

**Daniel Hänßgen**

# **Ein Orientierungssystem für Menschen mit Sehbehinderung auf Java ME**

**Konzeption und Implementierung**



**Diplomica Verlag**

Daniel Hänßgen

**Ein Orientierungssystem für Menschen mit Sehbehinderung auf Java ME: Konzeption und Implementierung**

ISBN: 978-3-8428-0494-4

Herstellung: Diplomica® Verlag GmbH, Hamburg, 2011

---

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die Informationen in diesem Werk wurden mit Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden und der Verlag, die Autoren oder Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für evtl. verbliebene fehlerhafte Angaben und deren Folgen.

© Diplomica Verlag GmbH

<http://www.diplomica-verlag.de>, Hamburg 2011

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Erläuterungen zur Gliederung . . . . .	8
1.2	Typografische Konventionen . . . . .	8
<b>2</b>	<b>Anforderungsanalyse</b>	<b>9</b>
2.1	Motivation . . . . .	9
2.2	Zielsetzung . . . . .	10
2.2.1	Abgrenzung . . . . .	10
2.2.2	Funktionale Anforderungen . . . . .	10
2.2.3	Informationsquellen . . . . .	11
2.2.4	Technische Anforderungen . . . . .	11
2.2.5	Nichtfunktionale Anforderungen . . . . .	12
2.3	OpenStreetMap . . . . .	12
2.3.1	Geschichte . . . . .	12
2.3.2	Lizenz . . . . .	13
2.3.3	Technik . . . . .	13
2.3.4	Datenquellen und Quellformate . . . . .	14
2.3.5	Details zum Datenformat . . . . .	15
2.3.6	Informationsdichte . . . . .	18
2.4	GPS . . . . .	19
2.4.1	NMEA-Protokoll . . . . .	19
2.5	Begriffserklärung: Kurs, Sollkurs und Peilung . . . . .	22
2.6	Richtungsabhängige Vibration . . . . .	23
<b>3</b>	<b>Analyse</b>	<b>25</b>
3.1	Paketstruktur . . . . .	25
3.2	Klassendiagramme . . . . .	28
3.2.1	View-Klassen . . . . .	28
3.2.2	Main-Klasse . . . . .	32
3.2.3	GPS-Paket . . . . .	33
3.2.4	OSM-Paket . . . . .	34
3.2.5	Util-Paket . . . . .	36
3.2.6	Datastore-Paket . . . . .	38
3.2.7	Entity-Paket . . . . .	40
3.3	Sequenzdiagramme . . . . .	42
3.3.1	Starten der Anwendung . . . . .	42
3.3.2	GPS-Daten lesen . . . . .	43

3.3.3	Umgebungssuche . . . . .	44
3.3.4	Stichwortsuche . . . . .	45
3.3.5	Favoriten speichern . . . . .	46
3.3.6	Punkt als Favoriten speichern . . . . .	47
3.3.7	Einfaches Beispiel für die Verwendung von SimpleInputView . . . .	48
<b>4</b>	<b>Entwurf</b>	<b>49</b>
4.1	Exkurs: Java ME . . . . .	49
4.1.1	MIDlet . . . . .	50
4.1.2	Erschwerte Bedingungen durch Signierung . . . . .	51
4.1.3	Screenreader-freundlich mit Java ME . . . . .	51
4.2	Beziehungen zwischen den Klassen . . . . .	53
4.3	Zustandsdiagramme . . . . .	55
4.3.1	Zustandsdiagramm – GPSCheck . . . . .	55
4.3.2	Zustandsdiagramm – Favoriten hinzufügen . . . . .	57
4.4	Zerlegung in Teilsysteme . . . . .	58
<b>5</b>	<b>Implementierung</b>	<b>59</b>
5.1	Binäres Kartenformat . . . . .	59
5.1.1	Format . . . . .	60
5.1.2	Tags-Tabelle . . . . .	60
5.1.3	Erstellung und Benutzung der binär-codierten Datei . . . . .	61
5.2	Implementierungsdetails . . . . .	62
5.2.1	Klasse FavoriteStore . . . . .	62
5.2.2	Klasse ActualPosition . . . . .	63
5.2.3	Klasse AlertManager . . . . .	63
5.2.4	Klasse NodeManager . . . . .	65
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>69</b>
6.1	Zusammenfassung . . . . .	69
6.2	Ausblick . . . . .	72
<b>7</b>	<b>Anhang</b>	<b>73</b>
<b>A</b>	<b>Anwendungsfälle</b>	<b>75</b>
A.1	Lizenzbedingungen . . . . .	75
A.2	Favoriten-Verwaltung . . . . .	76
A.3	GPS-Status anzeigen . . . . .	80
A.4	Wo bin ich? (Nächstgelegene Adresse und Ort) . . . . .	81
A.5	Umgebungssuche . . . . .	82
A.6	Stichwortsuche . . . . .	85
A.7	Einstellungen . . . . .	86
A.8	Sonstiges . . . . .	89
	<b>Literaturverzeichnis</b>	<b>91</b>

