

PROGRAMMERING-Java

TENTAMINA

Nicolina Månsson
2010-03-17

Tentamensinstruktioner

Poängsättning

Hela tentamen omfattar 42 poäng. Poäng för varje uppgift står angivet inom parentes före varje uppgift.

- För att få betyget 3, 4 eller 5 krävs: 20,28,36 poäng.

Redovisning

I varje uppgift står det vad som skall göras för att få angivet antal poäng.
Redovisa koden i den ordning den efterfrågats i texten med så mycket implementerat som möjligt.
Tänk på att vara noggrann och strukturerad. Det är Du som skall visa vad Du kan!

Hjälpmedel

Finns hos tentavakten.

Lycka till!

Uppgift 1 - Enkla program med variabler, metoder, val och repetition satser (4p+4p+4p+4p+4p)

a) När man skall försäkras en bil är det vanligt att man väljer s.k. helförsäkring eller halvförsäkring beroende på bilens ålder.
Skriv ett program som föreslår vilken typ av försäkring du skall välja. Programmet skall som indata få aktuellt år samt bilens årsmodell.
Utdata skall vara "Välj helförsäkring" om bilen är yngre än 8 år, "Välj halvförsäkring" om bilen är mellan 8 och 15 år gammal, samt "Köp bättre bil" om bilen är äldre än 15 år.

b) Skriv ett program som från användaren läser in ett namn (med förnamn och efternamn). Därefter skall programmet skapa en ny sträng bestående av första bokstaven i förnamnet följt av de första 4 i efternamnet samt ett slumpgenererat tal mellan 10 och 99. Den typen av algoritm kan system använda för att automatisk generera ett tillfälligt användarnamn för ett datorsystem.

T.ex. För Nicolina Månsson skall stängen bli NMåns45.

Några av följande metoder från klassen String kan vara till hjälp.

int	length() Returns the length of this string.
int	indexOf(String str) Returns the index within this string of the first occurrence of the specified substring.
String	substring(int beginIndex, int endIndex) Returns a new string that is a substring of this string.
char	charAt(int index) Returns the char value at the specified index.

c) Skriv ett program som beräknar och skriver ut tiden som krävs att resa mellan två städer. Programmet ska läsa in avståndet i km och hastigheten i km/h från användaren. Programmet skall beräkna och skriva ut tiden som krävs för resan.

Utskriften skall vara i format som vi vanligtvis anger tid, t ex 1h 10 min 3sec .

Ledning! Tiden beräknas med formeln $\text{tid} = \text{avstånd} / \text{hastighet}$. Värdet du får är tiden i decimal form vilket inte är tillräckligt bra. För att få full poäng skall du konvertera tiden till läsbart format.

(Omvandla allt till sekunder (multiplicera med 3600). Genom repeterade divisionen med 60 kommer du att räkna ut antalet timmar, minuter och sekunder. Använd % (moduls) operator som effektivt beräknar resten)

d) Skriv en metod `areaRektangel()` som argument tar längden och bredden av en rektangel och returnerar area.

Area räknas med formeln: $A = \text{längd} * \text{bredd}$

e) Uppskattningsvis så blir en m^3 snö 100 liter vatten. Skriv ett program som från användaren läser in längd och bredd (för en trädgård) samt höjden på snölagret. Därefter skall programmet skriva ut hur många liter vatten det blir kommer att bli när snön smälter.

Ledning! Volym beräknas med formel: $\text{längd} * \text{bredd} * \text{höjd}$. Du använda det redan implementerade `areaRektangel()` metod som du multiplicerar med höjden. Detta ger dig volymen.

f) Förklara vad följande metod gör.

```
public class TentaMetoder
{
    public static String förvirra( String text)
    {
        String nyText="";
        for(int i=text.length()-1; i>=0;i--)
            nyText=nyText+text.charAt(i);

        return nyText;
    }
}
```

Uppgift 2 – Egna klasser (6p+6p)

a) Programmet TestSquare fungerar inte riktigt som det är tänkt: Varför?
Gör nödvändiga ändringar (minst 3) i klassen Square och/eller TestSquare så att programmet fungerar. Motivera dina ändringar

```
public class Square
{
    private double side;

    public Square(int s)
    {
        side=s;
    }

    public void getArea()
    {
        return side*side;
    }

    public String toString()
    {
        System.out.println("Square:" + side);
    }

}

public class TestSquare
{
    public static void main(String [] args)
    {
        Square sq=new Square(3.5);
        System.out.println(sq.getArea());
        System.out.println(sq.toString());
    }
}
```

b) I en skyttetävling skjuter ett antal tävlande var och en ett visst antal skott på en 10-ringad tavla. Varje enskilt skott kan alltså ha ett resultat från 0 till 10. Följande klass beskriver en tävlande i en skyttetävling. Läs och förstå vad varje metod gör.

```
public class Tävlande{

    private String namn;
    private int antal_skott;
    private int summa_poäng;

    public Tävlande( String inamn)
    {
        namn=inamn;
        antal_skott=0;
        summa_poäng=0;
    }
    public void skjut( int poäng)
    {
        antal_skott++;
        summa_poäng=summa_poäng+poäng;
    }

    public int getSummaPoäng()
    {
        return summa_poäng;
    }
    public int getAntalSkott()
    {
        return antal_skott;
    }
    public String getNamn()
    {
        return namn;
    }

}
```

Skriv **ett program** som använder klassen `Tävlande` och som gör följande:

1. Skapar två `Tävlande`-objekt kalla dem `tv11` och `tv12`.
2. Båda tävlande ska skjuta två skott. Antalet poäng som den tävlande får ska slumpas. Max antal poäng som man kan få för ett skott är 10.
3. Programmet ska skriva ut namnet på den tävlande som har fått flest antal poäng. Om de har samma antal poäng ska de tävlande skjuta en gång till. Programmet skall avslutas med att skriva ut namnet på vinaren eller båda namnen om de `Tävlande` fortfarande har samma antal poäng.

	<code>Random ()</code> Creates a new random number generator.
<code>int</code>	<code>nextInt (int n)</code> Returns a pseudorandom, uniformly distributed <code>int</code> value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.

Uppgift 3 - Arrayer (6p)

I nedanstående program finns det deklarerad och skapad en tom array kallad `lotto_rad`. Du skall komplettera programmet och simulera ett lottospel på följande sätt.

- 1) Men hjälp av en loop låter användaren mata in 7 lotto nummer.
- 2) Med hjälp av en annan loop slumpa 20 olika nummer i intervallet 1-35. För varje slumpat nummer skall programmet söka i arrayen (`lotto_rad`) efter respektive värde. Om den hittas skall variabel `antal_rätt` uppdateras.
- 3) Programmet skall i slutet skriva ut antalet rätt spelaren hade.

```
import java.util.*;
public class LottoGame
{

    public static void main(String [] ar)
    {

        System.out.println (" SpelaLotto Simulator");

        Random generator=new Random();
        Scanner scan=new Scanner(System.in);
        int antal_rätt=0;

        int [] lotto_rad=new int[7];

        // din kod

    }

}
```