

Kernprüfung AAG

Betriebshandbuch

Stand: 05.09.2012

Autor: Michael Buchwald

Version	Datum	Kap./Seite	Grund der Änderung	Bearbeiter
1.0	29.09.2009		Erstellung	M. Buchwald
1.1	28.02.2011	Kapitel 3 und 4	Überarbeitung	M. Buchwald
1.2	31.05.2011	Kapitel 3	Dokumentation des Tester-Tools	M. Buchwald
1.3	05.09.2012	Kapitel 2 und 4	Überarbeitung	M. Buchwald

Inhalt

1	EINLEITUNG	4
2	VORAUSSETZUNGEN	4
3	INSTALLATION	4
4	VERWENDUNG	5

1 Einleitung

Dieses Dokument ist die technische Anleitung für die Installation und den Betrieb der Komponente für die Kernprüfung im Verfahren AAG (Erstattungsanträge nach dem Aufwendungsausgleichsgesetz).

2 Voraussetzungen

Folgende Laufzeit- / Entwicklungsumgebung wird vorausgesetzt:

- Java-Runtime bzw. JDK 1.6

Außer der Java-Runtime-Umgebung werden keine weiteren Komponenten oder Programme benötigt.

3 Installation

Die Lieferung besteht aus folgenden Bestandteilen:

- **Betriebshandbuch:** Dieses Dokument.
- **Quell-Dateien:** Alle Java-Sourcen der Komponente AAG.
- **Class-Dateien:** Alle übersetzten Java-Sourcen der Komponente AAG.
- **Javadoc-Dateien:** Aus den Sourcecode generierte Entwicklerdokumentation.
- **Tester:** Benutzungsoberfläche für den Aufruf der Kernprüfung.

Alle oben genannten Dateien werden in separaten JAR-Dateien zusammengefasst:

- **Quell-Dateien:** KernpruefungAAG_<Version>-src.jar
- **Class-Dateien:** KernpruefungAAG_<Version>-bin.jar
- **Javadoc-Dateien:** KernpruefungAAG_<Version>-doc.jar
- **Tester:** KernpruefungTester_<Version>.zip

Für den Betrieb müssen die Class-Dateien bzw. die entsprechende JAR-Datei durch die Java Virtual Machine zugreifbar sein.

Da auch Quell-Dateien geliefert werden, kann die Komponente geändert werden. Zu beachten ist, dass nur bei den Interfaces **kernpruefung.Kernpruefung** und **kernpruefung.Rueckgabe** garantiert wird, dass diese nie nachträglich geändert werden. Alle anderen Klassen und Interfaces können in weiteren Versionen geändert werden, insbesondere Signaturen von Methoden.

4 Verwendung

Wie in der Sitzung der Arbeitsgruppe Einheitliche Kernprüfungen am 22.07.2009 festgelegt, sollen alle Prüfungen (Aufrufe) durch den gemeinsamen Typ **kernpruefung.Kernpruefung** erfolgen, die Ergebnisse sollen in einem Objekt vom Typ **kernpruefung.Rueckgabe** gespeichert werden. Somit kann die Komponente AAG schon vor der eigentlichen Lieferung in anderen Programmen zur Entwicklungszeit verwendet werden. Zur Laufzeit muss ein konkretes Objekt instanziiert werden, das erfolgt mit dem Aufruf der statischen Methode

de.bitmarckservice.kernpruefung.aag.KernpruefungAAGImpl.getInstance(). Diese Methode liefert ein neues Objekt vom Typ **kernpruefung.Kernpruefung** (kein Singleton). Der Aufruf von **getInstance()** ist die einzige Stelle, an der ein verfahrensspezifischer Typ verwendet wird. Alternativ kann

- `new de.bitmarckservice.kernpruefung.aag.KernpruefungAAGImpl ()` oder
- `Class.forName("de.bitmarckservice.kernpruefung.aag.KernpruefungAAGImpl").newInstance()`

verwendet werden.

Beispiel:

```
Kernpruefung pruefer = KernpruefungAAGImpl.getInstance();
```

```
Rueckgabe rueckgabe = pruefer.pruefe(satz, vorlaufsatz);
```

Die Methode **getReturnCode()** des Rückgabe-Objektes liefert folgende Werte:

- **0:** Alle Daten-Sätze sind plausibel gemäß dem Fehlerkatalog.
- **1:** Prüfung ergab mindestens einen Hinweis auf mögliche Fehler. **Wird in der aktuellen Version nicht verwendet.**
- **2:** Es wurde mindestens ein Fehler erkannt.
- **4:** Prüfung konnte nicht durchgeführt werden und wurde abgebrochen.

Der Return-Code wird „geodert“, d.h. dass z.B. ein Return-Code mit dem Wert 3 zurückgegeben wird, falls es sowohl Hinweise als auch Fehler gibt.

Die Methode **getRueckgabeMeldungen()** des Rückgabe-Objektes liefert **null**, falls es keine Meldungen gibt oder ein String-Array mit den Meldungen über erkannte Fehler bzw. Hinweise. Werden mehr als neun Fehler/Hinweise erkannt, so werden nur die ersten neun berücksichtigt. Ausnahme: Wenn alle neun Meldungen Hinweise sind, so wird der letzte Hinweis durch die nächste Meldung ungleich Hinweis ersetzt.

Tritt eine Abbruchbedingung (Return-Code = 4) auf, werden alle zuvor erkannten Fehler gelöscht und nur die Abbruch-Meldung im Rückgabewert vermerkt.

Die toString()-Methode des Rückgabe-Objektes liefert einen String im Format:

<return code><anzahl fehler>[<meldungen>]

<return code>: Der mit getReturnCode() gelieferte Wert (siehe oben).

<anzahl fehler>: Anzahl der erkannten Fehler oder Hinweise. Maximal neun.

<meldungen>: Optional die Konkatenation aller Meldungen (Strings der Länge 76) in der Index-Reihenfolge des Arrays, der durch getRueckgabeMeldungen() geliefert wird.

Hinweis:

KernpruefungAAGImpl ist ab Version 01.03.00.00 thread-sicher.