



Dokumentation

Inhaltsverzeichnis

1	Hinweise zu AGB	1
2	Runterladen benötigter Dateien	1
3	Schnellstart Windows	1
4	Allgemeine Hinweise zum Programm	2
5	Ausgaben	3
6	Konfigurationsdatei	3
7	Programm-Ablaufplan	4
7.1	Angriff	5
7.2	Arbeiten	6
7.3	Außendienst	7
7.4	Beschreibung	7
7.5	Bank	8
7.6	Booster	8
7.7	Böse Spieler	9
7.8	Goldene Zwerge	9
7.9	Kampf	9
7.10	Minuten	10
7.11	Neustart	10
7.12	Stopp	10

8	Programm-Varianten	11
8.1	Tägliche Möglichkeiten	11
8.1.1	Tagesquiz	11
8.1.2	Rubbelos	11
8.1.3	Glücksrad	11
8.1.4	Weinkeller	11
8.1.5	Tagesspiel	12
8.2	Lottoschein	12
9	Weitere Hinweise	12
9.1	Medizin	12
9.2	Unerwartete Fehler	13
9.2.1	Bekannte Fehler	13
10	Beispielkonfigurationen	14

1 Hinweise zu AGB

Laut den Allgemeine Geschäftsbedingungen der Spielwerk GmbH¹ ist die **Benutzung** dieses Programms **verboten**. Benutzt man es trotzdem, kann man **vom Spiel ausgeschlossen** werden.

2 Runterladen benötigter Dateien

Auf der Webseite² kann man verschiedene Dateien runterladen. Java³ wird für die Ausführung benötigt. Für einen einfachen Betrieb braucht man dann nur zwei Dateien:

- BKampfBot.jar
- config.json

Die `.jar`-Datei heißt auf der Webseite, wie die aktuelle Version. Zur Zeit der Dokumentation lautet der Name `"1.1.alpha4.jar"`. Man kann diesen Namen umbenennen. Beide Dateien müssen im gleichen Verzeichnis liegen.

Die GZip-Datei mit dem Verzeichnis `src` beinhaltet den Quellcode. Diesen braucht man nur, falls man das Programm verändern möchte. Dazu sollte man beachten, dass der Code unter der Lizenz GPLv2⁴ steht.

3 Schnellstart Windows

Da schon viele Windowsnutzer beim ersten Starten scheitern, habe ich eine Zip-Datei mit allen nötigen Dateien erstellt (`schnellstartWindows.zip`). Dort ist auch eine Stapelverarbeitungsdatei (`start.bat`) enthalten. Nach dem Entpacken einfach einmal diese Datei starten. Es sollte sich ein Fenster öffnen, welches erst nach Bestätigung wieder schließt. Nach dem ersten Starten sollte die Ausgabe ungefähr so aussehen:

```
00:41 Login (meinName)
Login fehlgeschlagen. Vermutlich sind die Benutzerdaten nicht korrekt.
```

Der Login kann natürlich nicht stattfinden. Es fehlen die Logindaten in der Konfiguration. Diese Datei (`config.json`) ist zu öffnen zu die entsprechenden Einträge zu bearbeiten.

¹http://spielwerk.eu/index.php?option=com_content&view=article&id=90&yt_color=mint&Itemid=157

²<http://bundeskampf.georf.de>

³<http://www.java.com/de/download/>

⁴<http://www.gnu.org/licenses/gpl.html>

Sollte nun ein Login möglich sein, erklärt der Rest der Dokumentation, wie man den Bot einstellt. Sollte es nicht funktionieren, bitte auch zuerst dieses PDF lesen und dann beschweren.

