

1 Teori

Markera om följande påståenden är sanna eller falska:

(+1 för rätt svar, ingen förändring för fel svar)

Ett use case-diagram visar hur man använder en viss klass.

☐ Sant

☐ Falskt



Det är först när man har ett klassdiagram som man kan börja fundera på hur systemet skall testas.

☐ Falskt



☐ Sant

Ett klassdiagram beskriver hur klasser och objekt samarbetar.

☐ Falskt



☐ Sant

Design patterns beskriver hur man bäst strukturerar användargränssnitt.

☐ Sant

☐ Falskt



Interaktionsdiagram visar vilka metodanrop objekt gör på andra objekt.

☐ Sant



☐ Falskt

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

☐ Falskt



☐ Sant

Att klassen "Äpple" ärver från klassen "SpelElement" innebär att alla metoder och attribut i "SpelElement" också finns i "Äpple".

☐ Falskt

☐ Sant



"Unit Testing" är ett speciellt testramverk för spelmotorn "Unity".

☐ Falskt

☐ Sant



Totalpoäng: 8

2 GRASP Mönster

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

Man kan använda Polymorphism för att åstadkomma High Cohesion.

☐ Falskt

☐ Sant



En Controller behövs bara när användaren skall styra (kontrollera) ett spel.

☐ Sant

☐ Falskt



High Cohesion går ut på att varje klass skall ha så få och så välavgränsade ansvarsområden som möjligt.

☐ Sant

☐ Falskt



Low Coupling går ut på att man skall sträva efter att ha så få och så "lösa" associationer som möjligt mellan klasser i ett system.

☐ Sant

☐ Falskt



En Controller kan anropa Information Experts.

☐ Falskt

☐ Sant



Controller kräver Polymorfism för att fungera.

☐ Sant

☐ Falskt



En klass kan vara både en Information Expert och en Controller.

☐ Sant

☐ Falskt



Totalpoäng: 7

3 Design Patterns

Markera om följande påståenden är sanna eller falska:

(+1 för rätt svar, ingen förändring för fel svar)

Singleton betyder att man bara får anropa klassen en gång.

☐ Falskt



☐ Sant

Strategy pattern använder sig av polymorfism.

☐ Falskt

☐ Sant



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

Abstract Factory är i själva verket en variant av Strategy.

☐ Falskt

☐ Sant



En Observable är en klass med data som andra klasser kan vara intresserade av.

☐ Falskt

☐ Sant



Ett Strategy pattern består av minst tre klasser med rollerna Context, AbstractStrategy, och ConcreteStrategy

☐ Sant



☐ Falskt

Totalpoäng: 6

4 Design Patterns II

Välj rätt designmönster ur listan för varje påstående.

Priset för olika typer av tågbiljetter räknas ut på olika vis, till exempel för studenter och

pensionärer. För detta kan man använda designmönstret (Strategy, Observer, Factory, State).

När ett visst tåg är fullbokat skall det inte gå att köpa fler biljetter till tåget. För att meddela alla biljettautomater detta så snabbt det går så att de kan avbryta eventuella pågående transaktioner

kan man använda sig av designmönstret (Strategy, State, Observer, Factory)

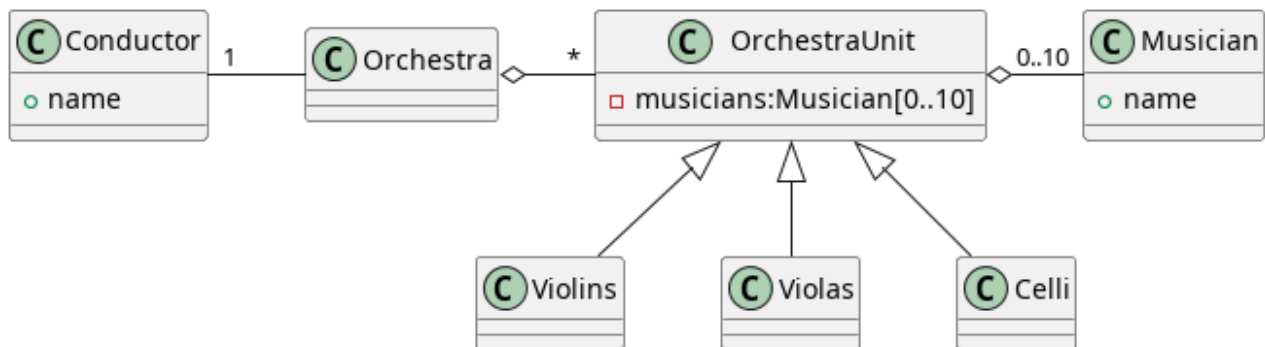
Biljetterna ser olika ut beroende på vad de är för typ. Samma information finns med, men skrivs på olika ställen och med olika storlek på texten, och med olika illustrationer så att man skall lätt kunna känna igen vilken biljett det är. Varje sak som finns med på biljetten representeras av ett

objekt, och systemet använder sig av designmönstret (Factory, Strategy, State, Observer) för att en gång för alla välja på vilket sätt informationen på biljetten skall skapas.

Totalpoäng: 3

5 Klassdiagram

Klassdiagrammet nedan beskriver en stråkorkester:



För varje påstående nedan, markera om diagrammet stödjer påståendet (sant) eller inte stödjer påståendet (falskt).

(+1 för rätt svar, ingen förändring för fel svar)

adam:Conductor är inte en Musician.

☐ Sant



☐ Falskt

bea:Conductor vet inte hur många Musician som spelar i en Orchestra.

☐ Sant



☐ Falskt

camera:Orchestra har ingen Conductor.

☐ Sant

☐ Falskt



dave:Musician spelar inget instrument.

☐ Falskt

☐ Sant



I first:Violins ingår tre Musician med namnen "Alice", "Bob", och "Cecilia".

☐ Sant



☐ Falskt

I c1:Celli ingår musicians[0]:Musician och musicians[1]:Musician.

☐ Sant



☐ Falskt

Man måste ha minst en Musician som spelar Viola för att kunna skapa object av typen Orchestra.

☐ Sant

☐ Falskt



Totalpoäng: 7

i Betygsgränser

Betygsgränserna för denna tenta är:

Betyg	Procent	Poäng
MAX	100%	31
A	90%	28
B	80%	25
C	70%	22
D	65%	20
E	60%	18

Lycka till!