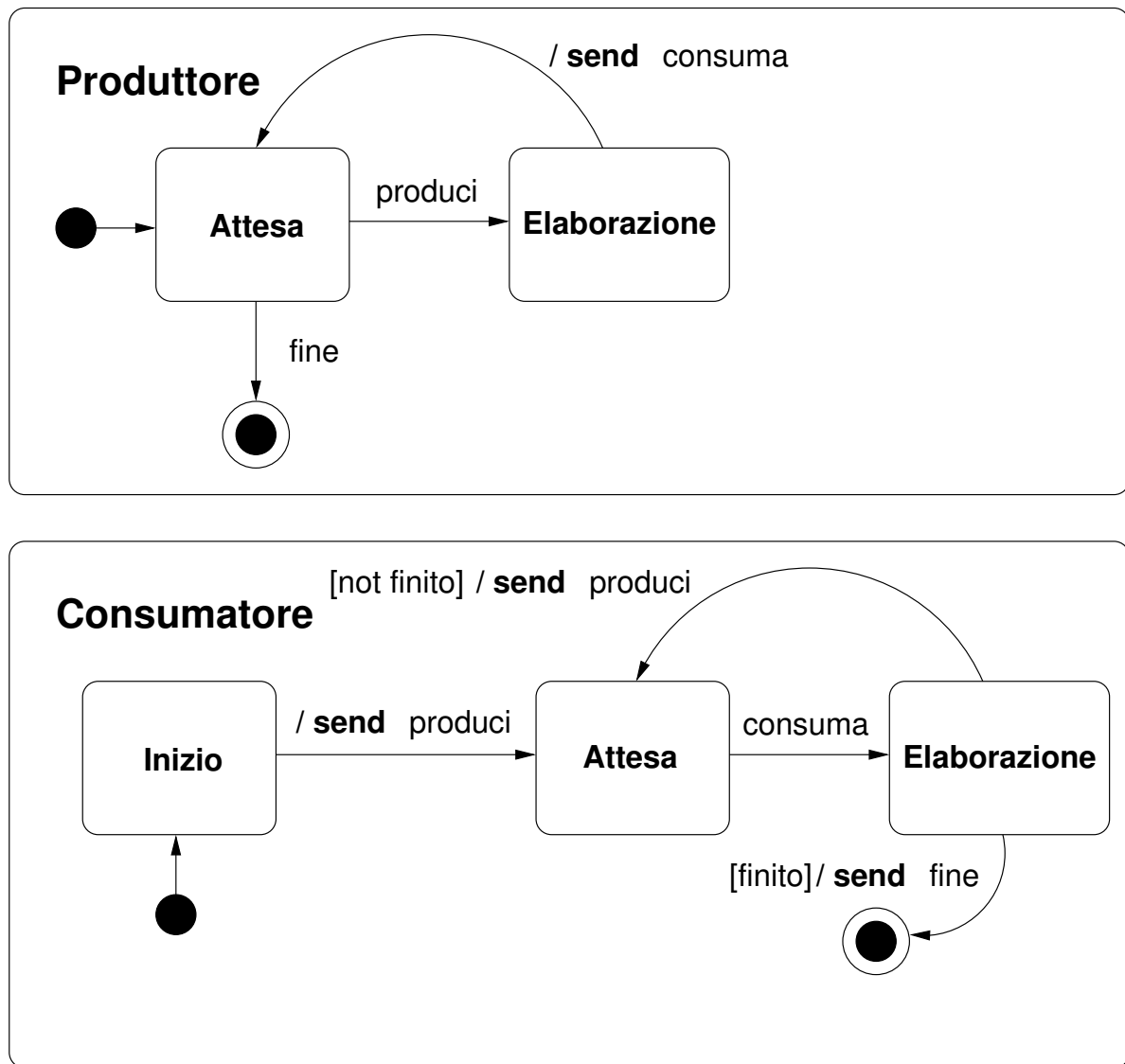


Fig. 6

Con riferimento alla Fig. 6, rispondere alle domande.

	V	F
F1 il Produttore inizia l'interazione	<input type="checkbox"/>	<input checked="" type="checkbox"/>
F3 il Produttore invia il segnale consuma	<input checked="" type="checkbox"/>	<input type="checkbox"/>
F2 il Produttore fa terminare l'interazione	<input type="checkbox"/>	<input checked="" type="checkbox"/>
F4 il Consumatore entra in Elaborazione prima del Produttore	<input type="checkbox"/>	<input checked="" type="checkbox"/>
F5 il Consumatore entra in Attesa quando riceve produci	<input type="checkbox"/>	<input checked="" type="checkbox"/>