# Abbildungsverzeichnis

2.1	Erklärung von Heading, Bearing und Course . . . . .	22
3.1	Paketdiagramm – Struktur der Pakete . . . . .	25
3.2	Klassendiagramm – View-Klassen . . . . .	29
3.3	Klassendiagramm – Main-Klasse . . . . .	32
3.4	Klassendiagramm – GPS-Paket . . . . .	33
3.5	Klassendiagramm – OSM-Paket . . . . .	34
3.6	Klassendiagramm – Util-Paket . . . . .	36
3.7	Klassendiagramm – Datastore-Paket . . . . .	38
3.8	Klassendiagramm – Entity-Paket . . . . .	40
3.9	Sequenzdiagramm – Starten der Anwendung . . . . .	42
3.10	Sequenzdiagramm – GPS-Daten lesen . . . . .	43
3.11	Sequenzdiagramm – Umgebungssuche . . . . .	44
3.12	Sequenzdiagramm – Stichwortsuche . . . . .	45
3.13	Sequenzdiagramm – Favoriten speichern . . . . .	46
3.14	Sequenzdiagramm – Punkt als Favoriten speichern . . . . .	47
3.15	Sequenzdiagramm – Beispiel für die Verwendung von SimpleInputView . .	48
4.1	Paketdiagramm – Abhängigkeiten der Pakete . . . . .	53
4.2	Zustandsdiagramm – GPSCheck . . . . .	55
4.3	Zustandsdiagramm – Favoriten hinzufügen . . . . .	57



# Kapitel 1

## Einleitung

Am 02.05.2000 wurde die „Selective Availability“ (SA) des amerikanischen satellitengestützten Positionierungssystems GPS abgeschaltet.<sup>1</sup> Unter SA versteht man die künstliche Verschlechterung der Satellitenbahndaten, sodass die Positionierungsgenauigkeit bei zivilen GPS-Empfängern nur bei etwa 100 Metern liegt. Durch das Abschalten wurde die Positionierung auf etwa zehn Meter genau. Spätestens seitdem erfährt die Verbreitung von GPS-Navigations-Geräten einen bedeutenden Anstieg.

Von der Routenführung bis hin zum Finden von so genannten *Orten von Interesse* ist für die Navigation mit dem Auto nahezu jeder erdenkliche Bereich abgedeckt. Allerdings vernachlässigen die meisten Hersteller die besonderen Bedürfnisse von Menschen mit Sehbehinderung, unter anderem, da diese in der Regel zu Fuß unterwegs sind. So sind zum Beispiel Orte von Interesse, die für Menschen mit Sehbehinderung von Bedeutung wären, wie etwa Bushaltestellen oder Straßenüberquerungen, häufig nicht in handelsüblichem Kartenmaterial enthalten. Auch die Routenführung für das Auto bezieht nicht sämtliche Informationen mit ein, die für FußgängerInnen - insbesondere für solche mit Sehbehinderung - relevant wären. So existiert zum Beispiel bei den wenigsten Navigationsgeräten eine Ausgabe der Namen der Straßen, in die man geführt wird, und Straßenüberwege werden gar nicht berücksichtigt.

