Mittuniversitetet
MID SWEDEN UNIVERSITY

Försättsblad Prov Original

Kurskod	Provkod	Tentamensdatum
DT007G	T104	2018-08-30
Kursnamn	Datateknik GR (A), Java II	
Provnamn	Tentamen	
Ort	Sundsvall	
Termin	H18	
Ämne	Datateknik	

8982538474

0027151

Mittuniversitetet
MID SWEDEN UNIVERSITY

Tentamen

dt007g Java II

Martin Kjellqvist*

2018.08.30

Instructions

Carefully read the questions before you start answering them. Note the time limit of the exam and plan your answers accordingly. Only answer the question, do not write about subjects remotely related to the question.

Write your answers on separate sheets, not on the exam paper. Only write on one side of the sheets. Start each question on a new sheet.

Make sure you write your answers clearly, if I cannot read an answer the answer will be awarded no points – even if the answer is correct. The questions are *not* sorted by difficulty.

Time 5 hours, no breaks.

Aids None.

Maximum points 49

Questions 9

Preliminary grades

E $\geq$ 50%, D $\geq$ 60%, C $\geq$ 70%, B $\geq$ 80%, A $\geq$ 90%.

Questions

(5p) 1. Describe the process of attaching a socket to another host, then reading, writing and closing the socket. List all relevant classes.

(4p) 2. Describe the problems associated with multithreaded applications, that are not an issue in single threaded applications. I.e. describe the unique difficulties inherent to multithreading.

(3p) 3. Describe and provide an example as to how the synchronized keyword can be used.

*martin.kjellqvist@miun.se

(3p) 4. The java language contains a restriction that renders the prospect of a swap-method impossible. Describe this restriction. Suggest a reason for this restriction.

```
int a = 42;
int b = 32;
swap( a, b );      // a = 32, b = 42
String s1 = "Hello";
String s2 = "World";
swap(s1, s2);     // s1 = "World", s2 = "Hello"
```

Both these calls to swap, common in other languages, are not possible in Java.

(4p) 5. Describe situations when a private constructor is useful.

(8p) 6. Consider a collection Vector<Parent> p. The class Parent is a superclass to a class Child. Vector is implementing the List generic interface (as per the jdk).
Comment on wether the following situations are feasible or problematic. Argue your position.

```
// 1- Assigning with
p = new Vector<Object>();

// 2- calling the following method with a valid p as argument
void method(Vector<Child> als){ ... }

// 3- calling the following method with a valid p as argument
void method(List<Parent> lp){ ... }

// 4- calling the following method with a valid p as argument
void method(Vector<Object> alo){ ... }
```

(4p) 7. There is an interface definition:

```
public interface IntValue{
        public int getValue();
}
```

Provide a *complete* sample implementation for the `getIntValue` method

```
public class ExamQ{
        public static IntValue getIntValue(int value){
                ...
        }
}
```

(12p) 8. Provide code for the following situations:

- You are defining a custom JButton-class. The JButton needs to intercept the action event using a namned inner class.
- You have a regular class Handler implementing ActionListener. A method inside a GUI class attaches objects of type Handler, to two button objects, button1 and button2. Both Buttons are members of the GUI class.
- An anonymous inner class that is a Socket whos close() method is an empty method, a no op. As well as a toString() method contains the usual socket info as well as the text "uncloseable". Provide surroinding code to show how it is constructed, and methods are called.

Note that if you need to make additional assumtions in the situations, state those assumtions. The code is graded on class structure, method calls, class relationships, instantiations and necessary clues to make the structure and code flow clear.

9. Provide an implementation for an exception that

(3p) (a) Is not enforced/checked by the compiler.

(3p) (b) Contains a message string as well as a timestamp when the exception occurred.

Good luck,
Martin