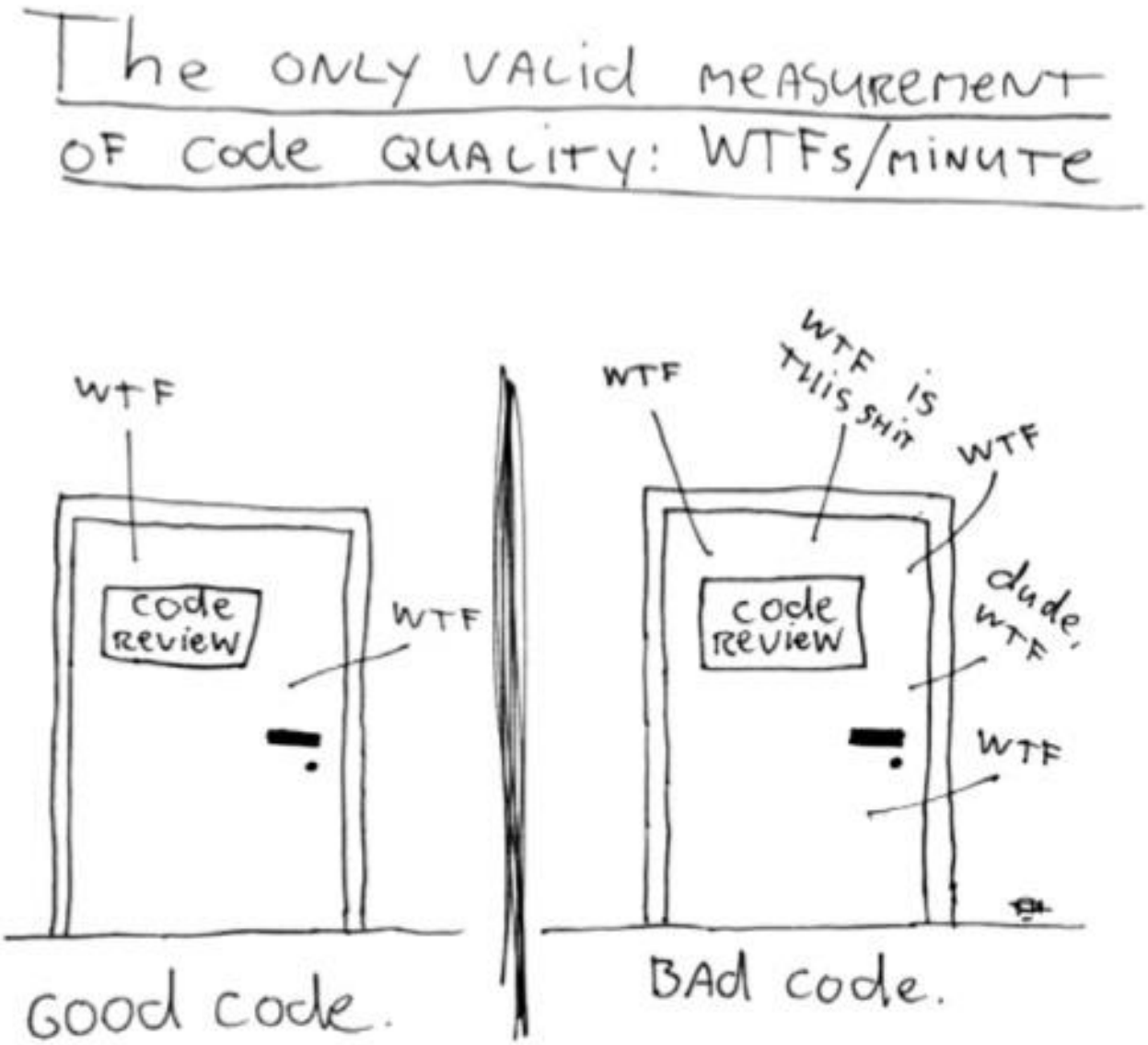
CLEAN CODE

Methods & Techniques for better Code

by Nicolas Wagner



Source: Clean Code - Refactoring, Patterns, Testen und Techniken für sauberen Code: Deutsche Ausgabe

CONTENT

- Importance of Clean Code
- Clean Code – Characteristics
  - Meaningful Names
  - Methods
  - Comments
  - Formatting
- Let‘s do it!

