

TENTAMEN OOP 2013-03-13

ANVISNINGAR

- Påbörja varje ny uppgift på nytt blad.
- Skriv endast på ena sidan av bladen.
- Skriv tydligt - oläsbara svar beaktas ej.

BETYGSÄTTNING

Max antal poäng är 30.

För att bli godkänd på tentan (minst betyg E) krävs dels minst 4 poäng sammanlagt på uppgift 1 och uppgift 2 och dels minst 15 poäng sammanlagt på hela tentan.

För högre betyg krävs:

- Betyg D: minst 18 poäng samt högst en uppgift med 0 poäng.
- Betyg C: minst 21 poäng samt ingen uppgift med 0 poäng.
- Betyg B: minst 24 poäng samt ingen uppgift med 0 poäng.
- Betyg A: minst 27 poäng samt uppgifterna lösta med korrekt användande av objektorienterade principer (t.ex. inkapsling, ej upprepning av kod).

Betyget Fx med möjlighet att komplettera ges till studenter som fått 12-14 poäng eller som fått mer än 14 poäng men missat att få 4 poäng på uppgift 1+2.

HJÄLPMEDEL

De enda tillåtna hjälpmedlen är en (1) valfri bok om Java och en (1) bok som inte behandlar programmering i någon form.

Lycka till!

Lösningförslag läggs upp i Moodle senast tre arbetsdagar efter tentatillfället.

UPPGIFT 1: KODFÖRSTÅELSE IMPERATIV PROGRAMMERING (6 POÄNG)

Om man exekverar följande programrader, vad kommer att skrivas ut på skärmen? Du skall i dina svar vara noga med vad som skrivs på vilken rad, alltså beakta skillnaden mellan print och println.

DELUPPGIFT A (2 POÄNG)

```
System.out.println("5+6");  
System.out.println(" "+5+6);  
System.out.println(" "+(5+6));  
System.out.println(5+" "+6);
```

KORREKT SVAR

5+6
56
11
5+6

VARFÖR BLIR SVARET VAD DET BLIR?

- "5+6" är en textsträng som skrivs ut precis som den står.
- ""+5+6 börjar med en tom textsträng. Till denna konkateneras (läggs till) en int (5) som först konverterats till en sträng. Resultatet av detta är en ny sträng "5" till vilken man konkatenerar ytterligare en int (6).
- ""+(5+6) börjar också med en tom textsträng till vilken man ska konkatenera en int, men eftersom (5+6) står inom parentes så läggs de ihop till 11 innan konkateneringen.
- 5+" "+6 är en variant av det andra fallet där en int (5) och en sträng (" ") konkateneras till en ny sträng: "5+" som konkateneras med ytterligare en int (6).

DELUPPGIFT B (2 POÄNG)

```
for (int n = 0; n < 10; n += 3) {  
    switch (n) {  
        case 0:  
            System.out.println("Litet");  
        case 2:  
        case 4:  
            System.out.println("Större");  
        case 6:  
        case 8:  
            System.out.println("Störst");  
            break;  
        default:  
            System.out.println("Det var udda");  
    }  
}
```

KORREKT SVAR

Litet
Större
Störst
Det var udda
Störst
Det var udda

VARFÖR BLIR SVARET VAD DET BLIR?

Det första varvet i loopen är $n=0$. Switch-satsen väljer därför det första valet (0) och skriver ut "Litet". Eftersom detta inte följs av ett break så görs det andra valet (2) också, men här finns ingen kod, så vi faller vidare ner till det tredje valet (4) och skriver ut "Större". Inte heller här finns något break så vi går vidare neråt, förbi 6 och in på 8 där vi skriver ut "Störst". Efter denna utskrift finns ett break, så switch-satsen avbryts och vi kör nästa varv i loopen.

Det andra varvet i loopen är $n=3$ vilket inte matchar något av fallen i switch-satsen som därför går in i default-grenen och skriver ut "Det var udda".

