

Laboration 3, Strömmar

Avsikten med laborationen är att du ska träna på att använda strömmar och då lagra och hämta data på en hårddisk.

På kurssidan hittar du följande filer: `Person.java`, `WriteToFile.java`, `heltal.dat`, `medlemmar.txt`.

Placera java-filerna i paketet ***laboration3*** och de övriga filerna i katalogen ***files*** vilken ska vara i projektet.

Uppgift 1 – Hur stor är summan?

Lägg filen **`heltal.dat`** i katalogen *files* och skriv sedan ett program som beräknar summan av heltalen i filen. Filen är formaterad så här:

int 1: Antal int som följer (nedan n st)

int 2, int 3, int 4, ..., int n

Exempel: 7 19 2 -13 5 3 5 8 7:an anger att det följer 7 st int (19, 2, ..., 8)

Lämpliga klasser att använda i lösningen är **`DataInputStream`**, **`FileInputStream`** och eventuellt **`BufferedInputStream`**.

Uppgift 2 – Hur många är medlemmarna? Hur många är kvinnor?

Lägg filen **`medlemmar.txt`** i katalogen *files*. Filen består av ett antal rader med text. Varje rad representerar en medlem och består av ett namn och avslutas med K (kvinna) eller M (man):

”Anders M”

”Siv K”

Skriv ett program som skriver ut samtliga medlemmars namn, antalet medlemmar och antalet kvinnliga medlemmar.

Aktuella klasser att använda är **`BufferedReader`**, **`InputStreamReader`** och **`FileInputStream`**. Teckenkodningen är "ISO-8859-1".

Uppgift 3 – Skriva till hårddisk

Komplettera nedanstående metoder i klassen **WriteToFile** som du hittar på kurssidan.

- **public void writePrimes(String filename) throws IOException**
writePrimes ska skriva primtalen i arrayen *primes* till filen *files/primes.dat*. Först ska antalet primtal skrivas som en *int* och sedan följer primtalen:
int, long, long, long, ...

När du testkör programmet *WriteToFile* och väljer ”Skriv primtal” så anropas metoden *writePrimes*. Sedan kan du kontrollera innehållet i filen du skapat genom att välja ”Läs primtal”.

- **public void writeInvited(String filename) throws IOException**
writeInvited ska skriva strängarna i *invited* till filen *files//invited.txt*. Teckenkodningen i filen ska vara ”UTF-8”. Strängarna ska skrivas som strängar med **write**-metoden i **BufferedWriter**. Glöm inte att anropa **newLine** efter varje sträng.

Kontrollera ditt resultat genom att välja ”Läs bjudna”.

- **public void writePersons(String filename) throws IOException**
writePersons ska skriva **Person**-objekten i *persons* till filen *files/persons.dat*. Först ska antalet objekt på filen skrivas som en *int* och sedan ska objekten skrivas:
int, Person, Person, Person, ...
Lämpliga klasser att använda i lösningen är **ObjectOutputStream** och **FileOutputStream**.

Kontrollera ditt resultat genom att välja ”Läs personer”.

Uppgift 4 – Hur gamla är medlemmarna?

Läs objekten i filen *personer.dat* (som du skrev i uppgift 3) och beräkna *Person*-objektens genomsnittliga ålder. Beräkna dessutom kvinnornas genomsnittliga ålder.

Lämpliga klasser att använda i lösningen är **ObjectInputStream** och **FileInputStream**. Tänk på att undantaget **ClassNotFoundException** måste fångas när man läser ett objekt via en objektström.

Uppgift 5 – Skriva till hårddisk, extra

Skriv metoderna **writePrimes2**, **readPrimes2**, **writeInvited2**, **readInvited2**, **writePersons2** och **readPersons2** i *WriteToFile.java* så att objektsamlingarna *primes*, *invited* respektive *persons* skrivs/läses som objekt. Det innebär t.ex. att hela arrayen *primes* skrivs som ett objekt i *writePrimes2* och hela arrayen läses som ett objekt i *readPrimes2*.

