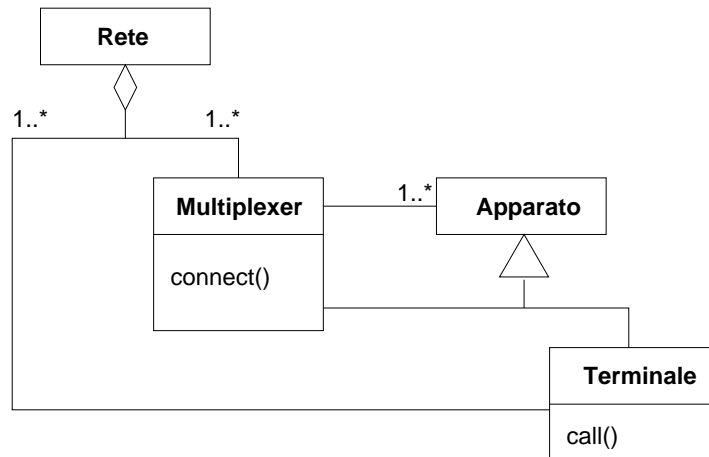


Scrivere le risposte (a, b, c oppure V, F) nelle rispettive caselle del file di testo allegato al messaggio inviato dal docente. I candidati devono consegnare entro 45 minuti dall'inizio della prova, inviando al docente il file di testo delle risposte, usando la funzione "rispondi" del cliente di posta elettronica. Chi si ritira dalla prova lo deve comunicare al docente per posta elettronica.



A1

- (a) La classe **Multiplexer** contiene classe **Terminale**. ☐
- (b) Un'istanza di **Multiplexer** può essere collegato a istanze di **Terminale**. ☒
- (c) La classe **Multiplexer** deriva dalla classe **Terminale**. ☐

A2

- (a) Un oggetto **Multiplexer** può contenere oggetti **Rete**. ☐
- (b) La classe **Multiplexer** deriva dalla classe **Rete**. ☐
- (c) Un oggetto **Rete** può contenere oggetti **Multiplexer**. ☒

A3

- (a) La classe **Terminale** è base della classe **Apparato**. ☐
- (b) Un oggetto **Terminale** può essere collegato ad un oggetto **Multiplexer**. ☒
- (c) La classe **Terminale** deriva dalla classe **Multiplexer**. ☐

A4

- (a) Un oggetto **Terminale** può contenere oggetti **Apparato**. ☐
- (b) La classe **Apparato** è base della classe **Terminale**. ☒
- (c) La classe **Apparato** contiene la classe **Terminale**. ☐

A5

- (a) **Multiplexer** offre l'operazione **connect**. ☒
- (b) **Multiplexer** eredita l'operazione **call**. ☐
- (c) **Multiplexer** eredita l'operazione **connect**. ☐

