



**Fonds für Unterrichts- und Schulentwicklung  
(IMST-Fonds)**

**S1 „Lehren und Lernen mit neuen Medien“**

---

# **PROGRAMMIEREN VON JAVA-ANWENDUNGEN FÜR MOBILTELEFONE**

**ID 532**

**Mag. Walther M. Stuzka**

**Mag. Andreas Resetarits**

**PG Neulandschule 1100 Wien  
L.-v.-Höhnel-G. 17-19 (910026)**

Wien, Juli 2007

# INHALTSVERZEICHNIS

<b>INHALTSVERZEICHNIS</b> .....	<b>2</b>
<b>1 EINLEITUNG</b> .....	<b>4</b>
1.1 Ausgangssituation.....	4
<b>2 ZIELE DES PROJEKTS</b> .....	<b>5</b>
2.1 Allgemeine Ziele .....	5
2.2 Persönliche Ziele .....	6
<b>3 UNTERSUCHUNGSFRAGEN</b> .....	<b>7</b>
3.1 Warum programmieren? .....	7
3.2 Warum JAVA? .....	7
3.3 Erwartungen.....	8
<b>4 PROJEKTVERLAUF</b> .....	<b>9</b>
4.1 Werkzeuge.....	9
4.1.1 Soft- & Hardware .....	9
4.1.2 Courseware.....	10
4.2 Unterricht .....	11
4.3 Seitenblicke.....	12
4.4 Lehrerfortbildung.....	13
4.5 Finanzielles .....	13
<b>5 FEEDBACK – EVALUATION &amp; REFLEXION</b> .....	<b>14</b>
5.1 Feedback .....	14
5.2 Evaluation & Reflexion .....	15
<b>6 LITERATUR</b> .....	<b>16</b>

## ABSTRACT

*Das IMST-Projekt "Programmieren von JAVA-Anwendungen für Mobiltelefone" untersucht Möglichkeiten und Wege, die genannte Thematik in den Informatikunterricht der Oberstufe einzubauen. Die praktischen Erfahrungen der Autoren in diesbezüglichem Unterricht einer Wahlpflichtfachgruppe (6. Klasse) sind festgehalten. Informationen, betreffend die Installierung und Handhabung der notwendigen Soft- und Hardware werden gegeben. Eine reichhaltige Sammlung von (unterrichtsrelevanten) JAVA-Programmen für Mobiltelefone, Ausarbeitungen, Erklärungen und Verweisen auf andere Quellen sind in der im Zuge des Projektes aufgebauten Internetplattform <http://www.nls.at/~jam> zu finden.*

Schulstufe: 10

Fächer: Informatik

Kontaktperson: Mag. Walther M. Stuzka

Kontaktadresse: PG Neulandschule, 1100 Wien, L.-v.-Höhnel-G. 17-19, 910026

# 1 EINLEITUNG

Programmieren von JAVA-Anwendungen für Mobiltelefone soll in den Informatik-Unterricht (Wahlpflichtfach) eingebaut werden. Warum? Es erscheint uns, den Projektmitarbeitern und Autoren dieses Berichts, interessant und für die Schüler motivierend, mit einem neuen Medium, mit einem komplexen, technischen Gerät zu arbeiten, das in der modernen Jugendkultur einen sehr hohen Stellenwert hat. Lernen, aber auch Lehren kann auf diese Weise eine neue Dimension erhalten.

## 1.1 Ausgangssituation

Zu Beginn des Schuljahres (2006/07), unmittelbar vor dem Projektstart stellte sich heraus, dass die für das Projekt vorgesehene Wahlpflichtfachgruppe (InformatikschülerInnen der achten Klassen mit zwei Jahren Erfahrung im Programmieren - ein Jahr davon JAVA für Applikationen und Applets) zusammengebrochen war und mit einer Wahlpflichtfachgruppe von SechstklasserInnen gearbeitet werden musste.

Die neue Gruppe bestand aus SchülerInnen des Gymnasial- bzw. des Realgymnasialzweiges. Alle SchülerInnen hatten den Pflicht-Informatik-Unterricht der fünften Klasse absolviert, waren aber, was das Programmieren - im speziellen das Programmieren in JAVA - betrifft, zum überwiegenden Teil AnfängerInnen. Darüber hinaus war die Gruppe, was die Erfahrung im Umgang mit Computern betrifft, nicht sehr homogen. JAVA, als ein immer häufiger anzutreffendes Feature von modernen Mobiltelefonen, war nahezu allen SchülerInnen bekannt, wenngleich auch nur als ein Instrument (um vor allem Spiele ausführen zu können).