4 Allgemeine Hinweise zum Programm

Das Programm arbeitet über die Kommandozeile. Öffnen Sie also zuerst eine Konsole und navigieren Sie in das Verzeichnis, wohin Sie die Datei geladen haben. ⁵⁶

Der allgemeine Aufruf dann sieht wie folgt aus:

```
eingabe# java -jar BKampfBot.jar
```

Ruft man das Programm nur mit der Hilfe auf, erscheint folgendes:

[illegible]

⁵Windows: http://www.carpelibrum.de/tutorials/windows_konsole.pdf

⁶Linux: <http://wiki.ubuntuusers.de/Terminal>

5 Ausgaben

Das Programm erzeugt Konsolenausgaben. Dabei schreibt es in stderr (Fehlerausgabe) und stdout (Normale Ausgabe). Es gibt nun Möglichkeiten diese Ausgaben getrennt, bzw. zusammen in eine Datei umzuleiten. Dadurch kann man mehrere Stunden zurückblicken. Die Konsole speichert nur eine bestimmte Anzahl von Zeilen, die man zurück scrollen kann. Die Ausgabeumleitung ist in Windows⁷ und Linux⁸ möglich.

6 Konfigurationsdatei

Normalerweise liegt die Datei `config.json` im gleichen Verzeichnis, wie das Programm. Es müssen bestimmte Angaben gemacht werden, ohne die das Programm beim Start einen Fehler ausgibt. Dazu gehören: "Benutzername" und "Passwort". Name und Passwort werden in `'` eingeschlossen und bezeichnen die Zugangsdaten.

"Ausgabe" bezeichnet die Menge der Ausgabe. 0 → nur Fehler; 1 → Hinweise auf aktuelle Funktionen; 2 → alle Ausgaben. Diesen Wert kann man auch mittels Option `-output` gesetzt werden. Die beiden weiteren Angaben werden als Listen bezeichnet und beinhalten die Funktionen, die im Abschnitt 7 erklärt werden.

Alle Zeilen, die mit der Raute (`#`) anfangen, werden nicht benutzt. Sie können also als Kommentare oder Ausklammeroperator genutzt werden.

Beispiele für Konfigurationsdateien finden Sie im Abschnitt 10.

Hier folgt eine minimale Konfigurationsdatei.

```
"Benutzername": "name",  
"Passwort": "geheim"
```

⁷<http://support.microsoft.com/kb/110930/de>

⁸<http://www.fibel.org/linux/lfo-0.6.0/node111.html>

Name	Beschreibung	Pflicht	Datentyp
"Benutzername"		ja	String
"Passwort"		ja	String
"Ausgabe"	Wert für die Menge der Ausgabe	nein	Int
"Plan0"	Plan, wenn der Außendienst verfügbar ist	nein	Array
"Plan1"	Plan, wenn der Außendienst nicht verfügbar ist	nein	Array
"Angriff nochmal"	Wenn der Wert beim Angriff überschritten wird, wird der Gegner nochmal angegriffen	nein	Int
"Info Pfad"	Verzeichnis, in dem erweiterte Kampfberichte gespeichert werden sollen. Wird es weggelassen, werden keine gespeichert.	nein	String

7 Programm-Ablaufplan

Die eigentliche Aufgabe des Programms beinhaltet das Abarbeiten eines Plans nach bestimmten Kriterien. Dieser Plan wird in der Konfigurationsdatei (`config.json`) definiert. Es gibt zwei unterschiedliche Pläne, in denen einzelne Funktionen aufgerufen werden.

Der Plan 0 (in der Version 1.0 als "Außendienst verfügbar" bezeichnet) wird ausgeführt, falls man am aktuellen Tag noch einen Außendienst machen darf. Nach jedem Aufruf einer Funktion wird erneut geprüft ob noch Außendienste gemacht werden dürfen. Ist das nicht der Fall, wird in den Plan 1 übergegangen.