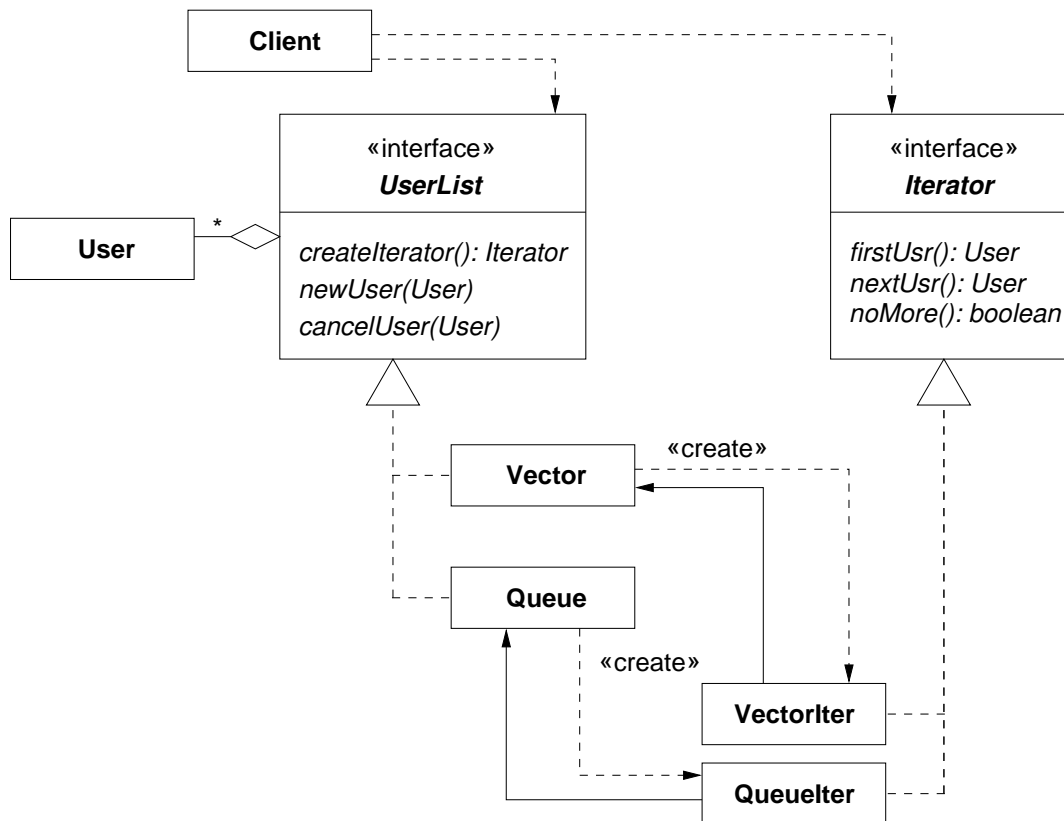
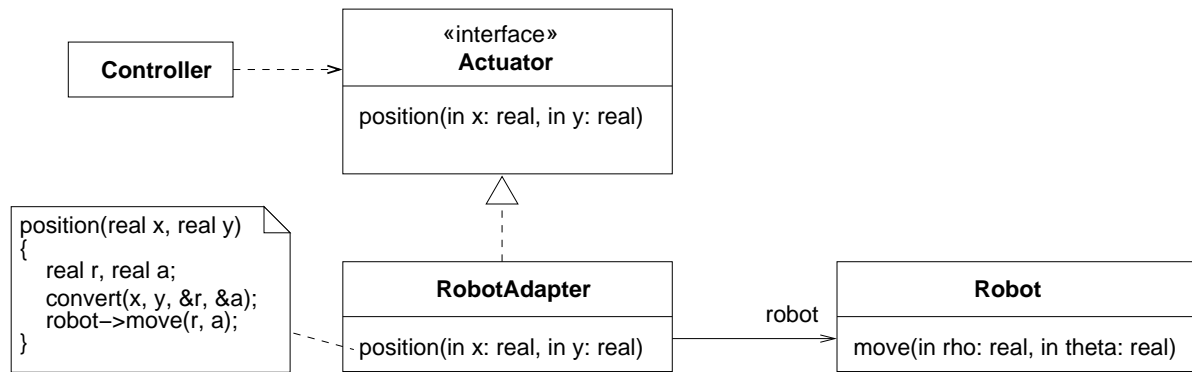


