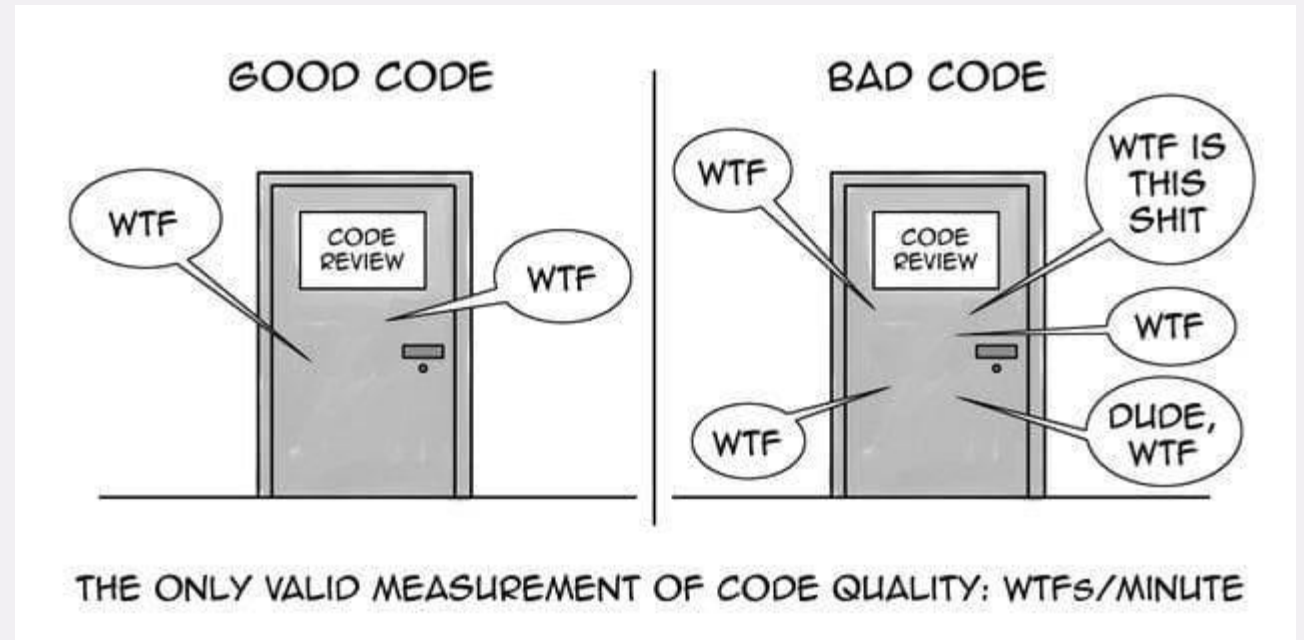




OBJECT CALISTHENICS

MOTIVATION



1. ONE LEVEL OF INDENTATION PER METHOD



r/ProgrammerHumor • vor 1 Jahr
Beano-69



Übersetzen

"NOOOOO! YOU SHOULDN'T INDENT MORE THAN 3 TIMES" Me:

Meme

```
662 // Add the buffers from each thread
663 for (uint32_t i = 0; i < numThreads; i++)
664 {
665     if (m_state.mustCancel)
666         break;
667
668     // I'm sorry for so many indentations. It drives me crazy too. I apologize.
669     if (m_threads[i])
670     {
671         std::vector<float> threadBuffer = m_threads[i]->getBuffer();
672         if (!m_state.mustCancel)
673         {
674             uint32_t redIndex;
675             for (uint32_t y = 0; y < inputHeight; y++)
676             {
677                 if (m_state.mustCancel)
678                     break;
679
680                 for (uint32_t x = 0; x < inputWidth; x++)
681                 {
682                     redIndex = (y * inputWidth + x) * 4;
683                     convBuffer[redIndex + 0] += threadBuffer[redIndex + 0] * CONV_MULTIPLIER;
684                     convBuffer[redIndex + 1] += threadBuffer[redIndex + 1] * CONV_MULTIPLIER;
685                     convBuffer[redIndex + 2] += threadBuffer[redIndex + 2] * CONV_MULTIPLIER;
686                     convBuffer[redIndex + 3] += 1.0f;
687                 }
688             }
689         }
690     }
691 }
692 }
```

