

## 1DV506, Problemlösning och programmering, ht 2017

---

# Laboration 2

---

**Problem?** Tveka inte att fråga er labbhandledare på labträffarna eller chat-forumet på Moodle (eller Jonas på lektionerna) om ni har problem med någon av uppgifterna.

**Eclipse förberedelser:** Skapa ett nytt paket med namnet `DittLnuAnvändarNamn_lab2` inne i Java-projektet 1DV506 och spara alla filer relaterade till denna lab i det paketet.

**Notera:** Varje uppgift skall lösas i form av ett komplett program. Det innebär att till VARJE uppgift skall det finnas åtminstone EN klass som innehåller en `main`-metod.

**VG-uppgifter:** Uppgifterna 7, 8, 9 och 13 är sk VG-uppgifter som bara är obligatoriska för dom av er som siktar på de högre betygen A och B.

## Lektion 4 - Styrande satser

Att kunna kombinera olika styrande satser för att lösa ett problem är mycket viktigt. Vi har därför ett ganska stort antal (korta) uppgifter på detta tema.

- **Uppgift 1**

Skriv ett program `RaknaA.java` som läser in en textrad från tangentbordet och sedan skriver hur ut många 'a' och 'A' det finns i raden. En körning kan se ut enligt följande:

```
Mata in en textrad: Alla barn fick högsta betyget A.  
Antal a: 3  
Antal A: 2
```

- **Uppgift 2**

Skriv ett program `Baklanges.java` som läser in en textrad från tangent bordet och sedan skriver ut den baklänges. En körning kan se ut enligt följande:

```
Mata in en textrad: Anakin Skywalker  
Baklänges: reklawykS nikanA
```

**Notera:** Du förväntas lösa denna uppgift genom att iterera över den inlästa strängen baklänges.

- **Uppgift 3**

Skriv ett program `StorstaK.java` som för ett godtyckligt positivt heltal N (läses in från tangentbordet) beräknar det största värdet K sådant att  $0+2+4+6+8+\dots+K < N$ . En körning kan se ut enligt följande:

```
Mata in ett positivt heltal: 25  
Största tal K sådant att  $0+2+4+6+\dots+K < 25 \Rightarrow K=8$ 
```

**Notera:** Om ett icke-positivt heltal matas in skall programmet avslutas med ett lämpligt felmeddelande.

- **Uppgift 4**

Skriv ett program `HighLow.java` som implementerar gissningsleken *High and Low*. Datorn väljer ett

slumptal mellan 1 och 100 och låter sedan spelaren gissa dess värde. Efter varje gissning ges en ledtråd av typen "högre" eller "lägre". En körning kan se ut enligt följande:

```
Gissning 1: 67
  Ledtråd: högre
Gissning 2: 82
  Ledtråd: lägre
Gissning 3: 77
  Rätt svar efter bara 3 gissningar - Strålande!
```

Programmet avbryts efter 10 gissningar med en lämplig slutkommentar.

### • Uppgift 5

Skriv ett program `Triangle.java` som för ett godtyckligt positivt *udda* heltal  $N$  (läses in från tangentbordet) skriver ut två olika trianglar. Först en rätvinklig sedan en likbent. Exempel på körning:

```
Mata in ett udda heltal: 7
Rätvinklig triangel:
  *
 **
***
****
*****
*****
*****

Likbent triangel:
  *
 ***
****
*****
*****
```

Notera, heltalet  $N$  bestämmer hur lång basen på varje triangel är. Programmet skall avslutas med ett felmeddelande om det inmatade talet inte är ett positivt udda heltal.

### • Uppgift 6

Skriv ett program `NastStorsta.java` som läser in tio heltal och hittar det näst största talet av dem och därefter skriver ut det. En körning kan se ut enligt följande:

```
Mata in 10 heltal: 67 -468 36 1345 -7778 0 34 7654 45 -666
Det näst största talet är: 1345
```

Försök att göra programmet så att det är lätt att ändra antalet inmatade heltal.

**Rekommendation:** Använd mindre än 10 heltal under tiden ni utvecklar programmet.

**Notera:** Du får inte använda arrayer (eller andra datastrukturer) för att temporärt lagra alla de inlästa heltalen innan du börjar leta efter det näst största heltalet.

### • Uppgift 7 (VG-uppgift)

Skriv ett program `RaknaSiffror.java` som för ett godtyckligt positivt heltal  $N$  (läses in från tangentbordet) skriver ut antalet nollor, antalet udda siffror och antalet jämna siffror. En körning kan se ut enligt följande:

```
Mata in ett heltal: 6789500
Nollor: 2
Udda: 3
Jämna: 2
```

**Notera:** Noll är (i denna uppgift) varken udda eller jämnt.

### • Uppgift 8 (VG-uppgift)

Skriv ett program `TartLjus.java` som beräknar hur många askar tårtljus en person behöver köpa varje år fram till hundraårsfirandet. Ni kan anta att antalet ljus i tårtan är samma som det år man fyller, att man sparar överblivna ljus från ett år till ett annat, och att varje ask innehåller 24 ljus. I slutet av programet skall ni också skriva ut det totala antalet askar man måste köpa och hur många ljus som finns kvar efter man firat sin 100:e födelsedag. En körning kan se ut enligt följande:

```
Before birthday 1, buy 1 box(es)
Before birthday 7, buy 1 box(es)
Before birthday 10, buy 1 box(es)
Before birthday 12, buy 1 box(es)
Before birthday 14, buy 1 box(es)

...

Before birthday 95, buy 3 box(es)
Before birthday 96, buy 4 box(es)
Before birthday 97, buy 5 box(es)
Before birthday 98, buy 4 box(es)
Before birthday 99, buy 4 box(es)
Before birthday 100, buy 4 box(es)

Total number of boxes: 211, Remaining candels: 14
```

**Notera:** I utskriften ovan redovisar vi bara de år då man måste köpa nya askar. De år som inte skrivs ut (tex 2-6, 8-9) klarar man sig utan att köpa några nya ljus.

