

1 **UML/RUP**

Markera om följande påståenden är sanna eller falska:
(+1 för rätt svar, ingen förändring för fel svar)

Ett klassdiagram beskriver hur klasser och objekt samarbetar.

- ☐ Sant
- ☐ Falskt ✓

Man skriver use cases efter att man implementerat systemet för att ha något att testa mot.

- ☐ Sant
- ☐ Falskt ✓

Design patterns beskriver hur man löser vanliga interaktioner med användarna av systemet i use cases.

- ☐ Sant
- ☐ Falskt ✓

Interaktionsdiagram visar på vilka metodanrop olika objekt gör på andra objekt.

- ☐ Sant ✓
- ☐ Falskt

kommunikationsdiagram visar hur användare kommunicerar med systemet.

- ☐ Sant
- ☐ Falskt ✓

I ett klassdiagram ser man alla värden på alla attribut i klasserna.

- ☐ Sant
- ☐ Falskt ✓

I ett sekvensdiagram kan man se vilka objekt som behövs för att lösa en viss systemhändelse.

- ☐ Sant ✓
- ☐ Falskt

I ett sekvensdiagram kan man se vilka klasser som objekten är instanser av.

- ☐ Sant ✓
- ☐ Falskt

Ett systemsekvensdiagram är ett specialfall av klassdiagram.

- ☐ Sant
- ☐ Falskt ✓

Metoder i en klass kan vara public, protected, eller private.

- ☒ Sant
- ☐ Falskt



En metod som är deklarerad som public får inte använda sig av attribut i samma klass som är private.

- ☐ Sant
- ☒ Falskt



Om objektet alpha av typen Greek skall kunna anropa metoder i objektet aleph av typen Hebrew så måste det åtminstone finnas en association mellan klasserna Greek och Hebrew.

- ☒ Sant
- ☐ Falskt



Ett system är inte färdigt för leverans om inte alla use cases är helt implementerade.

