Чист код

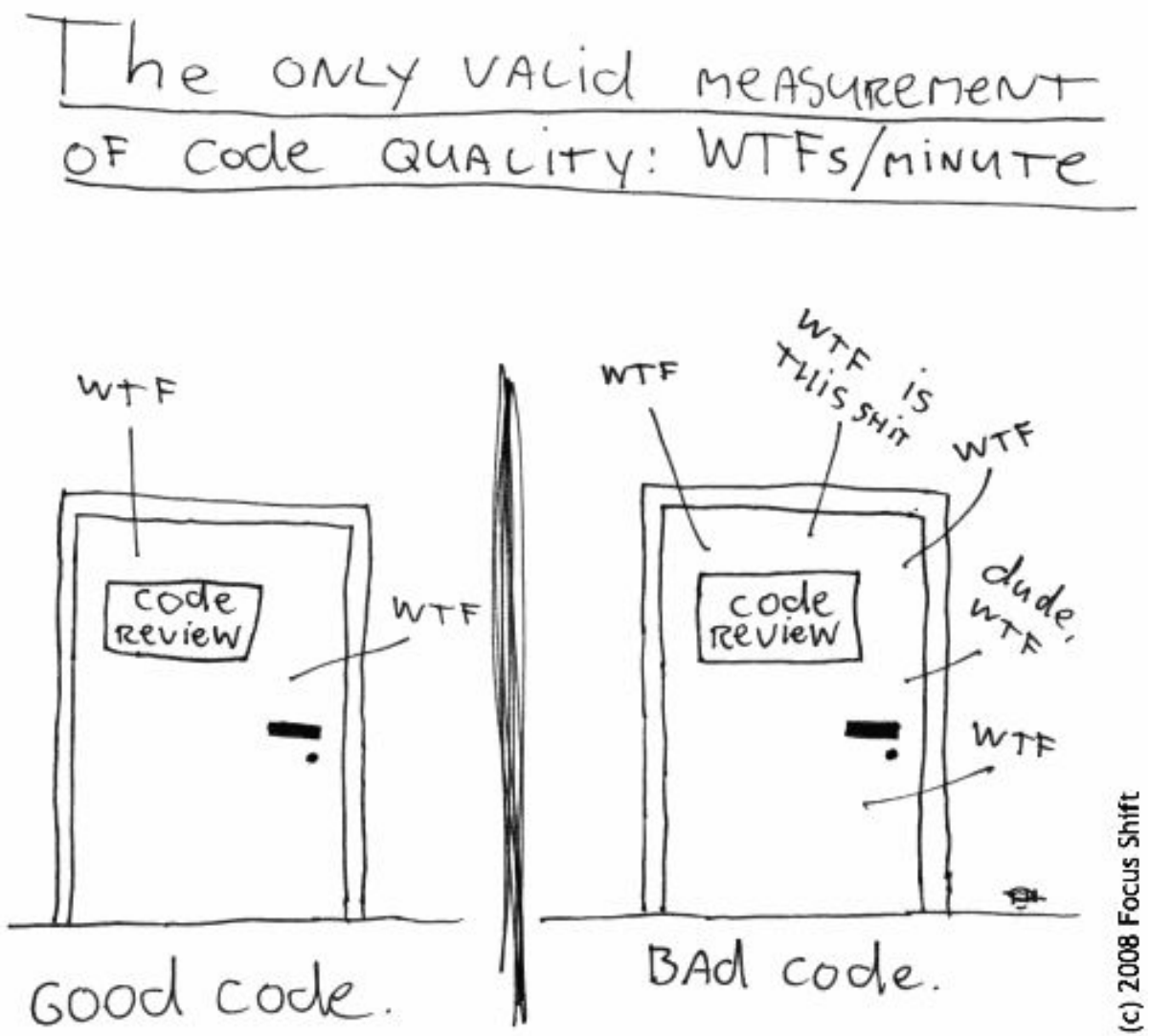
Емилиян Кадийски

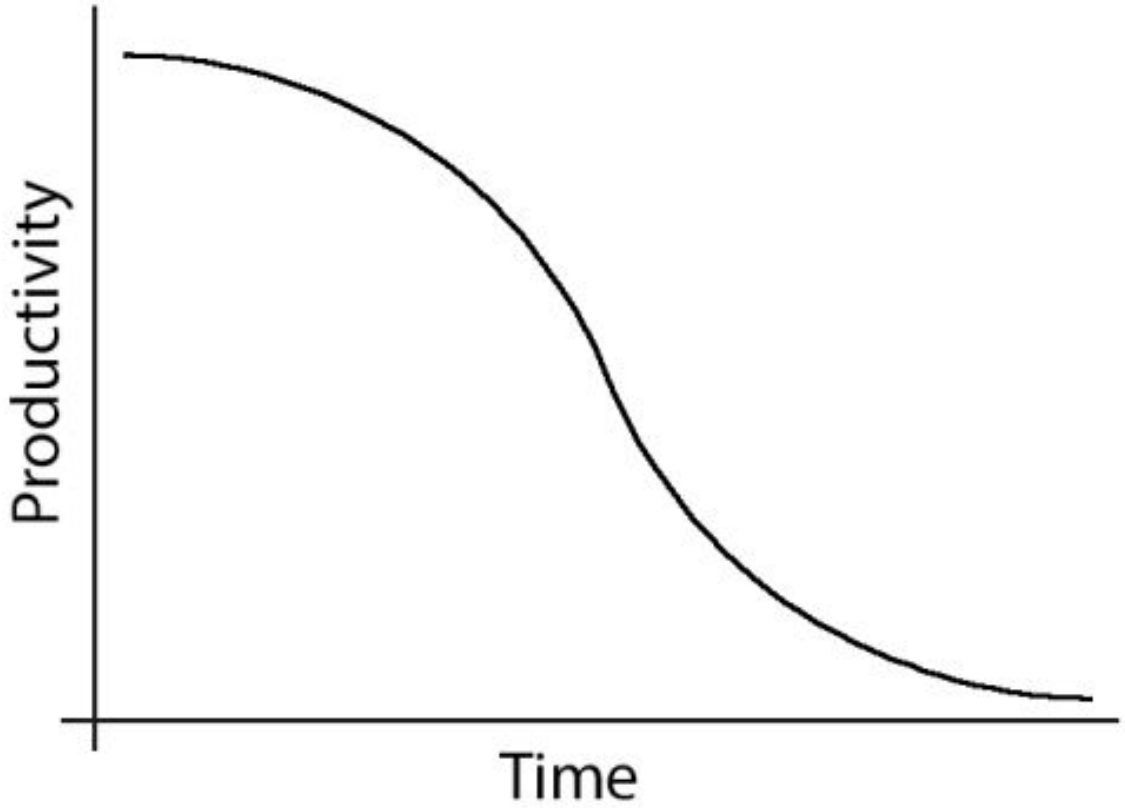
https://digitalrazgrad.org
https://digitaltargovishte.org

Планът за днес

1. Защо е нужен чистият код?
2. Именуване
3. Методи
4. Коментари
5. Форматиране
6. Упражнение

ЧИСТ КОД?


The ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE
WTF
code review
WTF
Good code.
WTF
WTF IS THIS SHIT
WTF
code review
dude, WTF
WTF
BAd code.
(c) 2008 Focus Shift


Productivity
Time

Правило

LeBlanc's law:

“LATER EQUALS NEVER”

ИМЕНУВАНЕ

Именуване на променливи

Имената **трябва да имат смисъл**

- `String n;`
- `String name;`

Имената **трябва да са лесни за четене** и произнасяне

- `String formatydhms;`

Трябва да **могат да се търсят**

- `int a;`
- `boolean bool;`

Трябва да **избягваме дезинформация**

- `String[ ] accountList;  //всъщност не е лист, а масив`
- `int index = arr.length; //всъщност не е индекс`

Конвенцията за именуване е **camelCase**

Именуване

Имена на класове	Имена на методи

- съществителни
- първата буква е главна
- PascalCase
 	
`Date`
`CarDealer`
`Human`
`FileReader`
	- глаголи
	- първата буква е малка
	- camelCase
	
	`getYear()`
	`sellCar()`
	`sleep()`
	`readFrom(File file)`

ЧИСТ КОД?

Методи

Трябва да са **малки**
- не повече от 20 реда
- колкото по-малки, толкова по-добре

Не трябва да **влагаме много блокове** един в друг