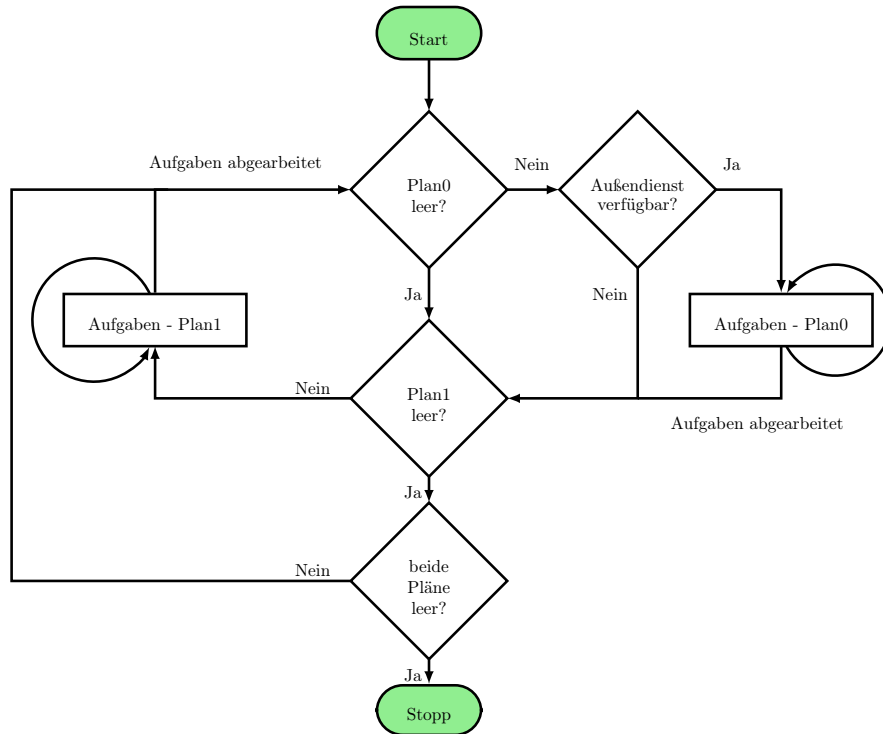
Der Plan 1 (in der Version 1.0 als "Kein Außendienst" bezeichnet) wird in zwei Fällen ausgeführt. Zum einen, wenn im Plan 0 keine Funktionen definiert sind, der Plan also leer ist. Zum anderen, wenn es für den aktuellen Plan keine Außendienste mehr ausgeführt werden können.

Die Pläne werden in der Config in einer Aufzählung beschrieben. Zwischen ihnen müssen also immer Kommata stehen. Hier ein Beispiel:

```
"Plan0": [
  { "Außendienst": 0 },
  { "Außendienst": 1 },
  { "Bank": 0 },
  { "Bank": 103 },
  { "Boese": 0 }
],
```

Wie man erkennen kann, werden die Funktionen in eckigen Klammern eingeschlossen. Nach der letzten Funktion kommt kein Kommata.

Es folgt ein Diagramm zur Erleuterung des Ablaufs der Pläne:



Nachfolgend werde alle implementierten Funktionen beschrieben. Sie können in beiden Plänen benutzt werden.

7.1 Angriff

Der Befehl "Angriff" bringt den Bot dazu sich einen geeigneten Gegner zu suchen und ihn anzugreifen. Dabei kann man zwei Parameter einstellen. "Stufe" bezeichnet die Anzahl der Level, die der Gegner Unterschied haben soll. Ist man selber im Level 8 und will jemanden aus Level 5 angreifen trägt man also -3 ein. Der zweite Parameter sagt was über die Vereinszugehörigkeit aus: -1 = Kein Verein; 0 = Egal; 1 = Im Verein.

```
"Plan0":[
# Angriff mit eigene Stufe+1 und im Verein
  {"Angriff":{"Stufe":1,"Verein":1}},
# Angriff mit eigene Stufe-5 und nicht im Verein
  {"Angriff":{"Stufe":-5,"Verein":-1}},
# Angriff mit eigene Stufe und egal ob im Verein
  {"Angriff":{"Stufe":0,"Verein":0}}
],
```

Ab Version *1.0.beta8* gibt es zusätzliche Optionen, die aber auch weggelassen oder kombiniert werden können:

```
"Plan0":[
# Kein Angriff gegen Freunde
  {"Angriff":{"Stufe":0,"Verein":0,"Freunde":true}},
# Angriff nur gegen weniger als 10000 Respekt
  {"Angriff":{"Stufe":0,"Verein":0,"Respekt":10000}}
],
```

Freunde sind Personen auf der Freundesliste (*Zu mir \Rightarrow Freundes Liste*)
Generell werden nur Feinde aus anderen Bundesländern angegriffen.

