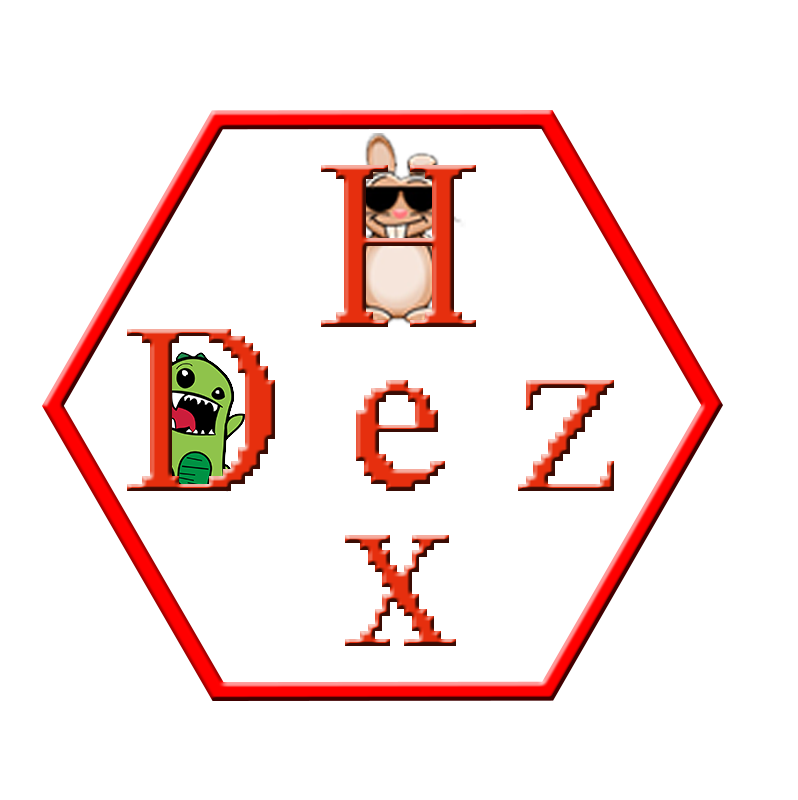
Pflichtenheft

DezHex



**Pflichtenheft**

„Entwicklung eines benutzerfreundlichen Programms für die Umwandlung von Dezimal- in Hexadezimalzahlen in Java“

***Stand:*** 01.02.2017

***Auftraggeber:***  Zuse-Grundschule Berlin

Bernhard-Weiß-Straße 6

10178 Berlin

Ansprechpartner: Herr Igor Münstermann

***Auftragnehmer:*** Institut für Schule und Medien Berlin

Uhlandstraße 114

10717 Berlin-Wilmersdorf

Ansprechpartner: Herr Johannes Schulz

Inhaltsverzeichnis

[1 Zielbestimmung 3](#_Toc473719606)

[1.1 Muss-Kriterien 3](#_Toc473719607)

[1.2 Kann-Kriterien 3](#_Toc473719608)

[1.3 Abgrenzungskriterien 3](#_Toc473719609)

[2 Produkteinsatz 4](#_Toc473719610)

[2.1 Anwendungsbereich 4](#_Toc473719613)

[2.2 Zielgruppen 4](#_Toc473719614)

[2.3 Produktumgebung 4](#_Toc473719615)

[2.4 Betriebsbedingungen 4](#_Toc473719616)

[3 Produktfunktionen/Anforderungen 4](#_Toc473719617)

[3.1 Funktionale Anforderungen 4](#_Toc473719618)

[3.1.1 Beschreibung der FA mit Rollen innerhalb der Geschäftsprozesse 4](#_Toc473719619)

[3.1.2 Aktivitäten mit Benutzeroberfläche (GUI) 5](#_Toc473719620)

[3.1.3 Fachliches Klassendiagramm (domain model) / Produktdaten 8](#_Toc473719621)

[3.2 Nichtfunktionale Anforderungen 8](#_Toc473719622)

