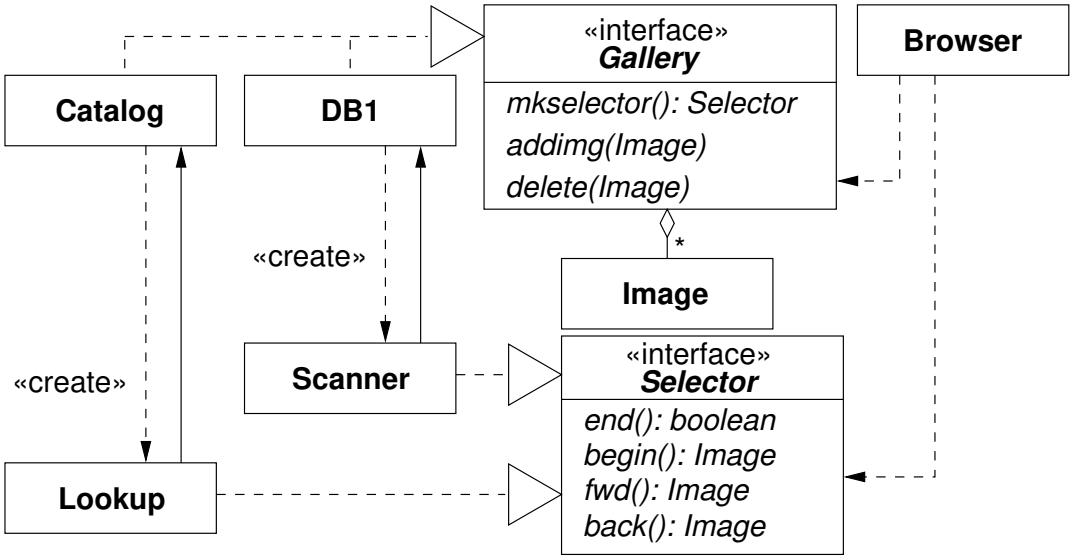
Esame di Ingegneria del software, 1^o luglio 2021

prova a distanza

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.



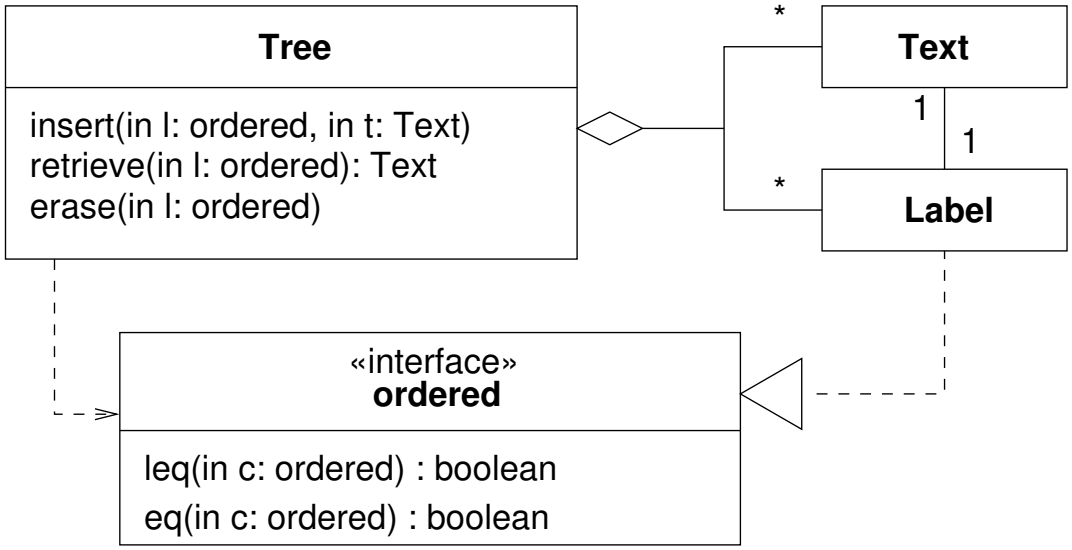
A1 **Scanner** ha operazioni che
(a) restituiscono oggetti di tipo **DB1**. ☐
(b) restituiscono oggetti di tipo **Image**. ☒
(c) restituiscono oggetti di tipo **Gallery** ☐