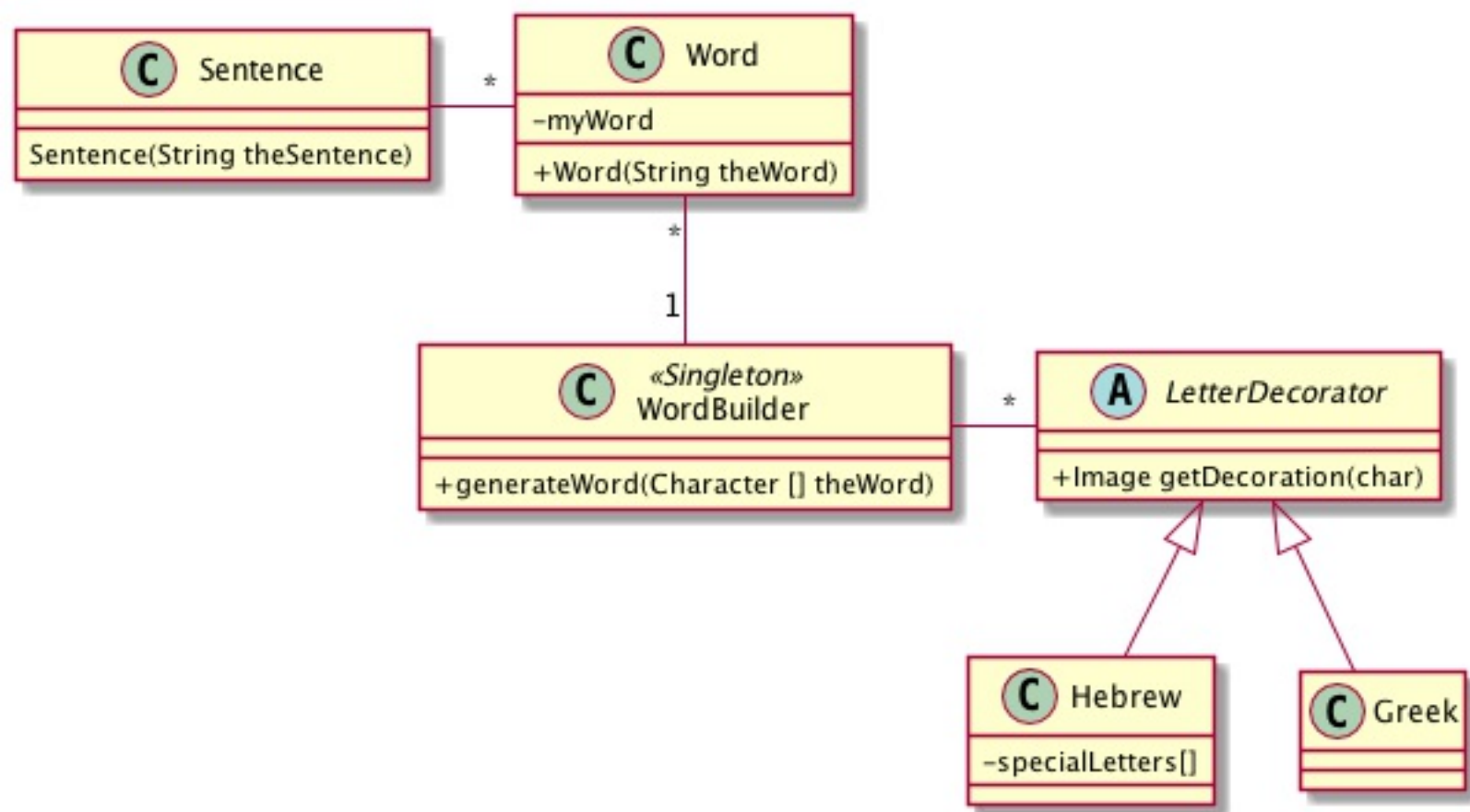
- ☐ Sant
- ☒ Falskt



Totalpoäng: 13

2 Klassdiagram

Givet följande klassidiagram:



Markera om följande påståenden är sanna eller falska:
(+1 för rätt svar, ingen förändring för fel svar)

En Sentence behöver inte innehålla några Word.

- ☐ Sant
- ☐ Falskt



Det får bara finnas ett objekt av typen WordBuilder i systemet.

- ☐ Sant
- ☐ Falskt



wb:WordBuilder kan användas av flera Word.

- ☐ Sant
- ☐ Falskt



Det måste finnas minst ett objekt av typen LetterDecorator i systemet.

- ☐ Sant
- ☐ Falskt



Eftersom Sentence bara har en konstruktor så måste den få en sträng med ord när den skapas.

- ☐ Sant
- ☐ Falskt



Ordet w:Word kan innehålla tecken från både Hebreiska och Grekiska.

- ☐ Sant
- ☐ Falskt



Word får enbart innehålla Hebreiska eller Grekiska tecken.

- ☐ Sant
- ☐ Falskt
- ✓

WordBuilder, LetterDecorator, Hebrew, och Greek bildar tillsammans designmönstret Strategy.

- ☐ Sant
- ☐ Falskt
- ✓

Metoden WordBuilder::generateWord() kan läsa attributet Hebrew::specialLetters för att avgöra om ordet är på Hebreiska.

- ☐ Sant
- ☐ Falskt
- ✓

Enligt GRASP är WordBuilder är en Controller.

- ☐ Sant
- ☐ Falskt
- ✓

Enligt GRASP är Sentence en Creator av Word-objekt.

- ☐ Sant
- ☐ Falskt
- ✓

Enligt GRASP-mönstret Information Expert skall WordBuilder avgöra vilket språk som skall användas eftersom den vet vilket ordet är.

- ☐ Sant
- ☐ Falskt
- ✓

Enligt GRASP-mönstret Information Expert skall LetterDecorator avgöra vilket språk som skall användas eftersom den vet vilka språk som finns.