[3.2.1 Benutzbarkeit 8](#_Toc473719623)

[3.2.2 Zuverlässigkeit 8](#_Toc473719624)

[3.2.3 Effizienz 8](#_Toc473719625)

[3.2.4 Sicherheit 8](#_Toc473719626)

[3.2.5 Normen 8](#_Toc473719627)

[4 Testung 8](#_Toc473719628)

[5 Monitoring/ Support bei Übergabe oder ähnliche Leistungen 9](#_Toc473719629)

[6 Dokumentation 9](#_Toc473719630)

[6.1 Anwenderdokumentation 9](#_Toc473719631)

[6.2 Administratorendokumentation 9](#_Toc473719632)

[6.3 Entwicklerdokumentation 9](#_Toc473719633)

[6.4 Weitere referenzierte Dokumente 9](#_Toc473719634)

[7 Vorgehen (Wie?) 10](#_Toc473719635)

[8 Entwicklungsumgebung (Womit?) 11](#_Toc473719636)

[9 Glossar 12](#_Toc473719637)

# Zielbestimmung

Für die Projektwoche der Zuse-Grundschule ist eine Java-Applikation zu entwickeln, die ganze Dezimal- in Hexadezimalzahlen umwandelt und anders herum. Dieses Programm soll durch ein Desktop-Icon startbar und über eine grafische Benutzeroberfläche bedienbar sein.

## Muss-Kriterien

|  |  |  |
| --- | --- | --- |
| MK-H-01 | Hilfetext | Implementierung von F1: das System soll nach Betätigung des Hilfebuttons, welcher auf der kinderfreundlichen Benutzeroberfläche positioniert ist einen Hilfetext in einem extra Fenster ausgeben. |
| MK-H-02 | Hilfetext | Der Hilfetext soll in leicht verständlicher Sprache geschrieben sein. |
| MK-K-01 | Konvertierung | Die Konvertierungsrichtung wird durch zwei beschriftete RadioButtons festgelegt. Diese befinden sich auf der Benutzeroberfläche. |
| MK-K-02 | Konvertierung | Implementierung von F2 & F3: das System erwartet eine Nutzereingabe einer ganzen Zahl in einem TextField mit Label und nachfolgendes Drücken des Umwandlungsbuttons auf der Benutzeroberfläche. |
| MK-K-03 | Konvertierung | Das System soll die eingegebene Zahl konvertiert in Form in einem zweiten TextField mit Label ausgeben, welches auch auf der Benutzeroberfläche positioniert ist. |
| MK-K-04 | Konvertierung | Die Ausgabe der Hexadezimalzahlen soll in Großbuchstaben erfolgen. |
| MK-IMPL-01 | Java Code | Die Implementierung soll in Java erfolgen. |
| MK-IMPL-02 | Code Style | Der Java-Code ist nach den Vorgaben des Google Java Style Guide zu implementieren. (https://google.github.io/styleguide/javaguide.html) |
| MK-ERR-01 | Fehlerbehandlung | Bei falscher Nutzereingabe soll ein Hinweistext in einem extra Fenster ausgegeben werden. |
| MK-ERR-02 | Fehlerbehandlung | Der Hinweistext soll dem Nutzer die richtige Eingabe in leicht verständlicher Sprache verdeutlichen. |

## Kann-Kriterien

|  |  |  |
| --- | --- | --- |
| KK-ERR-01 | Fehlerbehandlung | Das System gibt den Hinweistext nicht in einem extra Fenster aus, sondern als ToolTip an der entsprechenden Stelle. |

## Abgrenzungskriterien

|  |  |  |
| --- | --- | --- |
| AK-NK-01 | Non-Konsole | Das System muss nicht über die Konsole aufrufbar sein. |
| AK-K-01 | Konvertierung | Das System soll keine Konvertierung in andere Zahlensysteme (Binär, Oktal) vornehmen. |

# Produkteinsatz



## Anwendungsbereich

Die DezHex-Anwendung dient der Lehre im Bereich Informatik an der Zuse-Grundschule Berlin während einer Projektwoche.