Es gibt also einen Bedarf an Kartenmaterial und einer Routenführung, die den Bedürfnissen von Menschen mit Sehbehinderung gerecht werden. Erst das Abschalten der SA machte eine solche Routenführung möglich, doch bisher fehlt es an kostengünstigen Umsetzungen. Das OpenStreetMap-Projekt jedoch, welches sein Kartenmaterial unter einer freien Lizenz veröffentlicht, beinhaltet schon heute Orte von Interesse für Menschen mit Sehbehinderung und erweitert diese fortlaufend.

Dieses Buch befasst sich - unter Einbeziehung des erwähnten Kartenmaterials - mit der Konzeption und Implementierung einer freien Software, die speziell auf die Bedürfnisse von Menschen mit Sehbehinderung zugeschnitten ist.

---

<sup>1</sup> Vergleiche: [http://www.ngs.noaa.gov/FGCS/info/sans\\_SA/](http://www.ngs.noaa.gov/FGCS/info/sans_SA/)

### 1.1 Erläuterungen zur Gliederung

Die Gliederung dieses Buches orientiert sich größtenteils an den Schritten eines allgemeinen Software-Entwicklungsprozesses:

*Anforderungsanalyse - Analyse - Entwurf - Implementierung - Test*

Unterschiede finden sich in den Schritten „Anforderungsanalyse“ und „Test“: Die Anwendungsfälle, die sonst in der Anforderungsanalyse enthalten sind, befinden sich hier im Anhang, da ihr Inhalt bereits im Kapitel „Zielsetzung“ ausführlich behandelt wird. Des Weiteren werden die Ergebnisse des Schrittes „Test“ bereits im Kapitel „Zusammenfassung und Ausblick“ thematisiert.

### 1.2 Typografische Konventionen

Für dieses Buch gelten folgende typografische Konventionen:

- Besonders zu betonende Textstellen werden *kursiv* dargestellt.
- Der **Typewriter**-Stil gilt für Kommandos, Dateinamen, Pakete, Klassen, Attribute und Quelltexte im Allgemeinen.
- Querverweise werden in Kästchen mit abgerundeten Ecken dargestellt.



# Kapitel 2

## Anforderungsanalyse

Bereits im März 2009 begann die *Community* (siehe Informationsquellen(2.2.3)) damit, die Anforderungen, die an das System gestellt werden, zusammenzutragen. Daher sind alle *Entscheidungen und Begründungen*, die in diesem Kapitel zur Sprache kommen, bereits getroffen worden, bevor ich mich der Entwicklung des Systems annahm. Eine Bewertung dieser Entscheidungen und Begründungen erlaube ich mir im Ausblick.

### 2.1 Motivation

Beim Erfassen von Daten für eine Karte berücksichtigen die wenigsten KartografInnen Informationen, die für Menschen mit Sehbehinderung von Bedeutung sein könnten, weil sie davon ausgehen, dass die Karten optisch dargestellt werden. Daher fehlen handelsüblichen Karten Informationen wie Straßenüberquerungen, ob eine Bushaltestelle mit taktiler Pflasterung ausgestattet ist und Ähnliches. Zwar gibt es Möglichkeiten der nicht-visuellen Darstellung, jedoch sind derzeit am Markt verfügbare Systeme sehr teuer - beispielsweise kostet der *Trekker* umgerechnet ungefähr 2.300€<sup>1</sup> - und bieten weder blindenspezifische Punkte in ihren Kartendaten noch die Möglichkeit, diese selbst hinzuzufügen.

