

STUDENT

Ayubi Bilal 200312220155

TENTAMEN

20240103 DA339A 2202

Kurskod	--
Bedömningsform	--
Starttid	03.01.2024 13:15
Sluttid	03.01.2024 18:15
Bedömningsfrist	--
PDF skapad	01.02.2024 17:33

Information

Uppgift	Uppgiftstyp
---------	-------------

i	Dokument
---	----------

Uppgift 1 (10p) - Sant eller Falskt

Uppgift	Uppgiftstyp
---------	-------------

1.1	Sant/Falskt
-----	-------------

1.2	Sant/Falskt
-----	-------------

1.3	Sant/Falskt
-----	-------------

1.4	Sant/Falskt
-----	-------------

1.5	Sant/Falskt
-----	-------------

1.6	Sant/Falskt
-----	-------------

1.7	Sant/Falskt
-----	-------------

1.8	Sant/Falskt
-----	-------------

1.9	Sant/Falskt
-----	-------------

1.10	Sant/Falskt
------	-------------

1.11	Sant/Falskt
------	-------------

1.12	Sant/Falskt
------	-------------

1.13	Sant/Falskt
------	-------------

1.14	Sant/Falskt
------	-------------

1.15	Sant/Falskt
------	-------------

1.16	Sant/Falskt
------	-------------

1.17	Sant/Falskt
------	-------------

1.18	Sant/Falskt
------	-------------

1.19	Sant/Falskt
------	-------------

1.20	Sant/Falskt
------	-------------

Uppgift 2 (2p) - Grundläggande begrepp 1

Uppgift	Uppgiftstyp
---------	-------------

2.1	Essä
-----	------

Uppgift 3 (2p) - Grundläggande begrepp 2

Uppgift	Uppgiftstyp
---------	-------------

3.1	Essä
-----	------

Uppgift 4 (4p) - Läsa sekvensdiagram

Uppgift	Uppgiftstyp
4.1	Programmering

Uppgift 5 (2p) - Utskrift och Overloading

Uppgift	Uppgiftstyp
5.1	Essä

Uppgift 6 (2p) - Undantags hantering (Exception handling)

Uppgift	Uppgiftstyp
6.1	Programmering

Uppgift 7 (4p) - Skapa klass, använda arv m.m.

Uppgift	Uppgiftstyp
7.1	Programmering

Uppgift 8 (7p) - Skapa klassdiagram

Uppgift	Uppgiftstyp
8.1	Filinlämning

Uppgift 9 (7p) - Skapa sekvensdiagram

Uppgift	Uppgiftstyp
9.1	Filinlämning

Uppgift 10 - Uppgift för betyg VG

Uppgift	Uppgiftstyp
10.1	Programmering

1 Sant eller Falskt

Uppgift 1 10p 0,5/påstående i deluppgifter.

Vilka påståenden är korrekta/sanna och vilka är felaktiga/falsa?

Rätt svar i en deluppgift ger 0,5p (totalt 10p på hela uppgiften)
Fel svar på en deluppgift ger 0p, dvs inga minuspoäng.

För kod som förekommer i frågorna eller påståenden som relaterar till hur kod fungerar kan programspråket förutsättas vara Java.

1.1 Antag att metoderna `methodA` och `methodB` finns i en klass (se nedan):

```
public Person[] methodA( ) {  
    /*kod*/  
}  
public void methodB (Person person) {  
    /*kod*/  
}
```

Metoderna kan användas enligt metदानropet nedan:

```
methodB(methodA( ));
```

Välj ett alternativ:

☐ Sant

☒ Falskt

Totalpoäng: 0.5

1.2 En abstrakt metod behöver inte implementeras av alla subclasser till superklassen som innehåller den abstrakta metoden.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

1.3 Antag att klassen `ClassB` är en subclass till klassen `ClassA`. Klassen `ClassB` har en metod `foo` som inte finns i klassen `ClassA`.

Då är följande syntax korrekt:

```
ClassA ca = new ClassB();  
((ClassB)ca).foo();
```

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

- 1.4 En klass kan ärva från flera klasser och kan endast implementera ett interface.

Välj ett alternativ:

☐ Sant

☒ Falskt

Totalpoäng: 0.5

- 1.5 Om en klass är abstrakt kan man inte skapa objekt av klassen.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

- 1.6 En enum är en klasstyp och kan instansieras, dvs. det går att skapa ett objekt av en enum med nyckelordet **new**.

