Final lab113 2019 Conference

```
public class InvalidFeeException extends Exception
{
public InvalidFeeException(){
super("InvalidFeeException");
}

public InvalidFeeException(String s){
super(s);
}
}
```

===============

```
import java.io.Serializable;

public class Conference implements Serializable
{
private String cName;
private int cID;
private double fee;
private String [] papersArr;
private int nbP;

public Conference( String name , int id, double fee,int size)throws
InvalidFeeException
{
cName=name;
cID=id;
papersArr=new String[size];
nbP = 0 ;

if(fee<0 || fee>9999)
throw new InvalidFeeException();
else
this.fee=fee;
}

public boolean addPaper(String title)
{
if(nbP >= papersArr.length)
return false;
else
papersArr[nbP++]=title;
return true;
}

public String toString(){
String str=" cName: "+cName+" cID: "+cID+" fee: "+fee+" nbP: "+nbP +
"\n" ;
for(int i=0;i<nbP;i++)
str = str + papersArr[i]  + "\n";
return str;
}


public boolean checkC(Conference c){
if(c !=null &&  cID == c.cID && cName .equals(c.cName))
return true;

return false;


}
public String getName(){
return cName;
}
public int getID(){
return cID;}
public double getfee(){
return fee;}
}
==================================================
public class Node {
   private Conference data;
   private Node next;


   public Node(Conference obj) {
      data = obj;
      next = null;
   }

   public void setNext(Node n) {
      next = n;
   }

   public Node getNext() {
      return next;
   }

   public void setData(Conference c) {
      data = c;
   }

   public Conference getData() {
      return data;
   }

   public String toString() {
      return data.toString();
   }
} // end class
====================

   // Don't add extra setters and getters

public class List {
```

```
// Attributes
private Node head;

public List() {
   head = null;
}

public boolean isEmpty() {
return head == null;
}

public int size() {
if (isEmpty())
return 0;


Node current = head;
int n = 0;
while (current != null) {
 n++;
current = current.getNext();
}
return n;
}

public void insertAtFront(Conference c) {
   Node newnode = new Node(c);
   newnode.setNext(head);
   head = newnode;
}

public void insertAtBack(Conference c) {
   Node newnode = new Node(c);
   if (isEmpty())
      head = newnode;
   else {
      Node current = head;
      while (current.getNext() != null)
         current = current.getNext();
      current.setNext(newnode);
   }
}

public Conference removeFromfront() {
   if (isEmpty())
      return null;
   Node First = head;
   head = head.getNext();
   return First.getData();
}

public Conference removeFromBack() {
   Node current = head;
   Node pre = null;
   if (isEmpty())
      return null;
   else
```

```
            while (current.getNext() != null) {
               pre = current;
               current = current.getNext();
            }
         Conference e = current.getData();
         if (current == head)
            head = null;
         else
            pre.setNext(null);
         return e;
      }

      //   write method cheapConferences   (  for student )

      public Conference[] cheapConference()
      {
          if( isEmpty() == true)
          {
              return null;
          }

         Conference[] arr=new Conference[size()];
         int j=0;

         Node current = head;
         while( current != null )
         {
         if(current.getData().getfee()<1000)
         arr[j++]=current.getData();
         current = current.getNext();
         }
         return arr; }

   } // end class

   ============================


   import java.util.*;
   import java.io.*;

   public class TestList {
   static Scanner input=new Scanner(System.in);
   public static void main(String[] args){
   //a
   List cList=new List();    // stack

