Lucrare de laborator nr. 11
Fişiere (I)

1. Scopul lucrării

În această lucrare se studiază clasele *FileReader* şi *BufferedReader* folosite pentru a citi date date din fişiere text. *FileReader* este folosită pentru a citi un fișier text caracter cu caracter, iar *BufferedReader* pentru a citi linie cu linie.

2. Breviar teroretic

2.1 Generalitati.

Fişierele sunt de două tipuri:

- **fişiere text**
- **fişiere binare**

Diferenţa principală dintre ele este dată de modul de codificare a informaţiei.

Exemplu:

Fie un fișier text ce conține ca informație valoarea numerică 259 .

Informaţia din fişierul text este codificată binar pe baza codurilor Unicode ale celor 3 caractere: ‘2’, ‘5’ si ‘9’.

Codul caracterului ‘2’ este $50_{10}=2^5+2^4+2^1$=0 0 1 1.0 0 1 0

Deci, in binar: 00110010 .

Asemănător sunt codate și celelalte două caractere: ‘5’ si ‘9’.

Fie un fişier binar de numere întregi, ce conține aceeați informaţie: 259.

Informatia va fi reprezentată ca număr întreg pe 4 octeţi, corespunzător reprezentării în baza 2 a numărului intreg 259:

$259_{10}=2^8+2^1+2^0$

Deci in baza 2:

00000000 00000000 00000001 00000011

O altă diferenţă între fişierele text şi fişierele binare este aceea ca fişierele text au înregistrări (linii) de lungime variabilă, iar fişierele

binare au înregistrări de lungime fixă. Astfel, intr-un fişier binar de numere întregi, fiecare înregistrare ocupă 4 octeţi.

Putem totuşi interpreta un fişier text ca un fişier binar, ce are ca înregistrare de bază octetul.

În Java, clasele care prelucrează fişiere se găsesc în pachetul **java.io**.

În cazul tuturor prelucrărilor de fişiere, se pot genera excepţii care sunt obligatoriu de tratat. Cea mai des întâlnită excepţie face parte din clasa **IOException**, excepţie obligatoriu de tratat.

În Java, clasele ce prelucrează fişiere sunt specializate, atât pentru tipurile de fişiere (text sau binar), cât şi pentru operaţiile ce se fac asupra fişierelor.

2.2 Clase pentru citire din fişiere text.

Clasa FileReader.

Este o clasă folosită pentru a citi un fişier text caracter cu caracter.

Constructorul:

public FileReader(String numeFisier)

Metode:

a) **public int read( )** – citeşte şi returnează codul caracterului de pe poziţia curentă.
   Determină avansarea în mod automat pe următoarea poziţie a capului de citire.
   În caz de sfârşit de fişier returnează **-1**.
b) **public void close( )** – este folosită pentru a închide fişierul.

Clasa BufferedReader.

Este folosită pentru a citi un fişier text linie cu linie.

Constructorul:

public BufferedReader(FileReader fr)

Metode:

public String readLine( )

Returnează sub formă de String, linia curentă citită din fişier. În caz de sfârşit de fişier, returnează **null**.

3. Probleme rezolvate

Problema 1

Se citeşte de la tastatură numele unui fişier text. Să afişăm dacă este prezent caracterul **a** în fişier.

```
import java.io.*;
import javax.swing.*;
class F1
{
 public static void main(String args[ ])
 {
  String numeF=JOptionPane.showInputDialog("nume fişier=");
  FileReader fr=null;
  boolean este_a=false;
  try{
   fr=new FileReader(numeF);
   //citesc fişierul caracter cu caracter:
   for(;;) {
    int cod=fr.read( );
    if (cod==-1) break;
    if ((char)cod=='a') { este_a=true;
      break; }
   }
   fr.close( );
  }catch(IOException e){
     System.out.println(e);
     System.exit(1); }
  if (este_a== true) System.out.println("Este prezent a");
  else System.out.println("Nu este prezent a");
 }
}
```

