Entwicklung einer Klassenbibliothek zur Erzeugung autokorrelierter Zufallszahlen

SAD – Software Architecture Documentation

Abteilung Informatik

Hochschule für Technik Rapperswil

|  |
| --- |
| Herbstsemester 2017 |

Autor(en): Anthony Delay

Philipp Bütikofer

Betreuer: Prof. Dr. Andreas Rinkel

Lukas Kretschmar

# Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Änderung** | **Autor** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Inhalt

[Änderungsgeschichte 2](#_Toc496086712)

[Inhalt 3](#_Toc496086713)

[1. Einführung 4](#_Toc496086714)

[1.1 Zweck 4](#_Toc496086715)

[1.2 Gültigkeitsbereich 4](#_Toc496086716)

[1.3 Referenzen 4](#_Toc496086717)

[2. .NET Core – Architektonische Ziele und Einschränkungen 5](#_Toc496086718)

[2.1 Technische Plattform 5](#_Toc496086719)

[2.2 Performance 5](#_Toc496086720)

[3. Grundlagen 5](#_Toc496086721)

[4. Namespaces (Systemübersicht) 5](#_Toc496086722)

[5. Lösungsstrategie 5](#_Toc496086723)

[6. Assembly: ARTA 5](#_Toc496086724)

[6.1 Beschreibung 5](#_Toc496086725)

[6.2 Klassendiagramm 6](#_Toc496086726)

[7. Assembly: MathSubSet 6](#_Toc496086727)

[7.1 Beschreibung 6](#_Toc496086728)

[7.2 Klassendiagramm 6](#_Toc496086729)

[8. Assembly: Statistical Tests 6](#_Toc496086730)

[8.1 Beschreibung 6](#_Toc496086731)

[8.2 Klassendiagramm 6](#_Toc496086732)

[9. Patterns 6](#_Toc496086733)

[9.1 Factory-Pattern 6](#_Toc496086734)

# Einführung

## Zweck

Dieses Dokument zeigt die architektonische Sichtweise übe rdie Studienarbeit «Entwicklung einer Klassenbibliothek zur Erzeugung von autokorreltierten Zufallszahlen».

## Gültigkeitsbereich

Dieses Dokument ist im Rahmen der Studienarbeit 2017 «Entwicklung einer Klassenbibliothek zur Erzeugung autokorrelierter Zufallszahlen gültig.

## Referenzen

|  |  |
| --- | --- |
| **Bezeichnung** | **Link** |
|  |  |
|  |  |
|  |  |

# .NET Core – Architektonische Ziele und Einschränkungen

Dieser Abschnitt beschreibt Anforderungen und Einschränkungen, welche signifikanten Einfluss auf die Architektur von Arta.Core mit sich bringen.

## Technische Plattform

Das Fundament soll .Net Core bilden. Ziel ist es, eine möglichst schlanke Klassenbibliothek zu erzeugen. Diese soll möglichst auf allen Plattformen einsetzbar sein. .Net Core stellt eine modulare Version des .Net Framework dar, welches auf plattformübergreifende Portierbarkeit ausgelegt ist.

Jarta referenziert und nutzt einige externe mathematische Bibliotheken von Apache. Viele dieser Refenzen werden für Jarta selbst nicht benötigt und vergrössern den Umfang der Applikation unötig. Es stehen zwar für .Net vergleichbare Bibliotheken zur Verfügung, diese werden aber von .Net Core nicht unterstützt. Somit werden diese Bibliotheken für dieses Projekt von Java in C# übersetzt, um in der .Net Umgebung von ARTA.core die gleiche Funktionalität zu erreichen. Hierbei wird auch darauf geachtet die Applikation schlank zu halten und nicht benötigte Referenzen auszuschliessen.

## Performance

Dem Projekt sind keine Performance-Constraints gesetzt. Die Klassenbibliothek selbst soll sich in den Zeiten der bereits existierenden JARTA-Implementation richten.

# Grundlagen

# Namespaces (Systemübersicht)

# Lösungsstrategie

# Assembly: ARTA

## Beschreibung

Dieses Assmebly enthält die nötigen Klassen und Interfaces um den ARTA-Prozess korrekt abzubilden. Das Assembly ist wiederrum in mehrere Folders unterteilt.

## Klassendiagramm

# Assembly: MathSubSet

## Beschreibung

## Klassendiagramm

# Assembly: Statistical Tests

## Beschreibung

## Klassendiagramm

# Patterns

## Factory-Pattern