

Partea I: Exemple punctuale

(Sursă capturi ecran: <https://github.com/leonardolemie/clean-code-java>)

1. Denumiri variabile – sugestive si explicite

Bad:

```
String yyyyymmddstr = new SimpleDateFormat("YYYY/MM/DD").format(new Date());
```

Good:

```
String currentDate = new SimpleDateFormat("YYYY/MM/DD").format(new Date());
```

2. Evitare “mental mapping”

Don't force the reader of your code to translate what the variable means. Explicit is better than implicit. **Bad:**

```
String [] l = {"Austin", "New York", "San Francisco"};

for (int i = 0; i < l.length; i++) {
    String li = l[i];
    doStuff();
    doSomeOtherStuff();
    // ...
    // ...
    // ...
    // Wait, what is `li` for again?
    dispatch(li);
}
```

Good:

```
String[] locations = {"Austin", "New York", "San Francisco"};

for (String location : locations) {
    doStuff();
    doSomeOtherStuff();
    // ...
    // ...
    // ...
    dispatch(location);
}
```

3. Denumiri funcții/metode - consecvență

Bad:

```
getUserInfo();  
getClientData();  
getCustomerRecord();
```

Good:

```
getUser();
```

4. Valori constante (ușor de căutat)

Bad:

```
// What the heck is 86400000 for?  
setTimeout(blastOff, 86400000);
```

Good:

```
// Declare them as capitalized `const` globals.  
public static final int MILLISECONDS_IN_A_DAY = 86400000;  
  
setTimeout(blastOff, MILLISECONDS_IN_A_DAY);
```

5. Variabile ca parametru (in funcții/metode)

Bad:

```
String address = "One Infinite Loop, Cupertino 95014";  
String cityZipCodeRegex = "/^[^,\\\\\\\\]+[,\\\\\\\\\\\\\\\\s]+(\\.+?)\\\\s*(\\\\d{5})?$/";  
  
saveCityZipCode(address.split(cityZipCodeRegex)[0],  
                address.split(cityZipCodeRegex)[1]);
```

Good:

```
String address = "One Infinite Loop, Cupertino 95014";  
String cityZipCodeRegex = "/^[^,\\\\\\\\\\\\\\\\s]+[,\\\\\\\\\\\\\\\\\\\\s]+(\\.+?)\\\\s*(\\\\d{5})?$/";  
  
String city = address.split(cityZipCodeRegex)[0];  
String zipCode = address.split(cityZipCodeRegex)[1];  
  
saveCityZipCode(city, zipCode);
```

6. Denumire variabile membre - fără numele clasei

Bad:

```
class Car {  
    public String carMake = "Honda";  
    public String carModel = "Accord";  
    public String carColor = "Blue";  
}  
  
void paintCar(Car car) {  
    car.carColor = "Red";  
}
```

Good:

```
class Car {  
    public String make = "Honda";  
    public String model = "Accord";  
    public String color = "Blue";  
}  
  
void paintCar(Car car) {  
    car.color = "Red";  
}
```

7. Funcții/metode separate pentru acțiuni diferite

Bad:

```
public void emailClients(List<Client> clients) {  
    for (Client client : clients) {  
        Client clientRecord = repository.findOne(client.getId());  
        if (clientRecord.isActive()){  
            email(client);  
        }  
    }  
}
```

Good:

```
public void emailClients(List<Client> clients) {  
    for (Client client : clients) {  
        if (isActiveClient(client)) {  
            email(client);  
        }  
    }  
}  
  
private boolean isActiveClient(Client client) {  
    Client clientRecord = repository.findOne(client.getId());  
    return clientRecord.isActive();  
}
```

8. Denumirea metodelor - spun ceea ce fac

Bad:

```
private void addToDate(Date date, int month){  
    //..  
}  
  
Date date = new Date();  
  
// It's hard to tell from the method name what is added  
addToDate(date, 1);
```

Good:

```
private void addMonthToDate(Date date, int month){  
    //..  
}  
  
Date date = new Date();  
addMonthToDate(1, date);
```

9. Flag-uri pentru funcționalități multiple

Bad:

```
public int operation(int opType, int val1, int val2)  
{  
    switch (opType)  
    {  
        case 1: return val1+val2;  
        case 2: return val1-val2;  
        case 3: return val1*val2;  
        case 4: return val1/val2;  
        default: return 0;  
    }  
}
```

Good:

```
public int addValues(int value1, int value2)  
{  
    return value1+value2;  
}  
  
public int subtractValues(int value1, int value2)  
{  
    return value1-value2;  
}
```

10. Fără cod duplicat (evitare copy/paste)

Partea II: Exemplu proiect complet

- *Codul sursă inițial - în fișierul InitialSourceCode.txt*
- *Diagrama inițială a claselor - diagram.png*