

Boilerplate-Code

In der Softwareentwicklung ist **Boilerplate-Code** (oder einfach nur Boilerplate) die Bezeichnung für Code-Segmente, die an mehreren Stellen mit geringen bis keinen Änderungen wiederholt werden. Bei der Verwendung von Sprachen, die als wortreich (*verbose*) gelten, muss der Programmierer viel Code schreiben, um nur geringfügige Funktionalitäten zu implementieren. Ein solcher Code wird als Boilerplate bezeichnet.^[1]

Der Bedarf an Boilerplate-Code kann durch übergeordnete Mechanismen wie Metaprogrammierung (bei der der Computer den erforderlichen Boilerplate-Code automatisch schreibt oder beim Kompilieren einfügt), Konvention über Konfiguration (die gute Standardwerte liefert, und die Notwendigkeit der Programm-Details in jedem Projekt verringert) und modellgetriebenes Engineering (das Modelle und Modell-zu-Code-Generatoren verwendet, sodass kein manueller Code für das Boilerplate erforderlich ist) verringert werden.

Inhaltsverzeichnis

Begriff

Vorwort

In objekt-orientierter Programmierung (OOP)

Methoden-Boilerplate

HTML

Minimierung von Boilerplate-Code in Java, C# und Kotlin

Einzelnachweise

Begriff

Der Begriff entstand aus dem Zeitungsdruck. Spalten und andere Stücke, die von Druckkonsortien verteilt wurden, wurden in Form von vorbereiteten Druckplatten an abonnierende Zeitungen geschickt. Aufgrund ihrer Ähnlichkeit mit den bei der Herstellung von Kesseln verwendeten Metallplatten wurden sie als „Kesselplatten“ und der daraus resultierende Text „Kesselplatten-Text“ bekannt. Da die Geschichten, die von Kesselplatten verbreitet wurden, normalerweise eher „Füllstoffe“ als „ernsthafte“ Nachrichten waren, wurde der Begriff zum Synonym für unoriginalen, wiederholten Text.^{[2][3]}

Ein verwandter Begriff ist Buchhaltungscode, der sich auf Code bezieht, der nicht Teil der Geschäftslogik ist, aber mit diesem verschachtelt ist, um Datenstrukturen auf dem neuesten Stand zu halten oder sekundäre Aspekte des Programms zu behandeln.

Vorwort

Eine Form von Boilerplate-Code besteht aus Deklarationen, die zwar nicht Teil der Programmlogik oder der wesentlichen Syntax der Sprache sind, jedoch benutzerdefiniert am

Anfang einer Quelltextdatei hinzugefügt werden. Das folgende Perl-Beispiel zeigt das Boilerplate:

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

Die erste Zeile ist ein Shebang, der die Datei als Perl-Skript identifiziert, das direkt in der Befehlszeile (auf Unix / Linux-Systemen) ausgeführt werden kann. Die anderen beiden sind Compiler-Anweisungen, die Warnungen aktivieren, und der strenge Modus, die von modischem Perl vorgeschrieben werden Programmierstil. Dieses Beispiel ist ein C/C++ Programmiersprache boilerplate, #include Include-Guard.

```
#ifndef MYINTERFACE_H
#define MYINTERFACE_H

...

#endif
```

Dadurch wird ein globales Flag überprüft und eingerichtet, um dem Compiler mitzuteilen, ob die Datei myinterface.h bereits enthalten ist. Da möglicherweise viele voneinander abhängige Dateien an der Kompilierung eines Moduls beteiligt sind, wird vermieden, dass derselbe Header mehrmals verarbeitet wird (was aufgrund mehrerer Definitionen mit demselben Namen zu Fehlern führen würde).

In objekt-orientierter Programmierung (OOP)

In objektorientierten Programmen werden Klassen häufig mit Methoden zum Abrufen und Festlegen von Instanzvariablen bereitgestellt. Die Definitionen dieser Methoden können häufig als Boilerplate angesehen werden. Obwohl der Code von Klasse zu Klasse unterschiedlich ist, ist seine Struktur ausreichend stereotyp, sodass er besser automatisch generiert als von Hand geschrieben wird. In der folgenden Java-Klasse, die ein Haustier darstellt, ist beispielsweise fast der gesamte Code mit Ausnahme der Deklarationen von Haustier, Name und Eigentümer ein Boilerplate:

```
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;
    }
}
```

```
}
```

Der größte Teil der Boilerplate-Code in diesem Beispiel dient zur Kapselung. Wenn die Variablen **name** und der **owner** als öffentlich deklariert wären, wären die **set**- und **get**-Methoden nicht erforderlich. Um die Menge an Boilerplate-Code zu reduzieren, wurden viele Frameworks entwickelt, z. B. Lombok für Java.^[4] Der gleiche Code wie oben wird von Lombok mithilfe von Java-Annotations automatisch generiert, welche eine Art Metaprogrammierung ist:

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

In einigen anderen Programmiersprachen ist es möglicherweise möglich, dasselbe mit weniger Boilerplate-Code zu erreichen, wenn die Sprache solche gängigen Konstrukte unterstützt. Zum Beispiel kann das Äquivalent des obigen Java-Code in Scala mit nur einer Codezeile ausgedrückt werden:

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

Oder in C# mithilfe der automatischen Eigenschaften (*auto properties*) mit vom Compiler generierten Sicherungsfeldern:

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

Methoden-Boilerplate

Neben Deklarationen tragen auch Methoden in OOP-Sprachen zur Menge der Boilerplate-Code bei. Eine Studie aus dem Jahr 2015 zu beliebten Java-Projekten zeigt, dass 60 % der Methoden durch das Auftreten von 4,6 % ihrer Token eindeutig identifiziert werden können, sodass die verbleibenden 95,4 % der Boilerplate für die Logik irrelevant sind. Die Forscher glauben, dass dieses Ergebnis im Allgemeinen in Unterprogramme in prozedurale Programmiersprachen übersetzt werden würde.

