Instuderingsuppgifter läsvecka 6

1.

Om vi i ett program vill deklarera en lista av strängar bör vi skriva

```
List<String> theList = new ArrayList<String>();
```

och inte

```
ArrayList<String> theList = new ArrayList<String>();
```

Förklara varför!

2.

Om vi i ett program behöver lagra strängar i en mängd stringSet kan vi deklarera stringSet på följande vis:

i) Set stringSet;

ii) Set<String> stringSet;

Vilken av deklarationerna är att föredra och varför?

3.

Antag att vi har en klass

```
public class Point {
    . . .
}
```

för att avbilda punkter. Om man vill sortera en lista med objekt av Point kan man t.ex. använda sig av metodanropet

```
Collections.sort(listOfPoints);
```

där listOfPoints har typen ArrayList<Point>. Vad måste läggas till i klassen Point för att sorteringen ska gå att utföra?

4.

Klasserna i Java Collection Framework stödjer generiska typer från och med version 1.5. Skriv om nedanstående kod så att generiska typer används:

```
TreeSet mySet = new TreeSet( );
mySet.add( new Double(1.0) );
. . .
mySet.add( new Double(0.5) );
Double d = (Double) mySet.first();
```

5.

Antag att vi har följande lista:

```
List<String> list = new LinkedList<String>();
list.add("abc123");
list.add("trl673");
...
```

Iterera genom alla elementen i listan och skriv ut dem genom att använda en itetator resp. en förenklad **for**-sats.

6.

Betrakta nedanstående program.

```
public class Cigarette {
   //omitted
}
```

```
import java.util.*;
public class CigarettePack {
   private ArrayList cigs;
   public CigarettePack() {
      cigs = new ArrayList();
   }
   public void addCigarette(Cigarette c) {
      cigs.add(c);
   }
   public Cigarette getCigarette() {
      Cigarette c;
      if (cigs.size() > 0)
         c = cigs.remove(0);
      else
         c = null;
      return c;
   }
   public String toString() {
      String result = "pack of [";
      for (int i=0; i < cigs.size(); i++) {
         Cigarette c = cigs.get(i);
         result += ((i==0) ? "" : ",") + c.toString();
      }
      return result + "]";
   }
}
```

Klassen CigarettePack går inte att kompilera. Felutskriften blir något i stil med:

```
CigarettePack.java:13: incompatible types . Found  java.lang.Object , required Cigarette
CigarettePack.java:21: incompatible types . Found java.lang.Object , required Cigarette
```

Förklara vad som är fel! Åtgärda felet!

7.

Klassen Collections innehåller följande metod

```
public static void sort(List<T> list)
```

där det gäller att

```
T extends Comparable<? super T>
```

Ange för var och en av deklarationerna nedan huruvida den deklarerade samlingen kan eller inte kan sorteras genom att använda metoden sort ovan. Motivera ditt svar! Klassen Dog har följande utseende:

```
public class Dog {
   . . .
}
```

i. List<String>

ii. Set<String>

iii. ArrayList<String>

iv. List<Dog>

8.

Antag att följande klasser är givna: Betrakta nedanstående klasser (utlämnad kod är betydelselös för uppgiften):

```
public class Point { … }
public class MutablePoint extends Point { … }
public class Point3D extends Point { … }
public class ColorPoint extends Point  { … }
public class ColorPoint3D extends ColorPoint { … }
```

Antag vidare att vi i ett program har gjort följande deklarationer:

```
Object o;
Point p;
Point3D p3d;
ColorPoint cp;
ColorPoint3D cp3d;
List<? extends Point> lep;
List<? extends ColorPoint> lecp;
List<? super ColorPoint> lscp;
```

Ange för var och en av nedanstående satser om satsen är korrekt eller ger kompileringsfel. Motivera!

a)	lep.add(p);	b)	lep.add(cp);
c)	lecp.add(cp3d);	d)	lscp.add(cp);
e)	lscp.add(**null**);	f)	o = lep.get(1);
g)	p = lep.get(1);	h)	cp = lscp.get(1);
i)	cp = lecp.get(1);	j)	cp3d = lecp.get(1);

9.

