Instructor task (MMAN)

Course: 20441 - Introduction to Computer Science and the Java Language

Assignments: The basics of language

Study material for the assignment: 1-2 units

Task weight: 3 points

Number of questions: 2

Deadline for submission: 2021.11.13

Semester: 2022 a

) ÿ (

pay attention:

- **The names of the departments must be observed exactly as they were written.**
- **The programs must be documented in internal documentation in English only) at the beginning of the program The documentation explains what the program does in general and during the programs the documentation explains the code (.**
- **Do not add methods beyond those explicitly required in the task.**
- **Do not use advanced material and especially not loops.**
- **Constants should be used where possible.**
- **Care must be taken to indent (indentation) - correct (correct, and for names of variable variables) Meaning) in English (and according to the conventions in the course.**
- **The output format must be observed exactly as indicated in the question: correct spelling, letters Big and small, profits, etc.**
- **On the course website you will find a guide file for solving the programming assignments. It is very worthwhile to review it and follow it. The file is on the course website within "Unit 1" In "Auxiliary Guides and Links" and it is called "Guidelines for Writing Plans and Assignments The course".**
- **The submission of the task is done solely with the help of the online task system on the site The course.**
- **Do not forget to keep the reference number you will receive from the system after submission.**

Question 1 - to run) 40%)

The use of milliseconds is common in computers for measuring time. Millisecond is the millisecond. In this assignment you will write a program that converts from milliseconds to a more readable format for humans.

The program will read from the input entered by the user an integer representing milliseconds, and print on the screen the number of days, hours, minutes and seconds that are equal to the number of milliseconds recorded) should be rounded down If necessary (, in the following format:

d_days_h: m: s_hours

When instead of s, m, h, d the number of days, minutes and seconds should be printed respectively.) Should be read from left to right(.

_ Indicates a space, and should not be printed. The character

Examples:

0 days 0: 1: 40 hours will be printed 100000 input for •

1 days 3: 4: 50 hours will be printed 97490000 The input for •

1 days 3: 4: 50 hours will be printed 97490738 Record for •

Please note - the output format should be exactly as set!

It can be assumed that the user will enter a valid input, i.e. a non-negative integer.

It can be assumed that the input is greater than 1000 milliseconds.

To read from the input use the Scanner class .

To use it you need to write the line at the beginning of the program

```
import java.util.Scanner;
```

The Interface of the Scanner Department can be found within Section 6.2 Programs Interactive from the book Solutions Software Java found after video 4.2, where some of the methods are presented.

Explanations about the department and its use can be found on the course website within "Unit 2" in "Auxiliary Instructions and Links" in the file "Guide to working with the Scanner Department for user input".

The program you wrote should be in a department called Milliseconds.

For your help, we have written parts of the department here. You must fill in the blanks (also the missing documentation).

Do not forget to add constants as needed.

Below is the beginning of the Milliseconds class.

```
import java.util.Scanner;
public class Milliseconds
{
    public static void main (String [] args)
    {
        Appropriate constants should appear here ... //

        Scanner scan = new Scanner (System.in);
        System.out.println ("This program reads an integer which"                       +
            "represents Milliseconds and converts it to days," "hours, minutes and seconds.");    +

        System.out.println ("Please enter the number of Milliseconds");
        long ms = scan.nextLong ();

        Here you must continue ... //

    } // end of method main
} // end of class Milliseconds
```

Question 2 - to run) 60%)

background:

It is said of the well-known mathematician and physicist Albert Einstein, who greatly enjoyed confusing his friends with the

riddle described below. You too can have fun with it:

First of all, write the number 1089 on a piece of paper, fold it, and let a friend save it. No one will read what you have written

until the end of the process.

Now, ask the friend to write an integer (positive). Three digits. Emphasize that the first and last digits must be different. That

you do not see the number

Who wrote.

After the friend has written the number, ask him to turn it so that the first digit will be the last and vice versa. Now, ask him to

subtract the smaller number from the larger one.

For example, if the number you chose was 773, the inverse number is 377, and the difference is 396 = 377-773.

After doing so, ask the friend to reverse the difference (396 becomes 693).

Finally ask him to summarize the difference and the reverse difference. (1089 = 693 + 396 (

If you did everything right, the friend will be stunned - the number you first wrote on the note, which is folded

In his hands contains the number 1089 ...

Now he approaches the task of writing a plan that executes it. The program will play the Einstein game as detailed:

1. Print a short welcome message to the user (in the following format: Einstein the to Welcome)

.) magic game. ”

2. Ask the user to enter a 3-digit number (in the following format: digit 3 a enter Please “

.) positive number whose first and last digits are different: ”

3. Enter the number - If the number is not positive, 3 digits, print an error message) in the format

The program you entered is not a 3 digit positive number

If the number is 3 digits, but the unity digit is the same as the hundreds digit, print a message

“The first and last digits of the number should be: Next in format) Error

Different) and completed the program.

Otherwise, continue with the program and print the number entered by the user

.) "User number is: 773" : Next in format)

4. Invert the number and print the difference between the number entered by the user and the reverse number. Take care

That the difference will always be a positive number.) In the following format: "396: Difference)"

You can use the) x (abs.Math method, which is a Java method in the Math class

And returns the absolute value of x . To use it you do not need to import any class, but call it by its full name (x

(abs.Math when instead of the parameter x you write the expression

Who want to get its absolute value. If the x parameter of this method is of type

Integer (int , (the method will return an integer) int.)

5. Print the reverse difference.) In the following format: "693: difference Reversed)"

6. Check if the sum of the difference and the inverse difference is equal to **the fixed** number 1089, if so, print

SUCCEEDED , otherwise print FAILED.

For your help, we have written here an output obtained from running the program for the number 773

```
Welcome to the Einstein magic game.
Please enter a 3 digit positive number whose first and last digits are different:
773
User number is: 773
Difference: 396
Reversed difference: 693
SUCCEEDED
```

Remarks:

- It can be assumed that the user entered an integer.

- Each print should be in a separate line and accompanied by a message about what was printed (for example, the difference).

- If the difference between the number and its inverse is double digits (for example, for input 384, the difference is 99,)

It's OK. In this plan the "inversion" of 99 is 990.

- This time, too, use the Scanner class to read from the input.

- **Make sure that the program output is exactly as listed in the instructions! Do not add messages about what is written explicitly in the instructions and be precise) including uppercase / lowercase letters and punctuation (in the detailed messages.**

The program you wrote should be in a department called Einstein.

In this MMAN you have to write the plan of each of the two questions

In one class) one class for each question in the names Milliseconds and

Einstein,) and everything in the main method . **Do not add additional methods .**

Loops and / or arrays must not be used for this task!

submission

1. The submission of the financial statement is done only electronically, through the assignment submission system.

2. The solution to question 1 includes the java.Milliseconds file.

3. The solution to question 2 includes the java.Einstein file.

4. Pack the two files in a single zip file **(not rar)** and send it only.

5. Do not forget to keep the reference number you received from the system after submission. If you did not receive a reference number, it is a sign that the submission was not accepted.

6. Please note, you can repeatedly send the task in the system, if you want to fix something in it. Each submission overrides the previous submission. **But did so solely until the filing date.** After the date, it will be considered as if you submitted late, even if the first submission was on time! Also, if the facilitator has already downloaded your assignment from the system, you will not be able to send a copy

More informed.

Successfully