- ☐ Sant
- ☐ Falskt
- ✓

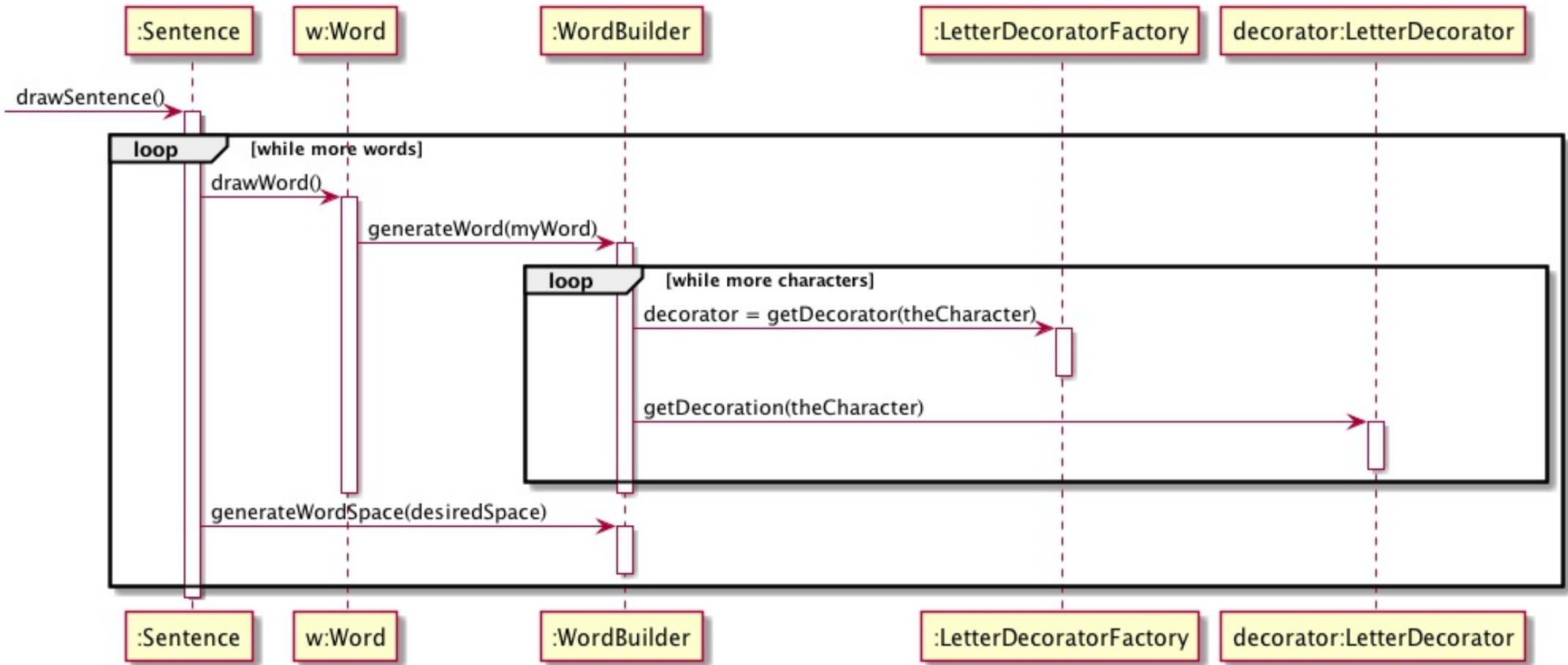
Enligt GRASP-mönstren Information Expert och Pure Fabrication borde vi skapa en ny klass LetterDecoratorFactory för att skapa rätt LetterDecorator för varje bokstav i ordet.

- ☐ Sant
- ☐ Falskt
- ✓

Totalpoäng: 14

3 Sekvensdiagram

Givet följande sekvensdiagram:



Markera om följande påståenden är sanna eller falska:
(+1 för rätt svar, ingen förändring för fel svar)

Detta sekvensdiagram börjar med systemhändelsen drawSentence()

- ☐ Sant
- ☐ Falskt
- ✓

Sentence::drawSentence() fortsätter köra samtidigt som Word::DrawWord() körs.

- ☐ Sant
- ☐ Falskt
- ✓

W:Word och decorator:LetterDecorator är de enda objekten i det här diagrammet. Alla andra metoder anropas direkt mot klassen.

- ☐ Sant
- ☐ Falskt
- ✓

Enligt GRASP-mönstret Information Expert skall metoden drawWord() ligga i klassen Sentence

- ☐ Sant
- ☐ Falskt
- ✓

Enligt GRASP är Sentence en Controller för att rita en mening.

- ☐ Sant
- ☐ Falskt
- ✓

4 GRASP och Design Patterns

Markera om följande påståenden är sanna eller falska:
(+1 för rätt svar, ingen förändring för fel svar)

Controllers kan anropa andra Controllers.

- ☐ Sant
- ☐ Falskt
- ✓

Det får bara finnas en instans av en Information Expert.

- ☐ Sant
- ☐ Falskt
- ✓

En Controller är ansvarig för att ta emot systemhändelser och delegera till andra klasser för att genomföra den önskade operationen.

- ☐ Sant
- ☐ Falskt
- ✓

Information Expert betyder att ansvaret för att arbeta med en viss information bör ligga i den klass som innehåller informationen.

- ☐ Sant
- ☐ Falskt
- ✓

Information Expert betyder att man skall lägga informationen i den klass som har metoderna för att hantera den.

- ☐ Sant
- ☐ Falskt
- ✓

Enligt High Cohesion skall man se till att varje klass skall ha så få olika ansvarsområden som möjligt.

- ☐ Sant
- ☐ Falskt
- ✓

Singleton betyder att man bara får anropa klassen en gång

- ☐ Sant
- ☐ Falskt
- ✓

Strategy pattern använder sig av polymorfism

- ☐ Sant
- ☐ Falskt
- ✓

Abstract Factory används för att skapa rätt typ av objekt givet ett visst kontext, där resten av systemet inte behöver veta exakt vilken typ objektet är.