Uppgift 1

172907

```
public class Exercisel {
    public void sum(String filename) throws IOException {
        try (DataInputStream dis = new DataInputStream(new
FileInputStream(filename))) {
            int sum=0;
            int n = dis.readInt();
            for(int i=0; i<n; i++) {
                System.out.println(sum);
            }
            System.out.println(sum);
        }
    }

    public static void main(String[] args) throws IOException {
        Exercisel ex1 = new Exercisel();
        ex1.sum("files/heltal.dat");
    }
}
```

Uppgift 2

```
public class Exercise2 {
    public void statistics(String filename) throws IOException {
        try (BufferedReader br = new BufferedReader( new InputStreamReader(
            new FileInputStream(filename),"ISO-8859-1"))) {
            int count=0, women=0;
            String name = br.readLine();
            while(name!=null) {
                count++;
                System.out.print(name);
                if(name.endsWith("K"))
                    women++;
                name = br.readLine();
                if(name!=null)
                    System.out.print(", ");
            }
            System.out.print("\nAntal medlemmar: "+count+" , därav "+
                women+" kvinnor.\n");
        }
    }

    public static void main(String[] args) throws IOException{
        Exercise2 ex2 = new Exercise2();
        ex2.statistics("files/medlemmar.txt");
    }
}
```

Uppgift 3

```
public void writePrimes(String filename) throws IOException {
    try (DataOutputStream dos = new DataOutputStream(
        new BufferedOutputStream(new FileOutputStream(filename)))) {
        dos.writeInt(primes.length);
        for (int i = 0; i < primes.length; i++) {
            dos.writeLong(primes[i]);
        }
        dos.flush();
    }
}

public void writeInvited(String filename) throws IOException {
    try (BufferedWriter bos = new BufferedWriter(new OutputStreamWriter(
        new FileOutputStream(filename), "UTF-8"))) {
        for (int i = 0; i < invited.size(); i++) {
            bos.write(invited.get(i));
            bos.newLine();
        }
        bos.flush();
    }
}

public void writePersons(String filename) throws IOException {
    try (ObjectOutputStream oos = new ObjectOutputStream(
        new BufferedOutputStream(new FileOutputStream(filename)))) {
        oos.writeInt(persons.size());
        for (Person p : persons) {
            oos.writeObject(p);
        }
        oos.flush();
    }
}
```

Uppgift 4

```
public class Exercise4 {
    public void statistics(String filename) throws IOException {
        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(filename))) {
            int n, women=0, sum=0, sumWomen=0;
            n = ois.readInt();
            Person[] p = new Person[n];
            for(int i=0; i<n; i++) {
                try {
                    p[i] = (Person) ois.readObject();
                    sum += p[i].getAge();
                    if(p[i].getSex()=='K') {
                        sumWomen += p[i].getAge();
                        women++;
                    }
                } catch(ClassNotFoundException e) {
                    System.out.println(e);
                }
            }
            System.out.println("Genomsnittlig ålder är "+(double)sum/n);
            System.out.println("Kvinnornas genomsnittlig ålder är "+
                (double)sumWomen/women);
        }
    }
}
```

```
    public static void main(String[] args) throws IOException {
        Exercise4 ex4 = new Exercise4();
        ex4.statistics("files/personer.dat");
    }
}
```

Uppgift 5

```
public void writePrimes2(String filename) throws IOException {
    try (ObjectOutputStream dos = new ObjectOutputStream(
        new FileOutputStream(filename))) {
        dos.writeObject(primes);
        dos.flush();
    }
}

public void readPrimes2(String filename) throws IOException {
    long[] primes=null;
    try (ObjectInputStream ois = new ObjectInputStream(
        new FileInputStream(filename))) {
        primes = (long[])ois.readObject();
    }catch(ClassNotFoundException e) {}
}

public void writeInvited2(String filename) throws IOException {
    try (ObjectOutputStream oos = new ObjectOutputStream(
        new FileOutputStream(filename))) {
        oos.writeObject(invited);
        oos.flush();
    }
}

public void readInvited2(String filename) throws IOException {
    try (ObjectInputStream ois = new ObjectInputStream(
        new FileInputStream(filename))) {
        invited = (LinkedList<String>) ois.readObject();
    }catch(ClassNotFoundException e) {}
}

public void writePersons2(String filename) throws IOException {
    try (ObjectOutputStream oos = new ObjectOutputStream(
        new FileOutputStream(filename))) {
        oos.writeObject(persons);
        oos.flush();
    }
}

public void readPersons2(String filename) throws IOException {
    try (ObjectInputStream ois = new ObjectInputStream(
        new FileInputStream(filename))) {
        persons = (ArrayList<Person>) ois.readObject();
    }catch(ClassNotFoundException e) {}
}
}
```