

DAT043 – Objektorienterad Programmering

Tentamen 2018-03-10

Tid: 08.30-12.30

Ansvarig lärare: Moa Johansson

Tfn: 031 772 10 78

Ansvarig lärare besöker tentamenssalarna ca klockan 9.30 samt 11.00.

Tentamensregler

Tentamen består av två delar. För att få **godkänt på tentan (betyg 3) måste man lösa minst fem av sju uppgifter i Del 1** (frågor numrerade 1–7). För att få högre betyg krävs att man utöver detta även löser uppgifter i Del 2 (frågor 8–9). För att få **betyg 4 ska, utöver godkänt på Del 1, även en fråga från Del 2 lösas**. För **betyg 5 skall, utöver godkänt på Del 1, dessutom båda frågorna i del 2 lösas**. Förutsatt att man klarat Del 1 får man självklart testa att lösa båda uppgifterna i Del 2. Lyckas man inte med någon har man ändå säkrat betyg 3, lyckas man bara med den ena så får man betyg 4. Minuspoäng ges ej.

Poäng på Del 2 kan *eventuellt* räknas mot godkänt om det skulle behövas (t.ex. fyra godkända lösningar på Del 1 och en godkänd lösning på Del 2 kan ge en trea i betyg), men det är osannolikt att man klarar frågorna i Del 2 om man inte samtidigt klarat Del 1.

Tillåtna hjälpmedel: Den tresidiga lathund som finns tillgänglig på kurssidan och upptryckt i ett exemplar till varje student vid tentamen (man ska alltså inte ta med sig en egen kopia).

Tentamensgranskning: Efter rättning finns tentamen tillgänglig på expeditionen på plan 4 i EDIT huset och kan granskas där. Önskar man diskutera rättningen kommer man kunna boka tid för detta genom att fylla i det formulär som kommer finnas länkat till från kursidensida efter att tentan är rättad. Notera att tentan i så fall ska lämnas kvar på expeditionen.

-
- Implementeringar ska skrivas i Java. Oväsentliga syntax- och namnfelel eller liknande, betyder inte att lösningen underkänns. Det viktiga är att lösningar bedöms vara tillfredställande, d.v.s. att studenten med sitt svar tydligt visar förståelse för problemet i uppgiften och dess lösning.
 - Skriv tydligt och välstrukturerat. Svårförståeliga lösningar kan underkännas.
 - Lösningar som är onödigt krångliga eller inte följer god programmeringsstil, dels allmänt och dels med tanke på idéerna med objektorienterad programmering, kan underkännas.
 - Om det inte uttryckligen står motsatsen i uppgiften kan du använda klasser och metoder i Java:s API.
 - Om det inte uttryckligen står motsatsen i uppgiften får du definiera egna hjälpmetoder. Importeringar av klasser i Java:s API behöver inte skrivas ut.

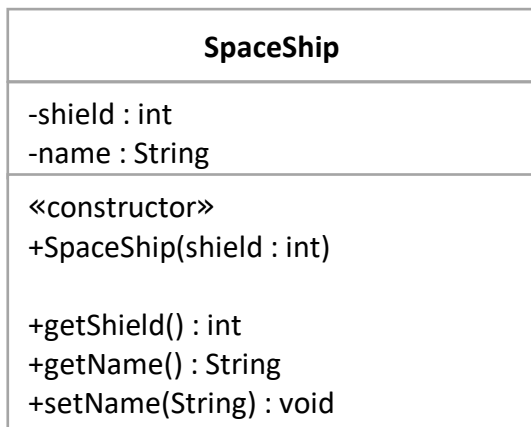
Lycka till!

DEL 1

För att få godkänt på tentan (betyg 3) måste du **lösa minst fem av sju uppgifter** i denna del.

Uppgift 1

Implementera en enkel klass `SpaceShip` som ska representera rymdskepp (i t.ex. ett spel). Klassen ska ha de instansvariabler och metoder som anges i UML-diagrammet nedan. Du ska inte lägga till några ytterligare metoder eller variabler. Lägga märke till vilken tillgänglighet (private eller public) instansvariabler och metoder ska ha. Konstruktorn ska ta ett argument som anger ett värde på styrkan hos skeppets sköld. Namnet behöver inte vara känt när ett `SpaceShip`-objekt skapas, och ska då vara en tom sträng.



DEL 1

För att få godkänt på tentan (betyg 3) måste du **lösa minst fem av sju uppgifter** i denna del.

Uppgift 2

En ordruta är en ruta där samma ord kan läsas (i samma ordning) både horisontellt och vertikalt. Ett exempel är denna ruta som innehåller orden ASK, SKO, KOL:

```
A S K
S K O
K O L
```

I Java kan vi representera en ordruta med hjälp av en tvådimensionell array innehållande chars. Skriv en metod:

```
public static boolean ärOrdruta(char[][] ruta)
```

som kontrollerar om en given ordruta verkligen innehåller ord som går att läsa både horisontellt och vertikalt. Du kan anta att `ruta.length > 0` och att `ruta.length == ruta[i].length` för alla tillåtna `i`, samt att alla chars i rutan representerar stora bokstäver (VERSALER). Metoden ska dock fungera för ordrutor av godtycklig storlek, alltså inte bara för rutor av trebokstavsord som i exemplet.