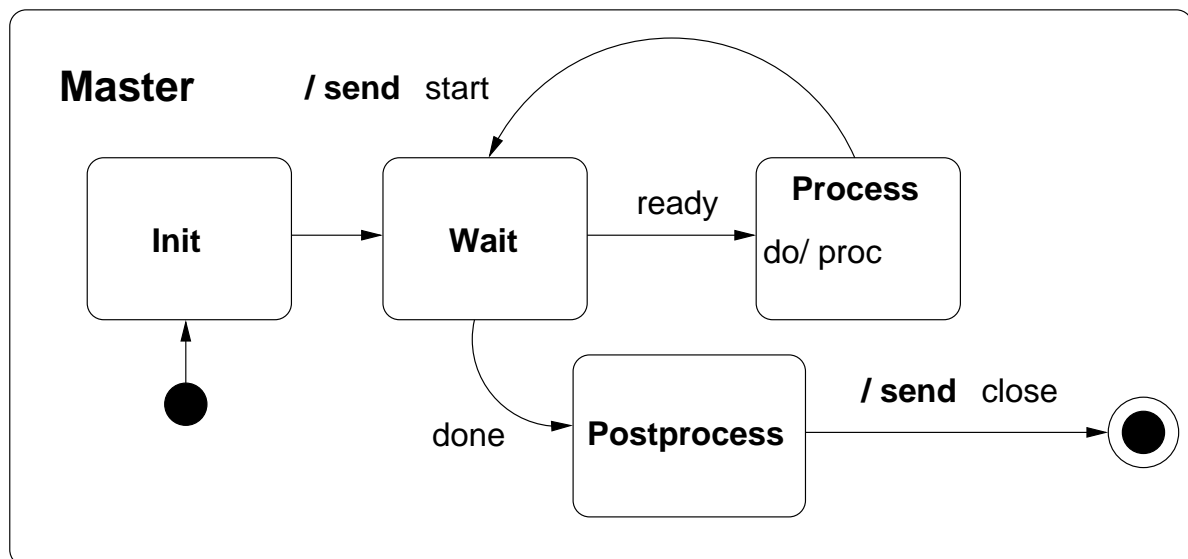
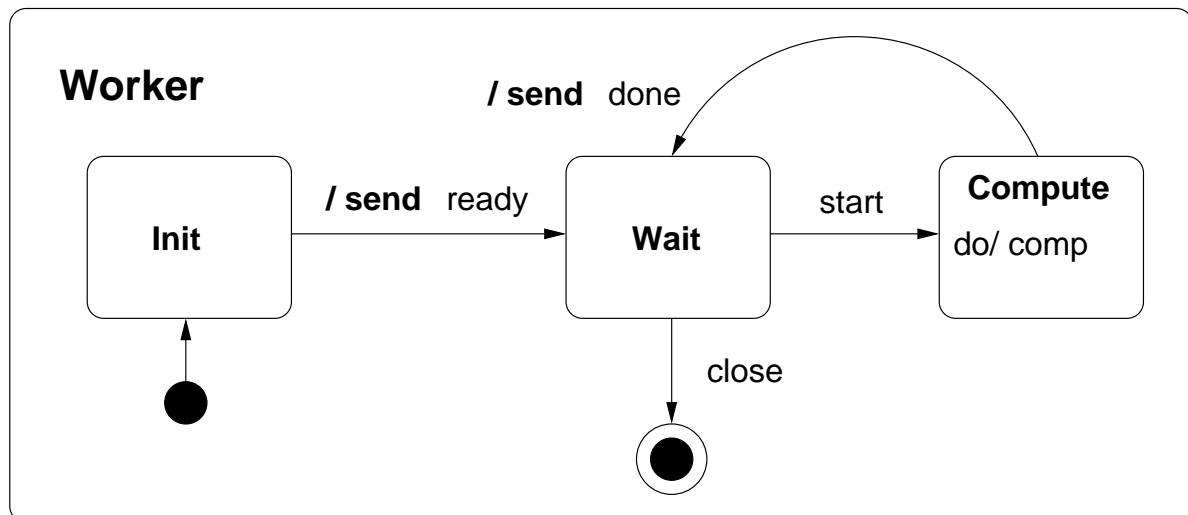
Fig. 4

- B1 VectorIter** ha operazioni che
- (a) restituiscono oggetti di tipo **UserList** ☐
 - (b) restituiscono oggetti di tipo **Vector**. ☐
 - (c) restituiscono oggetti di tipo **User**. ☒
- B2 Client**
- (a) usa puntatori a **UserList**. ☒
 - (b) usa puntatori a **Vector**. ☐
 - (c) usa puntatori a **VectorIter**. ☐
- B3 Queue**
- (a) usa **UserList**. ☐
 - (b) realizza **UserList**. ☒
 - (c) realizza **QueueIter**. ☐
- B4 Client**
- (a) usa puntatori a **Iterator**. ☒
 - (b) usa puntatori a **QueueIter**. ☐
 - (c) usa puntatori a **VectorIter**. ☐
- B5 createIterator()**
- (a) deve essere implementata da **Iterator**. ☐
 - (b) deve essere implementata da **UserList**. ☐
 - (c) deve essere implementata da **Vector**. ☒

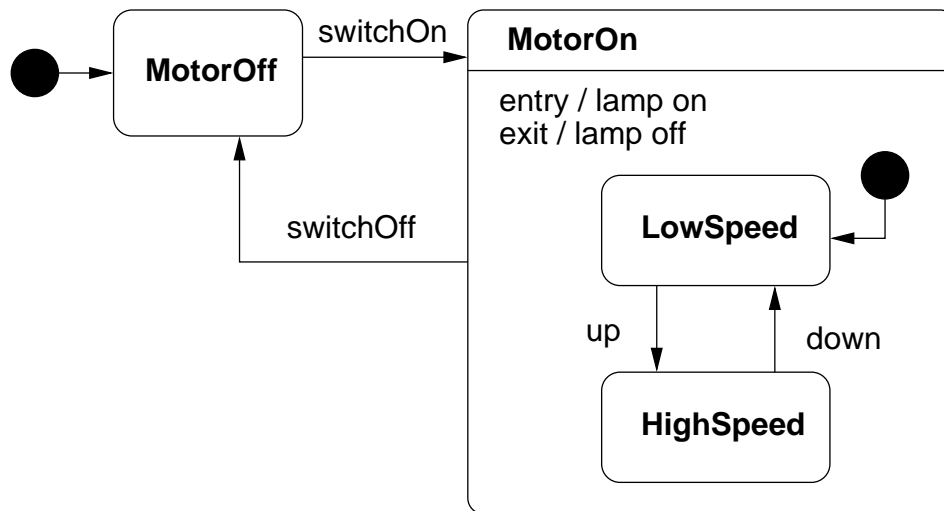


- C1 **RobotAdapter** usa l'interfaccia di **Robot**.
 C2 **Client** usa l'interfaccia di **Robot**.
 C3 **Actuator** implementa **Client**.
 C4 **Client** dipende da **Actuator**.
 C5 **Robot** realizza **Actuator**.