IMPORTANCE OF CLEAN CODE

CLEAN CODE - DEFINITIONS

elegant

no duplication

simple

direct

focussed

clear

efficient

purpose fulfilling

REASONS FOR BAD CODE

- Time Pressure?
- Desire of Finalization?
- Too much Work?

CLEAN CODE - PRINCIPLES

Read-Write: >10 : 1

- SRP – Single Responsibility Principle
- DIP – Dependency Inversion Principle
- OCP – Open-Closed Principle
- Style

Readable Code

for every Developer

CHARACTERISTICS OF CLEAN CODE

CHARACTERISTICS - OVERVIEW

- **Meaningful Names**
- Methods
- Comments
- Formatting

```java
17    public List<int[]> getThem() {
18        List<int[]> list1 = new ArrayList<int[]>();
19        for (int[] x : theList)
20            if (x[0] == 4)
21                list1.add(x);
22        return list1;
23    }
```

MEANINGFUL NAMES

- Choose pronounceable Names
- Avoid single-digit Names
- Describe the Purpose
- Code readable like a Newspaper
- Avoid part-names:
  "List", "Controller", "Manager", …

```java
17⊖     public List<int[]> getThem() {
18          List<int[]> list1 = new ArrayList<int[]>();
19          for (int[] x : theList)
20              if (x[0] == 4)
21                  list1.add(x);
22          return list1;
23      }
```

```java
25⊖     public List<int[]> getFlaggedCells() {
26          List<int[]> flaggedCells = new ArrayList<int[]>();
27          for (int[] cell : gameBoard)
28              if (cell[STATUS_VALUE] == FLAGGED)
29                  flaggedCells.add(cell);
30          return flaggedCells;
31      }
```

CHARACTERISTICS - OVERVIEW

- Meaningful Names
- **Methods**
- Comments
- Formatting

```java
    //NumberGuess Game
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        while (true) {
            System.out.print("Wie lautet Dein Name? ");
            String name = s.next();
            int random = (int) (Math.random()*100);
            int tries = 0;
            while (true) {
                tries++;
                System.out.print(name+", rate eine Zahl [0-100]: ");
                int guess = s.nextInt();
                if (guess > random) {
                    System.out.println("Versuch "+tries+": "+guess+" ist zu groß!");
                    continue;
                } else if (guess < random) {
                    System.out.println("Versuch "+tries+": "+guess+" ist zu klein!");
                    continue;
                } else {
                    System.out.println("Versuch "+tries+": "+guess+" ist korrekt!");
                    break;
                }
            }
            System.out.print("Möchtest Du erneut spielen (1) oder das Spiel beenden (0)? ");
            if (s.nextInt() == 1) {
                continue;
            } else {
                System.out.println("Danke fürs Spielen!");
                break;
            }
        }
        s.close();
    }
```

METHODS

- Size: Max. 50-100 lines
  Best Case: ~10 lines
- SRP – One Task per Method
- Multiple Abstraction Levels
- Names: Use of “is“, “get“, “set“, “add“, …
- Parameters: Maximum of 3

```java
48    public static void main(String[] args) {
49        scanner = new Scanner(System.in);
50        keepOnPlaying = true;
51        while (keepOnPlaying) {
52            initializeGame();
53        }
54        scanner.close();
55    }
56
57    public static void initializeGame() {
58        Random random = new Random();
59        username = getUsernameFromConsole();
60        amountOfTries = 0;
61        randomNumber = random.nextInt(101);
62        startGuessGame();
63        askUserForPlayingAnotherGame();
64    }
65
66    public static void startGuessGame() {
67        boolean isNumberCorrect = false;
68        while (!isNumberCorrect) {
69            int guessedNumber = getGuessedNumberFromConsole();
70            amountOfTries++;
71            if (guessedNumber > randomNumber) {
72                System.out.println("Try " + amountOfTries + ": Your number is too high!");
73            } else if (guessedNumber < randomNumber) {
74                System.out.println("Try " + amountOfTries + ": Your number is too low!");
75            } else if (guessedNumber == randomNumber) {
76                System.out.println("Try " + amountOfTries + ": Your number is correct!");
77                isNumberCorrect = true;
78            }
79        }
80    }
```

```java
    public static void askUserForPlayingAnotherGame() {
        System.out.print("Do you want to play again (1) or quit (0)? ");
        int userSelectionForNextNumber = scanner.nextInt();
        if (userSelectionForNextNumber == 0) {
            keepOnPlaying = false;
            System.out.println("Thanks for playing!");
        } else if (userSelectionForNextNumber == 1) {
            System.out.println("Next Game is starting...");
        } else {
            System.out.println("Invalid number! Please enter 0 for quit or 1 for another game.");
            askUserForPlayingAnotherGame();
        }
    }

    public static String getUsernameFromConsole() {
        System.out.print("Your name: ");
        return scanner.next();
    }

    public static int getGuessedNumberFromConsole() {
        System.out.print(username + ", guess a number: ");
        return scanner.nextInt();
    }
```