Problema 2

Să se afişeze dacă primul caracter dintr-un fişier text este egal cu ultimul.

```
import java.io.*;
class PrimulSiUltimulCaracter
{
 public static void main(String args[])
 {
  FileReader fr=null;
  //Presupunem ca fisierul are cel putin doua caractere!
  try{
```

```
   fr=new FileReader("date.txt");
   //citim primul caracter:
   int cod=fr.read();
   char primul=(char)cod;
   //citim urmatorul caracter si cu acesta initializam  pe ultimul:
   cod=fr.read();
   char ultimul=(char)cod;
    //citim restul caracterelor, actualizand ultimul caracter:
    for(;;){
      cod=fr.read();
      if(cod==-1)break;//s-a terminat fisier
      ultimul=(char)cod;
    }
    fr.close();
    if(primul==ultimul)System.out.println("da");
    else System.out.println("nu");
  }catch(IOException e){
  System.out.println(e);
  System.exit(1);}
 }//main
}
```

Problema 3

Pentru fişierul text test.txt să se afişeze numărul de apariţii pentru fiecare caracter din fişier. Afişarea se va face în ordine crescătoare a numărului de apariţii. Se consideră că în fişier codurile ASCII ale caracterelor fac parte din intervalul [0, 127] (deci sunt maxim 128 de caractere diferite).

```
import java.io.*;
class NrAparitii
{
public static void main(String args[])
{
 final int NR_CAR=128;
 int contor[]=new int[NR_CAR];
 int i;
 for(i=0;i<NR_CAR;i++)contor[i]=0;
 FileReader fr=null;
 i=0;//indexul caracterului curent citit din fisier
```

```
//citim fisierul caracter cu caracter:
try{
 fr=new FileReader("test.txt");
 for(;;){
   int cod=fr.read();//codul ASCII al caracterului citit
   if(cod==-1)break;//s-a terminat fisierul
   //numaram caracterul citit:
   contor[cod]++;
 }
}catch(IOException e){
  System.out.println(e);
  System.exit(1);
}
//sortare a 2 vectori paraleli:
char vectorCaractere[]=new char[NR_CAR];
for(i=0;i<NR_CAR;i++)
 vectorCaractere[i]=(char)i;
int j;
for(i=0;i<NR_CAR;i++)
  for(j=i+1;j<NR_CAR;j++)
    if(contor[i]>contor[j]){
     int aux=contor[i];
     contor[i]=contor[j];
     contor[j]=aux;
     char temp=vectorCaractere[i];
     vectorCaractere[i]=vectorCaractere[j];
     vectorCaractere[j]=temp;
    }
//afisare, doar cele ce apar cel putin o data:
for(i=0;i<NR_CAR;i++)
 if(contor[i]!=0)
    System.out.println(vectorCaractere[i]+" : "+contor[i] ) ;
 }
}
```

Problema 4

Pentru fişierul text *scrisoare.txt*, să se calculeze şi afişeze câte linii sunt în fişier şi câte linii încep cu caracterul **A** .

```
import java.io.*;
```

```
class CateLinii
{
 public static void main(String args[ ])
 {
  FileReader fr=null;
  BufferedReader bfr=null;
  int contorLinii=0;
  int contorLinii_A=0;
  try{
   fr=new FileReader("scrisoare.txt");
   bfr=new BufferedReader(fr);
   for(;;) {
     String s=bfr.readLine( );
     if (s==null) break;
     //am citit o linie:
     contorLinii++;
     //Incepe cu A ?
     if(s.charAt(0)=='A')contorLinii_A++;
   }//for;;
   bfr.close( );
   fr.close( );
  }catch(IOException e){
     System.out.println(e);
     System.exit(1); }
  System.out.println("numar linii ="+contorLinii);
  System.out.println("numar linii ce incep cu A ="+contorLinii_A);
 }//main
}//class
```

Problema 5

Să se calculeze şi afişeze care este cea mai lungă linie din fişierul text “scrisoare.txt”.

