Code Style

Yegor Bugayenko

Lecture #22 out of 24
80 minutes

"The harder it is for people to grasp the intent of any given section, the longer it will be before the program becomes operational. Trying to outsmart a compiler defeats much of the purpose of using one. Write clearly — don't sacrifice clarity for 'efficiency.'"
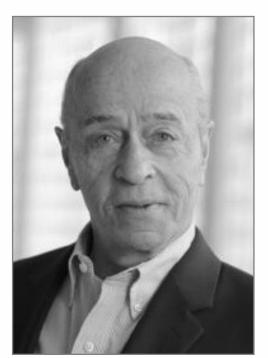
— Brian W. Kernighan and Phillip James Plauger. *The Elements of Programming Style.* McGraw-Hill, Inc., 1974

BRIAN KERNIGHAN

Which One Is Better?

```
1  int f(int n)
2  {
3    if (n == 1 || n < 2)
4      return 1;
5    int r = f (n-1);
6    int r2 = f(n - 2);
7    return r +r2;
8    }
```

```
1  int fibonacci(int n) {
2    if (n <= 2) {
3      return 1;
4    }
5    return fibonacci(n - 1)
6      + fibonacci(n - 2);
7  }
```

"An individual's body language helps clarify the spoken word. In a similar sense, the programmer relies on <u>white space</u>—what is not said directly—in the code to communicate logic, intent, and understanding."

— Robert Green and Henry Ledgard. Coding Guidelines: Finding the Art in the Science. *Communications of the ACM*, 54(12):57–63, 2011

HENRY LEDGARD

Jennifer Bauer

"While the perceptual processing of code is required to understand it, higher level processing, such as understanding its semantics and reasoning about its functionality, affect program comprehensibility more strongly. The influence of indentation could have been masked by these side effects, so it might well be that the effect of indentation comes more into play when the code is longer and more complex."

— Jennifer Bauer, Janet Siegmund, Norman Peitek, Johannes C. Hofmeister, and Sven Apel. Indentation: Simply a Matter of Style or Support for Program Comprehension? In *Proceedings of the 27th International Conference on Program Comprehension (ICPC)*, pages 154–164. IEEE, 2019

Figure 9. Examples of K&R, ANSI, and Whitesmiths coding styles.

```
if (expression) {
    statements
}
```

```
if (expression)
{
    statements
}
```

```
if (expression)
{
    statements
}
```

Source: Robert Green and Henry Ledgard. Coding Guidelines: Finding the Art in the Science. *Communications of the ACM*, 54(12):57–63, 2011

My Favorite Style Checkers

- ESLint for JavaScript

- Clang-Tidy for C++

- Pylint for Python

- Rubocop for Ruby

- PHP_CodeSniffer for PHP

- rustfmt for Rust

- Qulice by [Bugayenko, 2014] for Java (Checkstyle + PMD)

How Many Rules in Style Checkers?

- 690+ in Clang-Tidy (C++)

- 550+ in Rubocop (Ruby)

- 400+ in PMD (Java)

- 130+ in Checkstyle (Java)

- 120+ in Pylint (Python)

Some/most of the rules no only check style, but also find bugs

Some Exotic Style Checkers

- Shellcheck for Bash

- markdownlint for Markdown

- Checkmake for Makefile

- xcop for XML

References

Jennifer Bauer, Janet Siegmund, Norman Peitek, Johannes C. Hofmeister, and Sven Apel. Indentation: Simply a Matter of Style or Support for Program Comprehension? In *Proceedings of the 27th International Conference on Program Comprehension (ICPC)*, pages 154–164. IEEE, 2019.

Yegor Bugayenko. Strict Control of Java Code Quality. `https://www.yegor256.com/140813.html`, August 2014. [Online; accessed 26-02-2024].

Robert Green and Henry Ledgard. Coding Guidelines: Finding the Art in the Science. *Communications of the ACM*, 54(12):57–63, 2011.

Brian W. Kernighan and Phillip James Plauger. *The Elements of Programming Style*. McGraw-Hill, Inc., 1974.