CHARACTERISTICS - OVERVIEW

- Meaningful Names
- Methods
- **Comments**
- Formatting

```java
    //NumberGuess Game
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        //Until the user wants to stop the game-program
        while (true) {
            System.out.print("Your name: ");
            String name = s.next();
            int random = (int) (Math.random()*100);
            int tries = 0;
            while (true) { //user has to enter numbers until the random number has been found
                tries++;
                System.out.print(name+", guess a number [0-100]: ");
                int guess = s.nextInt();
                //if and else for looking what condition is fulfilled
                if (guess > random) {
                    System.out.println("Try "+tries+": "+guess+" is too high!");
                    continue;
                } else if (guess < random) {
                    System.out.println("Try "+tries+": "+guess+" is too low!");
                    continue;
                } else {
                    System.out.println("Try "+tries+": "+guess+" is correct!");
                    break;
                }
            }
            //ask user if another game should be started
            System.out.print("Do you want to play again (1) or quit (0)? ");
            if (s.nextInt() == 1) {
                continue;
            } else {
                System.out.println("Thanks for playing!");
                break;
            }
        }
        s.close();
    }
```

COMMENTS

- No Comment = Best Case
- Appropriate Comments:
  law, informative, warning, TODOs
- JavaDoc (public APIs)
- Code commented out = Worst Case

```java
46      // Copyright (C) 2003,2004,2005 by Object Mentor, Inc. All rights reserved.
47      // Released under the terms of the GNU General Public License version 2 or later.
48
49      //Extremely high run-time. Consider using another method.
50      public void mergeDatabases() {}
51
52      //TODO database connection needs to be defined.
53      public String getPackedUserdata() {return "";}
54
55      /**
56       * Deletes all userdata stored in the customer-database.
57       * @param username - unique username-key
58       */
59      public void removeAllUserdata(String username) {}
60
61
62
63
64
65
66
67
68
69
```

void com.nicolaswagner.cleancode.Comments.removeAllUserdata(String username)

Deletes all userdata stored in the customer-database.

Parameters:

username - unique username-key

Press 'F2' for focus

CHARACTERISTICS - OVERVIEW

- Meaningful Names
- Methods
- Comments
- **Formatting**

```
7⊖    public static void main(String[] args) {Scanner scan=new Scanner(System.in);System.out.print("Wie viele Fibonacci-Zahlen"
8             + " sollen generiert werden? ");int anzahlFibonaccis=scan.nextInt();if(anzahlFibonaccis<2){anzahlFibonaccis=2;}
9             int[] fibonaccis=new int[anzahlFibonaccis];fibonaccis[0]=1;fibonaccis[1]=1;fibonaccis=getFibonacci(fibonaccis,
10            anzahlFibonaccis-2);for(int i=0;i<fibonaccis.length;i++){System.out.println("F("+(i+1)+"): "+fibonaccis[i]);}scan.close();
11    }
12
13⊖   public static int[] getFibonacci(int[] fibonaccis,int anzahl){if(anzahl!=0){for(int i=0;i<fibonaccis.length;i++){if(fibonaccis[i]
14            ==0){fibonaccis[i]=fibonaccis[i-2]+fibonaccis[i-1];break;}}return getFibonacci(fibonaccis,anzahl-1);}else{return fibonaccis;}
15    }
```