Schülern soll anhand der Anwendung die Konvertierung von Zahlen greifbar gemacht werden.

## Zielgruppen

Benutzt wird die Anwendung von Schülern der Zuse-Grundschule Berlin. Dabei haben diese die Rolle des Hauptnutzers. Weitere Rollen - etwa eine Administratorenrolle - treten nicht auf*.*

## Produktumgebung

*Das System benötigt eine installierte Java Runtime ab Java-Version 6.0.***Das System benötigt eine installierte Java Runtime ab Java-Version 8.**

Des Weiteren wird ein Computer mit mindestens 128MB RAM sowie einem Pentium 2 mit über 266MHz benötigt.

## Betriebsbedingungen

Das System wird für die Anwendungsfälle F1 bis F3 in den Computerlaboren der Zuse-Grundschule Berlin verwendet*.* Hauptbenutzer bedienen das Programm über die grafische Benutzeroberfläche an den dort aufgestellten Rechnern*.* Ein unbeaufsichtigter Betrieb ist nicht vorgesehen*.* Die Rechner der Labore werden je nach finanzieller Möglichkeit mit aktueller Hardware erneuert, als Betriebssystem kommt Windows zum Einsatz. Die Installation von Betriebssystem und Software erfolgt durch den Administrator.Die Räume sind klimatisiert. Die Rechner haben keine unterbrechungsfreie Stromversorgung (USV).

# Produktfunktionen/Anforderungen

## Funktionale Anforderungen

### Beschreibung der funktionalen Anforderungen mit Rollen innerhalb der Geschäftsprozesse

Es gibt nur zwei Anforderungen: die Hauptfunktion und eine Hilfefunktion.  
Der am Betriebssystem angemeldete Nutzer führt das Programm aus. Der Aufruf erfolgt mittels Desktop-Java- Applikation.

|  |  |  |
| --- | --- | --- |
| F1 | Anzeige des Hilfe-Textes | Ein am Betriebssystem angemeldeter Benutzer führt das Programm über die Verknüpfung auf dem Desktop aus. Per Betätigung der „Hilfe“-Buttons auf der Benutzeroberfläche öffnet sich ein extra Fenster, in welchem ein Hilfetext ausgegeben wird. |
| F2 | Ausgabe der konvertierten Dezimalzahl | Ein am Betriebssystem angemeldeter Benutzer führt das Programm über die Verknüpfung auf dem Desktop aus. Per Betätigung des RadioButtons „in Hexadezimal“ wird die Konvertierungsrichtung gewählt. Bei Eingabe einer Dezimalzahl in das gekennzeichnete Eingabe-TextField und anschließender Betätigung des „Verwandle“-Buttons wird die konvertierte Dezimalzahl im Ausgabe-TextField ausgegeben. |
| F3 | Ausgabe der konvertierten Hexadezimalzahl | Ein am Betriebssystem angemeldeter Benutzer führt das Programm über die Verknüpfung auf dem Desktop aus. Per Betätigung des RadioButtons „in Dezimal“ wird die Konvertierungsrichtung gewählt. Bei Eingabe einer Hexadezimalzahl in das gekennzeichnete Eingabe-TextField und anschließender Betätigung des „Verwandle“-Buttons wird die konvertierte Hexadezimalzahl im Ausgabe-TextField ausgegeben. |