Es kann auch die *Medizin*-Option gesetzt werden.

```
"Plan0":[
  {"Angriff":{"Stufe":0,"Verein":0,"Medizin":40}}
],
```

7.2 Arbeiten

Der Befehl "Arbeiten" nimmt eine Zahl als Parameter. Diese stellt die Anzahl der Stunden ein.

```
"Plan0":[
  {"Arbeiten":1}
],
```

Mit der zusätzlichen Option "Stopp" wird der Bot nach dem Beginn der Arbeit beendet, um Systemressourcen zu sparen.

```
"Plan0":[
  {"Arbeiten":{"Stunden":10,"Stopp":true}}
],
```


7.3 Außendienst

Der Befehl "Aussendienst" ist nur in Plan 0 sinnvoll, kann also auch nur dort eingesetzt werden. Als Parameter kann 0 für leichten und 1 für schweren Außendienst gesetzt werden. (**Achtung:** In Version 1.0 hieß diese Funktion "Außendienst")

```
"Plan0": [
  { "Aussendienst": 0 },
  { "Aussendienst": 1 }
],
```

Es kann auch die "Medizin"-Option gesetzt werden. Dann lautet die Einstufung in Schwierigkeiten "Stufe". Dies macht natürlich nur beim schweren Außendienst Sinn.

```
"Plan0": [
  { "Aussendienst": { "Stufe": 1, "Medizin": 50 } }
],
```

7.4 Beschreibung

Diese Funktion ermöglicht es, den Bot zu steuern ohne ihn auszuschalten oder am richtigen Rechner zu sitzen. Man trägt die Funktion einfach in den Plan ein.

```
"Plan0": [
  { "Beschreibung": true }
],
```

Immer wenn der Plan zu der Funktion "Beschreibung" kommt, sucht das Programm auf der eigenen Profilseite in der Beschreibung nach einem Befehl. Dieser muss folgende Form haben:

```
{ "Befehl": [
] }
```

Zwischen den eckigen Klammern können beliebige Befehle stehen, wie in den anderen Plänen auch. Wichtig, dass das erste Zeichen die geschweifte Klammer ist. Der Bot schreibt dann Hinweise in diese Beschreibung, wenn er fertig ist oder die Eingabe nicht richtig verstehen konnte.



Seit der Version 1.0.beta5 gibt es eine Erweiterung hierzu. Der Platz in der Beschreibung ist begrenzt, also kann man Angriffe in einer einfachen Liste ausführen. Der folgende Befehl so direkt in die Beschreibung eingegeben, würde zwei Angriffe auf "_wolf_" ausführen, da die ID "CHEGGF", die von "_wolf_" ist.

```
[ "CHEGGF", "_wolf_" ]
```

7.5 Bank

Der Befehl "Bank" bringt Geld auf die Sparkasse. Mit dem Parameter 0 bringt man so viel wie möglich auf die Bank. Mit allen anderen positiven Zahlen wird genauso viel Geld eingezahlt, welches von der Zahl repräsentiert wird.

```
"Plan0": [  
# Bringt alles auf die Bank  
  { "Bank": 0 },  
# Bringt nur 205 DM auf die Bank  
  { "Bank": 205 }  
],
```