Tips: Fundera över vilka positioner i rutan som måste innehålla samma bokstav.

DEL 1

För att få godkänt på tentan (betyg 3) måste du **lösa minst fem av sju uppgifter** i denna del.

Uppgift 3

Vad skrivs ut av programmet nedan? Tänk efter noga!

```
public class Uppgift3 {  
  
    public static void minMetod(String[] p, String[] q, boolean r) {  
        p[1] = "Xylofon";  
        q[1] = "Yrväder";  
        r = false;  
    }  
  
    public static void main(String[] args) {  
        boolean b = true;  
        String[] x = {"Apa", "Björn", "Citron", "Docka", "Elefant"};  
        String[] y = x;  
        minMetod(x, y, b);  
        System.out.println(b);  
        System.out.println(x[1]);  
        System.out.println(y[1]);  
    }  
}
```

DEL 1

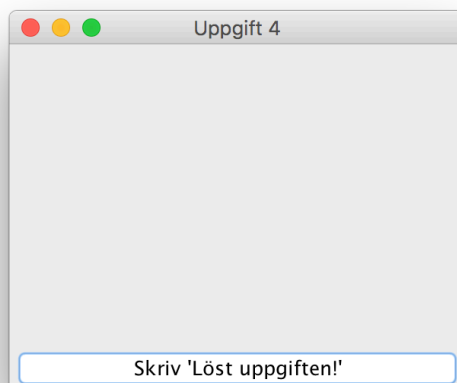
För att få godkänt på tentan (betyg 3) måste du **lösa minst fem av sju uppgifter** i denna del.

Uppgift 4

Koden i rutan hör till en enkel GUI-applikation (se nedan). Det är meningen att den ska skriva "Löst uppgiften" i terminalfönstret när man trycker på knappen. Men något saknas och knappen fungerar i nuläget inte som den ska.

Skriv den saknade koden. Ange även var i klassen du skulle lägga in den (radnummer).

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class Uppgift4 implements Runnable {
6
7     public static void main(String[] args) {
8         SwingUtilities.invokeLater(new Uppgift4());
9     }
10
11     public void run() {
12         JFrame frame = new JFrame("Uppgift 4");
13         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
14         JButton button = new JButton("Skriv 'Löst uppgiften!'");
15
16         frame.setPreferredSize(new Dimension(300, 250));
17         frame.add(button, BorderLayout.SOUTH);
18
19         frame.pack();
20         frame.setVisible(true);
21     }
22 }
```



DEL 1

För att få godkänt på tentan (betyg 3) måste du **lösa minst fem av sju uppgifter** i denna del.

Uppgift 5

Studera koden nedan noga. Vi har ett interface, `Doable`, samt klasserna `MyClass` och `MyClass2`. Ange samtliga rader (taggade med A-J) som **antingen orsakar kompileringsfel eller som kan orsaka ett runtime-fel** om programmet körs. Som svar räcker det att ange raden eller radernas bokstav.

- A. `MyClass a = new MyClass(3);`
- B. `MyClass2 b = new MyClass2(2, "a good deed");`
- C. `MyClass c = b;`
- D. `MyClass2 d = new MyClass(1);`
- E. `MyClass e = new MyClass2(1, "a tricky question");`
- F. `MyClass2 f = (MyClass2) a;`
- G. `Doable g = e;`
- H. `Doable h = (() -> System.out.println(42+3));`
- I. `MyClass2 i = (MyClass2) g;`
- J. `MyClass j = (MyClass) h;`

```
interface Doable {  
    public void doSomething();  
}
```

```
class MyClass implements Doable{  
    int number;  
  
    public MyClass(int number){  
        this.number = number;  
    }  
    public void doSomething(){  
        int i = number;  
        while(i > 0){  
            System.out.println("I did something!");  
            i--;  
        }  
    }  
}}
```

```
class MyClass2 extends MyClass{  
    String s;  
  
    public MyClass2(int number, String s){  
        super(number);  
        this.s = s;  
    }  
    public void doSomething(){  
        int i = number;  
        while(i > 0){  
            System.out.println("I did " + s + "!");  
            i--;  
        }  
    }  
}}
```

DEL 1

För att få godkänt på tentan (betyg 3) måste du **lösa minst fem av sju uppgifter** i denna del.

Uppgift 6

Studera klassen Uppgift6 i rutan nedan. Den ska ha en privat klassvariabel `counter` och en privat instansvariabel `x`. Dock har det skett ett misstag någonstans i klassens kod, och när vi försöker kompilera Uppgift6 får vi följande felmeddelande:

```
Uppgift6.java:13: error: non-static variable x cannot be
referenced from a static context
```

```
    x = newValue;
    ^
```

```
1 error
```

Vad är misstaget? Ange även rättad kod i ditt svar.

