

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 Interaktionsdiagram

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

Ett samarbetsdiagram beskriver samma sak som ett sekvensdiagram.

☐ Falskt

☐ Sant



Ett sekvensdiagram beskriver interaktionen mellan olika klasser.

☐ Sant

☐ Falskt



I sekvensdiagram listar man alla attribut och deras värden längst ner under varje objekts livlina.

☐ Sant

☐ Falskt



Man gör ett interaktionsdiagram för varje systemhändelse.

☐ Sant

☐ Falskt



Man extraherar systemhändelser ur use-cases genom att rita ett systemsekvensdiagram.

☐ Falskt

☐ Sant



Alla objekt i ett interaktionsdiagram måste ha ett variabelnamn och en klasstyp, t.ex.: namn:Typ.

☐ Sant

☐ Falskt



Man kan alltid använda tillståndsdigram (State diagrams) i stället för interaktionsdiagram.

☐ Falskt



☐ Sant

Man gör ett systemsekvensdiagram för varje Use Case.

☐ Falskt

☐ Sant



Syftet med systemsekvensdiagram är att identifiera hur olika aktörer interagerar med varandra.

☐ Sant

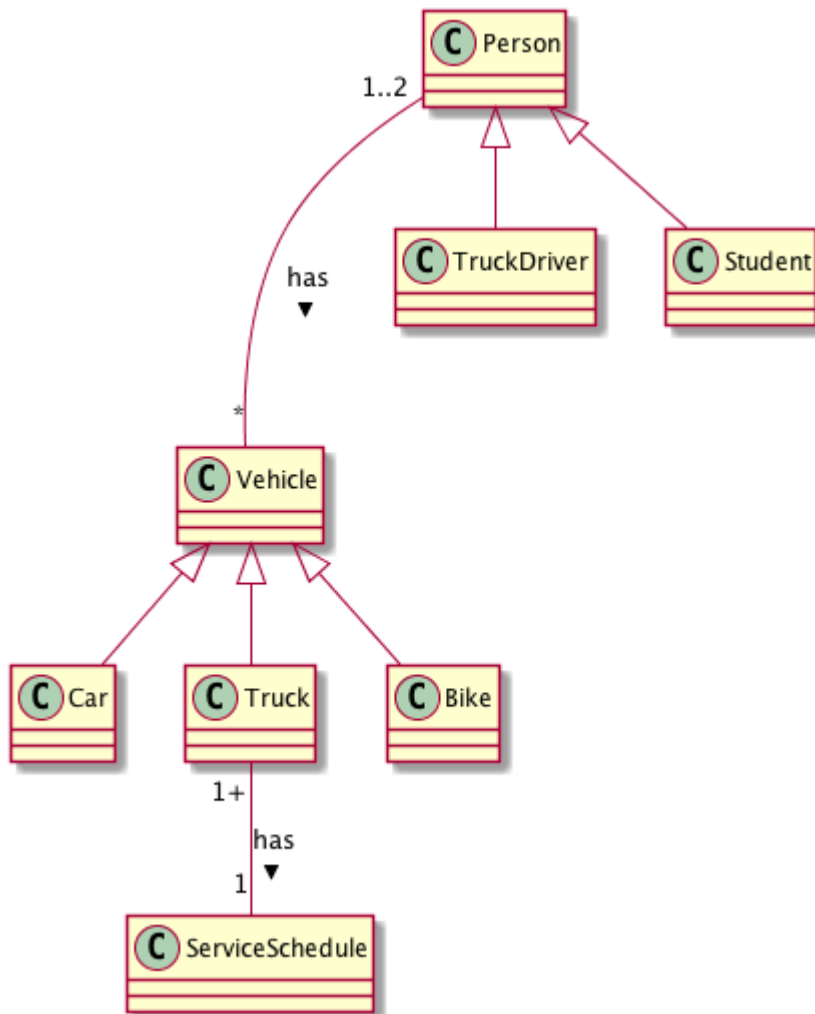
☐ Falskt



Totalpoäng: 9

3 Klassdiagram

Betrakta följande klassdiagram:



Markera sedan om följande påståenden stöds (är sanna) eller inte (=falskt) av diagrammet:

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

Tom, som är en Student, äger skrothögen:Truck.

- ☐ Sant
- ☐ Falskt



Grållen som är en Car ägs gemensamt av Eva:Person, och Egon:TruckDriver.

- ☐ Falskt
- ☐ Sant



Varje Truck har minst ett ServiceSchedule.

☐ Falskt



☐ Sant

Ett ServiceSchedule gäller bara för en Truck åt gången.

☐ Sant

☐ Falskt



Ett Vehicle måste vara antingen en Car, en Truck, eller en Bike.

☐ Sant

☐ Falskt



Gyllenstierna, som är en Person, äger Jaguar:Car, Landrover:Car, och Volvo:Truck.

☐ Sant



☐ Falskt

Studenterna Jack, Jill, och Jim äger tillsammans en Bike.