Nedan finns definitionen till klassen Person:

```
public class Person {
   private String name;
   private int birthYear;
   . . .
   public String getName() {
      return name;
   }
   public int getBirthYear() {
      return birthYear;
   }
   . . .
}
```

a) Skriva en metod

```
public static String[] bornThisYear1(Person[] people, int year)
```

som tar ett fält people av Person-objekt och ett årtal year, och returnerar ett fält som innehåller namnen på de personer i fältet people som är födda år year.

b) Skriva en metod

```
public static ArrayList<String> bornThisYear3(ArrayList<Person> people, int year)
```

som gör samma sak som metoden i deluppgift a) förutom att parametern person nu är en ArrayList istället för ett fält.

Gör två implementeringar av metoden, dels en som använder en *förenklad **for**-sats* vid genomlöpningen av listan people, dels en som använder en *iterator* vid genomlöpningen av listan people,

c) Skriva en metod

```
public static void removeNames(ArrayList<Person> people,  ArrayList<String>  names)
```

som tar en lista people och en lista names, och tar bort alla objekt i listan people vars namn finns med i listan names.

10.

Antag att du i ett program vill hålla reda på de låtar som ditt favoritband tänker spela på sin nästa konsert. Skall du spara låtarna i en lista eller i en mängd? Motiviera ditt svar! Om du behöver mer information för att fatta ditt beslut, ange vilken information du behöver.

11.

Vad är inre klasser? Hur kan (bör) dessa användas? Motivera och ge ett kort exempel.

12.

Betrakta följande klass:

```
public class Problem {
  private String str;

  private static class Inner {
    private void testMethod() {
      str = "Set from Inner";
    }//testMethod
  }//Inner
}//Problem
```

Klassen går inte att kompilera. Förklara varför, samt åtgärda felet.

13.

Para ihop var och en av följande fyra beskrivningar med en av de nedan angivna standardklasserna i Java.

a) Ström som används för att lagra data på ett skivminne.

b) Superklass för alla klasser som representerar utgående strömmar.

c) Superklass för flera av de klasser som utökar funktionaliteten hos en existerande ström.

d) Innehåller metoder för att skicka olika typer av primitiva datatyper, t.ex. heltal, flyttal.

Klasser: DataOutputStream, ObjectOutputStream, FileOutputStream, BufferedOutputStream, PrintStream, PrintWriter, FilterOutputStream, OutputStream.

14.

På en viss finmekanisk industri som tillverkar styraxlar mäter man hur mycket de tillverkade axlarna avviker från den specificerade diametern. Avvikelsen anges i millimeter och uppgifter lagras på en fil. För tillfället godkänns alla axlar som har en felmarginal på 0.0001mm. Din uppgift är att hjälpa kvalitetsavdelningen att skriva ett program som beräknar hur många procent fler av de tillverkade axlarna som inte skulle bli accepterade om felmarginalen minskades till 0.00001 mm.

a) Antag att mätvärdena ligger på en textfil med namnet data.txt och med ett mätvärde på varje rad.

b) Antag att mätvärdena ligger på en binärfil med namnet data.bin.

c) Varför skall du rekommendera kvalitetsavdelningen att lagra sina mätvärden på binärfiler?

15.

Betrakta nedanstående enkla klass för att avbilda personer:

```
public class  Person {
   private  String name;
   private boolean female;
   . . .
   public boolean isFemale() {
      return female;
   }
}
```

a) Vilka förändringar av klassen Person måste göras för att man skall kunna skriva ut objekt av klassen till en fil?

b) Skriv ett program som läser en fil som innehåller objekt av klassen Person och skapar två nya filer, en fil som innehåller alla kvinnor och en fil som innehåller alla män. Namnen på filen som skall läsas in och på de två filerna som skall skrivas anges som argument till programmet.