A2 **Browser**
(a) usa puntatori a **DB1**. ☐
(b) usa puntatori a **Scanner**. ☐
(c) usa puntatori a **Gallery**. ☒

A3 **Catalog**
(a) realizza **Gallery**. ☒
(b) realizza **Lookup**. ☐
(c) usa **Gallery**. ☐

A4 **Browser**
(a) usa puntatori a **Lookup**. ☐
(b) usa puntatori a **Scanner**. ☐
(c) usa puntatori a **Selector**. ☒

A5 mkselector()
(a) deve essere implementata da **Gallery**. ☐
(b) deve essere implementata da **DB1**. ☒
(c) deve essere implementata da **Selector**. ☐



B1 **Tree**

(a) implementa **ordered**. ☐

(b) richiede **ordered**. ☒

(c) offre **ordered**. ☐

B2 **Label**

(a) realizza **ordered**. ☒

(b) dipende da **ordered**. ☐

(c) appartiene a **ordered**. ☐

B3 Lasciando **Tree** immutata, si può sostituire **Label** con un'altra classe?

(a) no, **Tree** può usare solo chiavi **Label**. ☐

(b) sí, **Tree** può usare chiavi di altro tipo. ☒

(c) sí, **Tree** può usare chiavi di qualsiasi tipo. ☐

B4 **Text**

(a) implementa **Tree**. ☐

(b) deriva da **Tree**. ☐

(c) appartiene a **Tree**. ☒

B5 insert()

(a) è polimorfica. ☒

(b) è astratta. ☐

(c) è protetta. ☐

C1 Una tautologia è

(a) vera in qualsiasi interpretazione. ☒

(b) falsa in qualsiasi interpretazione. ☐

(c) indecidibile in qualsiasi interpretazione. ☐

C2 In UML, la relazione *A realizza B* significa:

