

# Tentamen i Imperativ Programmering med Grundläggande Objektorientering, DIT012

Joachim von Hacht

**Datum:** 2016-04-04

**Tid:** 08.30-12.30

**Hjälpmedel:** Engelskt-Valfritt språk lexikon

**Betygsgränser:**

- U: -23
- G: 24-43
- VG: 44-60 (max 60)

**Lärare:** Joachim von Hacht. Någon besöker ca 10.00 och 11.30, tel. 772 10 03

**Granskning:** Tentamen kan granskas på studieexpeditionen. Vi ev. åsikter om rättningen eposta mig och ange noggrant vad du anser är fel så återkommer jag (ta en bild och skicka).

**Instruktioner:**

- För full poäng på essä-frågor krävs ett läsbart, begripligt och heltäckande svar. Generellt 1p för varje relevant aspekt av problemet. Oprecisa eller alltför generella (vaga) svar ger inga poäng. Konkretisera och/eller ge exempel. Det är aldrig någon risk att vara övertydlig!
- Det räcker med enbart relevanta kodavsnitt, övrig kod ersätts med “...” (aldrig import, main-metod, etc....)
- Överkomplicerade lösningar kan ge poängavdrag.
- Vi utgår från att användaren alltid skriver rätt och/eller gör rätt (d.v.s ingen felhantering behövs). Om felhantering skall ingå anges detta specifikt.

**LYCKA TILL...**

1. Vad avses med (förklara med en eller ett par meningar, du får gärna förtydliga med en skiss eller med kod)? 2p

a) Tilldelning.

b) Returtyp.

2. Skriv ett program som beräknar volymen av en sfär (ingen enhet behövs). Volymen ges av formeln  $V = \frac{4}{3}\pi r^3$  där  $r$  är radien. Du kan använda den färdiga konstanten  $\text{PI}$  från `Math` om du vill. Programmet skall fungera som nedan: 2p

```
Input radius > 10      (enter)
Volume : 4188.790204786391
```

3. Skriv en metod som givet antalet termer,  $n > 0$ , returnerar summan av  $n$  termer från serien  $\frac{1}{1} + \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{8} + \frac{1}{13} + \dots$ . Nämnarna utgörs av Fibonacci-talen. 4p
- Exempel:

```
out.println(sumSeries(10)); // Ger 3.3122872225813405
```

4. Förklara i detalj skillnaden mellan variabler (parametrar) med primitiv typ repektive referenstyp. Hur fungerar t.ex. tilldelning, likhet, metदानrop, m.m. Du måste ge kodexempel och illustrera din förklaring (rita bilder, boxar, pilar, o.s.v.). 6p

5. Skriv en metod som givet en array av heltal dubblar alla jämna värden (d.v.s. man får två lika värden i följd i resultatet). Metoden skall inte förändra original-array:en utan skapa och returnera en ny array. Exempel: 8p

```
[1, 2, 2, 1, 1, 2, 1, 2, 2]    // Indata
[1, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2]    // Resultat
```

6. Skriv en method som givet ett antal rader ritar ut en halv rektangel med tal enligt bilden nedan. Lösningen skall inte använda strängar. För utskrifter används enbart `out.print()` och `out.println()`. 6p

```
// Exemplet gäller för rader = 13
1
121
12321
1234321
123454321
12345654321
...
12345678910111213121110987654321
```

7. Spelet "Tre i rad" (Tic Tac Toe) spelas på en spelplan bestående av ett rutnät om 3 x 3 rutor. Två spelare placerar i tur och ordning ut sina tre märken (X eller O). Därefter kan spelarna flytta sina märken till närliggande fria rutor. Den som först får alla märkena i rad: vertikalt, horisontellt eller diagonalt, vinner.

12p

- a) Skapa en klass för en spelplanen, TicTacToeBoard. Följande metoder skall finnas i klassen:

```
// Placera ch (d.v.s. 'X' eller 'O' ) på given rad och kolumn
public void put(int row, int col, char ch)
// Flytta märket från rad/kolumn r1/c1 till r2/c2
public void move(int r1, int c1, int r2, int c2)
// Sant om rutan tom
boolean isEmpty(int row, int col)
// Sant om rutan innehåller tecknet mark
boolean isOwner(int row, int col, char mark)

// Utskrift av spelplan. Behöver inte implementeras.
0 - -      (- = tom ruta)
X 0 -
X - -
```

- b) Visa hur du skapar en instans av klassen och hur man genom att använda metoderna låter aktuell spelare flytta sitt märke från en position till en annan. Koden skall undersöka om flyttningen är tillåten, i så fall flyttas märket annars händer inget. Du kan anta att det finns en färdig metod som givet rader och kolumner anger om rutorna är grannar. För att förenkla kan du hårdkoda alla värden.
8. Ibland vill man jämför hur lika, rent ljudmässigt, två namn är. Detta kan åstadkommas genom att koda namnen på ett visst sätt. Namn med samma eller närliggande kod "låter" lika. T.ex. får både "robert" och "rupert" samma kod, "r163". Kodningen ger alltid en sträng med en inledande bokstav som är samma som den inledande i namnet plus tre siffror. Bokstäver kodas enligt:

12p

- Om bokstaven är en konsonant eller h eller w stryks den.
- Annars ges kodsiffran enligt:

Bokstav	Kod
b, f, p, v	1
c, g, j, k, q, s, x, z	2
d, t	3
l	4
m, n	5
r	6

- Flera förekomster i rad av en bokstav kodas med en enda siffra t.ex "bb" blir bara "1".
- Skulle namnet var för kort för att ge tre siffror fyller man på med nollor på slutet. Skulle namnet ge mer än tre siffror så bortser man från dessa.

Implementera en metod som givet ett namn (String) returnerar koden (String) för detta.

TIPS: Skapa en hjälpmetod som givet ett tecken ger koden (i form av ett tecken).

Låt metoden använda två strängar enligt:

```
String keys    = "abcdefghijklmnopqrstuvwxyz";  
String values  = "01230120022455012623010202"; // Se tabell ovan
```

Du får använda följande metoder från String:

- charAt(int i), ger tecknet vid index i.
- indexOf(char ch), ger index för tecknet ch, -1 om tecknet saknas.
- length() ger längden av strängen.
- substring(int start, int end), ger en delsträng från start (inkl.) till end-1.
- substring(int start), ger en delsträng från start (inkl.) till strängens slut.

StringBuilder med metoden append() kan också användas (lägger till sist)

9. Ange vilka rader i koden nedan som inte att kompilerar och förklara varför. 8p

```
public class Mix {  
    public static int i = 1;  
    public int j = 2;  
    public static void doIt(){  
        i = 5;  
        i = j;  
        j = i;  
        doOther();  
        this.doOther();  
        new Mix().j = 5;  
        new Mix().doOther();  
    }  
    public void doOther(){  
        i = 5;  
        i = j;  
        j = i;  
        doIt();  
    }  
}}
```