```
1 public class Uppgift6{
2     private static int counter = 0;
3     private int x;
4
5     public Uppgift6(){
6         x = counter;
7         counter++;
8     }
9     public static void setCounter(int newCounter){
10        counter = newCounter;
11    }
12    public static void setVariable(int newValue){
13        x = newValue;
14    }
15 }
```

DEL 1

För att få godkänt på tentan (betyg 3) måste du **lösa minst fem av sju uppgifter** i denna del.

Uppgift 7

Skriv en **generisk metod** (en metod som använder sig av Java generics) med namnet **byt** som byter plats på två element i en array med element av godtycklig typ. Metoden ska vara statisk och ha tre argument: en array samt två argument av typ `int` som anger index för de två elementen som ska byta plats.

DEL 2

Du behöver **bara svara på dessa frågor om du aspirerar på betyg 4 eller 5**. Vill du ha fyra ska du lösa en uppgift (välj själv) och för femma ska du lösa båda uppgifterna i den här delen, utöver de uppgifter du löst i Del 1.

Uppgift 8

Skriv en statisk metod `filteredListFromFile`. Metoden ska:

- Läs in en fil som innehåller ord separerade med blanksteg, där filnamnet ges som ett argument till metoden, av typen `String`. Om filen inte kan hittas ska metoden hantera det undantag som då kastas på ett lämpligt sätt (t.ex. skriva ut ett felmeddelande och returnera `null` eller en tom lista).
- Filtrera dessa ord efter ett godtyckligt villkor (tips: detta ska ges som ett argument till metoden). Metoden ska alltså kunna användas för att filtrera orden i inputfilen på många olika sätt.
- Metoden ska sedan returnera en lista av de strängar som uppfyller det givna villkoret som en `List<String>`.
- Skriv även ner hur du skulle anropa `filteredListFromFile` för *ett av de tre exemplen (valfritt)* nedan (men kom ihåg att metoden även ska kunna filtrera efter andra villkor än de i exemplen nedan).

Exempel på tänkt användning:

Anta att vi har en textfil `Uppgift8.txt` som innehåller följande:

```
apa bus citron docka ek fot
gris häst is jos ko lo mys
näsa oxe päron q ros snigel
tåg ugglan varg xylofon
yngling zebra åsna ägg ö
```

Exempel 1: Om villkoret är att vi vill ha alla strängar med `längd==3`, så ska `filteredListFromFile` för filen `Uppgift8.txt` ovan producera en lista som innehåller orden:

`apa, bus, fot, jos, mys, oxe, ros, tåg, ägg`

Exempel 2: Om villkoret istället är att strängens första bokstav är "a" eller "b" så ska `filteredListFromFile` för filen `Uppgift8.txt` ovan returnera en lista som innehåller orden:

`apa, bus`

Exempel 3: Om villkoret är att stängen har fem eller fler bokstäver så ska `filteredListFromFile` för filen `Uppgift8.txt` ovan returnera en lista som innehåller orden:

`citron, docka, päron, snigel, ugglan, xylofon, yngling, zebra`

Tips 1: Studera klassen `Scanner` som finns listad i Lathunden.

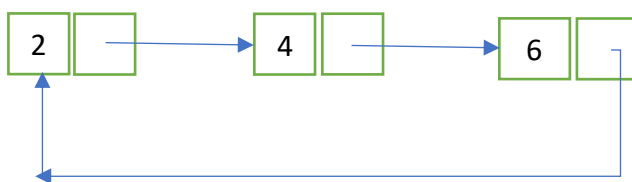
Tips 2: Studera även interfacet `Predicate<T>` i Lathunden.

DEL 2

Du behöver **bara svara på dessa frågor om du aspirerar på betyg 4 eller 5**. Vill du ha fyra ska du lösa en uppgift (välj själv) och för femma ska du lösa båda uppgifterna i den här delen, utöver de uppgifter du löst i Del 1.

Uppgift 9

En cirkulär länkad lista är en datastruktur som påminner om en vanlig länkad lista som vi sett under kursens gång, men med skillnaden att det sista länken länkar tillbaka till den första:



Listan ovan är ett exempel på en cirkulär lista som innehåller elementen [2,4,6]. Ditt jobb är att skriva en Javaklass `CircularList`. Klassen ska vara **generisk**. Du får **inte** använda någon biblioteksklass från Javas API i din lösning, du måste programmera datastrukturen för den cirkulära länkade listan själv.

Klassen ska implementera följande metoder:

- **public void add(E elem)**, som lägger till en länk med det nya elementet sist i listan.
- **public E get(int index)**, som returnerar det värde som finns på givet index. Listans första element ska som vanligt antas ha index 0. Till skillnad från en vanlig lista kan vi dock göra antagandet att om index är större än längden på listan -1, så fortsätter vi helt enkelt ytterligare ett "varv". Så i exemplet ovan ska
 - `get(0) == get(3) == get(6) osv,`
 - `get(1) == get(4) == get(7) osv,`
 - `get(2) == get(5) == get(8) osv.`

Tips 1: Fundera över hur du kan representera en länk i listan som en inre klass.

Tips 2: Det räcker att hålla reda på den sista länken i listan, eftersom listan är cirkulär.