WIKIPEDIA

Boilerplate code

In computer programming, **boilerplate code** or just **boilerplate** are sections of code that are repeated in multiple places with little to no variation. When using languages that are considered *verbose*, the programmer must write a lot of code to accomplish only minor functionality. Such code is called *boilerplate*.[1]

The need for boilerplate can be reduced through high-level mechanisms such as metaprogramming (which has the computer automatically write the needed boilerplate code or insert it at compile time), convention over configuration (which provides good default values, reducing the need to specify program details in every project) and model-driven engineering (which uses models and model-to-code generators, eliminating the need for manual boilerplate code).

Contents

Origin

The term arose from the newspaper business. Columns and other pieces that were distributed by print syndicates were sent to subscribing newspapers in the form of prepared printing plates. Because of their resemblance to the metal plates used in the making of boilers, they became known as "boiler plates", and their resulting text - "boilerplate text". As the stories that were distributed by boiler plates were usually "fillers" rather than "serious" news, the term became synonymous with unoriginal, repeated text.[2][3]

A related term is *bookkeeping code*, referring to code that is not part of the business logic but is interleaved with it in order to keep data structures updated or handle secondary aspects of the program.

Preamble

One form of boilerplate consists of declarations which, while not part of the program logic or the language's essential syntax, are added to the start of a source file as a matter of custom. The following Perl example demonstrates boilerplate:

```perl
#!/usr/bin/perl
use warnings;
use strict;
```

The first line is a shebang, which identifies the file as a Perl script that can be executed directly on the command line (on Unix/Linux systems.) The other two are pragmas turning on warnings and strict mode, which are mandated by fashionable Perl programming style.

This next example is a C/C++ programming language boilerplate, `#include` guard.

```
#ifndef MYINTERFACE_H
#define MYINTERFACE_H

...

#endif
```

This checks, and sets up, a global flag to tell the compiler whether the file myinterface.h has already been included. As many interdepending files may be involved in the compilation of a module, this avoids processing the same header multiple times (which would lead to errors due to multiple definitions with the same name).

In object-oriented programming

In object-oriented programs, classes are often provided with methods for getting and setting instance variables. The definitions of these methods can frequently be regarded as boilerplate. Although the code will vary from one class to another, it is sufficiently stereotypical in structure that it would be better generated automatically than written by hand. For example, in the following Java class representing a pet, almost all the code is boilerplate except for the declarations of *Pet*, *name*, and *owner*:

```
public class Pet {
    private String name;
    private Person owner;

    public Pet(String name, Person owner) {
        this.name = name;
        this.owner = owner;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Person getOwner() {
        return owner;
    }

    public void setOwner(Person owner) {
        this.owner = owner;
    }
}
```

Most of the boilerplate in this example exists to provide encapsulation. If the variables **name** and **owner** were declared as public, the accessor and mutator methods would not be needed.

To reduce the amount of boilerplate, many frameworks have been developed, e.g. Lombok for Java.[4] The same code as above is auto-generated by Lombok using Java annotations, which is a form of metaprogramming:

```
@AllArgsConstructor
@Getter
@Setter
public class Pet {
    private String name;
    private Person owner;
}
```

In some other programming languages it may be possible to achieve the same thing with less boilerplate, when the language has built-in support for such common constructs. For example, the equivalent of the above Java code can be expressed in Scala using just one line of code:

```
case class Pet(var name: String, var owner: Person)
```

Or in C# using Automatic Properties with compiler generated backing fields:

```
public class Pet
{
    public string Name { get; set; }
    public Person Owner { get; set; }
}
```

Method boilerplate

In addition to declarations, methods in OOP languages also contribute to the amount of boilerplate. A 2015 study on popular Java projects shows that 60% of methods can be uniquely identified by the occurrence of 4.6% of its tokens, making the remaining 95.4% boilerplate irrelevant to logic. The researchers believe this result would translate to subroutines in procedual languages in general.[5]

HTML

In HTML, the following boilerplate is used as a basic empty template and is present in most web pages:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8"/>
  <title></title>
</head>
<body>

</body>
</html>
```

The WHATWG HTML Living Standard defines that the `<html>`, `<head>` and `<body>` tags may be safely omitted under most circumstances.[6] The `<meta charset="UTF-8">` tag may also be omitted if the Web server is properly configured to send the character encoding along with the content type.[7] Google's HTML/CSS style guide recommends that all optional tags be omitted,[8] resulting in a much more compact boilerplate:

```
<!DOCTYPE html>
<title></title>
```

See also

- Directive (programming) – Language construct that specifies how a compiler should process its input
- General-purpose macro processor – Macro processor that is not tied to or integrated with a particular language or piece of software.
- "Hello, World!" program – Traditional beginners' computer program
- Library (computing) – Collection of non-volatile resources used by computer programs, often for software development.
- Macro (computer science) – In computer science, a concise representation of a pattern
- Preprocessor – Program which processes the input for another program
- Snippet (programming) – A small region of re-usable source code, machine code, or text
- Template processor – Software designed to combine templates with a data model to produce result documents
- Web template system – System in web publishing that lets web designers and developers work with web templates to automatically generate custom web pages
- Scaffold (programming) – A code generation technique or a project generation technique

References

1. Lämmel, Ralf; Jones, Simon Peyton (2003). "Scrap your boilerplate: a practical design pattern for generic programming" (https://www.researchgate.net/publication/221282345). *Proceedings of the 2003 ACM SIGPLAN International Workshop on Types in Languages Design and Implementation*. TLDI '03. New York: ACM. pp. 26–37. doi:10.1145/604174.604179 (https://doi.org/10.1145%2F604174.604179). ISBN 9781581136494. S2CID 9472305 (https://api.semanticscholar.org/CorpusID:9472305).
2. "Boilerplate" (https://www.dictionary.com/browse/boilerplate). *Dictionary.com*. Retrieved 2018-01-27.
3. "Boilerplate" (https://www.merriam-webster.com/dictionary/boilerplate). *Merriam-Webster*. Retrieved 2018-01-27.
4. Frankel, Nicolas (2009-12-07). "Lombok reduces your boilerplate code" (https://dzone.com/articles/lombok-reduces-your). *DZone.com*. Retrieved 2017-08-02.
5. Martin Velez; Dong Qiu; You Zhou; Earl T. Barr; Zhendong Su (5 Feb 2015). "On the Lexical Distinguishability of Source Code [was: A Study of "Wheat" and "Chaff" in Source Code]" (https://www.computerworld.com/article/2881655/your-code-is-far-more-chaff-than-wheat.html). arXiv:1502.01410 (https://arxiv.org/abs/1502.01410) [cs (https://arxiv.org/archive/cs)].
6. "HTML Standard - The HTML syntax - Optional tags" (https://html.spec.whatwg.org/multipage/syntax.html#syntax-tag-omission). *WHATWG*. 2017-05-05. Retrieved 2017-05-05.
7. "Is the charset meta tag required with HTML5?" (https://stackoverflow.com/q/14669352). *stackoverflow.com*. Retrieved 2017-05-05.
8. "Google HTML/CSS Style Guide" (https://google.github.io/styleguide/htmlcssguide.html#Optional_Tags). *google.github.io*. Retrieved 2017-05-05.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Boilerplate_code&oldid=1010523867"

This page was last edited on 5 March 2021, at 22:51 (UTC).