

A snail called Mob

Not about Mob-Programming

Jürg Weilenmann



Sorry guys, no code

```
import java.util.Scanner;

class Pascal{

public static void main(String[] args) {

    int kakao = 2;           //AnzahlKakao
    double kpreis = 2.30;    //Preis für 1 Kakao
    double kpg = kakao*kpreis; //KakaoPreisGesamt

    int wasser = 1;
    double wpreis = 1.66;
    double wpg = wasser*wpreis;

    int handy = 1;
    double hpreis = 370.0;
    double hpg = handy*hpreis;

    int butter = 3;
    double bpreis = 0.69;
    double bpg = butter*bpreis;

    double gesamt = kpg+wpg+hpg+bpg;
    double gelddabei = 400;

    if(gesamt == 0) {
        System.out.println("Nichts gekauft ");
    }else if(gelddabei > gesamt) {

        double gegeben = 0;
        double zurÜck = 0;

        Scanner g = new Scanner(System.in);

        System.out.println("KassiererIn: Das macht "+gesamt+" Euro bitte.");
        System.out.println("Du hast "+ gelddabei+" EUR dabei, wieviel möchtest du geben?");

        boolean akzeptiert = false;

        while(!akzeptiert) {
```

After last Friday

- Kind if disapointed last friday
- Long discussions but not a good result
- But why?
 - Mob not alligned ?
 - Bad programmers ?
 - Lack of knowledge ?
 - Teachers ;-) ?

Start an experiment

- Implement TicTacToe on Friday. Timebox: 2h
- Code it again in the Mob on Monday
- Analyse the differences

Rules

- Apply what I've learned so far
- Spezial attention to:
 - write test first
 - test behavior
 - code incremental
 - take care of Calisthenics
 - do only what's required
 - have the complete game as fast as possible
- Analyse the differences

- I started....
- Next time I checked the clock, I exceeded already the timebox
- But I was not happy with the code, so I gave me another hour
- After the extra hour I was still not happy -> another hour
- Now I thought it's ok, not everything was polished, but ok

The comparison

- After Monday, I've compared the Mob's solution with my one
- And shit , the Mob's solution was better ;-)

What's made the differences ?

- the MOB was guided by Alessandro ;-)
- I choose another order to implement the degrees of freedom
-> has result in a different model and I had more changes in the refactoring phase
- I applied the Calisthenics sometimes too late or too early
-> spent too much time implementing something that I had to refactor afterwards
- Not always used the simplest possible class
(own class instead of an enum)

- tried to make things ,nice‘ where not really needed
-> ended up in a too complex solution -> undo -> lost time
- still have some code (for convenience) used only for testing
- took (or mis-interpreted) the requirements too tight
- needs some practise

and of course, I will never do these mistakes again!

Thank you

and don't contact me and don't download my code ;-)