Välj ett alternativ:

☐ Sant

☒ Falskt

Totalpoäng: 0.5

- 1.7 Nyckelordet **throws** används tillsammans en metods signatur och tvingar metoden att implementera **try-catch** inne i metod-kroppen.

Välj ett alternativ:

☐ Sant

☒ Falskt

Totalpoäng: 0.5

- 1.8 Vid hantering av undantag, kan det finnas endast ett **try** block och ett **finally** block men flera **catch** block.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

1.9 Ett interface kan ärvas av en eller flera interfacer.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

1.10 Nyckelordet **final** kan användas för att förhindra arv av en klass.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

1.11 Det kan finnas flera associationer av samma typ mellan två klasser.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

1.12 Vid överskuggning/overriding har man samma metod som i superklassen men med olika implementation i subclasser i en klasshierarki.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

1.13 Ett sekvensdiagram visar vilka objekt som existerar under exekvering.

Välj ett alternativ:

☐ Sant

☒ Falskt

Totalpoäng: 0.5

1.14 När typen av objekt är bestämt vid kompilering, sker en dynamisk bindning.

Välj ett alternativ:

☐ Sant

☒ Falskt

Totalpoäng: 0.5

1.15 Polymorfism innebär att ett objekt kan bete sig som om det tillhör två olika klasser.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

1.16 Klassvariabler och klassmetoder finns endast i en uppsättning som delas mellan samtliga instanser av klassen. Modifieraren *static* används för att ange att en variabel eller en metod är en klassvariabel respektive klassmetod.

Välj ett alternativ:

☐ Sant

☒ Falskt

Totalpoäng: 0.5

1.17 Vid generalisering ärvs även associationer som superklassen har till andra klasser.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

1.18 Ett klassdiagram är ett dynamiskt diagram.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

- 1.19 Om man tillämpar konceptet Boundary-Control-Entity strikt för klasser i ett system ska där inte finnas någon association mellan klasser av typen Boundary och Entity.

Välj ett alternativ:

☒ Sant

☐ Falskt

Totalpoäng: 0.5

- 1.20 En klass av typen controller i MVC-mönstret hanterar interaktionen med användaren via ett grafiskt gränssnitt.

Välj ett alternativ:

☐ Sant

☒ Falskt

Totalpoäng: 0.5

- 2 Förklara hur du kan resonera för att avgöra om en association mellan två klasser lämpligtvis kan vara en aggregation eller inte (med aggregation avses här hur detta definieras i kurslitteratur av Bennet). Frågan bör kunna besvaras med ungefär 6-8 meningar. (2p)

Skriv in ditt svar här

låt säga att vi har klass A och B. Ett bra sätt att kolla är att först ta en titt på vilken relation som klass A har till klass B och vice versa. Man ställer sig frågan om ifall jag nu raderar objekt i klass A, är de då beroende i klass B och behöver tas bort också? Ifall svaret är ja så tillhör delen helheten vilket innebär att det är en komposition. Men skulle svaret vara nej på frågan så kollar man vidare på relationen, är det som så att ifall vi nu tar bort klass A, försvinner då klass B också? Stannar klass A kvar ifall klass B tas bort? Ifall svaret är Ja på båda frågor så har vi ett exempel på hur delen (klass B) är beroende av helheten klass (A) alltså är det en Aggregation där ett av klasserna har ett beroende utav det andra men inte tvärtom. Ett annat exempel som vi kan lyfta är människokroppen, majoriteten av människor föds med två armar men skulle det vara som så att vi förlorar en arm så händer ingenting med oss alltså är det en aggregation men skulle det vara som så att vi förlorar vår hjärna så dör vi nog ganska så snabbt och detta är då en komposition.

Totalpoäng: 2

- 3 Förklara kortfattat begreppet abstrakt/abstract klass och hur detta används i förhållande till begreppet generalisering. Motivera nyttan med att använda abstrakta klasser och när detta lämpar sig. Frågan bör kunna besvaras med ungefär 6-8 meningar. (2p)

Skriv in ditt svar här

En abstrakt klass är en klass som är uppbyggd på sina subklasser med olika abstrakta metoder, koden för ens abstrakta klass skrivs alltså i subklasser som är fördelade i förhållande till generalisering så är det överordnande klasser som delas inom flera klasser. Abstrakta klasser är bra klasser att använda vid tillfälle då man vill återanvända sin kod eller skapa ett uppfördelat klasshierarki.