Algoritmul:

Iniţializăm maximul (lungimea celei mai lungi linii din fişier) cu 0.
Citim pe rând fiecare linie şi comparăm lungimea ei cu maximul.
Dacă este mai mare, schimbăm maximul.

```
import java.io.*;
class LiniaCeaMaiLunga
{
```

```
 public static void main(String args[ ])
 {
  FileReader fr=null;
  BufferedReader bfr=null;
  int lMax=0;
  String linieMax="";//stringul vid
  try{
   fr=new FileReader("scrisoare.txt");
   bfr=new BufferedReader(fr);
   for(;;) {
     String s=bfr.readLine( );
     if (s==null) break;
     int l=s.length( );
     if (l>lMax) { lMax=l;
          linieMax=s; }
   }//for;;
   bfr.close( );
   fr.close( );
  }catch(IOException e){
        System.out.println(e);
        System.exit(1); }
  System.out.println(linieMax);
 }//main
}//class
```

Problema 6

Se citeşte numele unui fişier text ce conţine mai multe linii. Să se afişeze dacă toate liniile sunt diferite între ele sau nu.

```
import java.io.*;
import javax.swing.*;
class LiniiDiferite
{
 public static void main(String args[])
 {
  String numeF=JOptionPane.showInputDialog("nume fisier=");
  FileReader fr=null;
  BufferedReader bfr=null;
  //Vom copia toate liniile din fisier, intr-un vector de Stringuri:
  String s[]=new String[1000];//dimensiune acoperitoare
```

```
 //initializare pt. numarul de linii din fisier:
 int N=0;
 try{
   fr=new FileReader(numeF);
   bfr=new BufferedReader(fr);
   for(;;){
    String linie=bfr.readLine();
    if(linie==null)break;//s-a terminat fisierul
    //copiem linia in vectorul s[]:
    s[N]=linie;
    N++;
   }
  bfr.close();
 fr.close();
 }catch(IOException e){
    System.out.println(e);
    System.exit(1);}
//Verificam daca vectorul s[] are toate elementele diferite:
for(int i=0;i<N-1;i++)
  for(int j=i+1;j<N;j++)
     if(s[i].compareTo(s[j])==0){
       System.out.println("Nu are toate liniile diferite !");
       return;}
System.out.println("Are toate liniile diferite !");
 }//main
}
```

Problema 7

Pentru fişierul text *scrisoare.txt*, să se calculeze şi afişeze de câte ori apare cuvântul *este* în acest fişier.

```
import java.io.*;
import java.util.*;
class ContorizareCuvant
{
 public static void main(String args[])
 {
  FileReader fr=null;
  BufferedReader bfr=null;
  //initializari:
```

```
 int contor=0;
  try{
    fr=new FileReader("scrisoare.txt");
    bfr=new BufferedReader(fr);
    for(;;){
      String linie=bfr.readLine();
      if(linie==null)break;//s-a terminat fisierul
      //Extragem cuvintele din aceasta linie:
      StringTokenizer tk=new StringTokenizer(linie);
      int n=tk.countTokens();
      for(int i=0;i<n;i++){
        String cuvant=tk.nextToken();
        if(cuvant.compareTo("este")==0)contor++;
         }
     }

     bfr.close();
    fr.close();
   }catch(IOException e){
    System.out.println(e);
    System.exit(1);}
  System.out.println("numar aparitiii: "+contor);
 }//main
}
```

Problema 8

Să se afişeze care este cel mai lung cuvânt dintr-un fişier text dat. Se consideră că în fişier nu se află cuvinte despărţite în silabe, la cap de rând.

```
import java.io.*;
import java.util.*;
class CelMaiLungCuvant
{
 public static void main(String args[])
 {
  FileReader fr=null;
  BufferedReader bfr=null;
  //initializari:
```

```
 String cuvantMax="";
 int lungimeMax=0;//initializare pt. dimensiunea celui mai lung
                  //cuvant
  try{
    fr=new FileReader("referat.txt");
    bfr=new BufferedReader(fr);
    for(;;){
      String linie=bfr.readLine();
      if(linie==null)break;//s-a terminat fisierul
      //Extragem cuvintele din aceasta linie:
      StringTokenizer tk=new StringTokenizer(linie);
      int n=tk.countTokens();
      for(int i=0;i<n;i++){
        String cuvant=tk.nextToken();
        int lungime=cuvant.length();
        if(lungime>lungimeMax){
            lungimeMax=lungime;
            cuvantMax=cuvant;
         }
        }
      }
     bfr.close();
    fr.close();
   }catch(IOException e){
    System.out.println(e);
    System.exit(1);}
 System.out.println("Cel mai lung cuvant este: "+cuvantMax);
 }//main
}
```

