

Clean Code

What is Clean Code?

- Can be read and enhanced by a developer other than its original author
- Has unit and acceptance tests
- Does one thing well
- Looks like it was written by someone who cares
- Never obscures the designer's intent
- Each routine you read turns out to be pretty much what you expected
- Reads like well written prose
- Provides a clear and minimal API
- Elegant & efficient

Names

Intent

```
// ...  
player1.setmenOCom(1);  
player2.setmenOCom(0);  
// ...
```

🦇 *Can you figure out what these functions are supposed to do?*

The "Batman Mode" metaphor (🦇)

When there is a mystery or crime to be solved, Batman will utilize his brain and all kinds of fancy gadgets to get it done! He will analyze, investigate and deduce until he has the answer. For him as a *costumed Super Hero Detective* it's part of the job! *Software engineers* should **never** have to go into Batman Mode to investigate about names used in the code!

Clarifying (!?) declaration

```
public void setmenOCom(int a) {  
    this.menOCom = a;  
}
```

Comment () to the rescue

```
/**
 * setzt menOCom, 0 = Mensch, 1 = Computer
 *
 * @param int fuer menOCom
 */
public void setmenOCom(int a) {
    this.menOCom = a;
}
```

Reveal your intent

```
import static PlayerType.*;
// ...
player1.setType(HUMAN);
player2.setType(COMPUTER);
// ...
```

```
public void setType(PlayerType type) {
    this.type = type;
}

public enum PlayerType {
    HUMAN, COMPUTER
}
```


Functions

Comments

Formatting