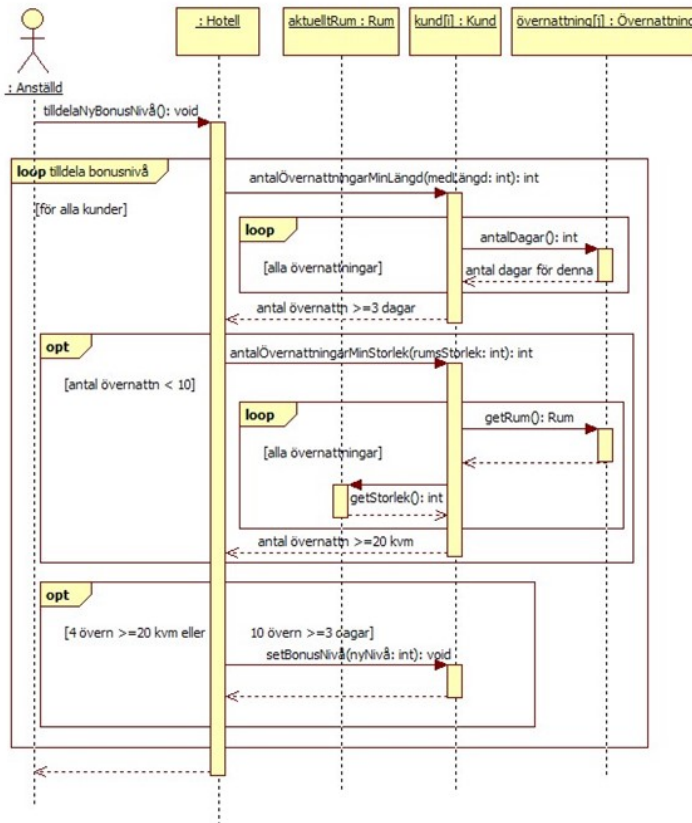
Totalpoäng: 2

- 4 Sekvensdiagrammet nedan visar vad som sker när en anställd vill köra den årliga översikten av bonusnivå för kunderna i ett system för hotell-övernattningar. Alla kunder som har gjort något av följande

- övernattat minst 10 gånger och dessa övernattningar varat minst 3 dagar
- övernattat minst fyra gånger i ett rum som är minst 20 kvm

ska sättas till bonusnivå 3. Skriv kod för de två klasserna **Hotell** och **Kund** som finns i diagrammet utifrån information om klassnamn och metoder i sekvensdiagrammet och informationen ovan.

Koden skall visa vilka metoder de två klasserna har enligt diagrammet samt den kod i metoderna som går att utläsa från diagrammet. Vissa antaganden kan göras om namn på lokala variabler, villkor i iterationer och selektioner samt villkor i dessa. (4p)



4 Skriv den kod som går att utläsa ur sekvensdiagrammet här:**Skriv in ditt svar här**

```
public class Hotell{
    public void tilldelaNyBonusNivå(){
        for(i = 0; i < kund[i].length; i++){
            int antalMin = kund[i].antalÖvernattningarMinLängd(3);
            int antalNatt = kund[i].antalÖvernattningarMinStorlek(10);
            int storlek = aktuelltRum.getStorlek();
            if(antalNatt && storlek >= 20 || antalMin){
                setBonusNivå(nyNivå);
            }
        }
    }

    public class Kund{
        int antalÖvernattningarMinLängd(int medLängd){
            for(j = 0; j<Övernattning[i]; j++){
                int dagar = övernattning[i].antalDagar();
            }
        }
        int antalÖvernattningarMinStorlek(int rumsStorlek){
            for(j = 0; j<Övernattning[i]; j++){
                int rum = övernattning[i].getRum();
            }
        }
        public void setBonusNivå(int nyNivå){...}
    }
}
```

Totalpoäng: 4

5 Metoderna foo och boo finns i någon klass:

```
public String foo(int a, int b){
    String text = ("#1 text = " + (a + b) + "c");
    return text;
}

public String foo(double a, double b){
    String text = ("#2 text = " + (a - b) + "c");
    return text;
}

public String foo(String a, String b){
    String text = ("#3 text = " + a + b + "c");
    return text;
}

public void boo(){
    int a = 2;
    int b = 2;
    System.out.println(foo(a, b));
}
```

5 Vad blir utskriften när metoden boo exekveras? (2p)

Metoderna foo och boo kan du se till vänster.

