

Instuderingsuppgifter läsvecka 6

1.

Betrakta klassen `Product` nedan:

```
public class Product {  
    private int nrInStock = 0;  
    private int outwardConnectionCount;  
    public void addToStock(int quantity) {  
        nrInStock = nrInStock + quantity;  
    } // addToStock  
    public void removeFromStock(int quantity) {  
        if (quantity > nrInStock)  
            throw new IllegalArgumentException("To big quantity");  
        nrInStock = nrInStock - quantity;  
    } // addToStock  
    public int numberInStock() {  
        return nrInStock;  
    } // numberInStock  
    //fler attribut och metoder som är oväsentliga för uppgiften  
} // Product
```

Klassen `Product` är inte trådsäker (*thread-safe*).

- Förklara vad som menas med att en klass inte är trådsäker. Beskriv med ett exempel vad som kan inträffa i klassen `Product`.
- Gör klassen `Product` trådsäker. Din lösning skall så långt som möjligt bibehålla parallella bearbetning.

2.

När man skriver ett program i Java som består av flera samverkande trådar uppstår ofta ett behov av att synkronisera trådarnas exekvering. Till exempel kan en tråd behöva vänta på att en annan tråd ska bli färdig med någon beräkning eller så behöver man se till att inte två trådar försöker använda sig av samma resurs samtidigt. En ofta använd teknik i Java är att använda sig av så kallade monitorer. För att stödja användandet av monitorer har man i Java infört ett antal språkkonstruktioner samt ett antal standardmetoder i klassen `Object`.

- Beskriv vad följande standardmetoder har för innebörd (semantik) i Java:
 - `wait()`
 - `notify()`
 - `notifyAll()`
- För att man ska kunna anropa metoderna `wait()`, `notify()` eller `notifyAll()` (utan att få en exception), måste ett mycket specifikt villkor vara uppfyllt. Vilket?

3.

Kalle Klåpare har skrivit ett Javaprogram som använder två trådar, **thread1** och **thread2**. Nu vill han att **thread1** ska kunna skicka över en sträng till **thread2**. Han har därför skrivit följande klass för att hantera överföringen:

```
public class Buffer {  
    private String message;  
    public void set(String m) {  
        message = m;  
    } //set  
    public String get() {  
        String m = message;  
        message = null;  
        return m;  
    } //get  
} //Buffer
```

thread1 anropar då och då **set()** för att lägga in en sträng i bufferten. **thread2** anropar hela tiden **get()** och när något annat än **null** returneras utför **thread2** någon typ av operation på den mottagna strängen. Tyvärr fungerar Kalles program väldigt dåligt.

- a) Ge exempel på två problem Kalle kan råka ut för med lösningen ovan.
- b) Skriv om klassen **Buffer** ovan så att den blir både trådsäker och effektiv.

4.

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

5.

Betrakta nedanstående program:

```
public class Red {  
    public static void main(String[] args) throws InterruptedException {  
        Thread t1 = new Thread(new Runnable() {  
            public void run() {  
                System.out.print("O");  
                System.out.print("Y");  
            }  
        });  
        Thread t2 = new Thread(new Blue());  
        System.out.print("R");  
        t1.start();  
        System.out.print("G");  
        t2.start();  
        System.out.print("I");  
        t1.join();  
        System.out.print("V");  
        t2.join();  
        System.out.print("K");  
    }  
} //Red  
  
public class Blue implements Runnable {  
    public void run() {  
        System.out.print("B");  
    }  
} //Blue
```

Vilka av nedanstående utskriftssekvenser är möjliga/omöjliga att erhålla från programmet?

- a) ROYGBIVK
- b) ROYBGIVK
- c) RGOYIBVK
- d) OYBRGIVK

6.

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.

7.

Denna uppgift handlar om djup kopiering av objekt.

- a) Antag att du av någon anledning vill förbjuda att objekt av en viss klass kopieras med `clone`. Vad behöver du addera till klassen? Visa hur det skall se ut!
- b) Överskugga metoden `clone()` i klassen `C` nedan:

```
public class B implements Cloneable {  
    public B clone() { ... }  
}  
  
public class C {  
    private int x;  
    private B b;  
    ...  
}
```

- c) Överskugga metoden `clone()` i klassen `D` nedan:

```
import java.util.ArrayList;  
public class D {  
    private int x;  
    private ArrayList<C> cs;  
}
```

Använd tekniken med *kopieringskonstruktor*, vilken givetvis då också skall implementeras, samt utnyttjar möjligheten till *kovariant returtyp*. Tips: Standard klassen `ArrayList` överskuggar `clone()`.

8.

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`.

9.

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?

10.

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.