HTML

In HTML wird der folgende Boilerplate-Code als grundlegende leere Vorlage verwendet und ist auf den meisten Webseiten vorhanden:

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

</body>
```

```
</html>
```

Der WHATWG HTML Living Standard definiert, dass die Tags `<html>`, `<head>` und `<body>` unter den meisten Umständen sicher weggelassen werden können.^[5] Das `<meta charset="UTF-8">` Tag kann auch weggelassen werden, wenn der Webserver so konfiguriert ist, dass die Zeichenkodierung zusammen mit dem Inhaltstyp gesendet wird.^[6] Der HTML/CSS-Style-Guide von Google empfiehlt, alle optionalen Tags wegzulassen^[7], was zu einem viel kompakteren Boilerplate-Code führt:

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

Minimierung von Boilerplate-Code in Java, C# und Kotlin

Verschiedene Programmiersprachen versuchen durch ihre Syntax Boilerplate-Code zu verhindern beziehungsweise zu minimieren. Hier sind drei Beispiele einer Daten-Klasse **Book** (siehe auch Transferobjekt, kurz: DTO sowie Plain Old Java Object, kurz: POJO sowie Plain old CLR object, kurz POCO).

Daten-Klasse **Book** in Java:

```
1  /** POJO-Klasse Book, bietet Getter u. Setter für: ISBN, Title, SubTitle und Author. */
2  public class Book {
3      private String m_ISBN;
4      private String m_Title;
5      private String m_SubTitle;
6      private String m_Autor;
7
8      public String getISBN() { return m_ISBN; }
9      public void setISBN(String pISBN) { m_ISBN = pISBN; }
10
11     public String getTitle() { return m_Title; }
12     public void setTitle(String pTitle) { m_Title = pTitle; }
13
14     public String getSubTitle() { return m_SubTitle; }
15     public void setSubTitle(String pSubTitle) { m_SubTitle = pSubTitle; }
16
17     public String getAutor() { return m_Autor; }
18     public void setAutor(String pAutor) { m_Autor = pAutor; }
19 }
```

Daten-Klasse **Book** in C#:

```
1  /** POCO-Klasse Book, bietet Eigenschaften (mit Getter- u. Setter-Methoden) für: ISBN, Title, SubTitle und
   Author. */
2  public class Book
3  {
4      public string ISBN { get; set; }
5      public string Title { get; set; }
6      public string SubTitle { get; set; }
7      public string Autor { get; set; }
8  }
```

Daten-Klasse **Book** in Kotlin:

```
1  /** Daten-Klasse Book: die Parameter in Konstruktor sind gleichzeitig die Eigenschaften. */  
2  data class Book(var ISBN: String, var Title: String, var SubTitle: String, var Autor: String)
```

Einzelnachweise

1. Ralf Lämmel, Simon Peyton Jones: *Scrap your boilerplate: a practical design pattern for generic programming*. In: *TLDI '03: Proceedings of the 2003 ACM SIGPLAN international workshop on Types in languages design and implementation*. ACM Press, 2003, [ISBN 978-1-58113-649-4](#), S. 26–37, doi:10.1145/604174.604179 (<https://doi.org/10.1145/604174.604179>).
2. *Boilerplate*. (<https://www.dictionary.com/browse/boilerplate>) In: *Dictionary.com*. Abgerufen am 27. Januar 2018.
3. *Boilerplate*. (<https://www.merriam-webster.com/dictionary/boilerplate>) In: *Merriam-Webster*. Abgerufen am 27. Januar 2018 (englisch).
4. Nicolas Frankel: *Lombok reduces your boilerplate code*. (<https://dzone.com/articles/lombok-reduces-your>) In: *DZone.com*. 7. Dezember 2009, abgerufen am 2. August 2017.
5. *HTML Standard – The HTML syntax – Optional tags*. (<https://html.spec.whatwg.org/multipage/syntax.html#syntax-tag-omission>) In: *WHATWG*. 5. Mai 2017, abgerufen am 5. Mai 2017.
6. *Is the charset meta tag required with HTML5?* (<https://stackoverflow.com/q/14669352>) In: *stackoverflow.com*. Abgerufen am 5. Mai 2017.
7. *Google HTML/CSS Style Guide*. (https://google.github.io/styleguide/htmlcssguide.html#Optional_Tags) In: *google.github.io*. Abgerufen am 5. Mai 2017 (englisch).

Abgerufen von „<https://de.wikipedia.org/w/index.php?title=Boilerplate-Code&oldid=229404830>“

Diese Seite wurde zuletzt am 2. Januar 2023 um 09:49 Uhr bearbeitet.

Der Text ist unter der Lizenz „Creative Commons Attribution/Share Alike“ verfügbar; Informationen zu den Urhebern und zum Lizenzstatus eingebundener Mediendateien (etwa Bilder oder Videos) können im Regelfall durch Anklicken dieser abgerufen werden. Möglicherweise unterliegen die Inhalte jeweils zusätzlichen Bedingungen. Durch die Nutzung dieser Website erklären Sie sich mit den Nutzungsbedingungen und der Datenschutzrichtlinie einverstanden.

Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.