Skriv in ditt svar här

#1 text = 4c

Ord: 4

Totalpoäng: 2

- 6 Koden nedan kommer att kasta ett `ArrayIndexOutOfBoundsException` undantag (exception). Skriv om metoden och visa hur du ska hantera undantaget för att undvika att programmet går ner (kraschar) under körning, samt att programmet ska skriva ut om det gick att returnera text eller inte. (2p)

```
public String returnStringElement(){
    String[] numberArray = new String[1];
    String text = numberArray[1];
    return text;
}
```

- 6 Koden nedan kommer att kasta ett `ArrayIndexOutOfBoundsException` undantag (exception). Skriv om metoden och visa hur du ska hantera undantaget för att undvika att programmet går ner (kraschar) under körning, samt att programmet ska skriva ut om det gick att returnera text (String variabeln i koden nedan) eller inte. (2p)

```
public String returnStringElement(){
    String[] numberArray = new String[1];
    String text = numberArray[1];
    return text;
}
```

Skriv in ditt svar här

```
1 answerToException = "Could convert: ";
2 public String returnStringElement(){
3     String[] numberArray = new String[1];
4     try{
5         String text = numberArray[1];
6     }catch(ArrayIndexOutOfBoundsException e){
7         answerToException = "Could not convert: ";
8     }finally{
9         system.out.println(answerToException + text + "to an array");
10    }
11    return text;
12 }
```

Totalpoäng: 2

7 Givna klasser Event och klassen Adress.**Event:**

```
public abstract class Event {
    private String name;
    private int date;
    private Adress adress;

    public Event(String name, int date, String city, String street){
        this.name = name;
        this.date = date;
        this.adress = new Adress(city, street);
    }

    public String toString() {
        String textOut = String.format("Name: %s | Date: %s | Adress: %s", name, date, adress.toString());
        return textOut;
    }
}
```

Adress:

```
public class Adress {
    private String city;
    private String street;

    public Adress(String city, String street){
        this.city = city;
        this.street = street;
    }

    public String toString(){
        String textOut = city + ", " + street;
        return textOut;
    }
}
```

Testkörning av metoden i Main ger resultatet:

```
public class Main {
    public static void main(String[] args) {
        Disco disco = new Disco("Tentamen Disco", 240103, "Malmö",
                                "Kranen street", "Tenta", "DA339A");
        System.out.println(disco);
    }
}
```

Output:**Name: Tentamen Disco | Created: 240103 | Adress: Malmö, Kranen street | Theme: Tenta | DJ playing: DA339A**

- 7 I följande uppgift ska du använda dig av den givna abstrakta klassen Event och klassen Adress (se till vänster). Inga ändringar eller tillägg är tillåtna i klasserna Event eller Adress. Alla instansvariabler ska vara deklarerade med modifieraren private. (4p totalt)

Skapa en klass Disco. Denna klass ska ha två instansvariabler theme med datatyp String och playingDJ med datatyp String. Klassen Disco skall ära klassen Event (se ovan). Alla instansvariabler ska vara deklarerade med modifieraren private. (1p)

Skapa en enda konstruktor för klassen Disco med nödvändiga parametrar för att initiera alla instansvariabler i klassen Disco och dess superklass. (2p)

Skriv en toString-metod i klassen Disco så att resultatet nedan erhålls. (1p)
En testkörning av metoden i Main samt output kan du se till vänster.

Du ska svara på deluppgifterna i samma svar – skriv allt som efterfrågas för klassen Disco i ett svar.

Skriv in ditt svar här

```
1 public class Disco extends Event{
2     private String theme;
3     private String playingDJ;
4 }
5
6 class Disco(String name, int date, String city, String street, String theme, String
7     super(name, date, city, street);
8     this.theme = theme;
9     this.playingDJ = playingDJ;
10 }
11
12 public String toString(){
13     String textOut = String.format("%s | Theme: %s | DJ playing: %s", super.toString()
14     return textOut;
15 }
```

Totalpoäng: 4

- 8 För denna uppgift gäller att du kan välja mellan att rita ditt diagram för hand på papper och lämna in ditt svar på papper eller rita diagrammet digitalt med angivet verktyg och lämna in ditt svar som en digital bild i Inspira (se instruktioner nedan för detta).

Du ska endast lämna in svar på denna uppgift på ett (1) av de två sätten. Om du skulle lämna in svar på bägge sätten sätts poäng efter det svar som bedöms få minst antal poäng.

Uppgiftstext

Konstruera ett lämpligt klassdiagram utifrån systembeskrivningen nedan. Identifiera lämpliga **entity-klasser** för systemet. Använd dig av så lämpliga associationstyper som möjligt för att beskriva hur klasserna relaterar till varandra. Använd generalisering, aggregation och/eller komposition där detta är lämpligt. Skriv ut namn på alla associationer för att förtydliga dessa. Om det går att ange en multiplicitet för en association så skall detta finnas – gör lämpliga antaganden där detta kan vara rimligt. Gör inga tillägg med information som överhuvudtaget inte finns i systembeskrivningen.