Daraus folgt auch, dass neue Objekte anderen Nutzenden nicht verfügbar gemacht werden können. Die Motivation für die Entwicklung eines neuen Systems liegt darin, diese Mängel zu beheben.

Bereits heute gibt es das *OpenStreetMap*-Projekt, welches sich um eine freie Weltkarte bemüht. In dieser Karte ist es möglich, Daten zu ändern und neue Daten hinzuzufügen. Auch bietet das geodatensammelnde Projekt schon jetzt einige blindenspezifische Informationen. Jedoch gibt es kaum Möglichkeiten für Menschen mit Sehbehinderung, am Projekt unmittelbar mitzuwirken.

---

<sup>1</sup> <http://www.abacuscity.ch/abashop?p=hierarchyoutline&hi=1290.00&hl=2&i=0U5KKcFL7h1h2KA0xCWD&s=32>

### 2.2 Zielsetzung

Das *Ziel* ist, ein mobiles System zu entwickeln, das es Menschen mit Sehbehinderung ermöglicht, die freien Daten aus dem OpenStreetMap-Projekt zu nutzen und diese Daten außerdem selbst zu editieren und zu erweitern. Es soll den Namen *LoroDux* tragen. Er setzt sich zusammen aus dem spanischen Wort für Papagei (Loro) und aus dem lateinischen Wort für Führer (Dux).

#### 2.2.1 Abgrenzung

Die *Kernanforderungen* wurden von der Community in mehrere Ausbaustufen unterteilt. Dieses Buch befasst sich ausschließlich mit der ersten Ausbaustufe. Daher findet sich in diesem Kapitel keine Beschreibung der für spätere Versionen erforderlichen Anforderungen.<sup>2</sup>

#### 2.2.2 Funktionale Anforderungen

Beim Starten der Anwendung muss ein Hinweis auf die verwendete *Lizenz* der Kartendaten angezeigt werden. Diese Funktion darf nicht abschaltbar sein.

Das System soll speziell für die Bedürfnisse von *Menschen mit Sehbehinderung* entwickelt werden. Daher ist es erforderlich, dass *keine Grafiken* verwendet werden. Die Ausgabe darf nur aus Text bestehen, der von einem Screenreader gelesen werden kann.

Ein *Screenreader* ist eine Anwendung, die im Hintergrund läuft und vorliest, was auf dem Display eines Gerätes als Text dargestellt wird. Für eine solche Software ist die Erkennung von Text in Bildern nicht möglich. Zum einen ist der rechnerische Aufwand zu groß, als dass ein mobiles Endgerät eine Texterkennung durchführen könnte. Zum anderen ist die Auflösung der Grafiken aufgrund des kleinen Displays zu gering, um eine hohe Erkennungsrate der Buchstaben zu erzielen. Konkret wird vorausgesetzt, dass mindestens der Screenreader „Talks“ der Firma Nuance unterstützt wird.

Es soll die Möglichkeit bestehen, sich ausgeben zu lassen, wo man sich gerade befindet. Dies soll mittels Angabe von Straße und Hausnummer sowie Ort beziehungsweise Stadtteil geschehen. Falls keine Hausnummer vorhanden ist, soll die nächste Kreuzung oder Einmündung angegeben werden.

Mindestens für Testzwecke soll es möglich sein, sich die Koordinaten ansagen zu lassen, die gerade vom GPS-Empfänger kommen.

Es soll möglich sein, *persönliche Favoriten* sowohl über die direkte Eingabe von Name und geografischen Koordinaten als auch durch einen vorhandenen Punkt aus den Kartendaten abzulegen.

Es soll möglich sein, sowohl die *Distanz* zu einem Punkt aus den Kartendaten als auch die *Richtung*, in der er sich befindet, berechnen zu lassen. Dies gilt auch für persönliche Favoriten. Die Richtung soll dabei in Himmelsrichtungen oder auf der Uhr, also relativ

---

<sup>2</sup> <http://wiki.openstreetmap.org/wiki/LoroDux/Requirements>