   //b
   int i = 0;
   boolean enter =true;
   while(i < 2 )
   {
   try{
   System.out.println("enter name ,  id, size");
   String name=input.next();
   int id=input.nextInt();
   int size=input.nextInt();
```

```
enter = true;
while( enter ){
    try{
    System.out.println("enter fee,");
    double fee=input.nextDouble();
    Conference c1=new Conference(name,id,fee,size);
    cList.insertAtFront(c1);
    enter = false;
    }
    catch(InvalidFeeException e){
    System.out.println("wrong fee,you should Enter fee between 0 and
9999 ");
    }

}  // while ( enter )

i++;

}
catch(InputMismatchException e)
{
input.next();
System.out.println(e.toString());
}
}


//c
Conference con3 = null;
try{
con3 = new Conference("Artificial intelligence international
Conference",52114,3000,10);
con3.addPaper("Bioinfomatics");
con3.addPaper("Cognitive system");
cList.insertAtFront(con3);
}
catch(InvalidFeeException e){
System.out.println(e.toString());
}

//d,e
List tempList = new List();
int len=cList.size();

try{
File f=new File("Conferences.data");
FileOutputStream f1=new FileOutputStream(f);
ObjectOutputStream file=new ObjectOutputStream(f1);

for(int j=0;j<len;j++)  // or    while( ! clist.isEmpty() )
{
Conference tempObj=cList.removeFromfront();
if(tempObj.getID()==22112)
 tempObj.addPaper("Ethics in AI");

System.out.println( tempObj.toString());  // on screen
file.writeObject(tempObj);                // on file
```

```
tempList.insertAtFront(tempObj);
}

file.close();
}catch(IOException e)
{System.out.print(e.toString());}

// return all object to cList
for(int r=0;r<len;r++){       //  or  while( ! tempList.isEmpty() )
cList.insertAtFront(tempList.removeFromfront() );
}
//==================================================

//f
try{
Scanner read= new Scanner(new  File("newConf.txt"));

while(read.hasNext())
{
String name=read.next();
int id2=read.nextInt();
double fee2=read.nextDouble();
int size2=read.nextInt();
try{
Conference con =new Conference(name,id2,fee2,size2);
cList.insertAtFront(con);
}
catch(InvalidFeeException e)
{
    System.out.print("invalid");
}
}  // hasNext

read.close();
}
catch(IOException e)    // or FileNotFoundException
{System.out.print(e.toString());
}

//g
System.out.println("Conference have  fee less than 1000:");
Conference[] array=cList.cheapConference();

if( array == null )
        System.out.println("Empty list ");
else
for( i =0;i<array.length ;i++){
if(array[i]!=null)
System.out.println(array[i].toString());
}


//h
boolean found=false;

 len =cList.size();
```

```
for( i = 0 ; i < len ; i++ )
{
 Conference con =cList.removeFromfront();
if(con.equals(con3)){
found=true;
System.out.println(" found  con3 in clist");}
tempList.insertAtFront(con);
}

if(found==false)
System.out.println("con3 not found ");

// return all object to clist
for( int j = 0 ; j < len ; j++)
cList.insertAtFront(tempList.removeFromfront());

}//end main
}//end class
```

===================

Output :

```
   │
   └  ----jGRASP: process ended by user.

   ┌  ----jGRASP exec: java TestList
   │ enter name ,  id, size
▶▶ │ hala
▶▶ │ 5555
▶▶ │ 3
   │ enter fee,
▶▶ │ 15000
   │ wrong fee,you should Enter fee between 0 and 9999
   │ enter fee,
▶▶ │ 1500
   │ enter name ,  id, size
▶▶ │ yara
▶▶ │ 3434
▶▶ │ 3
   │ enter fee,
▶▶ │ 900
   │  cName: Artificial intelligence international Conference cID:
52114 fee: 3000.0 nbP: 2
   │ Bioinfomatics
   │ Cognitive system
   │
   │  cName: yara cID: 3434 fee: 900.0 nbP: 0
   │
   │  cName: hala cID: 5555 fee: 1500.0 nbP: 0
```

```
Conference have  fee less than 1000:
 cName: lama cID: 5555 fee: 900.0 nbP: 0

 cName: maha cID: 1234 fee: 500.0 nbP: 0

 cName: yara cID: 3434 fee: 900.0 nbP: 0

 found  con3 in clist

 ----jGRASP: operation complete.
```