

Practice sheet 8 – due on 2013-01-14 and 2013-01-21

Please prepare your solutions / answers in written form. Make sure to always prepare them in a way that you are able to present them to your class mates and discuss your solution process effectively.

Please remember to always list your reference sources.

Learning aims: Gaining practical experiences with Cleanroom (aha-effect)

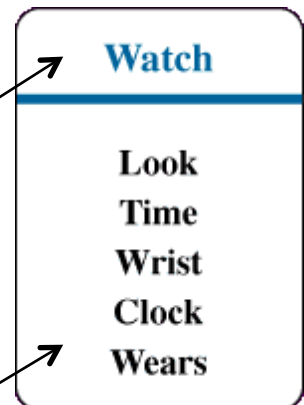
Task 8 – 1: Taboo-Cards (due on 2013-01-14)

Create at least two Taboo cards for each of the following areas:

1. CMMI
2. Software Engineering Economics
3. Software Measurement
4. Opens Source Software Development
5. Cleanroom Software Engineering
6. V-Modell XT
7. Errors and Defects

Each card should contain

- one keyword at the top and
- below 5 terms that are not allowed to be used by a person who wants to explain the keyword.



<http://www.monzy.org/taboo/>

Please use the document template offered in the Blackboard System for creating the cards.

Send an e-mail containing your cards in a .doc(x) or similar editable Format (not pdf, txt, etc.) to julia.schenk@fu-berlin.de

Task 8 – 2: Questions concerning the Cleanroom method (due on 2013-01-14)

Note down the answers to the following questions in a few brief and precise sentences.

1. Name and characterize the main idea of the Cleanroom method?
2. How does the verification take place in Cleanroom?
3. When is Cleanroom well suited and when not? Why so?
4. To what extend is the Cleanroom method suited for task 8-3?

Practice sheet 8 – due on 2013-01-14 and 2013-01-21

Task 8 – 3: Implement Mastermind with Cleanroom (due on 2013-01-21)

Develop the logistic game Mastermind by using the Cleanroom method (without statistical testing).

Program requirements:

1. *Rules of the game:* Comply with the rules of Mastermind on <http://de.wikipedia.org/wiki/Mastermind>. Implement the variation with 4 slots, 5 colors and no double usage of colors.
2. *Play modes:* Your program should offer two play modes to start:
 - a. Play mode 1: The program determines a combination of colors the user has to "guess". After each guess the program judges the moves.
 - b. Play mode 2: The program guesses a combination of colors the user made up. After each guess the user judges the moves.
In the tutorial we will compare the strength of the programs. Think about a reasonable strategy yourselves¹.
3. *Programming language:* Write your program in Java.
4. *User interface:* Implement a sign-based (no graphic!) user interface for the program. Use the following descriptions on the console:
 - Use the numbers 1,2,3,4,5 instead of colors.
 - Use the letters s and w for the colors black and white.
 - Example for the board's output after the third move:

```
1.  1234  ww..
2.  3526  wws.
3.  5316  ws..
```
 - It is up to you whether you display the board entirely after each move or the moves are displayed consecutively.

Procedure requirements:

- *Cleanroom method:* Implement the program by using the Cleanroom method. Have another good look at the corresponding slides of the lecture first and use the **half-formal verification** presented there. In case the slides' information does not suffice, refer to the web for further sources.
- *No tests: Do not execute* the program during development! You should, however, develop your program in an IDE and therefore use a syntax checker. The first test will take place **during the tutorial!**
- *Teamwork: Work in pairs.* Your half-formal description should suffice to convince each other and later your fellow students in the tutorial of each step's correctness.

Bring the following to the tutorial:

- The finished program, **not tested before** but executable on your laptop.
- The entire stepwise decomposition ("boxes") with the corresponding verifications as text document on your laptop. Be prepared to present it in the tutorial.

¹ You can definitely find various hints (or even implementations) on the web. However, do not spoil the party and learning effect and do **not** look at these solutions.