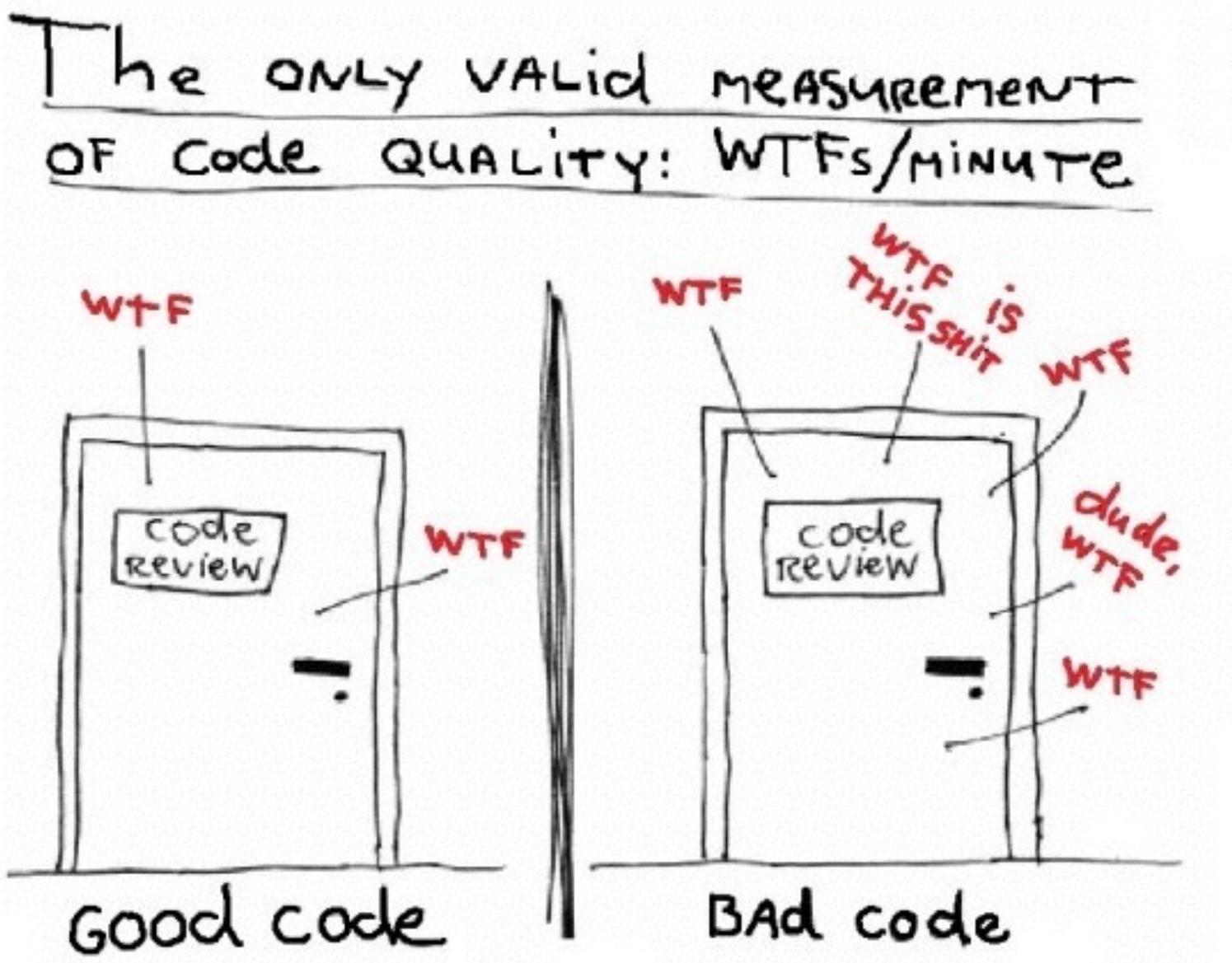
Why Coding **Conventions** ?

“a set of **guidelines** for a specific programming language that recommend **programming style**, practices and methods for each aspect of a piece program”

–Wikipedia

in most cases,
we'll **write**
in a **style** that we **want**

```
$userName $username $user_name
```


The ONLY VALID MEASUREMENT OF Code QUALITY: WTFs/MINUTE
WTF
code review
WTF
Good code
WTF
WTF
THIS SHIT
IS
WTF
code review
dude, WTF
WTF
BAd code

consistent

easier to understand

maintainable code

"coding style not because one is better than another, but because we need a standard by which to **collaborate**"