Die Projektmitarbeiter hatten zum Zeitpunkt der Projekteinreichung zwar (Unterrichts-)Erfahrung mit JAVA, aber nur geringe Erfahrung mit JAVA für Mobiltelefone.

## 2 ZIELE DES PROJEKTS

### 2.1 Allgemeine Ziele

Das Projekt verfolgte mehrere Ziele, die stufenweise erreicht werden sollten. Zuerst sollten den SchülerInnen der Wahlpflichtfachgruppe die grundlegende Technik des Programmierens von JAVA-Anwendungen für Mobiltelefone vermittelt werden.

Im Weiteren wurde versucht, eine Eigendynamik anzustoßen, die die SchülerInnen dazu bewegt, gemeinsam mit uns, das Mobiltelefon als Lernhilfelinstrument zu entdecken und Anwendungsideen zu entwickeln, die aus der, nur jungen Menschen eigenen, intensiven Verwendung des Mobiltelefons resultieren.

Als ein innerhalb der offiziellen Projektlaufzeit erreichbares, konkretes Ziel erschien es uns, gemeinsam mit unseren SchülerInnen jene Fertigkeiten zu entwickeln, die es ermöglichen, auf einem Mobiltelefon, eigene Texte zu verwalten (speichern, editieren, löschen), eigene Grafiken anzuzeigen und eigene Tonobjekte (Musik, Sprache) ein- und abzuspielen.

Erklärtes Endziel war und ist – da bis dato noch nicht vollständig erreicht – die Implementierung der Internetplattform <http://www.nls.at/~jam> (jam = JAVA-Anwendungen für Mobiltelefone). Diese Website soll interessierten SchülerInnen und LehrerInnen als reichhaltige Quelle zur Thematik "Programmieren von JAVA-Anwendungen für Mobiltelefone" dienen. Ein von den Projektmitarbeitern verfasstes Tutorium, Musterbeispiele mit Quelltexten, Links zu anderen, themenähnlichen Websites sowie die gesamten Projektunterlagen stehen bereits zur Verfügung und werden auch in Zukunft ergänzt, erweitert und gewartet. Im günstigen Fall entwickelt sich diese Plattform dahingehend, dass BenutzerInnen in einem noch einzurichtenden Forum, Informationen und Meinungen posten sowie Files down-, aber auch uploaden können.

Unser Vorsatz war, dass bei allen Vorhaben, SchülerInnen mit verschiedenen gestalterischen und verwaltenden Agenden betraut werden sollten, besser noch, sich dieser selbst annehmen bzw. sich selbst welche finden, also selbstständig und eigenverantwortlich arbeiten sollten.

Letztendlich sollte das Projekt ein wenig mithelfen, mobile Technologien in unserer bzw. in der Schule überhaupt "salonfähig" zu machen, wenn man eindeutige Vorteile für das Lernen entdeckt bzw. herausarbeitet. (Derzeit ist in Schulen und unter Schulverantwortlichen eine Polarisierung von Handy-Verbot bis Tolerierung, kaum aber proaktive Förderung festzustellen.)

Da in MNI-Projekten stets der konkrete Unterricht im Auge behalten werden soll, wurden Untersuchungen und Evaluierungen betreffend die Entwicklung der Empfindlichkeiten der SchülerInnen, aber auch der beiden Projektleiter, über die gesamte Arbeitsperiode hinweg geplant (siehe 5. Feedback – Evaluation & Reflexion).

## 2.2 Persönliche Ziele

Neben den eben formulierten, allgemeinen Projektzielen, verfolgten die Projektmitarbeiter (natürlich) auch persönliche Ziele.

Um einen anspruchsvollen Informatikunterricht bieten zu können, der mit seinen Inhalten auf der Höhe der Zeit ist, bedarf es ständiger Weiterbildung. Eine Form davon ist das Selbststudium und diese(s) muss vor allem dann praktiziert werden, wenn Angebote von außen nur in geringen Maßen oder garnicht vorhanden sind. Die Aussicht, durch mehr fachliches Können und Wissen kompetent zu sein, unseren Unterricht zu professionalisieren, war und ist uns sicherlich eine sehr große Motivation. Diese lässt sich – zumindest gilt das für uns – noch vergrößern, indem wir zu einem Thema, das uns am Herzen liegt, einen Artikel in einer (wissenschaftlichen) Zeitschrift verfassen, ein Projekt (wie dieses) durchführen, eine öffentlich zugängliche Internetplattform einrichten oder ein Seminar für die Lehrerfortbildung anbieten (siehe 4.3 Lehrerfortbildung). Durch diese Formen der Öffnung kann eine Breitenwirkung erzeugt werden, um Rückmeldungen und neue Impulse zu erhalten und um inhaltliche Verbesserungen zu erzielen.