Det är inte nödvändigt att visa attribut eller operationer i klassdiagrammet. Det räcker att skriva ut klassnamn och namn på associationer och multiplicitet. Korrekt UML-notation ska användas (se notationshjälp nedan).

Systembeskrivning

Man vill skapa ett program som stödjer skrivandet av böcker, vilket är en stor, mer eller mindre seriös, hobbyverksamhet för många. Detta är den inledande beskrivningen man har efter några första intervjuer med presumtiva användare av systemet.

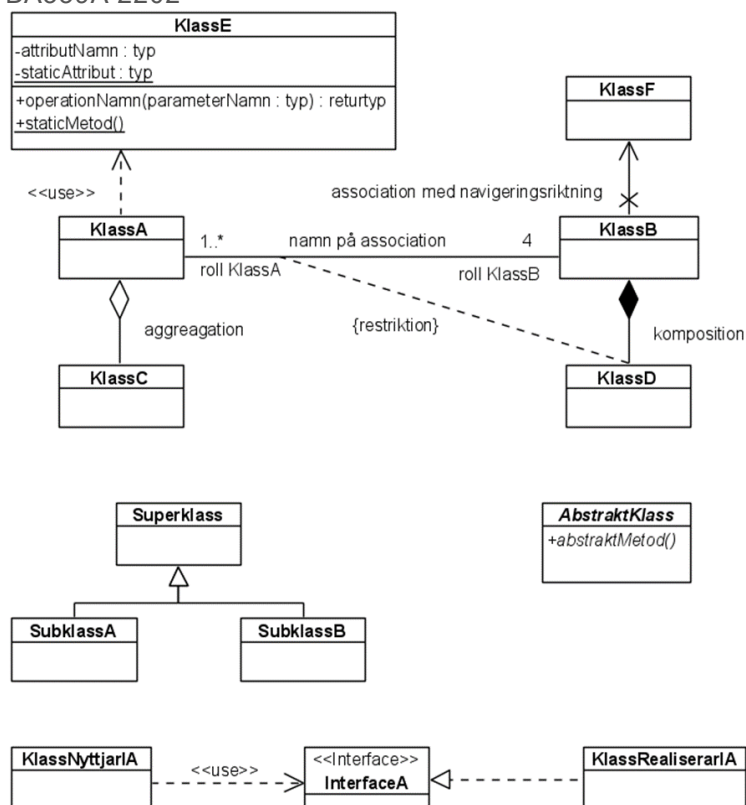
Man hittade två tydliga typer av böcker som skrevs - romaner och lyrik. Dessa är böcker men som skiljer sig åt väsentligt i hur detaljerna ser ut för deras uppbyggnad. Flera av författarna man intervjuat arbetar med andra författare på samma bok. Så en bok kan ha flera författare till den innehållande texten.

Lyrik innehåller verser. De som författade lyrik ville kunna hantera verser för sig och kunna sätta ihop verser till böcker med lyrik. En vers kan kanske komma att ingå i flera olika böcker i slutändan.

De författare som skrev romaner vill kunna hantera flera romaner som följer på varandra i en serie. Ingen har dock gett uttryck för att det behövs en trädstruktur utan det räcker att kunna hantera en kedja av romaner. Så man behöver veta vilken roman som utspelar sig före en annan och vilken som kommer efter. Givetvis så förekommer böcker som inte ingår i någon serie.

Det finns en mängd karaktärer som behöver hanteras gällande romanerna. Man behöver hålla reda på vilka karaktärer som förekommer i vilka romaner men också på en mer detaljerad nivå i vissa fall. I de mer detaljerade fallen behöver man veta i vilka kapitel som en karaktär förekommer. Romaner utan direkt kapitelindelning beräknas ha endast ett kapitel. Man ansåg också att det kunde finnas kapitel som inte innehåller någon väsentlig karaktär.

Notationshjälp klassdiagram



Instruktioner för att rita och spara diagram digitalt

Rekommendationen är att du först ritar ett utkast av ditt diagram för hand på anteckningspapper. När du är nöjd med detta ritar du rent diagrammet med det digitala verktyget. Detta för att minska risken att det digitala verktyget börjar bete sig konstigt om du gör många ändringar när du ritar med detta.

Använd resursen som du hittar via länken längst ner på sidan för att rita ett klassdiagram digitalt.