7.6 Booster

Mit der Vereinsvitrine kam auch das Spiel "Booster" hinzu. Es kann mit folgenden Befehl ausgeführt werden. Funktioniert natürlich nur, wenn man am aktuellen Tag noch Booster ausführen darf.

```
"Plan0": [  
# Booster mit Ergebnis 5 ausführen  
  { "Booster": 5 }  
],
```

7.7 Böse Spieler

Der Befehl "Boese" greift eine Person aus der "Böse Spieler"-Liste an. Wählt man als Parameter 0, wird zufällig jemand angegriffen. Wird 1 gewählt, werden alle aus der Liste gleichmäßig angegriffen. Pro Befehl wird ein Angriff ausgeführt. (**Achtung:** In Version 1.0 hieß diese Funktion "Böse")

```
"Plan0": [
# Böse-Liste-Person angreifen (zufällig = 0)
    {"Boese":0},
# Böse-Liste-Person angreifen (von oben = 1)
    {"Boese":1}
],
```

Es kann auch die Medizin-Option gesetzt werden. Die Standardoption wird zu "Liste".

```
"Plan0": [
    {"Boese":{"Liste":1,"Medizin":40}}
],
```

7.8 Goldene Zwerge

Der Befehl "Golden" greift den nächsten Goldenen Zwerg an. Dabei ist diese Option nur Sinnvoll, wenn die Konfiguration mit der erweiterten Ausgabe eingestellt ist. Außerdem sollte man es nur in der Beschreibung benutzen.

```
{"Befehl": [
    {"Golden":true}
]}
```

~~Es kann auch die Medizin-Option gesetzt werden.~~

```
"Plan0": [
    {"Golden":{"Medizin":40}}
],
```

7.9 Kampf

Der Befehl "Kampf" erzeugt im Gegensatz zu "Boese" und "Angriff" eine Keilerei mit einem bestimmten Gegner. Als Parameter wird entweder die ID oder der Name des Gegners übergeben. Die ID ist das Ende der URL, wenn man auf dem Profil ist. Bei der URL des Profils von "_wolf_" (1. Platz) lautet:

<http://www.bundeskampf.com/characters/profile/CHEGGF>

Die ID lautet also: CHEGGF Man könnte auch den Namen angeben, allerdings wird der bei zu kurzen Namen nicht immer gefunden. Hier beide Beispiele:

```
"Plan0": [
    {"Kampf": "CHEGGF"},
    {"Kampf": "_wolf_"},
],
```

Es kann auch die **Medizin**-Option gesetzt werden. Das feindliche Profil wird mit "Gegner" angegeben.

```
"Plan0": [
    {"Angriff": {"Gegner": "_wolf_", "Medizin": 40}},
],
```

7.10 Minuten

Der Befehl "Minuten" bewirkt ein Schlafen des Bots. Als Parameter nimmt er die Anzahl der Minuten, die er Schlafen soll. Dieser Befehl muss nicht für die anderen Funktionen benutzt werden, sondern ist dafür da, dass man selber etwas machen kann.

```
"Plan0": [
# Warte 3 Minuten
    {"Minuten": 3},
# Warte 2 Stunden
    {"Minuten": 120},
],
```

7.11 Neustart

Will man an einer Stelle den internen Speicher säubern oder ist durch irgendeinen Bug genervt, kann man den Bot kontrolliert neu starten.

```
"Plan0": [
    {"Neustart": true},
],
```

7.12 Stopp

Normalerweise arbeitet der Bot die Listen nacheinander ab und fängt wieder von vorne an. Um dies zu Unterbrechen gibt es den Befehl "Stopp".

```
"Plan0": [
    {"Stopp": true},
],
```

8 Programm-Varianten

Das Programm implementiert verschiedene Hilfsmittel. Hier werden Sie aufgeführt und beschrieben.

8.1 Tägliche Möglichkeiten

Folgende tägliche Möglichkeiten können kombiniert werden. Maximal zu:

```
eingabe# java -jar BKampfBot.jar los quiz glueck wein spiel
```

Sie werden in der jeweiligen Reihenfolge mit nur einem Login ausgeführt.

