The importance of naming

Tien Martin Trinh

Motivation

Why should we be better at naming?

- Communication
- Reduce code smells (e.g. Comments)
- Uncover problems

Meaningless Variable Names

```
var x = 42;
var data = new List<string> { "apple", "orange",
"banana" };
```

```
var age = 42;
var fruitList = ['apple', 'orange', 'banana'];
```

Ambiguous Function Names

```
public void doSomething(a, b) {
    // ...
}
```

```
public void calculateSum(num1, num2) {
    // ...
}
```

Misleading Class Names

```
class HandleData {
    // ...
}


class DataManager {
    // ...
}
```

Inconsistent Naming Styles

```
var maxVal = 100;
public void calculate_value() {
    // ...
}
```

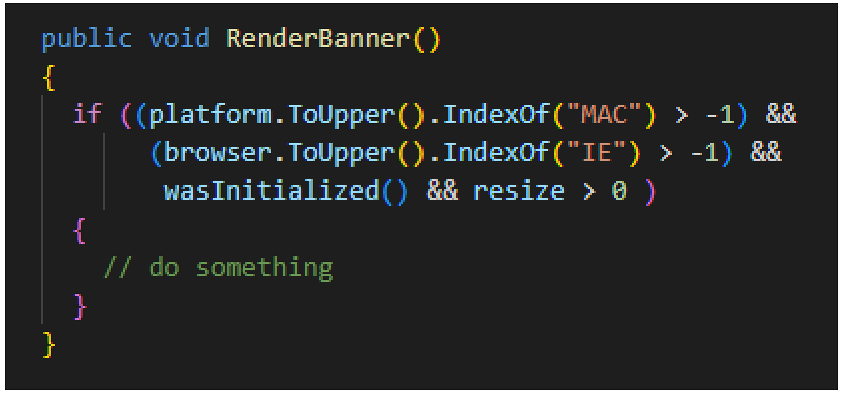
```
var maxValue = 100;
public void calculateValue() {
    // ...
}
```

Code smell: Comments

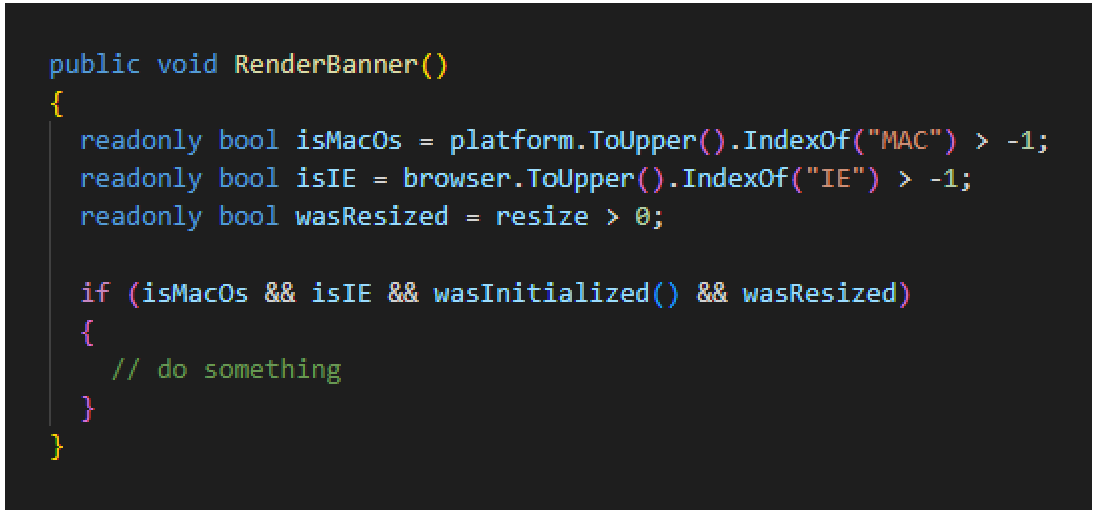
- Why do we use comments?

- How can we get rid of these comments?