När du ritat ditt diagram och ska lämna in väljer du "Export" uppe till höger i ritverktyget -> "Save as PNG"-> gör inga ändringar av standardinställningarna i nästa dialog "Image" utan tryck bara "Export" -> I dialogen "Save as" sätter du ett filnamn och sedan väljer du "Download" och spara filen lokalt på datorn.

Ladda sen upp filen till ditt svar genom att trycka på *Bläddra* nedan.

Filen hittar du under: **User->[ditt användar-ID]->Downloads**.

Skulle något strula för dig med detta verktyg på tentan kan tentavakterna hjälpa dig att spara filen och hitta den.

8 Se instruktioner för uppgiften till vänster. Om du lämnar in svar digitalt laddar du upp din fil här.



Ladda upp din fil här. Max en fil.

Följande filtyper är tillåtna: **.png** Maximal filstorlek är **1 GB**

Välj fil att ladda upp

Totalpoäng: 7

- 9 För denna uppgift gäller att du kan välja mellan att rita ditt diagram för hand på papper och lämna in ditt svar på papper eller rita diagrammet digitalt med angivet verktyg och lämna in ditt svar som en digital bild i Inspera (se instruktioner nedan för detta).

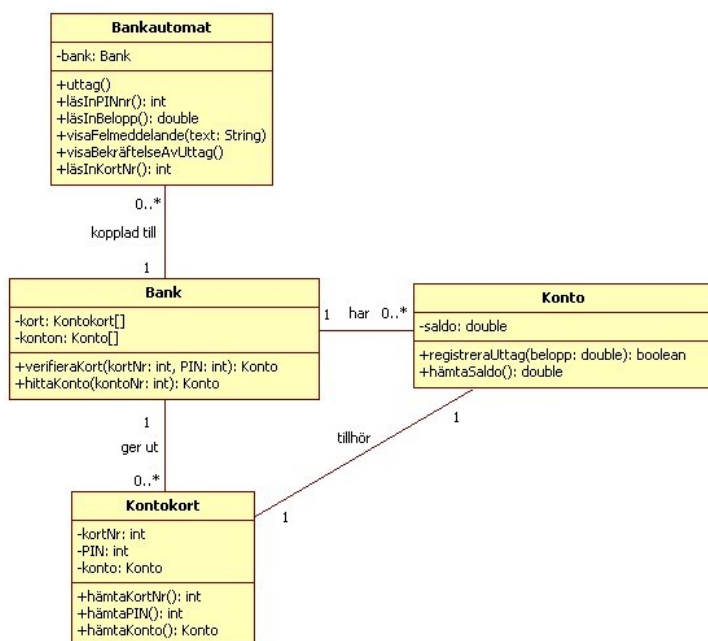
Du ska endast lämna in svar på denna uppgift på ett (1) av de två sätten. Om du skulle lämna in svar på bägge sätten sätts poäng efter det svar som bedöms få minst antal poäng.

Uppgiftstext

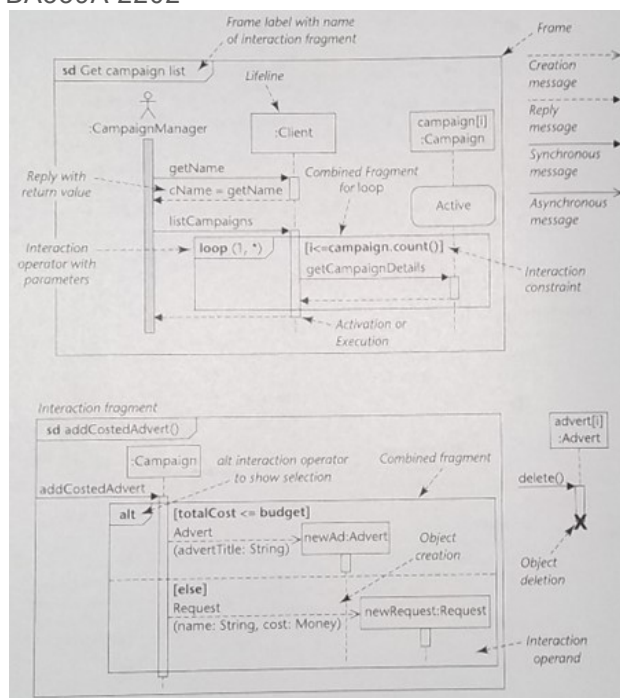
Klassdiagrammet nedan beskriver ett system som hanterar en uttagsautomat där man kan ta ut pengar från ett konto kopplat till ett bankkort.