2. DON'T USE THE ELSE KEYWORD

Definitely the latter. The former doesn't look bad right now, but when you get more complex code, I can't imagine anyone would think this:

```
public int SomeFunction(bool cond1, string name, int value, AuthInfo perms)
{
    int retval = SUCCESS;
    if (someCondition)
    {
        if (name != null && name != "")
        {
            if (value != 0)
            {
                if (perms.allow(name)
                {
                    // Do Something
                }
                else
                {
                    reval = PERM_DENY;
                }
            }
            else
            {
                retval = BAD_VALUE;
            }
        }
        else
        {
            retval = BAD_NAME;
        }
    }
    else
    {
        retval = BAD_COND;
    }
    return retval;
}
```

is more readable than

```
public int SomeFunction(bool cond1, string name, int value, AuthInfo perms)
{
    if (!someCondition)
        return BAD_COND;

    if (name == null || name == "")
        return BAD_NAME;

    if (value == 0)
        return BAD_VALUE;

    if (!perms.allow(name))
        return PERM_DENY;

    // Do something
    return SUCCESS;
}
```

I fully admit I never understood the advantage of single exit points.

Share

edited Jul 7, 2017 at 16:44

community wiki
6 revs, 5 users 81%
Jason Viers

3. WRAP ALL PRIMITIVES AND STRINGS

```
@TinyType
@EqualsAndHashCode
@ToString
@AllArgsConstructor(access = AccessLevel.PRIVATE)
class Foo {

    @Getter
    @NonNull
    private final String value;

    no usages
    public static Foo foo(@NonNull String value) {
        return new Foo(value);
    }
}
```

4. FIRST CLASS COLLECTIONS

```
1  import java.util.ArrayList;
2  import java.util.List;
3
4  ▼ public class WinningConditions {
5      private final List<WinningCondition> winningConditions;
6
7  ▼  public WinningConditions() {
8      winningConditions = new ArrayList<>();
9      winningConditions.add(new WinningCondition(Position.TOP_LEFT, Position.TOP_CENTER, Position.TOP_RIGHT));
10     winningConditions.add(new WinningCondition(Position.MIDDLE_LEFT, Position.MIDDLE_CENTER, Position.MIDDLE_RIGHT));
11     winningConditions.add(new WinningCondition(Position.BOTTOM_LEFT, Position.BOTTOM_CENTER, Position.BOTTOM_RIGHT));
12 }
13
14     public List<WinningCondition> getConditions() {
15         return winningConditions;
16     }
17 }
```

5. ONLY ONE DOT PER LINE

```
// bad
a.getFoo()
  .getBar()
  .getFoobar()
  .doAll();

// okay
fooBuilder.bar("value")
  .foobar("value")
  .barFoo("value")
  .build();
```