- не повече от три нива на влагане

Всеки метод трябва да **прави само едно нещо**
- не трябва да има странични ефекти
- или променя състоянието на даден обект, или отговаря на даден въпрос, никога и двете

Duplication may be the root of evil in software.

Страничен ефект

Един метод трябва да прави само **едно единствено нещо**!

Не дезинформирайте!

```
public static double calculateRectArea(double sideA, double sideB) {
    double rectArea = sideA * sideB;
    System.out.println(rectArea);
    return rectArea;
}
```

Добри практики

Оптималният брой параметри на даден метод е **нула.** След това е един, два и в редки случаи три. Това е така, защото:

- кодът става по-разбираем
- тестването е много по-лесно

Колко пъти ви се е случвало да размените местата на аргументите?

Булеви променливи не трябва да се подават.

Методи

Методите трябва да са колкото се може по-кратки и по-прости. Да имат колко се може по-малко аргументи.

```
public static String personInfo(String firstName, String secondName,
String homeTown, int age, int petsCount, String middleName, String
mothersName) {
    String fullName = firstName + “ “ + middleName + “ “ + lastName;
    String result = fullName + “\nAge: ” + age + “\n” + homeTown +
“\nSon of: ” + mothersName + “\nTotal pets: ” + petsCount;
    return result;
}
```

Правило

“FUNCTIONS SHOULD DO ONE THING.

THEY SHOULD DO IT WELL.

THEY SHOULD DO IT ONLY.”