Givet klassdiagrammet nedan rita ett sekvensdiagram som visar vad som sker när ett uttag sker enligt följande modell: Bankautomaten som är representerad av ett Bankautomat-objekt känner endast till ett Bank-objekt och måste hämta all sin information och övriga objekt via detta. Bankomaten läser in PIN-kod från kortinnehavaren och läser av kortnumret från plastkortet. Plastkortet kan dock inte i sig bekräfta något om PIN-nummer utan detta måste bekräftas via banken. Lösningen ska innebära att bankomaten visar felmeddelanden för om kontot kopplat till kortet inte kan hittas i banken, om PIN inte stämmer eller saldot är för litet för uttaget (du behöver inte visa upprepade försöka att slå in rätt PIN – att visa att Pin anges en gång räcker). Lösningen ska visa vad som sker för att hitta ett konto och verifiera PIN samt uttag om så kan göras.

Korrekt UML-notation ska användas (se notationshjälp nedan). Du skall inte skapa några nya klasser, lägga till operationer eller attribut i befintliga klasser utan skall använda dig av det som visas i klassdiagrammet. Du får däremot skapa en aktör som representerar kortinnehavaren som interagerar med Bankautomat-objektet och initierar sekvensen med anrop till metoden *uttag()*.



Notationshjälp sekvensdiagram



Instruktioner för att rita och spara diagram digitalt

Rekommendationen är att du först ritar ett utkast av ditt diagram för hand på anteckningspapper. När du är nöjd med detta ritar du rent diagrammet med det digitala verktyget. Detta för att minska risken att det digitala verktyget börjar bete sig konstigt om du gör många ändringar när du ritar med detta.

Använd resursen som du hittar via länken längst ner på sidan för att rita ett diagram digitalt.

När du ritat ditt diagram och ska lämna in väljer du "Export" uppe till höger i ritverktyget - > "Save as PNG" -> gör inga ändringar av standardinställningarna i nästa dialog "Image" utan tryck bara "Export" - > I dialogen "Save as" sätter du ett filnamn och sedan väljer du "Download" och spara filen lokalt på datorn.

Ladda sen upp filen till ditt svar genom att trycka på *Bläddra* nedan.

Filen hittar du under: **User->[ditt användar-ID]->Downloads**.

Skulle något strula för dig med detta verktyg på tentan kan tentavakterna hjälpa dig att spara filen och hitta den.

- 9 Se instruktioner för uppgiften till vänster. Om du lämnar in svar digitalt laddar du upp din fil här.



Ladda upp din fil här. Max en fil.

Följande filtyper är tillåtna: **.png** Maximal filstorlek är **1 GB**

Välj fil att ladda upp

Totalpoäng: 7

10 Uppgifter för betyg VG

Uppgift 10a-d

Enumerationen **ContainerShapeType** är given nedan. Du ska skriva en klass **Container** som använder enumerationen, dess metoder och implementerar interfacet **IContainer**. Senare i uppgiften skall du också skriva ett par konkreta subclasser till Container. Klasserna är inte stora; du ska skriva så mycket kod som frågorna kräver. Figuren nedan visar de källfiler som skall kompletteras i uppgiften.

Observera: Glöm inte att deklarera de variabler du använder, strukturera metoderna på rätt sätt och använda rätt syntax. Alla instansvariabler ska vara deklarerade med modifieraren `private`.

```
public enum ContainerShapeType {

    Cube(5, "Bronze"),
    Triangle(10, "Silver"),
    Sphere(30, "Gold");

    private final double price;
    private final String material;

    ContainerShapeType(double price, String material){
        this.price = price;
        this.material = material;
    }

    public double getPrice(){
        return price;
    }

    public String getMaterial(){
        return material;
    }
}
```

Uppgift 10a: Interface

Komplettera interfacet vid markeringarna 1 till 7.

```
public interface IContainer {

    //1. En låda (Container) måste ha en typ av enum-typen ContainerShapeType.
    //2. Komplettera med en getter- och en setter-metod som subclasserna
    //   (i detta fall klassen Container) ska implementera.

    //3. En låda (Container) skall räkna ut och returnera sin volym (volume)
    //   utifrån sin längd, höjd, bredd samt typ av låda.
    //   - Om det är av typen Cube: Volym = bredd*höjd*längd.
    //   - Om det är av typen Triangle: Volym = (bredd*höjd*längd)/2.
    //   - Om det är av typen Sphere: Volym = (4*PI*r^3)/3. Kan använda
    //     höjd/2 som ditt r. (r^3 = r*r*r).
    //4. Komplettera med metod som returnerar volymen som en double.

    //5. En låda (Container) skall räkna ut och returnera sitt totala pris
    //   (totalPrice) utifrån sin volym samt vilken ContainerShapeType lådan
    //   är med sitt unika pris.
    //6. Komplettera med metod som returnerar totala priset som en double
    //   eller int.

    //7. Skriv en metod getBuildingBlockInfo som skall returnera en String
    //   (mer om denna metod i uppgift 10b).
}
```

