

- "Clean code always looks like it was written by someone who cares."
- ~ Robert C. Martin

VARIABLES

"Later equals never."

~ Robert C. Martin

Common mistakes

```
int a = 0;
String b = "abc";
var XYZControllerForEfficientHandlingOfStrings = "";
var XYZControllerForEfficientStorageOfStrings = "";
// 3.
var timeHHmmSS = "12:04:40";
```

Good practices

Simple, short and to the point!

```
int middleItemFromList;
var ageInYears = 21;
String dogName = "Reksio";
```

FUNCTIONS

The first rule of functions is that they should be small.

```
private static void foo(List <Integer> list) {
     int countTwo = 0;
     for (int \underline{i} = 0; \underline{i} < list.size(); \underline{i} + +) {
          sum += list.get(i);
     for (int \underline{i} = 0; \underline{i} < list.size(); \underline{i} + +) {
          amount += 1;
     for (int i = 0; i < list.size(); <math>i++) {
          if (list.get(<u>i</u>) == 2) {
                countTwo += 1;
```

Good practices in functions

One function = One task

Correct naming

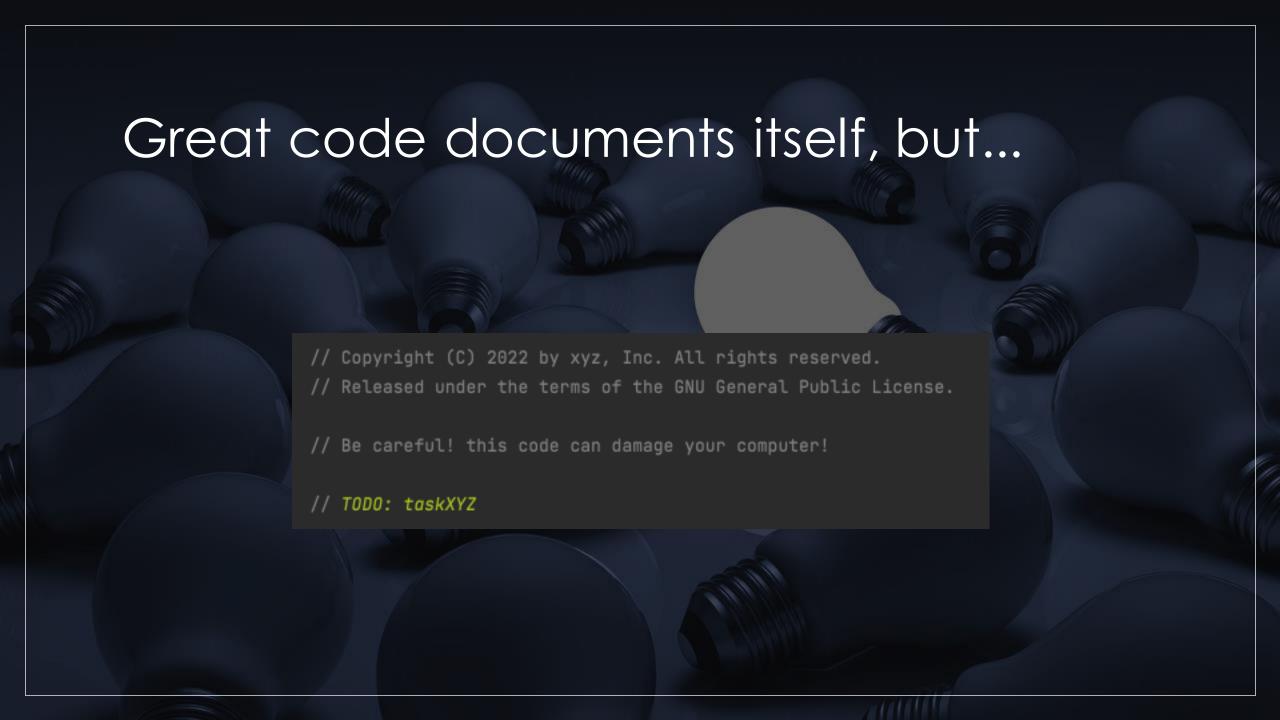
The more arguments the worse.

```
private static void sumAmountAndCountN(List<Integer> list, int n) {
    int sum = sum(list);
    int amount = amount(list);
    int countN = countN(list, n: 2);
orivate static int <mark>sum(</mark>List<Integer> list) {
    for (int \underline{i} = 0; \underline{i} < list.size(); \underline{i} + +) {
         sum += list.get(i);
    return sum;
orivate static int amount(List<Integer> list) {
    int amount = 0;
    for (int \underline{i} = 0; \underline{i} < list.size(); \underline{i} + +) {
         amount += 1;
    return amount:
private static int countN(List<Integer> list, int n) {
    int result = 0;
    for (int \underline{i} = 0; \underline{i} < list.size(); \underline{i} + +) {
         if (list.get(<u>i</u>) == n) {
              result += 1;
    return result;
```