КОМЕНТАРИ

Коментари

Коментарите не компенсират за лош код!!!!

- `int d; //elapsed time in dates`
- `int elapsedTimeInDays;`

Добри коментари

- Полезна информация
- *TODO*
- Предупреждения за последствия

Лоши коментари

- Остарели
- Коментари, които не са на английски
- **Закоментиран код**
- Тези, които трябва да са имена, не коментари
- Маркери

Добри коментари vs лоши коментари

```
// format matched kk:mm:ss EEE, MMM
//dd, yyyy


//TODO: split this method into two
separate ones


//Use with caution! Takes huge
//amount of time to run!
```

```
//Trqbwa da se razdeli na o6te
//metodi


//int number = 0;


int a; //elapsed time in seconds


//___________constants__________
```

Добри коментари vs лоши коментари

Нищо не е по-полезно от коментар на място

- описание на сложен алгоритъм
- TODO

Нищо не пречи повече от коментар не на място

- описва очевиден код
- обяснението на лошо написан код с коментар не променя факта, че е лошо написан код
- описва действието на един метод вместо името на метода да върши това действие

Правило

Brian W. Kernighan and P.J. Plaugher:

“DON’T COMMENT BAD CODE - REWRITE IT.”

ФОРМАТИРАНЕ

```
public class FitNesseServer implements SocketServer { private FitNesseContext
context; public FitNesseServer(FitNesseContext context) { this.context =
context; } public void serve(Socket s) { serve(s, 10000); } public void
serve(Socket s, long requestTimeout) { try { FitNesseExpediter sender = new
FitNesseExpediter(s, context);
sender.setRequestParsingTimeLimit(requestTimeout); sender.start(); }
catch(Exception e) { e.printStackTrace(); } } }
```

VS

Форматиран код

```
public class FitNesseServer implements SocketServer {
  private FitNesseContext context;

  public FitNesseServer(FitNesseContext context) {
    this.context = context;
  }

  public void serve(Socket s) {
    serve(s, 10000);
  }

  public void serve(Socket s, long requestTimeout) {
    try {
      FitNesseExpediter sender = new FitNesseExpediter(s, context);
      sender.setRequestParsingTimeLimit(requestTimeout);
      sender.start();
    }
    catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

CTRL + ALT + L

Групиране

Смислово свързан код трябва да е групиран заедно вертикално

```
Scanner input = new Scanner(System.in);
String firstName = input.nextLine();
String lastName = input.nextLine();

String fullName = firstName + “ “ + lastName;

System.out.println(fullName);
```

Декларация и използване

Декларирайте променливи, близо до мястото на използване

```
String result;
int tenFactoriel = 1;
for (int i = 1; i <= 10; i++) {
   tenFactoriel *= i;
}



result = “Ten factories equals = “ + tenFactoriel;
System.out.println(result);
```

Методи

Методи, които се викат един друг, трябва да са близо

```
public static void printLetter(String[] data) {
    String formattedLetter = formatLetter(data);
    System.out.println(formatterLetter);
}
public static String formatLetter(String[] data) {
    return String.format(“letter format...”, data[0], data[1], ...);
}
public static double calculateSomething(double arg1, double arg2) {...}
```

Празни редове

Кратки редове. Използвайте празни редове, но никога не оставяйте допълнителни

```
public static String formatLetter(String[] data) {
    String formattedLetter = String.format(“letter format...”, data[0],
data[1], ...);

    return formattedLetter;
}
```

Скоби

След скоба отваряща блок от код ({), винаги започвайте следващия ред 1 таб навътре. Винаги има празен ред след скоба затваряща блок от код (}).

```
public static String isPersonTooYoung(int age) {
    if (age < 18) {
        return “Yes, the person is too young”;
    }

    return “No, the person isn’t too young”;
}
```


Robert C. Martin Series
Clean Code
A Handbook of Agile Software Craftsmanship
Robert C. Martin

УПРАЖНЕНИЕ

Най-добрият начин, по който можем да се научим да пишем добър код, е да рефакторираме вече написан код.

Въпроси?

ДИГИТАЛНО ОБЩЕСТВО ТЪРГОВИЩЕ

ДИГИТАЛНО ОБЩЕСТВО РАЗГРАД

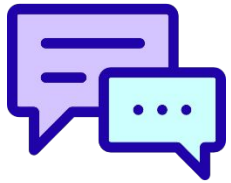
ДА ОТВОРИМ КРЪГА

ОБЩЕСТВЕН ДАРИТЕЛСКИ ФОНД ЗА РАЗГРАД • ФОНДАЦИЯ

КЛУБ НА НСО

Trainings @ Digital Razgrad & Digital Targovishte

- Digital Razgrad
  - https://digitalrazgrad.org
  - https://facebook.com/digitalrazgrad.org
  - digitalrazgrad.slack.com



- Digital Targovishte
  - https://digitaltargovishte.org
  - https://facebook.com/digitaltargovishte.org
  - digitaltargovishte.slack.com