



Antal blad /
Number of sheets

11

TENTAMEN / EXAMINATION

Anvisningar: Skriv din anonymitetskod på varje blad.
Endast en uppgift får lösas på varje blad.
Var vänlig skriv tydligt!

Instructions: Write your anonymous code on each sheet.
Answer only one question on each sheet.
Please write clearly!

Vänligen texta anonymitetskoden i textboxen enligt exempel nedan!
Please write the Anonymous Code clearly in the textbox like example below!

Bokstäver/Letters:

A-B-C-D-E-F-G-H-I-J-K-L-M-N-O
P-Q-R-S-T-U-V-W-X-Y-Z-Å-Ä-Ö

Siffror/Numbers:

Ø-1-2-3-4-5-6-7-8-9

Exempel:

A B C 1 7 Ø - Ø 1 7

DUGA09 Programutveckling och design

Kurskod + Kurs / Course Code + Course:

Delkurs / Part course:

Anonymitetskod / Anonymous code =
Kurskod + kodnr / course code + code number

D V G A Ø 9 - Ø 1 2

Tentamensdatum /
Examination date:

31/03/2017

Behandlade uppgifter / Solved problems

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X	X	X	X	X	X	X	X							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Ifylles av lärare / To be completed by the examiner

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4,5	3,5	5	2	5,5	4	3,5	5							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Poäng / Marks gained:

33
VG

Betyg / Grade:

Katarina Dahl
Examin. lärare / Kursansvarig signatur / Signature of the examiner

Max poäng / Total marks gained:

Namnförtydligande / Clarification of the signature

För Gk poäng / Marks gained to be passed:



a) Ett gränssnitt är en mängd abstrakta metoder som en klass kan implementera, exempelvis:

```
public class Apelsin implements Eatable
```

Detta gör att klassen Apelsin kan använda metoderna i gränssnittet Eatable men en programmerare måste först skriva egna metoder av de metoder som implementeras med hjälp av gränssnittet. Fördelen med detta är att man kan gruppera klasser som är olika men som utför samma roll. Exempel:

Både en myra och en människa kan förflytta sig men de gör det på olika sätt. När gränssnittet Moveable implementeras i båda klasserna så får klasserna enhetliga metoder som detaljerat beskriver hur de båda förflyttar sig. 2

b) En abstrakt klass är en klass som du inte kan skapa några objekt ifrån. Syftet med detta är att ibland kan en klass vara för generell för att ett objekt ska kunna skapas. Detta syfte förklarar jag bättre med ett exempel. Tänk dig att du har klasserna



Djur, Hund och Apa. När vi ska illustrera hur dessa klasser låter så kan vi enkelt föreställa oss hur en hund och hur en apa låter, men man kan inte riktigt säga att vi vet hur ett djur låter eftersom djur är för generell för uppgiften. I detta fall skulle det därför vara passande att klassen djur är en abstrakt klass.

c) Inkapsling är ett verktyg som en programmerare kan använda för att göra ett bra och hållbart datorprogram. När programmeraren kodar variabler och metoder så kan hen välja om datan ska vara synlig och åtkomlig ifrån andra klasser än den klassen datan skapas i. Detta händer med hjälp av public, private och protected.

7,5

14,5



```
public class Bankkonto
{
    private int kontonummer;
    private int saldo;
    private String ägare;

    public Bankkonto(int _kontonummer, String
    _ägare)
    {
        kontonummer = _kontonummer;
        ägare = _ägare;
        saldo = 0;
    }

    public void addSaldo(int _saldo)
    {
        saldo += _saldo;
    }

    public int getKundnummer()
    {
        return kundnummer;
    }

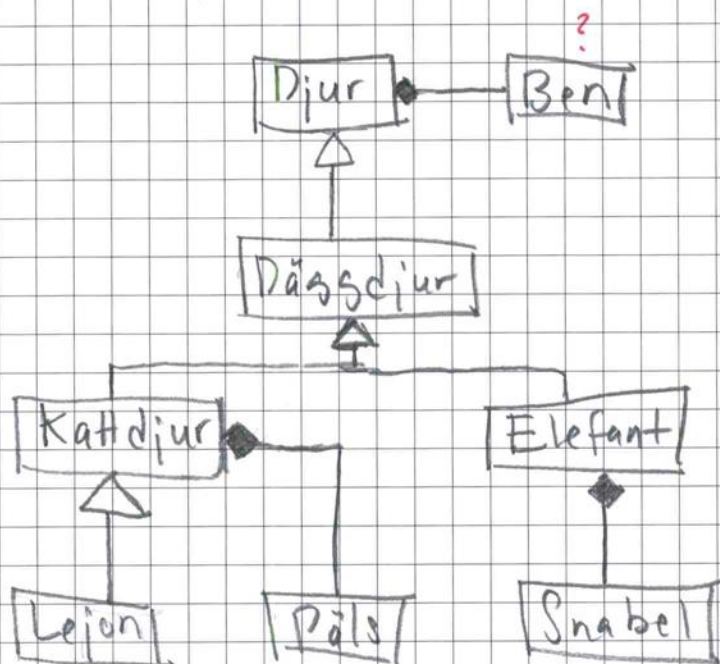
    public int getSaldo() { return saldo; }

    public String getÄgare() { return ägare; }
}
```

13,5



- a) Skillnaden mellan arv och komposition är att i arv så är en klass en annan klass, exempelvis ett lejon är ett kattdjur. I komposition så har en klass en annan klass, exempelvis en elefant har en snabel.