If a comment is there to explain an expression that is hard to understand, the expression should be split into separate variables that are self-explanatory.



```
public void RenderBanner()
{
  if ((platform.ToUpper().IndexOf("MAC") > -1) &&
      (browser.ToUpper().IndexOf("IE") > -1) &&
       wasInitialized() && resize > 0 )
  {
    // do something
  }
}
```

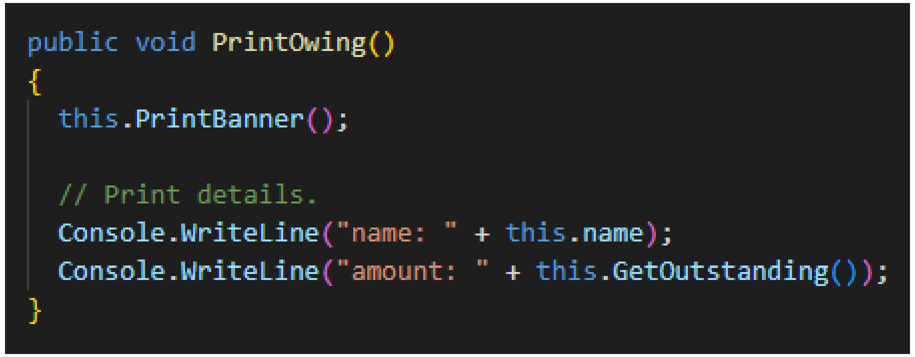


```
public void RenderBanner()
{
  readonly bool isMacOs = platform.ToUpper().IndexOf("MAC") > -1;
  readonly bool isIE = browser.ToUpper().IndexOf("IE") > -1;
  readonly bool wasResized = resize > 0;

  if (isMacOs && isIE && wasInitialized() && wasResized)
  {
    // do something
  }
}
```

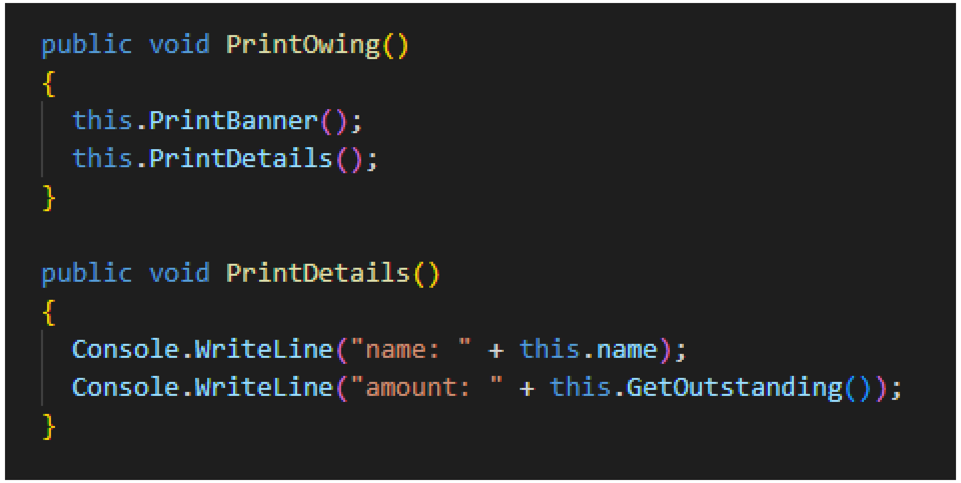
If a comment is there to explain a section of a code and it can be grouped together.

You can move the code to a separate method and replace the old code with a call to the new method.



```
public void PrintOwing()
{
  this.PrintBanner();

  // Print details.
  Console.WriteLine("name:  " + this.name);
  Console.WriteLine("amount:  " + this.GetOutstanding());
}
```



```
public void PrintOwing()
{
  this.PrintBanner();
  this.PrintDetails();
}

public void PrintDetails()
{
  Console.WriteLine("name:  " + this.name);
  Console.WriteLine("amount:  " + this.GetOutstanding());
}
```

If a method has already been extracted, but you still think comments are still necessary then you haven’t given it a good enough name.

Uncover Problems

- Naming helps with uncovering problems
  - Methods or class that are doing too much

  →Code smells

- Single responsibility

- Good naming → express intent → clear logic

What have we learned?

- Naming can be hard
- Naming is communication
- Naming can make it easier to discover problems
- Naming is a tool to help you write cleaner code

Thank you for your attention!