8.1.1 Tagesquiz

Mit Hilfe des Programms kann das Tagesquiz ziemlich gut gelöst werden. Und es geht viel schneller, als wenn man es selber macht. Der Aufruf sieht wie folgt aus:

```
eingabe# java -jar BKampfBot.jar quiz
```

Die Fragen werden direkt (richtig) beantwortet.

8.1.2 Rubbelos

Wer schon mal selber versucht hat, die Rubbellose von ihrer Beschichtung zu befreien, der lässt es sehr schnell sein, denn es dauert zu lange. Will man es schneller erledigen, nimmt man folgenden Befehl:

```
eingabe# java -jar BKampfBot.jar los
```

8.1.3 Glücksrad

Diese Option funktioniert nur in einem Verein. Sie dreht das Rad auf die Gewinnposition.

```
eingabe# java -jar BKampfBot.jar glueck
```

8.1.4 Weinkeller

Diese Option funktioniert nur in einem Verein. Sie spielt das Weinkellerspiel und gewinnt es.

```
eingabe# java -jar BKampfBot.jar wein
```

8.1.5 Tagesspiel

Das Tagesspiel wird mit voller Punktzahl ausgeführt.

```
eingabe# java -jar BKampfBot.jar spiel
```

8.2 Lottoschein

Genau wie im Fernsehen könnt ihr es euch Sonntags Abend gemütlich machen und darauf hoffen, dass die Glücksfee euch diese Woche hold ist. Und ihr - zumindest in Bundeskampf - nicht mehr zu arbeiten braucht.

Das Ausfüllen eines Scheins dauert auch ein wenig. Und welche Zahlen soll man nehmen? Mit dem folgenden Aufruf erledigt es das Programm:

```
eingabe# java -jar BKampfBot.jar lotto
```

Beispielausgabe (die Zahlen werden zufällig gewählt):

```
Login  
visit: http://www.bundeskampf.com/lotto/history  
Added numbers: 2,5,10,38,40  
Logout  
visit: http://www.bundeskampf.com/signups/logout
```

9 Weitere Hinweise

9.1 Medizin

Man kann bei verschiedenen Optionen die Wartezeit verkürzen, indem man Medizin kauft. Im Spiel wird diese "Direkt Fit" genannt. Folgende Pläne benutzen die Aktion: **Angriff**, **Außendienst**, **Böse Spieler**, **Goldene Zwerge** und **Kampf**.

In der Konfiguration gibt man eine ganzzahlige Zahl an, die als Prozent der restlichen Lebensenergie angesehen wird, ab der Medizin gekauft wird. Die Rechnung lautet:

$$\frac{\text{Lebenspunkte}}{\text{MaximaleLebenspunkte}} = \text{Prozent}$$

Am besten ein Beispiel zur Anschauung. Jemand hat folgenden Plan eingetragen:

```
"Plan0": [  
    {"Angriff": {"Stufe": 0, "Verein": 0, "Medizin": 40}}  
],
```

Es soll also Medizin gekauft werden, wenn nach dem Kampf die Lebensenergie unter 40% gefallen ist. Nun kämpft er und erhält folgendes Ergebnis:

$$\frac{345LP}{1000LP} = 35\%$$

Die Medizin wird also gekauft.

Trägt man 0 ein, wird gekauft, wenn man den Kampf verloren hat und die Lebenspunkte komplett runter sind. Wird 100 eingetragen, wird die Medizin immer gekauft.

9.2 Unerwartete Fehler

Sollte ein unerwarteter Fehler auftreten, welches zu einem Absturz des Systems führt, wird der Fehler mit mehreren Hinweisen ausgegeben. Danach startet sich das Programm neu. Das Programm versucht dann erneut seine Aufgaben zu erledigen. Passiert zu oft ein Absturz beendet sich das Programm.