☐ Sant

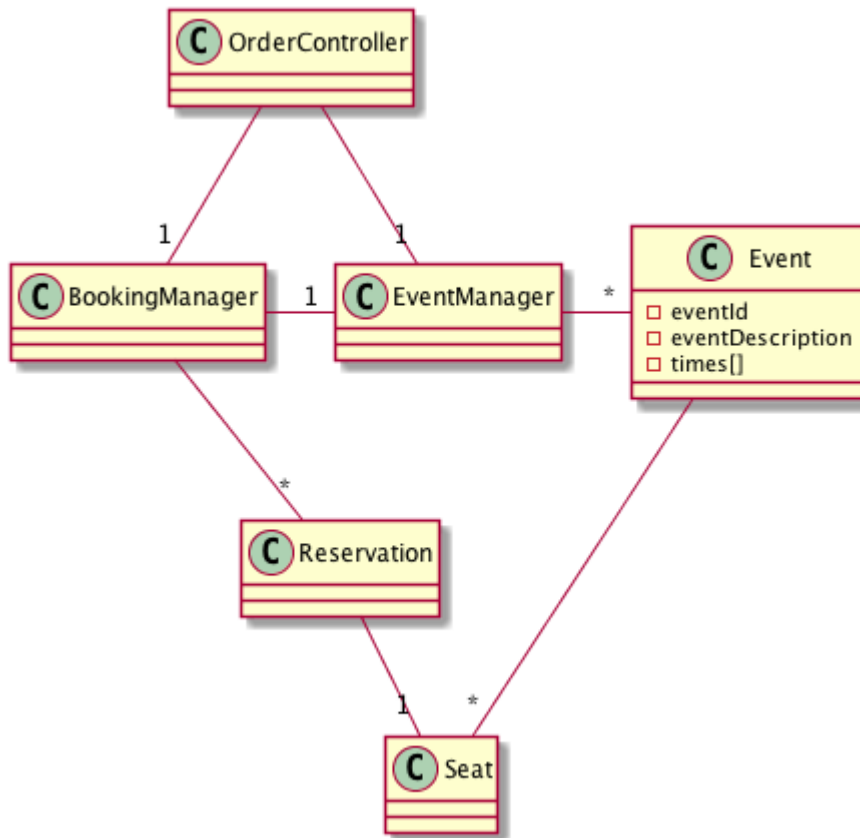
☐ Falskt



Totalpoäng: 7

4 GRASP mönster

Betrakta först klassdiagrammet. Välj sedan vilka klasser som ansvarar för vad, och varför.



Klassen (OrderController, Event, Seat, Reservation, BookingManager, EventManager) är bäst lämpad att ta emot systemhändelsen **createBooking()** enligt GRASP mönstret (Information Expert, Creator, Controller, Low Coupling, High Cohesion).

Klassen (OrderController, BookingManager, EventManager, Event, Reservation, Seat) skall ha metoden **searchBooking()** eftersom den är en (Information Expert, Creator, Controller).

När en ny **Reservation** skall skapas så måste först (OrderController, BookingManager, Reservation, Seat) söka fram en **Seat** på ett **Event**. Denna klass får ansvaret enligt principen (Low Coupling, High Cohesion, Polymorphism, Pure Fabrication).

Totalpoäng: 6

5 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

- ☐ 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 har man flera Modeller som presenteras i samma View

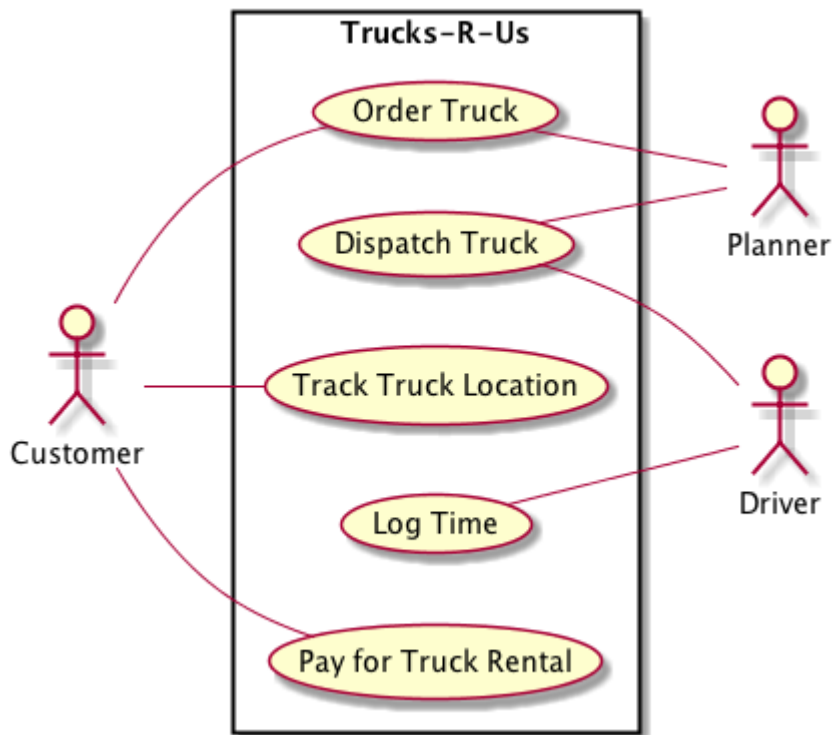
- ☐ Sant
- ☐ Falskt



Totalpoäng: 5

6 Use Case Diagram

Betrakta följande Use Case Diagram:



Markera sedan om följande påståenden stöds (är sanna) eller inte (falskt) av diagrammet:

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

Kunden möter aldrig Föraren

- ☐ Falskt
- ☐ Sant

✓

Planeraren interagerar med Föraren för att skicka iväg en lastbil.

- ☐ Sant
- ☐ Falskt

✓

För att kunna spåra en lastbil måste man först ha beställt en lastbil.

- ☐ Sant
- ☐ Falskt

✓

För att betala en lastbilshyra måste Föraren först ha loggat tidsåtgången i systemet.

☐ Falskt



☐ Sant

Totalpoäng: 4

i Betygsgränser

Betygsgränserna för denna tenta är:

Betyg	Procent	Poäng
MAX	100%	44
A	90%	39
B	80%	35
C	70%	30
D	65%	28
E	60%	26

Lycka till!