Uppgift 10b: Abstrakt klass och abstrakt metod

Skriv en klass **Container** som implementerar interfacet ovan. Klassen skall innehålla all kod som implementation av interfacet kräver, med undantag av metoden **getContainerInfo** som skall definieras som en abstrakt metod i klassen. Du får deklarera variabler och skriva flera metoder (metoderna får dock inte ersätta givna metoder så att de inte behöver användas) ifall du behöver dem för din lösning (instansvariabler ska vara deklarerade med modifieraren `private`).

Observera: det är **inte** obligatoriskt att skapa och använda konstruktörer i denna och de andra klasser som du skriver i denna uppgift. Setter- och getter-metoder skriver du efter behov för lösningen. Du behöver använda konstruktörer eller get/set-metoder för att lösa uppgiften.

Definiera **getContainerInfo** i klassen Container som skall returnera en string.

I denna del, skriv bara de ändringar som du behöver göra i klassen Container.

Uppgift 10c: Konkreta klasser

Skriv tre klasser, **Cube**, **Triangle** och **Sphere** som subklasser till Container. Deklarera en valfri instansvariabel i varje klass. Ifall du inte kommer på något bra attribut, använd följande:

Klassen Cube: numOfCubeSides (int)

Klassen Triangle: numOfTriangleSides (int)

Klassen Sphere: numOfSphereSides (int)

Alla klasserna skall överskugga (override) den abstrakta metoden **getContainerInfo**. Metoden skall returnera en String; en text, värdet på den valfria instansvariabeln, namnet och priset på materialet som används (beroende på vilken ContainerShapeType), volymen på lådan samt det totala priset. Exempel på output finns sist i uppgiften.

Klasserna behöver inte vara stora utan de skall bara vara så pass kompletta att de tillsammans med Container och IContainer kan kompileras. Små syntax-fel överses vid rättning om de inte påverkar arkitekturen för lösningen.

Uppgift 10d: Dynamisk bindning och Type Casting

Skriv färdigt metoden createContainer (se nedan) så main-metodens anrop fungerar för följande testvärden:

Cube: num of cube sides = 6

Triangle: num of triangle sides = 5

Sphere: num of sphere sides = 1

För Cube: $b \times h \times l = 2 * 4 * 8$ (bredd x höjd x längd)

För Triangle: $(b \times h \times l) / 2 = (2 * 4 * 6) / 2$ ((bredd x höjd x längd) / 2)

För Sphere: $(4 * PI * (r*r*r)) / 3 = 4 * PI * (2*2*2) / 3$

```
public class MainProgram {
    public static void main(String[] args) {
        createContainer(ContainerShapeType.Cube);
        createContainer(ContainerShapeType.Triangle);
        createContainer(ContainerShapeType.Sphere);
    }

    public static void createContainer(ContainerShapeType csType) {
        //Komplettera
        //Krav:
        // - dynamisk bindning (och type casting, beroende på din lösning).
        // - metoderna (alla tre metoderna måste anropas):
        //     getContainerInfo(),
        //     calculateContainerVolume(ContainerShapeType csType) och
        //     calculateContainerPrice(ContainerShapeType csType)
        // (två sista metoderna som räknar volymen och total pris, och får in
        // vilken typ av låda det är) måste anropas.
    }
}
```

Utdata från main-metoden:

Nice triangle

triangle has 5 sides

triangle material: Bronze

price on material: 5.0kr

Container has a volume of 48cm³

Total price for container: 240.0kr

Fantastic cube

cube has 6 sides

cube material: Silver

price on material: 10.0kr

Container has a volume of 24cm³

Total price for container: 240.0kr

Perfect sphere

sphere has 1 sides

sphere material: Gold

price on material: 30.0kr

Container has a volume of 33cm³

Total price for container: 990.0kr

- 10** Kom ihåg att dela upp koden så att det är bra strukturerat och lätt läst. Om du vill skriva kommentarer använd dig av // som vanligt när du skriver kod.

Skriv in ditt svar här

1	
---	--

Totalpoäng: 100