Det tredje varvet är $n=6$. Så vi går in i switch-satsen på "case 6" och faller ner till "case 8" där vi skriver ut "Störst". Efter denna utskrift finns ett break, så switch-satsen avbryts och vi kör nästa varv i loopen.

Det fjärde varvet är $n=9$ och vi får samma beteende som under det andra varvet.

Efter det fjärde varvet blir $n=12$, vilket är större än 10 så loopen avbryts.

DELUPPGIFT C (2 POÄNG)

```
int i = 7;  
  
System.out.println(i++ / 4);  
  
System.out.println(i++ / 4.0);
```

KORREKT SVAR

1
2.0

VARFÖR BLIR SVARET VAD DET BLIR?

Variabeln i är 7 från början. I den första utskriften så delas detta värde med 4 vilket är 1.75, men eftersom både täljaren och nämnaren är heltal så görs en heltalsdivision. Vid heltalsdivision så kastas alla decimaler och kvar blir bara heltalsdelen 1 som skrivs ut.

Efter detta ökas i med 1 till 8. Om den första utskriften hade sett ut så här istället:

```
System.out.println(++i / 4);
```

Så hade i ökat först och divisionen varit $8/4$ vilket är 2.

I den andra utskriften så är i alltså 8 vilket blir 2 när det delas med 4, men eftersom vi delar med 4.0, ett flyttal, så blir även resultatet ett flyttal. Det är därför resultatet skrivs ut med en decimal.

UPPGIFT 2: KODFÖRSTÅELSE KLASSER OCH OBJEKT (6 POÄNG)

Vad kommer att skrivas ut om man kör nedanstående Java-program?

```
public class Tomteland {  
  
    public static void main(String[] args) {  
        Tomte[] tomte = new Tomte[5];  
        tomte[0] = new Gniden("tras");  
        tomte[1] = new Nisse("blå");  
        tomte[2] = new Gniden("barbie");  
        tomte[3] = new Nisse("grön");  
        tomte[4] = new Gniden("lego");  
  
        for (Tomte t : tomte) {  
            t.ge();  
        }  
  
        System.out.println(Tomte.utdelat());  
        System.out.println(tomte[0]);  
        System.out.println(tomte[1]);  
    }  
}
```

```
abstract class Tomte {  
  
    private static int klappar = 0;  
    private String favorit="docka";  
  
    public Tomte() {  
    }  
  
    public Tomte(String typ) {  
        favorit = typ + favorit;  
    }  
  
    public static int utdelat(){  
        return klappar;  
    }  
  
    public void ge() {  
        klappar++;  
    }  
  
    public String toString() {  
        return favorit + klappar;  
    }  
}
```

```
class Nisse extends Tomte {  
  
    private String färg = "grå";  
  
    public Nisse(String färg) {  
        this.färg = this.färg + färg;  
    }  
  
    public String toString() {  
        return färg;  
    }  
}  
  
class Gniden extends Tomte {  
  
    public Gniden(String str) {  
        super(str);  
    }  
  
    public void ge() {  
    }  
  
    public String toString() {  
        return "Jättemånga " +  
            super.toString();  
    }  
}
```

KORREKT SVAR

2

**Jättemånga trasdocka2
gråblå**

VARFÖR BLIR SVARET VAD DET BLIR?

```
System.out.println(Tomte.utdelat());
```

Skriver ut värdet (2) av den statiska variabeln klappar i tomtar. Denna ökas med ett i metoden ge() i Tomte som anropas i loopen i huvudprogrammet. Anledningen till att den inte är 5 är att Gniden överskuggar metoden och inte delar ut några klappar. Eftersom tre av tomtarna är gnidna så är det bara två kvar som kan ge några klappar.

```
System.out.println(tomtar[0]);
```