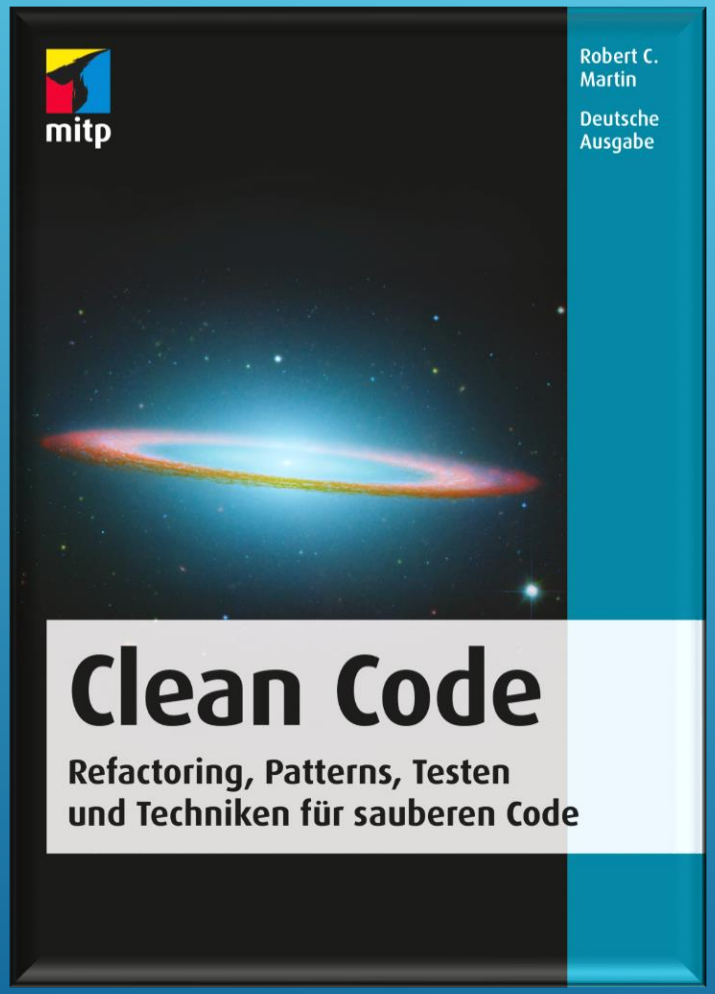
FORMATTING

- Maximum: 200-500 lines ea. 100 characters per file
- Common → Detail
- Teamwork:
  Make use of a Formatter and guidelines

```java
17    public static void main(String[] args) {
18        Scanner scan = new Scanner(System.in);
19        System.out.print("Wie viele Fibonacci-Zahlen sollen generiert werden? ");
20        int anzahlFibonaccis = scan.nextInt();
21        if (anzahlFibonaccis < 2) {
22            anzahlFibonaccis = 2;
23        }
24        int[] fibonaccis = new int[anzahlFibonaccis];
25        fibonaccis[0] = 1;
26        fibonaccis[1] = 1;
27        fibonaccis = getFibonacci(fibonaccis, anzahlFibonaccis-2);
28
29        for (int i = 0; i < fibonaccis.length; i++) {
30            System.out.println("F(" + (i+1) + "): " + fibonaccis[i]);
31        }
32        scan.close();
33    }
34
35    public static int[] getFibonacci(int[] fibonaccis, int anzahl) {
36        if (anzahl != 0) {
37            for (int i = 0; i < fibonaccis.length; i++) {
38                if (fibonaccis[i] == 0) {
39                    fibonaccis[i] = fibonaccis[i-2] + fibonaccis[i-1];
40                    break;
41                }
42            }
43            return getFibonacci(fibonaccis, anzahl - 1);
44        } else {
45            return fibonaccis;
46        }
47    }
```

RECOMMENDATION

& SOURCE



THANK YOU FOR YOUR ATTENTION

LET'S DO IT!

(HTTPS://GITHUB.COM/NIWA99/CLEAN-CODE-PRESENTATION-SE/BLOB/MASTER/TASKDESCRIPTION.MD)