## 3 UNTERSUCHUNGSFRAGEN

### 3.1 Warum programmieren?

Die Antwort auf die Grundsatzfrage „Warum programmieren?“ holen wir uns bei Donald E. Knuth. In seinem berühmten Sammelband „The Art of Computer Programming“ schreibt er: *„The process of preparing programs for a digital computer is especially attractive, not only because it can be economically and scientifically rewarding, but also because it can be an aesthetic experience much like composing poetry or music.“*

Während Programmieren im Lehrplan der AHS im Pflichtfach Informatik für die fünfte Klasse nicht vorgesehen ist, ist im Lehrplan für das Wahlpflichtfach (6. – 8. Klasse) z. B. von *„Konzepten von Programmiersprachen“*, *„grundlegenden Algorithmen und Datenstrukturen“* und *„formaler Modellierung von Sachverhalten, welche aus Analyse, Beschreibung in verschiedenen Darstellungsformen, Implementation, Überprüfung und Interpretation besteht“* zu lesen. Diese Bildungs- und Lehraufgabe kann unserer Meinung nach durch praktisches Programmieren in einer modernen Programmiersprache besonders gut erfüllt werden.

### 3.2 Warum JAVA?



JAVA ist eine von SUN-Microsystems <http://www.sun.com> ursprünglich für das Programmieren und Vernetzen von Geräten des täglichen Lebens entwickelte Sprache. (Schlagwort "Smart Home". Eine Wohnung mit elektronisch vernetzten Geräten wie z. B. Klimaanlage, Rollläden, Waschmaschine, Kühlschrank.) JAVA als Plattformen übergreifende Programmiersprache für Applikationen (damit meint man eigenständige PC-Programme), für Applets (JAVA-Programme, die von einem Web-Browser zur Ausführung gebracht werden) und zuletzt auch für MIDlets (JAVA-Programme, die in einem Mobiltelefon laufen. MID = Mobile Information Device. MID-Applets → MIDlets) haben in den vergangenen zehn Jahren, begünstigt von Internet-Boom und Handy-Hype, große Bekanntheit erlangt.

Lautet eine häufig gestellte Frage bei der Auswahl eines Mobiltelefons „Ist dieses Handy JAVA-fähig?“, so stellten wir uns die Frage „Ist JAVA für Mobiltelefone ein geeignetes Thema für den Informatikunterricht (16 – 18 jähriger AHS-SchülerInnen)?“

In diesem Zusammenhang interessierte uns:

- Was ist im Rahmen des Informatikunterrichts (zwei Stunden pro Woche) auf dem Gebiet JAVA-Programmierung für Mobiltelefone überhaupt möglich?
- Sind (unsere) SchülerInnen bereit, ihr „geliebtes“ Mobiltelefon von einer ganz anderen Seite als der des reinen Bedienens und Anwendens, kennen zu lernen? Einer Seite nämlich, die sich sowohl mit programmiertechnischen Aspekten als auch mit Problemen der Informationsdarstellung – einer, wie wir meinen, Kreativität und Phantasie erfordernden Tätigkeit – befasst.
- Wie schnell kann man beim Programmieren von Mobiltelefonen mit JAVA erste Erfolgserlebnisse vermitteln?

- Welche Werkzeuge (hard- & software-tools) gibt es zu welchem Preise und wie kompliziert sind sie (auf den Arbeitsplätzen unseres Schulnetzwerks zu installieren bzw. von SchülerInnenhand zu bedienen)?
- Ändert sich etwas im Lernverhalten der SchülerInnen, wenn es im Unterricht um ein so wichtiges Objekt wie das e i g e n e Handy geht?
- Motiviert unsere SchülerInnen die Arbeit an dem Projekt, auch über den Einsatz ihres Handys als Lerninstrument nachzudenken, vielleicht sogar eigene Ideen in Richtung "e-learning & micro-devices" zu entwickeln?

