

Inlämningsuppgifter omgång 1

Allmänna instruktioner: Det finns två delar i detta dokument: Del 1 som innehåller uppgifter som måste göras för att få godkänt, och Del 2 som innehåller uppgifter för att få bonuspoäng till tentan för högre betyg. För att bli godkänd på inlämningsuppgifterna räcker det således att göra Del 1.

Döp era Javaprogram till de filnamn som specificeras i uppgifterna. Detta innebär t.ex. att programmet i Uppgift 1 ska heta NameReversed.java.

Spara alla java-filer i en katalog "<användarnamn>_in1", t.ex. johthu19_in1. Komprimera denna katalog till en zip-fil och skicka in zip-filen.

Deadline för att lämna in uppgifterna på Blackboard är söndagen 8/11 klockan 23:59:59.

Uppgifterna presenteras muntligen på examinationstillfället för din grupp 9/11 - 10/11.

Del 1 – uppgifter för godkänt

I dessa inlämningsuppgifter så övar vi oss på primitiva och inbyggda datatyper, värden uttryck och kommandon, och typkonvertering. I denna första omgång inlämningsuppgifter är vi flexibla vad gäller hur ni väljer att ge input till era program. I varje uppgift nedan så kan ni välja att ange input till programmet antingen via kommandoraden (terminalen), eller genom att använda metoderna i Scanner.

Uppgift 1: NameReversed.java

Skriv ett program där användaren anger ett förnamn och sedan ett efternamn, varpå programmet skriver ut namnet i omvänd ordning. T.ex. om namnet är Anders Andersson så ska programmet skriva ut: Namnet baklänges är: Andersson Anders.

Tips: Input till ett program kan ges via

terminalen

```
public class NameReversed{
    public static void main(String[] args){
        String firstName = args[0];
        String surName   = args[1];
        System.out.println("Ditt namn baklänges är: ");
    }
}
```

Scanner

```
import java.util.Scanner;

public class NameReversed_scanner
{
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);
        System.out.print("Ange förnamn:");
        String first_name = obj.nextLine();
        System.out.print("Ange efternamn:");
        String surname = obj.nextLine();
        System.out.println("Namnet baklänges är: ");
    }
}
```

Uppgift 2: Bmi.java

Skriv ett program som beräknar Body Mass Index (BMI) för en person med en viss vikt och en viss längd. BMI beräknas enligt följande:

$$\text{BMI} = \text{vikt} / (\text{längd} * \text{längd}),$$

Där vikt anges i kilo (kg) och längd anges i meter (m).

Till exempel kommer BMI för en person som väger 70 kg och är 1.68 m lång vara $(70 / (1.68 * 1.68))$.

Programmet läser in vikten och längden för en person och skriver sedan BMI-värdet på skärmen.

Uppgift 3: CashRegister (cashregister.py)

Detta är en övning i att använda sig av heltalsdivision / och rest-operatorn.

Du ska skapa ett program som fungerar som en kassaapparat. Inputen till programmet är hur mycket man har handlat för och hur mycket man har betalat till kassören.

Outputen är hur mycket man skall få tillbaka och även hur många sedlar och mynt av varje valör man skall få tillbaka. Valörer är: 100, 50, 20, 10, 1.

Exempel: om användaren anger att denne betalade 581 för något som kostade 100 så skrivs det ut på skärmen:

Du får tillbaka: 4 hundralappar, 1 femtiolappar, 1 tjugolappar, 1 tior, och 1 enkronor.

Tips: Om du sparar kostnaden i en variabel `cost` och det du har betalat i en variabel `pay` så är det du ska få tillbaka:

```
rest = pay - cost;
```

Antalet hundralappar som ska erhållas är:

```
rest/100
```

Sen uppdaterar du rest:

```
rest = rest%100;
```

och räknar ut antalet 50-lappar som ska fås tillbaka genom:

```
rest/50
```

Sen uppdaterar du rest:

```
rest = rest%50;
```

osv...

Uppgift 4: Time.java

Detta är ytterligare en övning i att använda sig av heltalsdivision / och rest-operatorn.

