

**20 / 80**

# **Добри правила при писане на КОД**

# 20

20 % от живота в  
имплементация

# 80

80 % от живота в  
поддръжка

- *Пишете добре структуриран код.*
- *Пишете код, който лесно да се реструктурира.*

# 20

# 80

само 20 % от  
написалите кода ще  
участват в  
поддръжката

- *Пишете кода за „другите след вас“.*

# 20

20 % от метода е  
същинската работа

# 80

80 % от метода е  
трансформации и  
проверки

- *Пишете „къси“ методи*
- *Пишете методи с „малко“ аргументи*

# 20

# 80

- *Къс метод – 7 реда*

```
public void method(int param) {  
    1  проверка/инициализация  
    2  проверка/инициализация  
    3  проверка/инициализация  
    4  работа  
    5  проверка/трансформация  
    6  проверка/трансформация  
    7  проверка/трансформация  
}
```

# 20

# 80

- *Къс метод – 10 реда*

```
public void method(int param) {  
    try {  
        1  проверка/инициализация  
        2  проверка/инициализация  
        3  проверка/инициализация  
        4  работа  
        5  проверка/трансформация  
        6  проверка/трансформация  
        7  проверка/трансформация  
    } catch (Exception e) {  
        logger.error(e);  
    }  
}
```

# 20

# 80

- *Колкото повече са аргументите толкова по често ще се добавят нови или махат стари*
- *До 4(5) аргумента*

```
public void method(int param1, int param2, int  
param3, int param4) {  
  
}
```

# 20

# 80

- *Над 4(5) аргумента – създайте нов клас*

```
public void method(ParamsHolder paramsHolder) {  
  
}
```

```
public class ParamsHolder {  
    public int getParam1() {return param1;}  
    public int getParam2() {return param2;}  
    ...  
    public int getParamN() {return paramN;}  
}
```



# 80

# 20

80 % от стойността

идва от 20 % от  
кода

- *„The simplest thing that could possibly work, but not simpler“*
- *Не пишете повече от необходимото*
- *Намалете излишното*

# 80

в 80 % от времето

# 20

се изпълнява 20%  
от кода

- *Подобрейте производителността първо на тези 20 %*

80

20

80 % методите на обект са скрити за другите обекти

20 % от методите са достъпни за външни обекти

- *Ако клас съдържа много public методи разделете класа на два*

# 100

- *Не поставяйте „много“ коментари*
- *Кодът трябва да говори сам за себе си*
- *Коментарът не трябва да повтаря кода*
- *Ако трябва да сложите коментар опитайте се да преработите кода, така че да няма нужда от коментар*

# 100

- *Коментарът не трябва да повтаря кода.  
Не правете това:*

```
public void method(int param) {  
    // check if param is less than 4  
    if(param < 4)  
        createFibonacciEvaluator();  
}
```

# 100

- *Кодът трябва да говори сам за себе си. Не правете това:*

```
public void method(int param) {  
    for(int i =0; i<10; i++) {  
        for(int k =0; k<10; k++) {  
            if(k == i) {  
  
                } else if(k== i-1) {  
                    while(k<4) {  
                        }// end of while  
                    }// end of if  
                }// end of for  
            }// end of for  
        }  
    }  
}
```

# 100

- *Кодът трябва да следва приета конвенция*

```
public class GraphicalEditor //capital letter for class
{
    private Button rectangleButton; //lower case for fields
    private static String BUTTON_NAME; //upper case for
static
    public void createEditor(Composite parent) {
//lower case for methods
        Composite composite = ...;
        createCanvas(composite);
    }
}
```

# 100

- *Кодът трябва да следва приета конвенция*

```
public class GraphicalEditor {  
  
    public Color getColor() {...} //get prefix  
  
    public void setColor(Color color) {...} //set prefix  
  
    public boolean isColorSet() {...} //is prefix for method  
    returning boolean
```



# 100

Всеки обект върши само една точно определена задача.

- *Ако метод не използва вътрешното състояние на обекта, то този метод не трябва да е в този клас*

# 100

100 % от кода трябва да има тестове