### 3.3 Erwartungen

Die Hypothese ist: Mobiltelefone können im Informatikunterricht sinnvoll eingesetzt werden. Vor allem das Programmieren eben dieser, in der modernen Jugendkultur allgegenwärtigen und mit einem hohen Stellenwert behafteten Geräte, könnte für SchülerInnen interessant sein, ihre Kreativität fördernd und somit ertragreich. Lehrer könnten es schätzen, den mühsamen Niederungen des Selbststudiums früher zu entkommen und auf erprobte und didaktisch aufbereitete Unterrichtsmaterialien zurückgreifen zu können.

Zu diesem Zweck soll die Website <http://www.nls.at/~jam> interessierten SchülerInnen und LehrerInnen als Quelle zur Thematik "Programmieren von JAVA-Anwendungen für Mobiltelefone" dienen.

# 4 PROJEKTVERLAUF

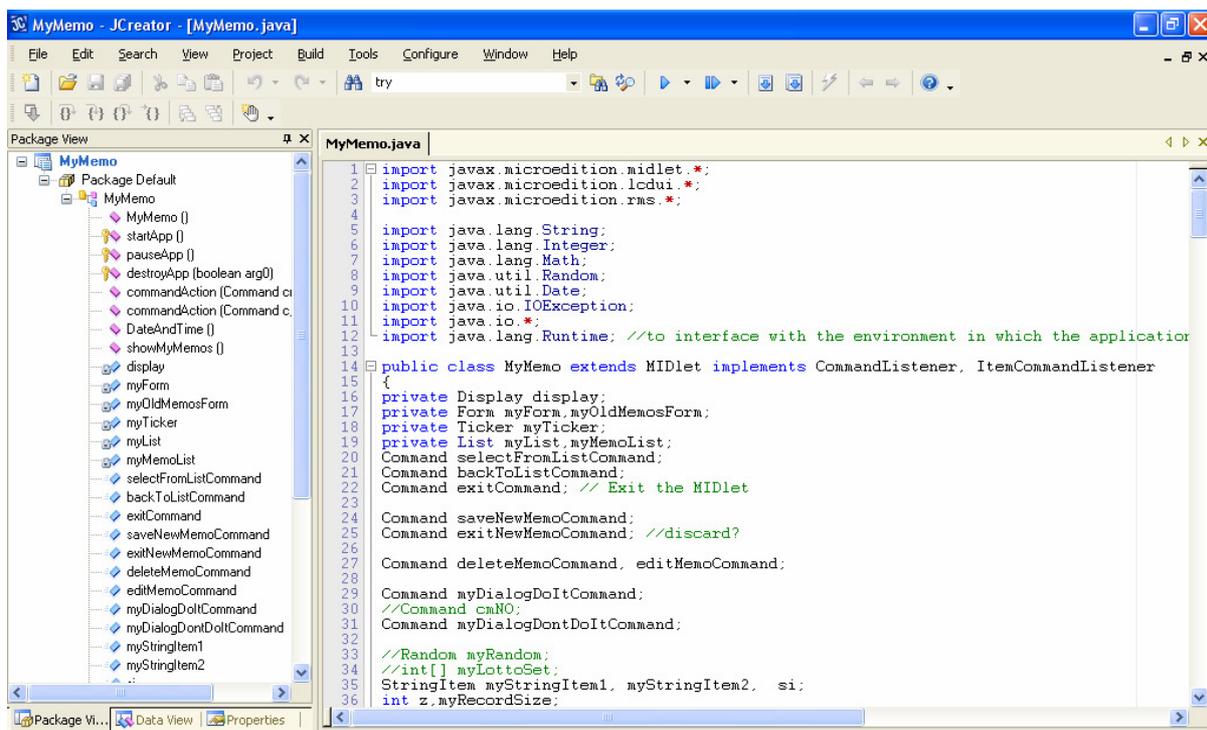
## 4.1 Werkzeuge

### 4.1.1 Soft- & Hardware

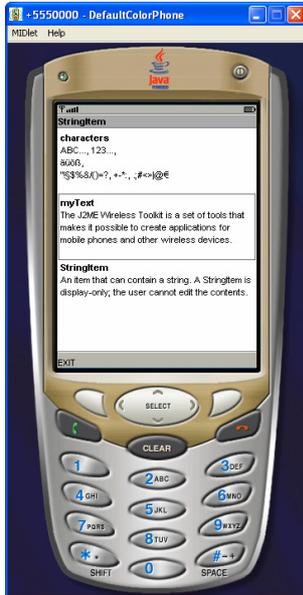
Damit auf einem PC (mit dem Betriebssystem MS Windows) JAVA-Programme ausgeführt werden können, muss das JRE (JAVA Runtime Environment) installiert sein. Damit man JAVA-Programme selbst entwickeln kann, benötigt man das JAVA SDK (Software Development Kit), auch unter dem Namen J2SE (JAVA 2 Standard Edition bekannt). Für die Entwicklung von Anwendungen für Mobiltelefone benötigt man die J2ME (JAVA 2 Micro Edition). Alle drei Softwarepakete können von <http://java.sun.com/> frei heruntergeladen werden.