Anropar toString() på den första tomten i arrayen. Denna tomte är gniden så resultatet blir "Jättemånga" plus resultatet av ett anrop på toString() i superklassen Tomte. Denna i sin tur ger favoriten "docka", som ändrats till "trasdocka" i konstruktorn, plus värdet av klappar som är detsamma som vi precis skrev ut, alltså 2.

```
System.out.println(tomtar[1]);
```

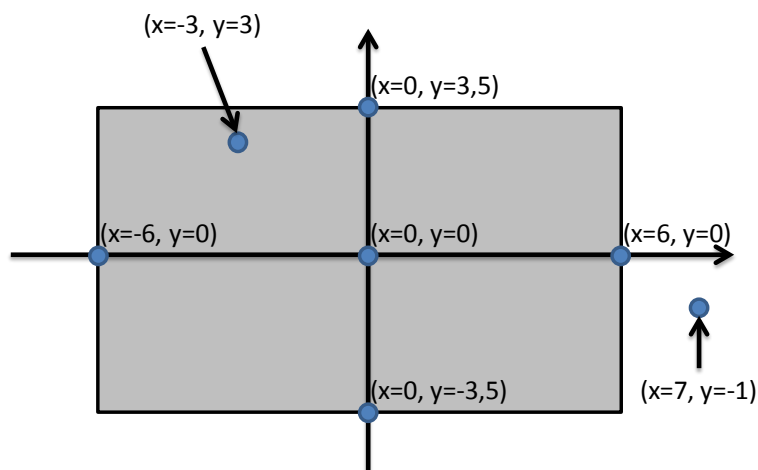
Anropar toString() på den första tomten i arrayen. Denna tomte är en nisse vars färg satts till "gråblå" i konstruktorn, och det enda toString() gör är att returnera denna färg.

UPPGIFT 3: PUNKTER I REKTANGEL (6 POÄNG)

Skriv ett program som läser in koordinaterna för en punkt i form av två tal från användaren och som skriver ut om punkten ligger inom en rektangel med bredden 12 och höjden 7 centrerad på koordinaterna (0, 0) enligt figuren till höger.

Ett exempel på hur programmet ska fungera:

Skriv x-koordinatens värde: `-3`
Skriv y-koordinatens värde: `3`
Punkten finns inom rektangeln



Eller om punkten man skriver in inte finns:

Skriv x-koordinatens värde: `7`
Skriv y-koordinatens värde: `-1`
Punkten finns INTE inom rektangeln

LÖSNINGSFÖRSLAG 1

```
import java.util.Scanner;

public class InomRektangel1 {

    public static void main(String[] args) {

        Scanner keyboard = new Scanner(System.in);

        System.out.print("Skriv x-koordinatens värde: ");
        double x = keyboard.nextDouble();
        System.out.print("Skriv y-koordinatens värde: ");
        double y = keyboard.nextDouble();

        if(x>=-6 && x<=6 && y>=-3.5 && y<=3.5){
            System.out.println("Punkten finns inom rektangeln");
        }else{
            System.out.println("Punkten finns INTE inom rektangeln");
        }
    }
}
```

UPPGIFT 4: KODLEDTRÅDAR (6 POÄNG)

Många av er har antagligen spelat spelet Mastermind, där en spelare ska gissa en hemlig kod bestående av fyra färgade pinnar. Spelet påminner starkt om ett betydligt äldre pappersbaserat spel som på engelska heter "Bulls and Cows".

"Bulls and Cows" spelas av två spelare och går till så att bägge spelarna skriver varsin fyrsiffrig kod på ett papper. Ingen siffra får förekomma mer än en gång i koden. Därefter turas de om att gissa varandras kod och får ledtrådar till hur bra deras gissningar var i form av antal "Bulls": rätt siffra på rätt plats, och antal "Cows": siffror som förekommer i koden men på fel plats.

Exempel:

Om den hemliga koden som ska gissas är 1385, och gissningen är 8325 så är två siffror på rätt plats (3 och 5) och ytterligare en siffra (8) ingår i koden men står på fel plats.