Ett antagande jag gjort i UML-diagrammet ovan är att jag tolkar Ben som skelettben. Eftersom jag inte är en zoolog så antar jag att alla djur har en eller flera skelettben.

- b) De har relationen komposition eftersom objektet `_b` som är av typen BBB skapas i AAA:s konstruktor. Detta betyder alltså att om klassen AAA tas bort så försvinner även objektet `_b`. Det betyder att AAA består av



Ange anonymitetskod / Write your anonymity code
(Vid icke anonym tentamen ange kurskod + namn + personnummer)
(For non-anonymous exams write the course code + name + civic registration number)

DVG409-012

Löpande sidnr
Consecutive no:

5

Uppgift nr /
Question no:

3

Poäng / Points
awarded:

2

Lärarens
anteckning
Examiner's remarks:

1/5



DVGA09-012

6

4

Relationen den deklarerade typen och den faktiska typen måste ha vid Polymorfism är antingen arv, eller att en klass implementerar ett gränssnitt. Skillnaden som blir är att du kan endast använda de metoder som finns i den deklarerade typen. Objektet p kan bara använda metoderna i klassen Pine och objektet träd har metoderna i träd. Vårt kan bara använda metoderna i Plant och objektet g använder metoderna i gränssnittet Growable

```
public class Pine extends Träd implements
Growable
public class Träd extends Plant
```

1/2



a) För att hantera knapptryckningar använder jag mig utav en JButton. JButton ärver metoden addActionListener och när den metoden anropas så läggs det till en observatör som kan registrera exempelvis klick. Klassen där JButton instanseras måste implementera gränssnittet ActionListener som innehåller metoden actionPerformed. För att klassen ska få använda metoden måste den överskrivas "override". När de inträffar en händelse i programmet anropas actionPerformed och all kod i den metoden exekveras.

1,5

b)

```

public void actionPerformed(Event e)
{
    if(e.getSource() == knapp1)
        System.out.println("Du tryckte på första knappen");
    if(e.getSource() == knapp2)
        System.out.println("Du tryckte på andra knappen");
}
  
```

2



c) Med GridLayout skapas ett osynligt rutnät där komponenter kan positioneras i de olika rutorna som skapas. Med FlowLayout läggs komponenterna ut i ett flöde som börjar högst upp till vänster. Komponenterna läggs ut efter varandra och när en rad tar slut så läggs nästa komponent på raden under. Med BorderLayout läggs komponenterna ut med hjälp av vädersträcken. För att lägga en komponent använder man BorderLayout.NORTH exempelvis. Samma gäller för WEST, EAST och SOUTH och man kan också använda CENTER.

2

18,5



I try - kodblocket exekveras massa kod och om det skulle kastas ett undantag så fångas de upp av antingen `catch (InputMismatchException e)` eller `catch (Exception e)` beroende på vilket undantag det är som kastas. Skillnaden som inträffar om de två catch - blocken byter plats är att bara `Exception e` kommer att fångas upp eftersom `Exception` är superklassen till alla andra undantag.

/4



a) Sambandet mellan ADT och datastruktur är
att när man skapar en ADT så använder
man sig av en datastruktur för att
lagra data. 0,5

b)

```
// pre: true
// post: en stack skapas
public Stack();

// pre: true
// post: element läggs till
public void addElement(E element);

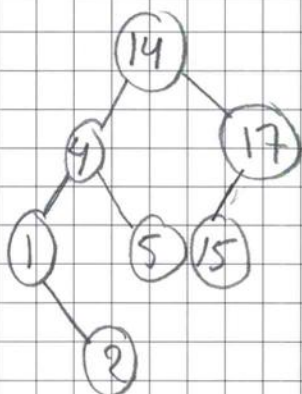
// pre: pos < 0 && !isEmpty
// post: tar bort ett element
public void removeElement(int pos);

// pre: true
// post: returnerar true om tom, annars false
public boolean isEmpty();
```

3
3,5



a)



b) Post order: 2, 1, 5, 4, 15, 17, 14
In order: 1, 2, 4, 5, 14, 15, 17

c) Rotnod är noden som är högst upp i ett träd. Lövnod är de noder som inte har några barn. Noddjup är hur djupt ner en nod är i ett träd. I trädet ovan har nod 2 ett noddjup på 3. Nodgrad är hur många barn en nod har.