COMMENTS

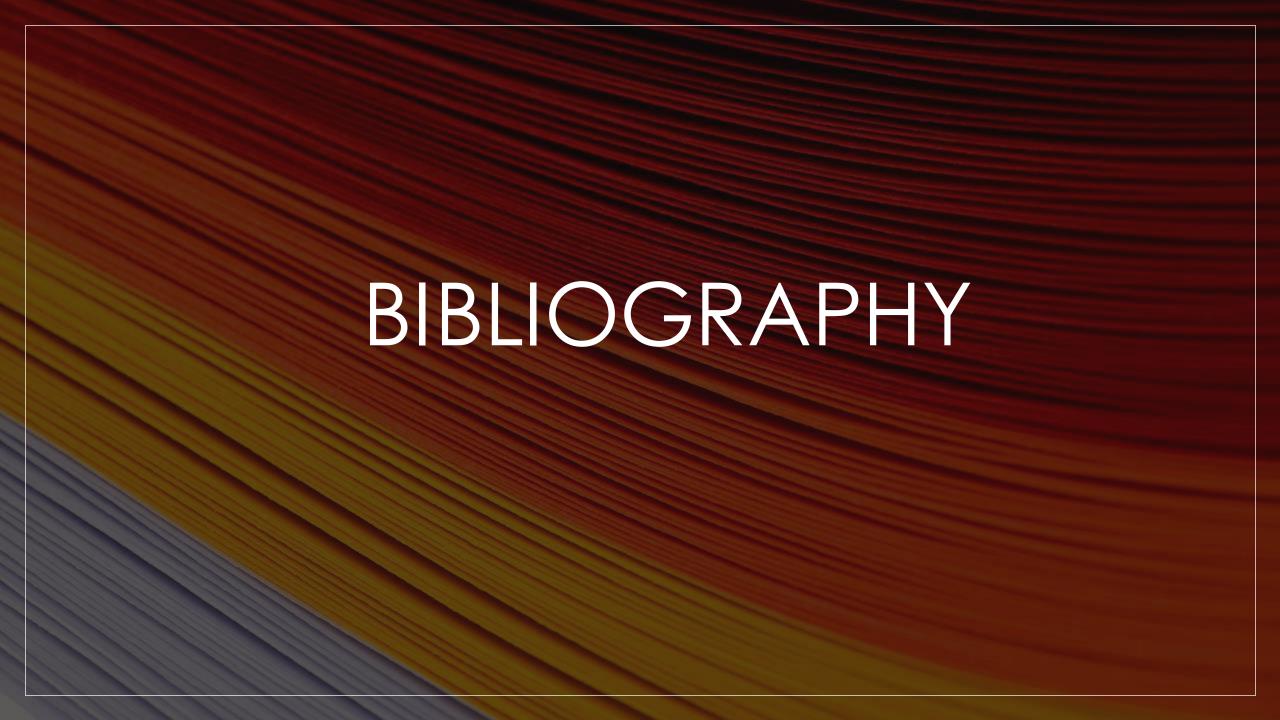
"Don't comment bad code - rewrite it."

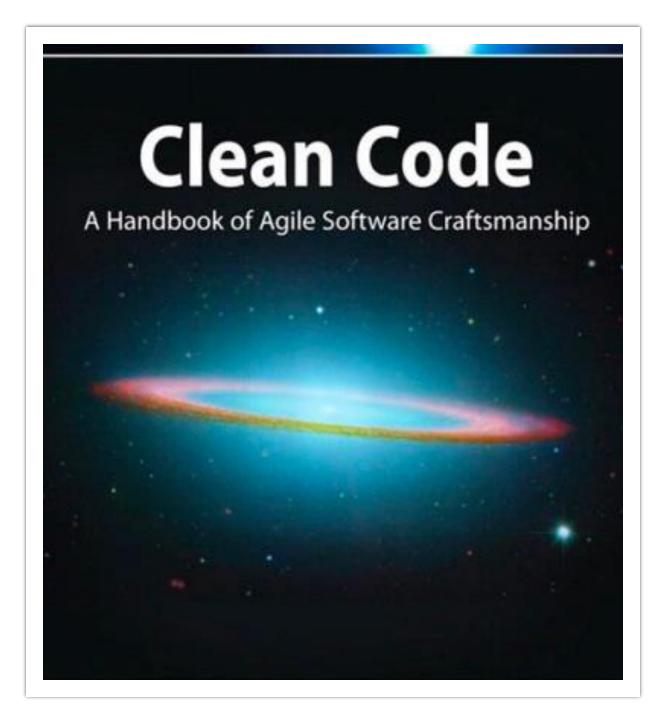
~ Brian W. Kernighan and P. J. Plaugher



KEP SHORT and SIMPLE

Summary





"Clean Code. A Handbook of Agile Software Craftsmanship"



```
peration == "MIRROR_X":
mirror_mod.use_x = True
mlrror_mod.use_y = False
drror mod.use z = Fal
___rror_mod.use_x = False
mirror_mod.use_y = True
"Irror_mod.use_z = False
  operation == "MIRROR Z"
  rror mod.use x = False
 HANK YOU FOR
            RATTENTION!
  ata.objects[one.name].se
  int("please select exactle
  -- OPERATOR CLASSES ----
```

x mirror to the select
ject.mirror_mirror_x"
ror X"

are not