Ytterligare exempel, om gissningen är 1357 och rätt kod 3456 så skall er metod skriva ut:

1 Bull (korrekt siffra på rätt plats, avser i exemplet siffran 5)

1 Cow (korrekt siffra men på fel plats, avser i exemplet siffran 3)

Enligt Wikipedia, från vilken ovanstående information är tagen, så fanns det datoriserade versioner av spelet redan i slutet av 60-talet, och vi vill naturligtvis föra den traditionen vidare. En fullständig implementation skulle vara för stor för en tentauppgift, så vi koncentrerar oss på en enda metod; den som ger ledtrådarna.

Din uppgift är att skriva klart nedanstående metod vars uppgift det är att jämföra en gissning med den rätta koden och skriva ut ledtrådarna. Du behöver inte göra några indatakontroller utan kan anta att bägge arrayerna innehåller fyra stycken unika heltal. Du får lägga till hjälpmetoder om det underlättar uppgiften.

```
void kontrollera(int[] gissning, int[] rätt) {  
    // skriv klart denna metod  
}
```

LÖSNINGSFÖRSLAG

```
void kontrollera(int[] rätt, int[] gissning) {  
    // Att kontrollera Bulls, alltså rätt siffra på rätt plats, är enkelt  
    int bulls = 0;  
    for (int n = 0; n < 4; n++) {  
        if (rätt[n] == gissning[n]) {  
            bulls++;  
        }  
    }  
  
    // Att kontrollera Cows, alltså rätt siffra, men på fel plats, är  
    // aningen mer komplicerat  
    int cows = 0;  
    for (int n = 0; n < 4; n++) {
```



```

    for (int m = 0; m < 4; m++) {
        if (n != m && rätt[n] == gissning[m]) {
            cows++;

            // Detta break är inte nödvändigt för att programmet ska
            // fungera, men om vi har hittat en matchande siffra är det
            // ingen ide att leta efter flera eftersom det inte får
            // förekomma dubletter, så vi avbryter den inre loopen i
            // för tid.
            break;
        }
    }
}

System.out.println(bulls+" Bulls");
System.out.println(cows+" Cows");
}

```

KOMPLETT PROGRAM

```

import java.util.Random;
import java.util.Scanner;

public class BullsAndCows {

    // Detta är den enda delen av lösningsförslaget som
    // ingick i frågan. Resten är till för att göra ett komplett program för den
    // som vill testa.
    private static void kontrollera(int[] rätt, int[] gissning) {
        // Att kontrollera Bulls, alltså rätt siffra på rätt plats, är enkelt
        int bulls = 0;
        for (int n = 0; n < 4; n++) {
            if (rätt[n] == gissning[n]) {
                bulls++;
            }
        }

        // Att kontrollera Cows, alltså rätt siffra, men på fel plats, är
        // aningen mer komplicerat
        int cows = 0;
        for (int n = 0; n < 4; n++) {
            for (int m = 0; m < 4; m++) {
                if (n != m && rätt[n] == gissning[m]) {
                    cows++;

                    // Detta break är inte nödvändigt för att programmet ska
                    // fungera, men om vi har hittat en matchande siffra är det
                    // ingen ide att leta efter flera eftersom det inte får
                    // förekomma dubletter, så vi avbryter den inre loopen i
                    // för tid.
                    break;
                }
            }
        }

        System.out.println(bulls+" Bulls");
        System.out.println(cows+" Cows");
    }
}