6. NO ABBREVIATIONS



Lewis Menelaws



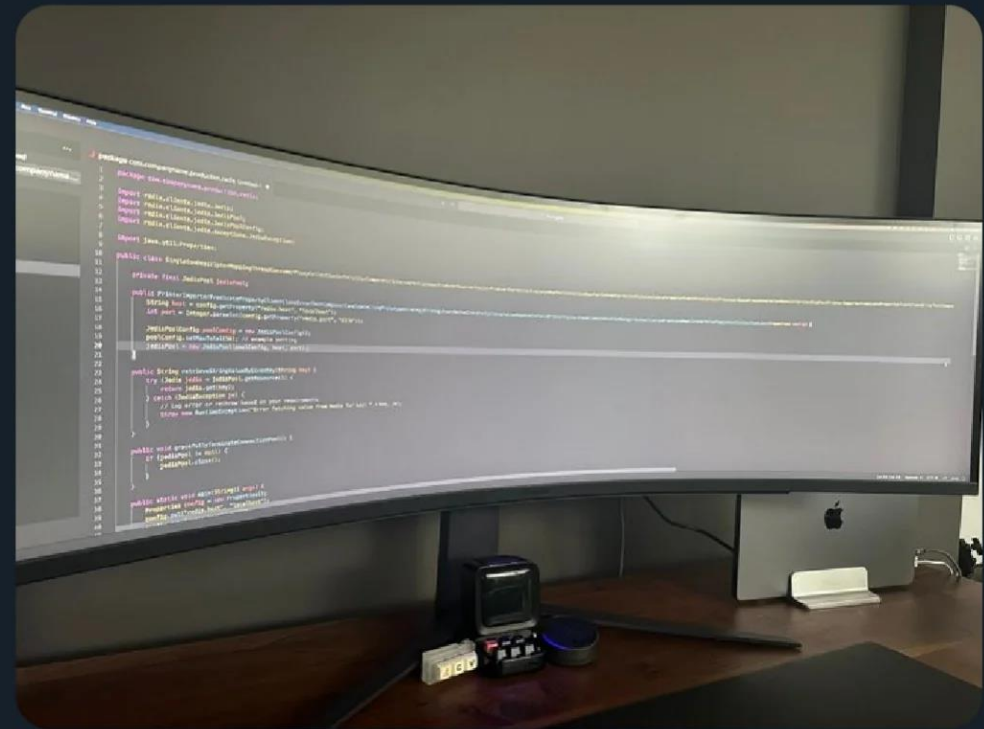
@LewisMenelaws

Follow



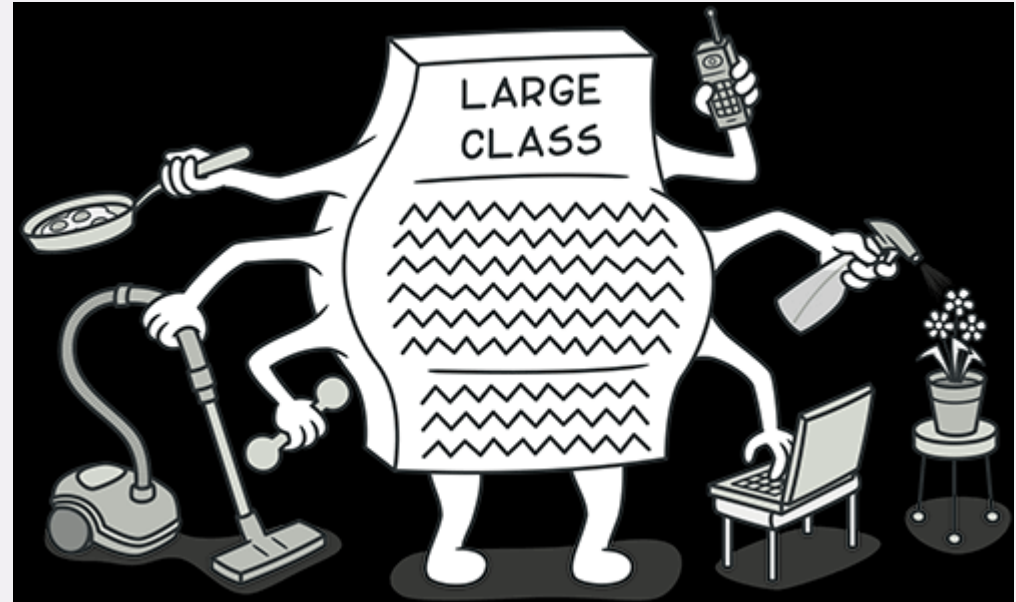
Just bought a new monitor.

Still can't fit my Java Class names in it.