Um JAVA nicht von der Konsole aus bedienen zu müssen, verwenden wir einer IDE (Integrated Development Environment). Im (von uns für die Arbeit mit SchülerInnen favorisierten) Freeware-Bereich, bieten sich da mehrere Produkte an. Netbeans (<http://www.netbeans.org> ein Open-Source-Projekt der Firma Sun), der Java Studio Creator 2 (<http://developers.sun.com/prodtech/javatools/jscreator/>) sowie Eclipse (<http://www.eclipse.org/>), ein IBM-Offspring, erschienen für unsere Zwecke zu umfangreich und komplex. BlueJ (<http://www.bluej.org/>) ist ein aus dem universitären Bereich stammendes JAVA-Lernwerkzeug, mit dem AnfängerInnen die Grundlagen der objektorientierte Programmierung auf einem besonderen Weg nähergebracht werden soll.

Der von uns letztlich als IDE gewählte JCreator (<http://www.jcreator.com>), in der freien Variante JCreator Light, erschien uns aus folgenden Gründen als die richtige Wahl: Die IDE macht einen übersichtlichen, aufgeräumten Eindruck (Abb. unten) und ihre Bedienung ist einfach zu erlernen.



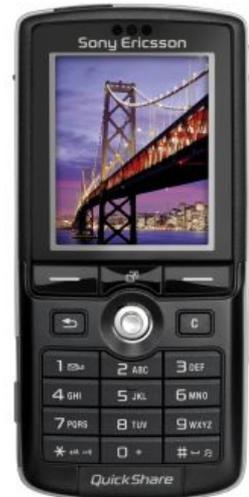
Das Programm unterstützt Konzepte wie Project- und Code-Templates, also wieder-einsetzbare - auch selbst generierte und erweiterbare - Muster und Schablonen, was gerade für AnfängerInnen ein zügiges und rasch Erfolg bringendes Arbeiten ermöglicht. Neben der Möglichkeit JAVA-Applikationen und JAVA-Applets zu erzeugen, ist es einfach, die zum Kompilieren von JAVA-Anwendungen für Mobiltelefone (MIDlets) benötigte JAVA-Microedition (J2ME) zu integrieren.



Mit einem Mobiltelefon-Emulator (Abb. links) können die entwickelten Programme, bevor sie auf das Mobiltelefon übertragen werden, auf einem PC getestet werden. Wir verwenden die KToolbar aus dem J2ME Wireless Toolkit - freier Download von <http://java.sun.com/javame/>.

Die genannten Softwarepakete wurden von den Projektmitarbeitern getestet und auf den Rechnern des Schulnetzwerks installiert.

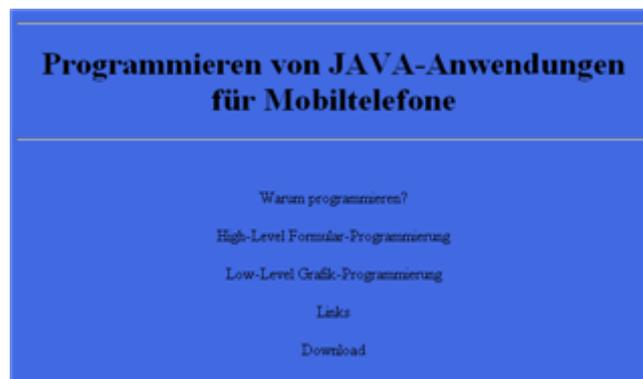
Funktioniert ein MIDlet in der Emulatorumgebung, kann/muss es auf ein Mobiltelefon übertragen und dort eingehen getestet werden. Als sehr praktisch für den Transfervorgang erwies sich die Bluetooth-Technik.



Damit konnte (für den Unterricht) eine Umgebung geschaffen werden, in der MIDlets von einem PC auf ein (Bluetooth fähiges) Handy, bzw. von Handy zu Handy (kabellos) überspielt werden konnten. Wir, die Projektmitarbeiter, verwendeten zu Testzwecken ein gängiges, auf dem Markt erfolgreiches, JAVA-fähiges Handy, ein Sony-Ericsson K750i (Abb. rechts oben).