### Aktivitäten mit Benutzeroberfläche (GUI)

|  |  |
| --- | --- |
| Anwendungsfall ID | F1 |
| AF Name | Anzeige des Hilfe-Textes |
| Akteur | am Betriebssystem angemeldeter Nutzer |
| Vorbedingung | Programm per Desktop-Icon geöffnet |
| Auslösendes Ereignis | Betätigung eines „Hilfe“-Buttons per Mausklick |
| Nachbedingung Erfolg | Ausgabe des Hilfetextes erfolgt in einem extra Dialog |
| Ablauf | - Programm per Desktop-Icon öffnen  - „Hilfe“-Button drücken  - Textausgabe erfolgt in extra Fenster |
| Benutzeroberfläche |  |

|  |  |
| --- | --- |
| Anwendungsfall ID | F2 |
| AF Name | Ausgabe der konvertierten Dezimalzahl |
| Akteur | am Betriebssystem angemeldeter Nutzer |
| Vorbedingung | Programm per Desktop-Icon geöffnet, Auswahl des „in Hexadezimal“-RadioButtons, Eingabe einer ganzen Dezimalzahl in das per Label gekennzeichnete Eingabe-TextField |
| Auslösendes Ereignis | Betätigung des „Verwandle“-Buttons per Mausklick |
| Nachbedingung Erfolg | Ausgabe der konvertierten Dezimalzahl erfolgt in per Label gekennzeichnetem Ausgabe-TextField |
| Ablauf | - Programm per Desktop-Icon öffnen  -ganze Dezimalzahl in Eingabe-TextField eingeben  - „in Hexadezimal“-RadioButton per Mausklick auswählen  - „Verwandle“-Button drücken  -Ausgabe der konvertierten Zahl in Ausgabe-TextField |
| Benutzeroberfläche |  |

|  |  |
| --- | --- |
| Anwendungsfall ID | F3 |
| AF Name | Ausgabe der konvertierten Hexadezimalzahl |
| Akteur | am Betriebssystem angemeldeter Nutzer |
| Vorbedingung | Programm per Desktop-Icon geöffnet, Auswahl des „in Dezimal“-RadioButtons, Eingabe einer ganzen Hexadezimalzahl in das per Label gekennzeichnete Eingabe-TextField |
| Auslösendes Ereignis | Betätigung des „Verwandle“-Buttons per Mausklick |
| Nachbedingung Erfolg | Ausgabe der konvertierten Hexadezimalzahl erfolgt in per Label gekennzeichnetem Ausgabe-TextField |
| Ablauf | - Programm per Desktop-Icon öffnen  -ganze Hexadezimalzahl in Eingabe-TextField eingeben  - „in Dezimal“-RadioButton per Mausklick auswählen  - „Verwandle“-Button drücken  -Ausgabe der konvertierten Zahl in Ausgabe-TextField |
| Benutzeroberfläche |  |

### Fachliches Klassendiagramm (domain model) / Produktdaten

extends

extends

Number

#value : int

+getValue() : int

DezNumber

+toString() : String

HexNumber

+toString() : String

Stage

+initStyle(arg0 : Stage) : void

+setTitle(value : String) : void

+setScene(value : scene) : void

+setResizeable(value : boolean) : void

extends

Application

+setTitle() : void

+setScene(:scene) : void

+setResizeable(Boolean) : void

+show() : void

+getClass() : class

+getResource(String) : resource

GradleDezHex

+main(args[] : String) : void

+start(primaryStage: Stage) : void

DezHexController

-numberinput : TextField

-numberoutput : TextField

-toHex : RadioButton

-toDez : RadioButton

-ToggleGroup1 : ToggleGroup

#convert(event : ActionEvent) : void

#openhelpdez(event : ActionEvent) : void

 #openhelphex(event: ActionEvent) : void

HilfeFenster

+HilfeFenster(auswahl : int) : void

DezHexTextField

+replaceText(start: int, end : int, text : String) : void

+replaceSelection(text : String) : void

-verify() : void

TextField

extends

extends

## Nichtfunktionale Anforderungen

### Benutzbarkeit

Das Programm muss sich leicht mit einer grafischen Oberfläche bedienen lassen. Die Benutzung soll ausschließlich über diese Oberfläche geschehen. Die Oberfläche muss so angepasst sein, dass Grundschülern die Funktionsweise der jeweiligen Bedienelemente sofort ersichtlich wird.