```

```

private static Random rnd = new Random();
private static Scanner keyboard = new Scanner(System.in);

private static boolean innehållerDubletter(int[] arr) {
    for (int n = 0; n < 4; n++) {
        for (int m = n + 1; m < 4; m++) {
            if (arr[n] == arr[m]) {
                return true;
            }
        }
    }
    return false;
}

private static int[] slumpaNod() {
    int[] rätt = new int[4];

    do {
        for (int n = 0; n < 4; n++) {
            rätt[n] = rnd.nextInt(10);
        }
    } while (innehållerDubletter(rätt));

    return rätt;
}

private static int[] läsInGissning() {
    int[] gissning = new int[4];

    do {
        for (int n = 0; n < 4; n++) {
            System.out.print("Skriv tal " + (n + 1) + " i din gissning: ");
            gissning[n] = keyboard.nextInt();
        }

        if (innehållerDubletter(gissning)) {
            System.out
                .println("Din gissning innehåller dubletter, försök igen");
        }
    } while (innehållerDubletter(gissning));
    return gissning;
}

public static void main(String[] args) {
    int[] rätt = slumpaNod();

    while (true) {
        int[] gissning = läsInGissning();
        kontrollera(rätt, gissning);
    }
}
}

```

UPPGIFT 5: KVITTO (6 POÄNG)

Här nedanför finns skelettet till två klasser som representerar ett kvitto och en vara på kvittot. Din uppgift är nu att skriva klart klasserna så att det går att lägga till varor på kvittot, och alla de i koden givna metoderna fungerar. Du får gärna lägga till ytterligare konstruktorer, metoder, variabler, etc. som hjälper dig att lösa uppgiften.

```
public class Kvitto{
    public void läggTill(Vara v){
    }
    public double totaltBelopp(){
    }
    public String toString(){
    }
}

public class Vara{
    public Vara(String namn,
                double belopp)
    {
    }
    public String namn(){
    }
    public double belopp(){
    }
}
```

De enda klasser du ska skriva någon kod för är de två ovanstående. Nedanstående exempel visar bara hur klasserna skulle kunna användas.

```
Kvitto k = new Kvitto();
k.läggTill(new Vara("Juice", 25.50));
k.läggTill(new Vara("Godis", 12.75));

System.out.println("Totalbelopp " + k.totaltBelopp());
System.out.println(); // ger en tom rad

k.läggTill(new Vara("Lott", 10));
System.out.println(k); // skriver hela kvittot
```

Ovanstående kod ger en utskrift i stil med nedanstående. Du behöver inte ha exakt detta utseende. Det viktiga är att alla varorna kommer med, med namn och pris, och att totalbeloppet finns med.

```
Totalbelopp 38.25 <- System.out.println("Totalbelopp " + k.totaltBelopp());

Juice    25.5
Godis    12.75
Lott     10.0
-----
Totalt:  48.25 } <- System.out.println(k);
```

LÖSNINGSFÖRSLAG FÖR KLASSEN KVITTO

Det vanligaste misstaget i Kvitto-klassen är att låta toString()-metoden skriva ut med System.out.println(). Detta ska en toString()-metod **ALDRIG** göra. Metodens uppgift är att göra om objektet till en strängrepresentation som sedan kan användas till olika saker, till exempel, att skrivas ut på skärmen. Vad som ska göras med den är dock upp till den som använder metoden att bestämma.

```
import java.util.ArrayList;

class Kvitto {

    private ArrayList<Vara> varor = new ArrayList<Vara>();

    public void läggTill(Vara v) {
        varor.add(v);
    }

    public double totaltBelopp() {
        double summa = 0.0;
        for (Vara v : varor) {
            summa += v.belopp();
        }
        return summa;
    }

    public String toString() {
        String kvittotext = "";
        for (Vara v : varor) {
            kvittotext += v + "\n";
        }
        kvittotext += "-----\n";
        kvittotext += "Totalt:\t" + totaltBelopp();
        return kvittotext;
    }
}
```

LÖSNINGSFÖRSLAG FÖR KLASSEN VARA

```
class Vara {  
  
    private String namn;  
    private double belopp;  
  
    public Vara(String namn, double belopp) {  
        this.namn = namn;  
        this.belopp = belopp;  
    }  
  
    public String namn() {  
        return namn;  
    }  
  
    public double belopp() {  
        return belopp;  
    }  
  
    public String toString() {  
        return namn + "\t" + belopp;  
    }  
}
```