V	F
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>

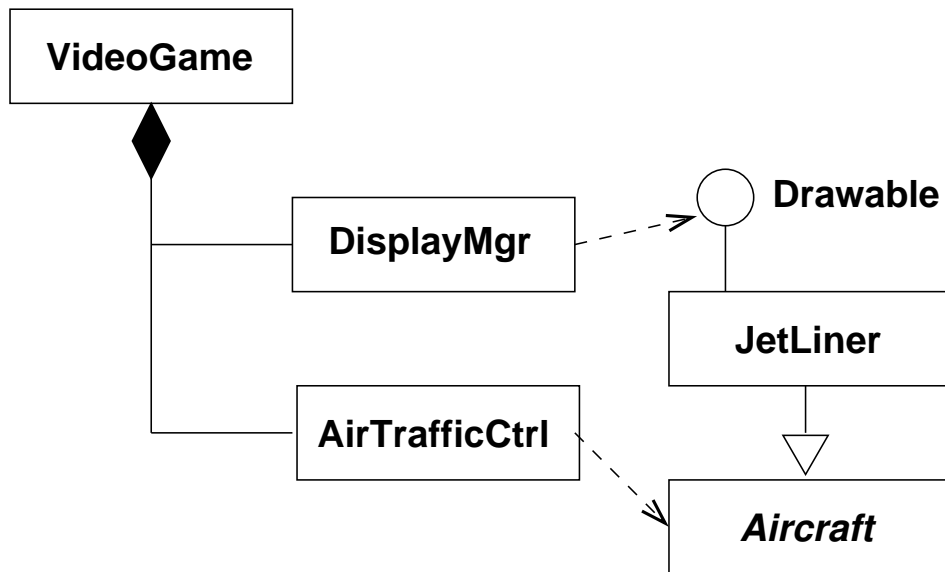


- | | V | F |
|------------------------------------------------------------------------------------|-------------------------------------|-------------------------------------|
| D1 Il processo Master manda l'evento done. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| D2 Il primo messaggio viene mandato da Worker . | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| D3 Il processo Worker può essere interrotto mentre esegue send done. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| D4 Il processo Master termina quando riceve l'evento done. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| D5 L'attività comp segue l'attività proc . | <input type="checkbox"/> | <input checked="" type="checkbox"/> |



- E1** L'evento **switchOn** causa sempre l'accensione del motore.
- E2** Lo stato iniziale del sistema è **LowSpeed**.
- E3** Durante il funzionamento c'è sempre una luce accesa.
- E4** Il sistema non ha uno stato finale.
- E5** Il motore si può spegnere (**switchOff**) solo nello stato **LowSpeed**.

V	F
<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>



F1

- (a) **Drawable** è un'interfaccia implementata da **Aircraft**.
- (b) **Drawable** è un'interfaccia richiesta da **Jetliner**.
- (c) **Drawable** è un'interfaccia implementata da **Jetliner**.

☐
☐
☒

F2

- (a) **JetLiner** usa l'interfaccia di **Aircraft**.
- (b) **JetLiner** offre l'interfaccia di **Aircraft**.
- (c) **Aircraft** deriva da **JetLiner**.

☐
☒
☐

F3

- (a) **DisplayManager** deriva da **Drawable**.
- (b) **DisplayManager** offre l'interfaccia di **Drawable**.
- (c) **Drawable** è un'interfaccia richiesta da **DisplayManager**.

☐
☐
☒

F4

- (a) **VideoGame** fa parte di **AirTrafficCtrl**.
- (b) **AirTrafficCtrl** fa parte di **VideoGame**.
- (c) **AirTrafficCtrl** fa parte di **DisplayManager**.

☐
☒
☐

F5

- (a) **AirTrafficCtrl** usa **Aircraft**.
- (b) **AirTrafficCtrl** implementa **Aircraft**.
- (c) **Aircraft** implementa **AirTrafficCtrl**.

☒
☐
☐