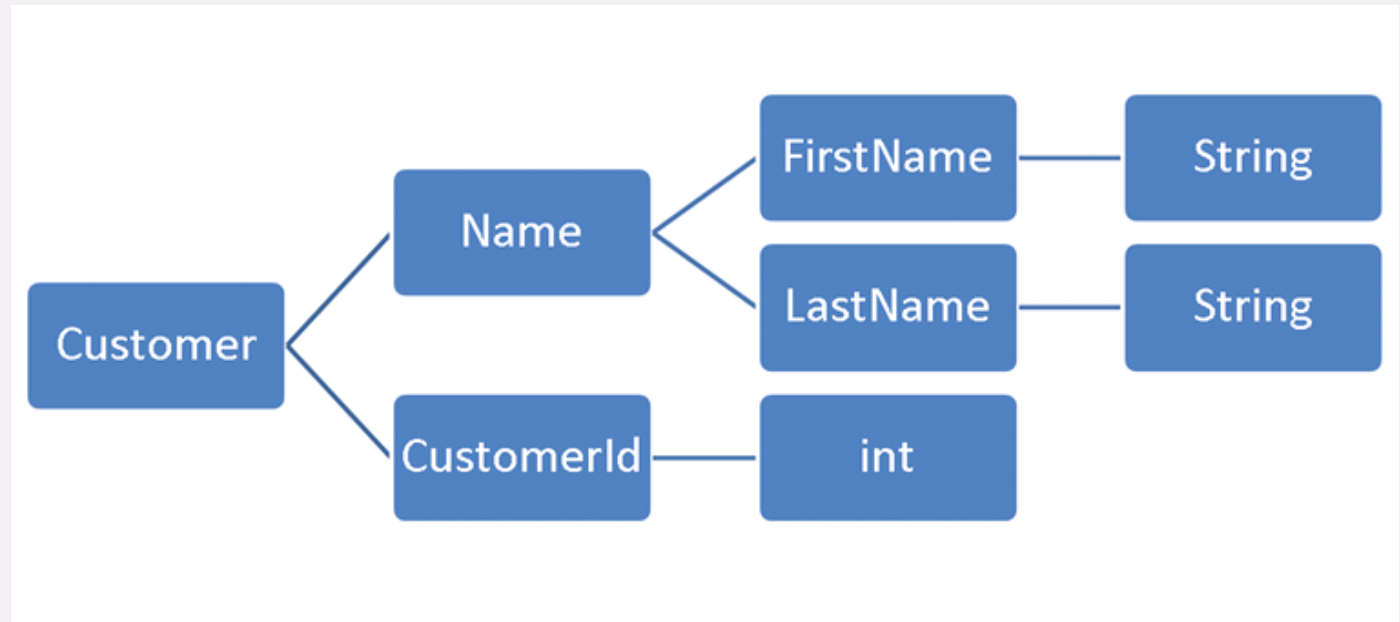


0:01 · 21 Oct 23 · 529K Views

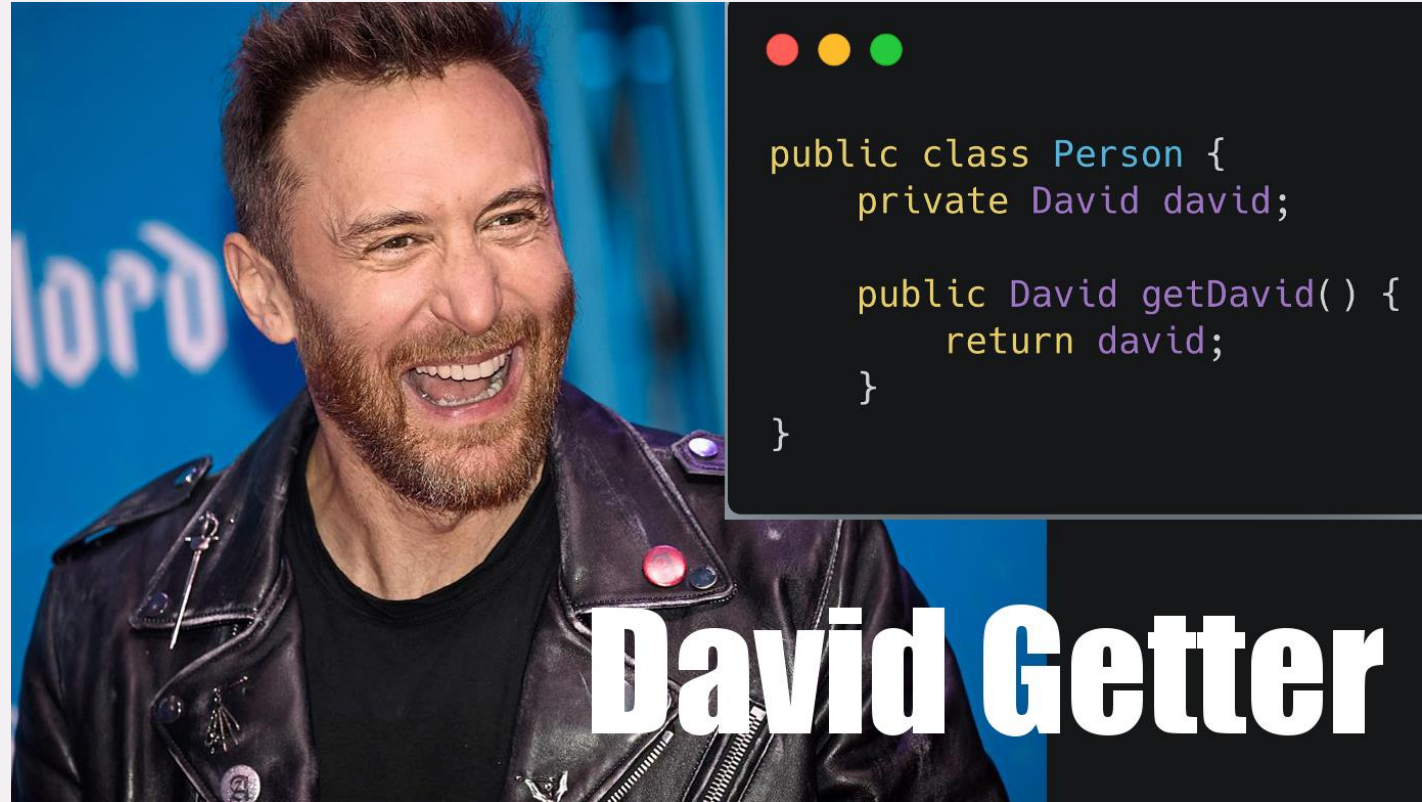
7. KEEP ALL ENTITIES SMALL



8. NO CLASSES WITH MORE THAN TWO INSTANCE VARIABLES



9. NO PUBLIC GETTERS SETTERS PROPERTIES





THANK YOU!

QUESTIONS?