Sollten Sie sich den Fehler durch die Ausgaben des Bots und durch selbstständigen Prüfen auf der Webseite nicht erklären lassen, wird darum gebeten ein Ticket auf der Entwicklerseite⁹ zu erstellen. Dort können auch Wünsche für neue Funktionen geäußert werden.

9.2.1 Bekannte Fehler

Tritt ein Fehler auf, in dem von einem *Internal Server Error - 500* die Rede ist, liegt es an dem Bundeskampf-Server und nicht am Bot. Die genaue Fehlermeldung ist auf der Webseite zu finden.

⁹<https://warnow.mgvmedia.com/trac/BKampfBot/newticket>

10 Beispielkonfigurationen

Die folgenden Konfigurationen sind auch auf der Webseite zum Download verfügbar.

Eine Konfiguration für jemanden, der nur die Außendienste (schwer) machen will und danach das Programm beenden will, sieht dann so aus:

```
"Benutzername": "meinName",
"Passwort": "geheim",
"Plan0": [
    {"Außendienst": 1}
],
"Plan1": [
    {"Stopp": true}
]
```

Konfiguration: Nur Außendienste leicht, wenn fertig, regelmäßig prüfen, ob es schon ein neuer Tag ist und dann wieder Außendienste machen:

```
"Benutzername": "meinName",
"Passwort": "geheim",
"Plan0": [
    {"Außendienst": 0}
],
"Plan1": [
    {"Minuten": 30}
]
```

Konfiguration: Keine Außendienste; Nur Angreifen, gleiche Stufe, im Verein.

```
"Benutzername": "meinName",
"Passwort": "geheim",
"Plan1": [
    {"Angriff": {"Stufe": 0, "Verein": 1}}
]
```

Konfiguration: Keine Außendienste; Nur Arbeiten, dann alles auf Bank bringen.

```
"Benutzername": "meinName",
"Passwort": "geheim",
"Plan1": [
    {"Arbeiten": 1},
    {"Bank": 0}
]
```

Konfiguration: Keine Außendienste; Angreifen, dann einen bösen Angreifen und dann eine Pause.

```
"Benutzername": "meinName",
```



```
"Passwort": "geheim",  
"Plan1": [  
  {"Angriff": {"Stufe": 0, "Verein": 1}},  
  {"Boese": 1},  
  {"Minuten": 10}  
]
```



Commons Deed

Namensnennung - Keine kommerzielle Nutzung - Weitergabe unter gleichen Bedingungen 3.0

Es ist Ihnen gestattet:

- das Werk vervielfältigen, verbreiten und öffentlich zugänglich machen
- Abwandlungen bzw. Bearbeitungen des Inhaltes anfertigen

Zu den folgenden Bedingungen:

- Namensnennung. Sie müssen den Namen des Autors/Rechtsinhabers in der von ihm festgelegten Weise nennen.
- Keine kommerzielle Nutzung. Dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden.
- Weitergabe unter gleichen Bedingungen. Wenn Sie den lizenzierten Inhalt bearbeiten oder in anderer Weise umgestalten, verändern oder als Grundlage für einen anderen Inhalt verwenden, dürfen Sie den neu entstandenen Inhalt nur unter Verwendung von Lizenzbedingungen weitergeben, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind.
- Im Falle einer Verbreitung müssen Sie anderen die Lizenzbedingungen, unter die dieser Inhalt fällt, mitteilen.
- Jede der vorgenannten Bedingungen kann aufgehoben werden, sofern Sie die Einwilligung des Rechteinhabers dazu erhalten.
- Diese Lizenz lässt die Urheberpersönlichkeitsrechte unberührt.

Das Commons Deed ist eine Zusammenfassung des Lizenzvertrags in allgemeinverständlicher Sprache. Um den Lizenzvertrag einzusehen, besuchen Sie die Seite

<http://creativecommons.org/licenses/by-nc-sa/3.0/de/>

oder senden Sie einen Brief an Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