- ☒ Sant ✓
- ☐ Falskt

När man använder Layered så har man alltid ett GUI-lager, ett logik-lager, och ett data-lager.

- ☐ Sant
- ☒ Falskt ✓

I **Model-View-Controller** måste man ha flera Modeller som presenteras i samma View.

- ☐ Sant
- ☒ Falskt ✓

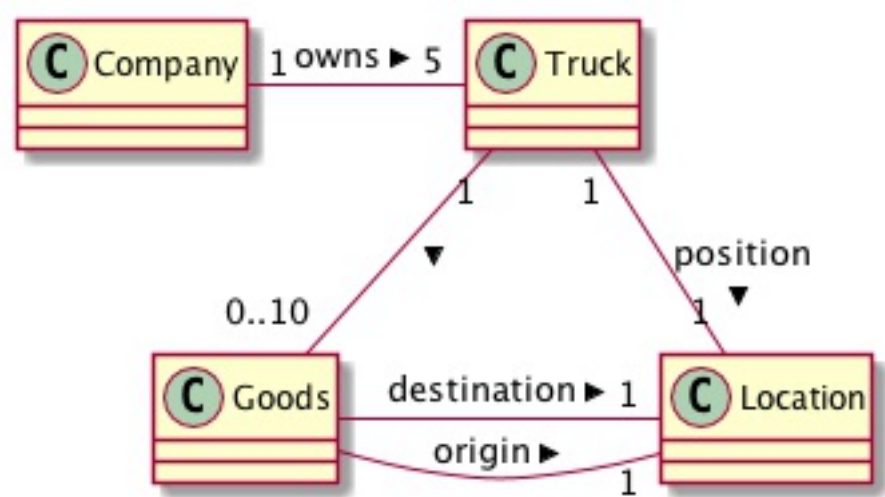
Indirection är en testningsteknik där man "smyger sig på" metoderna och hoppas att de skall bli överraskade och ge fel svar.

- ☐ Sant
- ☒ Falskt ✓

Totalpoäng: 12

5 **Klassdiagram II**

Givet följande klassdiagram:



Markera om följande påståenden är sanna eller falska:
(+1 för rätt svar, ingen förändring för fel svar)

Det får bara finnas fem Truckar i systemet.

- ☐ Sant
- ☐ Falskt



Enbart en Truck åt gången får ha en Location.

- ☐ Sant
- ☐ Falskt



Varje objekt av typen Goods har två relationer till objekt av typen Location.

- ☐ Sant
- ☐ Falskt



Alla Company måste äga exakt fem Truckar.

- ☐ Sant
- ☐ Falskt



En Truck vet inte direkt vart den är på väg utan att kolla med sin last av Goods först.

- ☐ Sant
- ☐ Falskt



I det här systemet kommer det alltid finnas fem gånger fler objekt av typen Truck än objekt av typen Company.

- ☐ Sant
- ☐ Falskt



Man kan flytta objekt av typen Goods till en annan Truck.

- ☐ Sant
- ☐ Falskt



Man kan inte lasta av objekt av typen Goods och lämna dem utan Truck.

☒ Sant

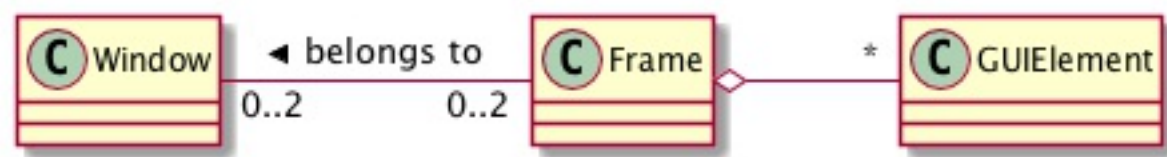


☐ Falskt

Totalpoäng: 8

6 Klassdiagram III

Givet följande klassdiagram:



Markera om följande påståenden stöds (sant) eller inte (falskt) av klassdiagrammet:
(+1 för rätt svar, ingen förändring för fel svar)

stray:Frame hör inte till något Window

- ☐ Sant
- ☐ Falskt
- ✓

win:Window har bara en main:Frame

- ☐ Sant
- ☐ Falskt
- ✓

white:Window har inga Frame

- ☐ Sant
- ☐ Falskt
- ✓

f:Frame används i w1:Window och w2:Window

- ☐ Sant
- ☐ Falskt
- ✓

main:Window använder setup:Frame och control:Frame

- ☐ Sant
- ☐ Falskt
- ✓

dummy:Frame har inga GUIElement

- ☐ Sant
- ☐ Falskt
- ✓

Totalpoäng: 6