Problema 9

Se dă fişierul text date.txt, care conţine numere întregi (în fiecare linie sunt unul sau mai multe numere întregi separate prin unul sau mai multe spaţii). Să se calculeze şi afişeze care este cel mai mare număr din fişier.

```
import java.io.*;
import java.util.*;
class NumarMaxim
{
```

```
public static void main(String args[])
{
 FileReader fr=null;
 BufferedReader bfr=null;
 try{
   fr=new FileReader("date.txt");
   bfr=new BufferedReader(fr);
   //initializam maximul cu primul numar citit din fisier, de  aceea,
   // prima linie o citim separat (presupunem ca fisierul are cel putin
   // o linie):
   String linie=bfr.readLine();
   //Extragem numerele din din aceasta linie:
   StringTokenizer tk=new StringTokenizer(linie);
   int n=tk.countTokens();
   //initializez maximul cu primul numar:
   int max=Integer.parseInt(tk.nextToken());
   //extragem restul numerelor din prima linie:
   int i,nr;
   for(i=1;i<n;i++){
      nr=Integer.parseInt(tk.nextToken());
      if(nr>max)max=nr;
   }
   //citim restul liniilor din fisier:
   for(;;){
     linie=bfr.readLine();
     if(linie==null)break;//s-a terminat fisierul
     //Extragem numerele din din aceasta linie:
     tk=new StringTokenizer(linie);
     n=tk.countTokens();
     for(i=0;i<n;i++){
       nr=Integer.parseInt(tk.nextToken());
       if(nr>max)max=nr;
     }
    }
    bfr.close();
   fr.close();
   System.out.println("Maxim="+max);
  }catch(IOException e){
   System.out.println(e);
   System.exit(1);}
```

```
 }//main
}
```

Problema 10

Se dă fişierul text cuvinte.txt, care conţine câte un cuvânt pe fiecare linie. Să se copieze toate cuvintele într-o colecţie ArrayList, şi apoi să se afişeze prin parcurgerea ei, acestă colecţie.

```
import java.io.*;
import java.util.*;
class IncarcaInArrayList
{
 public static void main(String args[ ])
 {
 FileReader fr=null;
 BufferedReader bfr=null;
 ArrayList<String> al=new ArrayList<String>();
 try{
  fr=new FileReader("cuvinte.txt");
  bfr=new BufferedReader(fr);
  for(;;) {
    String s=bfr.readLine( );
    if (s==null) break;
    //am citit un cuvant, il adaugam in colectia al:
    al.add(s);
  }
  bfr.close( );
  fr.close( );
 }catch(IOException e){
    System.out.println(e);
    System.exit(1);}
 //Afisare colectie:
 Iterator it=al.iterator();
 while(it.hasNext())
    System.out.println(it.next());
 }//main
}//class
```

3. Probleme propuse

Problema 1

Se citeşte de la tastatură numele unui fişier text. Să se calculeze şi afişeze de câte ori apare caracterul A în fişier.

Problema 2

Se citeşte de la tastatură numele unui fişier text. Să se calculeze şi afişeze care vocală apare de cele mai multe ori în fişier.

Problema 3

Se citeşte un număr natural N. Se citesc numele a N fişiere text. Să se afişeze în care fişier apare de cele mai multe ori caracterul **A**.

Problema 4

Se dă fişierul text date.txt, care conţine numere întregi (în fiecare linie sunt unul sau mai multe numere întregi separate prin unul sau mai multe spaţii). Să se calculeze şi afişeze media aritmetică a numerelor din fişier.

Problema 5

Se dă fişierul text date.txt, care conţine numere întregi (în fiecare linie sunt unul sau mai multe numere întregi separate prin unul sau mai multe spaţii). Să se afişeze dacă toate numerele din fişier sunt diferite între ele sau nu.

Problema 6

Se dă fişierul text clasa.txt ce conţine pe fiecare linie numele complet al unui elev şi media lui anuală. Să se afişeze numele elevului care are cea mai mare medie.

Problema 7

Se citeşte de la tastatură numele unui fişier text. Să se calculeze şi afişeze care este linia din fişier care conține cele mai multe cuvinte.

Problema 8

Se dă fişierul text date.txt, care conţine numere întregi (în fiecare linie sunt unul sau mai multe numere naturale separate prin unul sau mai multe spaţii). Să se calculeze şi afişeze care este cel mai mare număr prim din fişier.