### Zuverlässigkeit

Es gilt einen Absturz zu vermeiden. Durch jeweilige Fehlbedienung soll das Programm zuverlässig weiterarbeiten und eine entsprechende Meldung ausgeben. Mehrfaches Starten und Beenden darf im Speicher keine Probleme verursachen.

### Effizienz

Da das Programm auch auf älterer Hardware ohne größere Einschränkungen laufen muss, sollte mit den Ressourcen sparsam umgegangen werden. Unnötige Rechenoperationen sind zu vermeiden, um stets eine schnelle Reaktion vom Programm zu bekommen.

### Sicherheit

Es liegen keine Anforderungen bezüglich der Sicherheit vor.

### Normen

|  |  |  |
| --- | --- | --- |
| NF-B1 | Benutzung | DezHex soll nur über eine grafische Oberfläche bedienbar sein. |
| NF-E1 | Effizienz | Die Ausgabe der umgewandelten Zahlen soll unmittelbar erfolgen. |
| NF-S1 | Sicherheit | Es liegen keine Anforderungen bezüglich der Sicherheit vor. |
| NF-N1 | Normen | Das Programm muss keine besonderen Normen erfüllen. |

# Testung

Es wird ein einfacher Funktionstest für die Anwendungsfälle F1 bis F3 auf der zu erwartenden Umgebung durchgeführt.

# Monitoring/ Support bei Übergabe oder ähnliche Leistungen

Das Programm wird bereits eine Woche vor der Projektwoche der Schule in Betrieb genommen.   
Folgende Leistungen werden erbracht:

* Rufbereitschaft, sowohl per Telefon als auch per E-Mail, Montag – Freitag
* Individueller Support in der Projektwoche
* Bereitstellung des Repositories

# Dokumentation

## Anwenderdokumentation

*Die Anwenderdokumentation wird als „readme.txt“ sowie als „readme.pdf“ Datei in deutscher Sprache im Repository zur Verfügung gestellt.*

**Die Anwenderdokumentation wird als „Anwenderdokumentation.pdf“ Datei in deutscher Sprache im Repository zur Verfügung gestellt.**

## Administratorendokumentation

*Die Administratordokumentation wird als „admin.txt“ sowie als „admin.pdf“ Datei in deutscher Sprache zur Verfügung gestellt.*

**Die Administratorendokumentation wird als „Administratorendokumentation.pdf“ Datei in deutscher Sprache im Repository zur Verfügung gestellt.**

## Entwicklerdokumentation

*Als Entwicklerdokumentation werden die mit javadoc generierten HTML-Dokumente im Repository zur Verfügung gestellt.*

**Als Entwicklerdokumentation werden die mit Doxygen generierten HTML-Dokumente im Repository zur Verfügung gestellt.**

## Weitere referenzierte Dokumente

Das Pflichtenheft wurde auf Basis des „Lastenheftes – Entwicklung eines benutzerfreundlichen Programms für die Umwandlung von Dezimal- in Hexadezimalzahlen in Java“ erstellt. Geforderte Dokumentationen wie Anwender-, Administrator- und Entwicklerdokumentation befinden sich im Installationsordner.

# Vorgehen (Wie?)

Für die Anwendungsfälle F1, F2 und F3 wird ein Prototyp erstellt, der gemäß den nicht funktionalen Anforderungen erweitert werden kann. Der Funktionstest folgt unmittelbar danach. Die als „Release Candidate“ festgelegte Testversion dient zur Erstellung der Dokumentation (code freeze). Nach den Schritten erfolgt die Übergabe.

Die Meilensteine wurden Ende März 2017 vereinbart.