–Paul M. Jones

indentation

tabs vs spaces

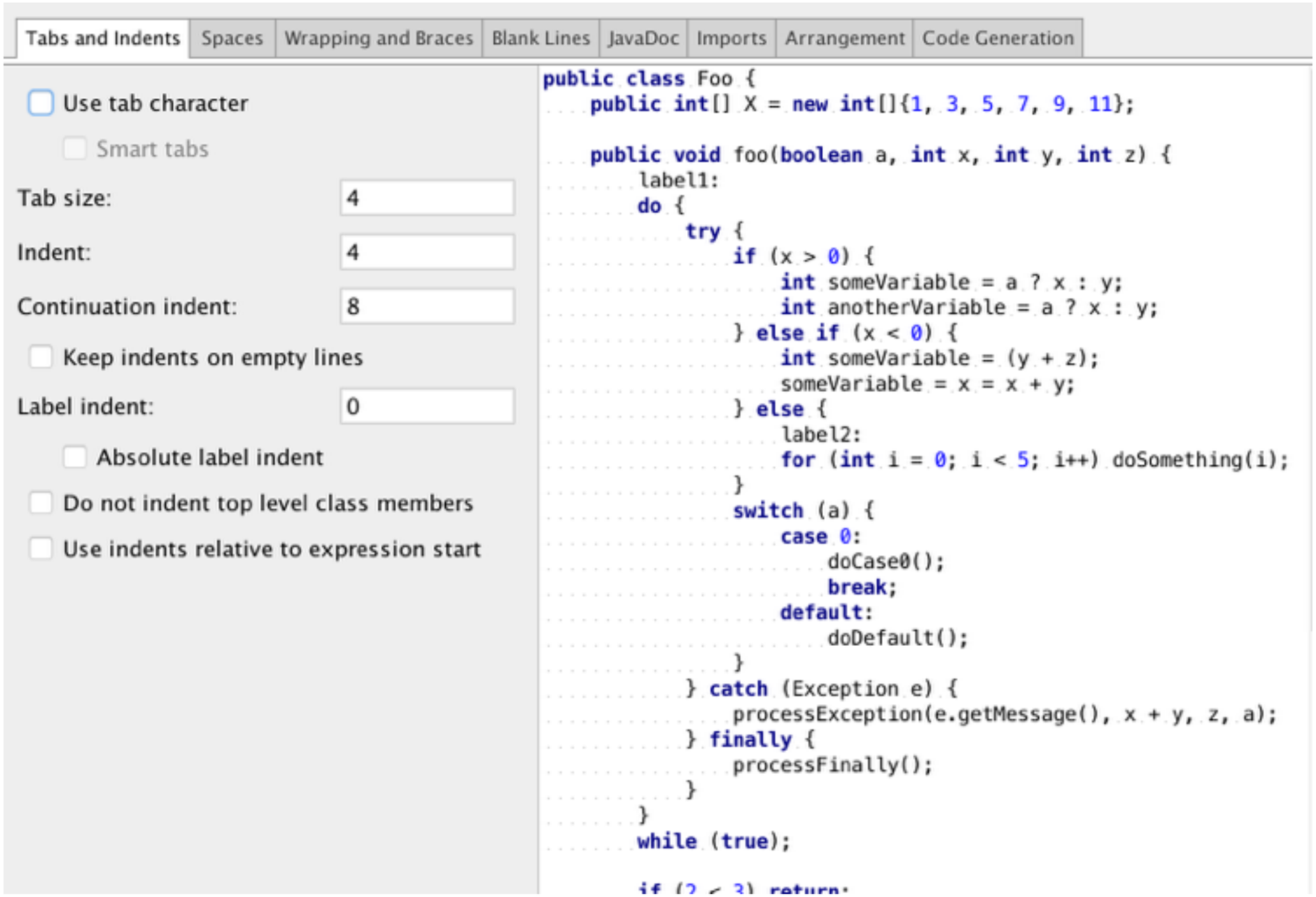
```
public class Foo {
	public int[] X = new int[]{1, 3, 5, 7, 9, 11};

	public void foo(boolean a, int x, int y, int z) {
		label1:
		do {
			try {
				if (x > 0) {
					int someVariable = a ? x : y;
					int anotherVariable = a ? x : y;
				} else if (x < 0) {
					int someVariable = (y + z);
					someVariable = x = x + y;
				} else {
					label2:
					for (int i = 0; i < 5; i++) doSomething(i);
				}
				switch (a) {
					case 0:
						doCase0();
						break;
					default:
						doDefault();
				}
			} catch (Exception e) {
				processException(e.getMessage(), x + y, z, a);
			} finally {
				processFinally();
			}
		}
		while (true);
```

```
public class Foo {
    public int[] X = new int[]{1, 3, 5, 7, 9, 11};

    public void foo(boolean a, int x, int y, int z) {
        label1:
        do {
            try {
                if (x > 0) {
                    int someVariable = a ? x : y;
                    int anotherVariable = a ? x : y;
                } else if (x < 0) {
                    int someVariable = (y + z);
                    someVariable = x = x + y;
                } else {
                    label2:
                    for (int i = 0; i < 5; i++) doSomething(i);
                }
                switch (a) {
                    case 0:
                        doCase0();
                        break;
                    default:
                        doDefault();
                }
            } catch (Exception e) {
                processException(e.getMessage(), x + y, z, a);
            } finally {
                processFinally();
            }
        }
        while (true);
```

or…hit TAB and let the IDE do the conversion for the appropriate number of space



brace style

Allman vs 1TBS

```
public class ThisIsASampleClass extends C1 implements I1, I2, I3, I4, I5
{
    private int f1 = 1;
    private String field2 = "";

    public void foo1(int i1, int i2, int i3, int i4, int i5, int i6, int i7)
    {
    }
```

```
public class ThisIsASampleClass extends C1 implements I1, I2, I3, I4, I5 {
    private int f1 = 1;
    private String field2 = "";

    public void foo1(int i1, int i2, int i3, int i4, int i5, int i6, int i7) {
    }
```

http://www.terminally-incoherent.com/blog/2009/04/10/the-only-correct-indent-style

naming convention

CamelCase vs underscore_case

`toCamelCase`

`or_to_underscore`

www.terminally-incoherent.com/blog/2009/04/10/the-only-correct-indent-style

Android Code Style Guidelines

in the end
it doesn't matter which
coding styles we pick.

what does matter is
everyone sticks with those
conventions and uses them
consistently.