### • Uppgift 9 (VG-uppgift)

Skriv ett program `Palindrom.java` som testat om en textrad (läses in från tangentbordet) är ett palindrom. Dvs, att textraden har samma sekvens av bokstäver vare sig man läser den framifrån eller bakifrån. Ni skall bortse från stora och små bokstäver, siffror och andra tecken som inte är bokstäver. Exempel på palindrom

```
"Anna"  "x"  "Ff"  "A1 n2%3na"  "Ni talar bra latin."
```

**Ledtråd:** De två statiska metoderna `Character.isLetter(char c)` och `Character.toLowerCase(char c)` kan här komma till nytta!

## Lektion 5 - Arrayer och ArrayList

### • Uppgift 10

Gör klart programmet `BytaPlats.java` nedan så att:

1. Först skrivs innehållet i arrayen text ut.
2. Ändra sedan innehållet i arrayen text, så att det första elementet byter plats med det sista, det andra byter plats med det näst sista o.s.v.
3. Sedan skrivs innehållet i arrayen text ut igen.

```
public class BytaPlats {
    public static void main(String[] args) {
        char[] text = { 't', 't', 'ä', 'l', ' ', 'r', 'a', 'v', ' ', 'a',
                        't', 't', 'e', 'D' };

        // Fortsätt här ...
    }
}
```

Observera att du inte bara ska skriva ut tecknen baklänges utan verkligen byta plats på dem i arrayen!

- **Uppgift 11**

Skriv ett program `OmvandOrdning.java` som läser ett godtyckligt antal positiva heltal från tangentbordet och sedan skriver ut dem baklänges. Inmatningen sker fram tills det att användaren matar in ett negativt tal. En körning kan se ut enligt följande:

```
Mata in positiva heltal. Avsluta med ett negativt.
tal 1: 5
tal 2: 10
tal 3: 15
tal 4: 20
tal 5: -7

Antal positiva: 4
Baklänges: 20, 15, 10, 5
```

Notera: Man skall inte behöva ange hur många tal man ämnar mata in.

- **Uppgift 12**

Skriv ett program `Frekvenstabell.java` som simulerar 6000 tärningskast. För att visa att sannolikheten att få en etta, tvåa, trea, fyra, femma eller sexa är ungefär lika stor, skall man skapa en frekvenstabell som lagrar hur många ettor, tvåor, o.s.v. som slumpats fram. (Använd en array för att lagra hur många ettor, tvåor, o.s.v. som slumpats fram.) En körning kan se ut enligt följande:

```
Frequencies when rolling a dice 6000 times.
1: 1025
2: 996
3: 980
4: 1006
5: 1035
6: 958
```

- **Uppgift 13 (VG-uppgift)**

Då en del fackförbund informerar om utfallet av löneförhandlingar redovisar de medianlönen (d.v.s. den mittersta lönen), genomsnittslönen och lönespridningen (d.v.s. skillnaden mellan den högsta och lägsta lönen). Skriv ett program `Lonerevision.java` som kan läsa in ett godtyckligt antal löner och sedan beräkna medianlön, medellön och lönespridning.

Medianlönen är den mittersta av de angivna lönerna och för att kunna bestämma den måste lönerna sorteras. Enklaste sättet att sortera en `ArrayList` är att använda sig av den statiska metoden `sort` som finns i klassen `Collections` (i Javabiblioteket `java.util`). Är det ett jämt antal löner så bestäms medianlönen genom att medelvärdet av de två mittersta lönerna beräknas. Två körningar kan se ut enligt följande:

```
Mata in löner (och avsluta med 'X'): 21700 28200 26300 25100 22600 22800 19900 X
Medianlön: 22800
Medellön: 23800
Lönespridning: 8300
```

```
Mata in löner (och avsluta med 'X'): 22100 29800 27300 25400 23100 22300 X
Medianlön: 24250
Medellön: 25000
Lönespridning: 7700
```

Notera, följande teknik kan användas för att läsa in ett godtyckligt antal heltal från tangentbordet

```
Scanner scan = new Scanner(System.in);
...
while (scan.hasNextInt()) {
```

```
    int salary = scan.nextInt();  
    ...  
}
```

---

## Redovisning

Alla uppgifter skall lämnas in och vi är bara intresserade av era .java- filer (och VG-uppgifterna 7,8,9 och 13 är inte obligatoriska). Zippa därför ihop .java-filerna i katalogen DittLnuAnvändarNamn\_lab2 (som finns inuti katalogen src) och lämna in dem mha Moodles redovisningssystem.