Anhand folgender Indikatoren erfolgt die Fortschrittskontrolle:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Indikator |  | Kick-Off |  | Projektplan  und  Pflichtenheft | Erstellen der  Entwicklungs-  umgebung | Prototyp | Funktions-  test | Release  Candidate | Übergabe |
| Pflichtenheft  [% erledigte  Gliederungs-  Punkte] | Soll |  |  | 100 |  |  |  |  |  |
| Ist |  |  |  |  |  |  |  |  |
| Umgebung  [Tools] | Soll |  |  |  | 10 |  |  |  |  |
| Ist |  |  |  |  |  |  |  |  |
| Diagramme  [Anzahl] | Soll |  |  | 1 |  |  |  |  |  |
| Ist |  |  |  |  |  |  |  |  |
| Quellcode  [LOC] | Soll |  |  |  |  |  |  |  |  |
| Ist |  |  |  |  |  |  |  |  |
| Verhältnis  LOC/  Kommentare | Soll |  |  |  |  | 4:1 |  |  |  |
| Ist |  |  |  |  |  |  |  |  |
| Tests  [Fälle/  Methoden] | Soll |  |  |  |  |  |  |  |  |
| Ist |  |  |  |  |  |  |  |  |
| Anwenderdoku  [Worte] | Soll |  |  |  |  |  | 50 |  |  |
| Ist |  |  |  |  |  |  |  |  |
| Entwicklerdoku  [Worte] | Soll |  |  |  |  |  | 50 |  |  |
| Ist |  |  |  |  |  |  |  |  |
| Administratordoku  [Worte] | Soll |  |  |  |  |  | 50 |  |  |
| Release  [Artefakte] | Soll |  |  |  |  |  |  | 6 | 6 |
| Ist |  |  |  |  |  |  |  |  |

# Entwicklungsumgebung (Womit?)

*Für die Entwicklung des Programmes wird ein Texteditor sowie ein Java-Compiler benötigt.*

V*erwendet wurde GNU nano 2.2.6 und javac 1.8.0\_91.*

**Für die Entwicklung des Programmes wurde die Entwicklungsumgebung Eclipse Neon 3 (4.6.3) genutzt.**

*Das Produkt wurde einem einfachen Funktionstest ohne Testwerkzeug unterzogen.*

**Das Produkt wurde mehrfachen Funktionstests mit JUnit 4 unterzogen.**

*Die Entwicklerdokumentation wurde mit javadoc erstellt, der Quellcode ist entsprechend kommentiert.*

**Die Entwicklerdokumentation wurde mit Doxygen erstellt.**

An die Hardware bestehenden die Mindestanforderungen von der Java Runtime sowie mindestens 20mb freien Speicher.

Bezüglich der Orgware bestehen keine Anforderungen.

# Glossar

GNU: unixähnliches Betriebssystem & vollständig Freie Software

Javadoc: Software-Dokumentationswerkzeug, erstellt aus Java-Quelltexten automatisch HTML-Dokumentationsdateien

HTML: Hypertext Markup Language

Java Runtime (Environment): Java-Laufzeitumgebung

Dezimal(-system): Zahlen(-system), auf der Basis 10

Hexadezimal(-system): Zahlen(-system), mit der Basis 16

Binär(-system): Zahlen(-system), mit den Grundzahlen 0 & 1,

auch Dualsystem genannt

Oktal(-system): Zahlen(-system), mit der Basis 8

Tooltip: Ist ein kleines zusätzliches Fenster in Anwendungsprogrammen, welches eine Beschreibung zu einem Element der grafischen Benutzeroberfläche bereitstellt.

Radiobutton: Grafisches Benutzeroberflächenelement,

welches eine Auswahlmöglichkeit bereitstellt.

Textfield: Grafisches Benutzeroberflächenelement,

bei dem Text eingetragen und verändert werden kann

Label: Grafisches Benutzeroberflächenelement,

welches Text anzeigt.

Repository: Zentrale Ablage für Daten, Dokumente, Objekte und Programme mit ihren Metadaten.

code freeze: Zeitpunkt innerhalb eines Software-Projekts, ab dem sich Quellcode  
der Software bis zur endgültigen Veröffentlichung nicht mehr ändern soll.

RAM: Arbeitsspeicher