(a) A eredita da B. ☐

(b) A e B hanno la stessa interfaccia. ☐

(c) A implementa l'interfaccia di B. ☒

C3 Nel calcolo proposizionale la *funzione di valutazione*

(a) assegna un valore ai simboli proposizionali. ☒

(b) assegna un valore ai connettivi. ☐

(c) assegna un valore alle formule. ☐

C4 Nel modello orientato agli oggetti, un *legame* è

(a) un'istanza di un'associazione. ☒

(b) un'istanza di una generalizzazione. ☐

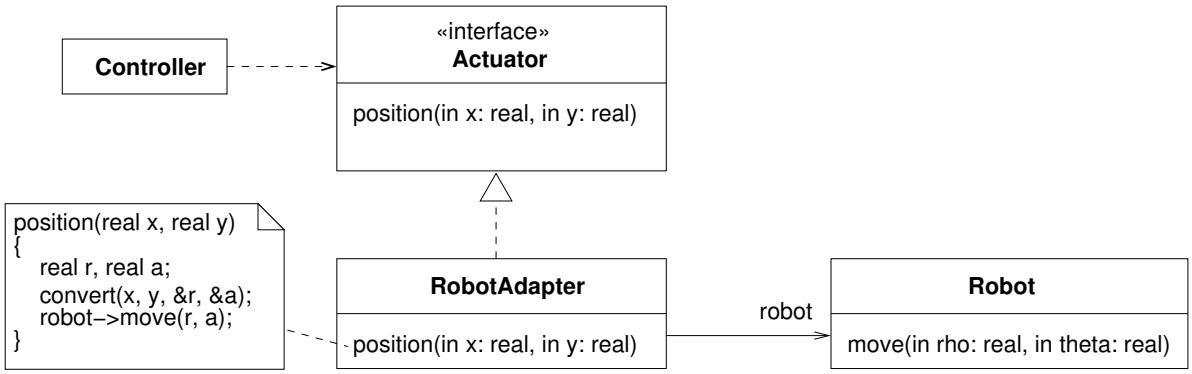
(c) uno stereotipo di associazione. ☐

C5 In un sistema formale completo

(a) tutte le formule dimostrabili sono valide. ☐

(b) tutte le formule valide sono dimostrabili. ☒

(c) tutti gli assiomi sono validi. ☐



D1

(a) **Controller** usa l'interfaccia di **Robot**. ☐

(b) **Controller** è implementato da umlnameRobotAdapter. ☐

(c) umlnameRobotAdapter usa l'interfaccia di **Robot**. ☒

D2

(a) **Controller** dipende da **Actuator**. ☒

(b) **Actuator** implementa **Controller**. ☐

(c) **Actuator** dipende da **RobotAdapter**. ☐

D3

(a) **Robot** realizza **Actuator**. ☐

(b) **RobotAdapter** realizza **Actuator**. ☒

(c) **RobotAdapter** realizza **Robot**. ☐

D4

(a) **Robot** dipende da **Controller**. ☐

(b) **Controller** non dipende da **Robot**. ☒

(c) **Controller** usa **Robot**. ☐

D5

(a) **RobotAdapter**::position è implementata per polimorfismo. ☐

(b) **RobotAdapter**::position è implementata per incapsulamento. ☐

(c) **RobotAdapter**::position è implementata per delega. ☒

E1 Un modulo fisico è costituito da

(a) uno o piú circuiti integrati. ☐
(b) uno o piú file. ☒
(c) uno o piú diagrammi dei componenti. ☐

E2 Un'operazione *protetta*

(a) viene usata solo nella sua classe e classi derivate. ☒
(b) viene usata solo nelle classi dello stesso package. ☐
(c) chiede la password. ☐

E3 Il criterio di copertura delle condizioni si usa

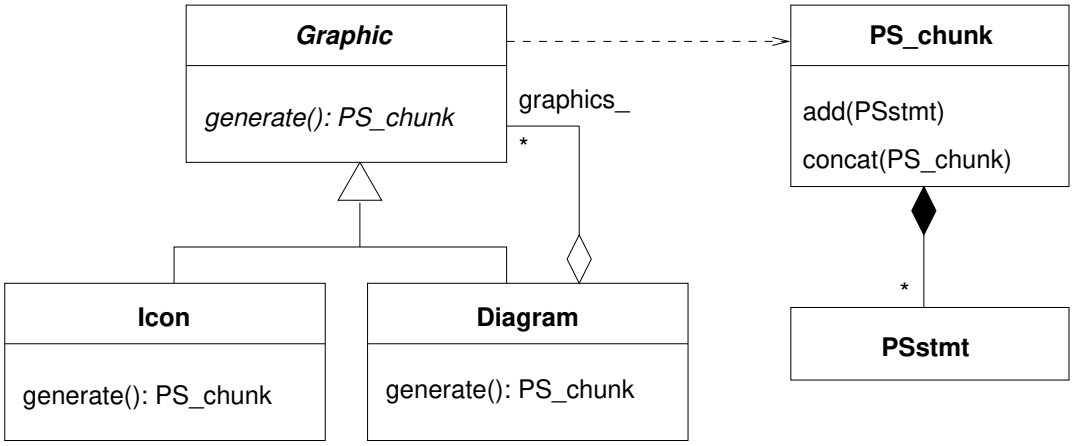
(a) nel test funzionale. ☐
(b) nel test di accettazione. ☐
(c) nel test strutturale. ☒

E4 Gli strumenti *CASE* servono

(a) a fare dei diagrammi. ☐
(b) a creare interfacce grafiche. ☐
(c) a definire dei modelli. ☒

E5 I *design pattern* sono

(a) dei moduli orientati agli oggetti. ☐
(b) degli schemi di soluzioni per problemi ricorrenti. ☒
(c) un linguaggio di progetto. ☐



F1

(a) *Graphic* implementa **PS_chunk**. ☐
(b) *Graphic* dipende da **PS_chunk**. ☒
(c) **PS_chunk** implementa *Graphic*. ☐

F2

(a) una **Icon** può contenere dei **PS_chunk**. ☐
(b) una **Icon** può contenere dei **Diagram**. ☐
(c) un **Diagram** può contenere delle **Icon**. ☒

F3

(a) un **Diagram** può contenere dei **PSstmt**. ☐
(b) un **Diagram** può contenere dei **PS_chunk**. ☐
(c) un **PSstmt** fa parte di un **PS_chunk**. ☒

F4

(a) tutti i *Graphic* sono **Icon**. ☐
(b) tutte le **Icon** sono *Graphic*. ☒
(c) tutti i **Diagram** sono **Icon**. ☐

F5

(a) generate() restituisce un oggetto di tipo **PS_chunk**. ☒
(b) generate() ha un argomento di tipo **PS_chunk**. ☐
(c) generate() ha un argomento di tipo *Graphic*. ☐