Skriv ett program som läser in en tid angiven i sekunder och sedan räknar ut hur många dagar, timmar, minuter och sekunder detta blir. Till exempel: 180122 sekunder blir 2 dagar, 2 timmar, 2 minuter och 2 sekunder.

Uppgift 5: LeapYear.java

Detta är en övning i att använda sig av logiska operatorer, samt slumptal.

Skriv ett program som skapar ett slumpmässigt årtal mellan 2000 och 3000. Sedan tar programmet reda på om detta är ett skottår eller inte enligt den gregorianska kalendern. Programmet skriver ut true på skärmen om året är ett skottår, om så inte är fallet skriver programmet ut false på skärmen.

Enligt den gregorianska kalendern är ett år ett skottår om det är jämnt delbart med 400, eller om det är jämnt delbart med 4 men inte jämnt delbart med 100.

Tips: om du ska generera ett slumptal **tal** mellan heltalen **min** och **max**, så kan du göra ett av två alternativ:

```
1. int tal = min + (int)(Math.random()*(1 + (max - min)));
```

```
2. import java.util.Random; ...  
   Random r = new Random();  
   int tal = min + r.nextInt(max - min);
```

För att undersöka om slumptalet är ett skottår, så översätter ni beskrivningen ovan till ett uttryck vars värde ni sparar i en variabel av typ boolean.

Del 2 – uppgifter för bonuspoäng

När ni löser uppgifterna får endast metoder som har gåtts igenom hittills i kursen användas. Villkorssatser (if, else etc.) är t.ex. inte tillåtna.

Uppgift 1: OrderCheck.java (2p)

I denna uppgift ska användaren slumpa ett heltal mellan 100 och 999. Sedan ska det undersökas om de tre siffrorna i talet är i fallande ordning eller i stigande ordning. Om detta är fallet ska talet samt true på skärmen, annars skrivs talet samt false ut på skärmen.

Vad som menas är att om talet är abc, där a, b, och c är siffror, så ska man undersöka om $a \geq b \geq c$ (vilket är fallande ordning) eller om $a \leq b \leq c$ (vilket är stigande ordning).

Ett exempel är talet 379. Eftersom $3 \leq 7 \leq 9$ (stigande ordning) så skrivs följande ut på skärmen:

```
379:true
```

Ett annat exempel är 313. Eftersom de tre talen 3, 1, och 3 inte förekommer i varken fallande eller stigande ordning så skrivs följande ut på skärmen

```
313:false
```

Uppgift 2: DayOfTheWeek.java (2p)

I denna uppgift ska vi använda [Zellers metod](#) för att räkna ut vilken veckodag ett angivet datum motsvarar. Detta är en kreativ (och knepig) övning i att använda de matematiska operatorerna. Lös alla delproblem, saker som ska räknas ut, genom endast matematiska yttryck (med hjälp av operatorerna +, -, *, /, och %) och typkonvertering.

Användaren anger:

- en månad **month**, som är ett heltal mellan 1 och 12 (1 för januari, 2 för februari osv.)
- en dag **day**, som är ett heltal mellan 1 och antalet dagar i månaden.
- Ett år **year** som är ett heltal.

Sedan räknas följande ut:

- Heltal **Y** som är lika med **year** om **month** > 2, annars **year-1**.
- Heltal **y** som är de sista två siffrorna i **Y**. Om t.ex. **Y** = 1999, så är **y** = 99.
- Heltal **c** som är de första två siffrorna i **Y**. Om t.ex. **Y** = 1999, så är **c** = 19.
- Heltal **m** som är ett slags förskjutning av **month**. Så att Mars motsvarar 3, april motsvarar 4, ..., januari motsvarar 13 och februari motsvarar 14.

Sedan räknas veckodagen D ut genom följande uttryck

$$D = ((13*(m+1))/5 + y/4 + c/4 + day + y - 2*c) \% 7$$

Där 0 motsvarar lördag, 1 motsvarar söndag, ..., 6 motsvarar fredag. Om ni skulle få negativa tal för något datum så motsvarar -1 fredag, -2 torsdag,...

Heltalet D skrivs ut på skärmen.