PRG – Recovering Lab Second Partial Exam

ETSInf – Academic Year 2012/2013 – June 17th, 2013 – Duration: 50 minutes

NAME AND LAB GROUP:

1. 4 points When trying to read an integer value by means of the method `readInt()` (see the code below), an exception of the class `NumberFormatException` could be thrown if the sequence of chars typed by the user is not valid to be converted into an integer value.

```
public static int readInt( Scanner t, String message ) {
    System.out.print( message );
    String str = t.nextLine();
    int value = Integer.parseInt( str );
    return value;
}

public static void main( String[] args ) {
    Scanner kbd = new Scanner( System.in );
    int value = readInt( kbd, "Type an integer number: " );
    System.out.println( "The read value is: " + value );
}
```

To be done:

1. [1 point]: Modify the method `readInt()` in order to explicitly propagate that exception.
2. [3 points]: Modify the method `main()` to catch this exception and ask again the value until the read operation is performed without error.

Solution:

```
public static int readInt( Scanner t, String message )
    throws NumberFormatException
{
    System.out.print( message );
    String str = t.nextLine();
    int value = Integer.parseInt( str );
    return value;
}

public static void main( String[] args )
{
    Scanner kbd = new Scanner(System.in);
    int value = 0;
    boolean readOK = false;
    do {
        try {
            value = readInt( kbd, "Type an integer number: ");
            System.out.println( "The read value is: " + value );
            readOK = true;
        } catch( NumberFormatException e ) {
            System.out.println( "Be sure to enter an integer number!" );
        }
    } while( !readOK );
}
```

2. 1 point The following code is the beginning of the definition of the class `Account`. This class has been studied in lab practises.

```
import java.io.Serializable;
/** The class Account is used for representing bank accounts. */
public class Account implements Serializable {
    private double balance;
    private int    id;
```

To be done: Justify in a brief explanation what means that one class implements the interface `Serializable`.

Solution: The goal of such a declaration is being able to store objects of this class into binary files by means of an object of the class `ObjectOutputStream`, and also to be able to read objects of this class from binary files by means of an object of the class `ObjectInputStream`.

3. 5 points We need to include a new operation into the class `Concordance` in order to return the numbers of the lines where a given word appears, or a message indicating that such word doesn't exist in the concordance.

Remind, the attributes of the classes `Concordance` and `NodeCnc` are the following:

Concordance		NodeCnc	
private NodeCnc	first;	String	word
private int	size;	QueueIntLinked	numbersOfLine;
private boolean	sorted;	NodeCnc	next;
private String	delimiters;		

and finally, remind that the class `QueueIntLinked` has the method `toString()`, which returns an object of the class `String` with the integer values stored in the queue.

To be done: add a new method to the class `Concordance` with the profile:

```
public String getNumbersOfLine( String word )
```

Solution:

```
public String getNumbersOfLine( String word ) {
    NodeCnc p = first;
    while( p!=null && !word.equals(p.word) ) p = p.next;
    if ( p != null ) return p.numbersOfLine.toString();
    else return "Word not found!";
}
```