## 4.1.2 Courseware

Alle von den Projektmitarbeitern erstellten Lehr- und Lernmaterialien sind (seit Juli 2007) im Internet unter <http://www.nls.at/~jam> (Screenshot Abb. unten) öffentlich zugänglich. Erweiterungen sind geplant, der Erhalt und die Pflege der Website über das offizielle Ende des IMST-Projekts hinaus ist Vorsatz.



## 4.2 Unterricht

Wie eingangs erwähnt, musste die Projektarbeit aus organisatorischen Gründen an eine Wahlpflichtfachgruppe von SechstklasserInnen, also überwiegend ProgrammieranfängerInnen übergeben werden. Es standen zwei Unterrichtseinheiten à 50 Minuten pro Woche zur Verfügung. Während der ersten drei Monate mussten wir unseren SchülerInnen daher die Grundlagen der Programmierung überhaupt und jene der Programmierung in JAVA im Speziellen vermitteln. Durch die Beschränkung auf einige wenige, grundlegende Konzepte (z. B. wurden (vorerst) nur eine Schleifen-Konstruktionen oder nur Einfach-Verzweigungen besprochen), durch einen Schwerpunkt im Bereich von Grafik-Applets und dadurch, dass wir Programm-Muster (Templates) zur Verfügung stellten, konnten wir die Einarbeitungszeit auf 9 Doppelstunden drücken. (Schüler sind also wirklich weit belastbarer, als man gemeinhin annimmt!) Ein eingehendes Studium von Konzepten der Objektorientierten Programmierung (wie z. B. im Buch von Barnes und Kölling, „JAVA lernen mit BlueJ“ (siehe 6. Literatur) dargeboten oder <http://www.u-helmich.de>, Menüpunkt Informatik) konnte so (leider) nicht durchgeführt werden.

Dies Betonung der Programmierung im grafischen Bereich begründen wir wie folgt: Das JAVA MID-Profil (MID = Mobile Information Device) zerfällt in zwei Bereiche: High-Level-API (unterstützt Formulare und Formularelemente) und Low-Level-API (ermöglicht direktes grafisches Gestalten auf dem Display eines Mobiltelefons). Gerade die Low-Level-API erweist sich vom Konzept her sehr ähnlich der J2SE-API zur Programmierung von einfachen JAVA-Applets und Applikationen.

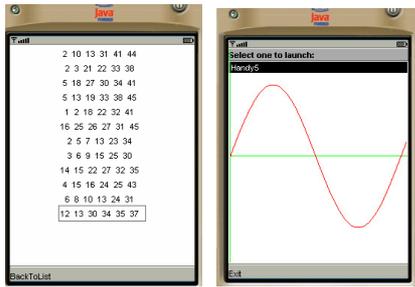
Dann, im Jänner 2007, viel später als ursprünglich geplant, konnte mit dem Programmieren von JAVA-Anwendungen für Mobiltelefone begonnen werden. Die Werkzeuge waren jedenfalls einsatzbereit. Parallel dazu stellten wir unser Projekt im Informatikunterricht der fünften Klassen sowie SchülerInnen der Informatikübungsgruppe der dritten und vierten Klassen vor, in der Hoffnung Anwendungsideen von "Handy-Powerusern" zu bekommen.

Wir erhielten u. a. folgende Vorschläge: Handy-Dilemma (Ist das Handyverbot in der Hausordnung ein Widerspruch zur einer am Handy veröffentlichten Hausordnung?), Buzzer (selbst gesteuerte Handy-Vibrationen), Selbstabschaltung via Timertask, Vokabeltrainer, Sammlung wichtiger Zugangsdaten (Passwörter, PIN-Codes, Lösungsworte), Cocktail- und Eismanager (Der nahe unserer Schule gelegene Salon lässt grüßen.), Krawattenbinder, Moonlander, Anwesenheitsliste (elektron. Klassenbuch).

Den Unterricht selbst hielten wir in dieser Phase (teilweise) im Team ab. Abwechselnd trug ein Lehrer vor, der andere stand den SchülerInnen als Helfer in der Not zur Verfügung. (Schätzten die Jugendlichen diesen Aufwand, diesen Luxus? Siehe 5. Evaluation)

Als große Belastung für die SchülerInnen erwies sich nämlich die praktische Arbeit mit zwei, wenn man die ständige Verwendung des Windows-Explorers (für Dateitransfers) und die eines Web-Browsers (für Recherchen und Nachschlagen in der JAVA-API) dazuzählt, mit vier Anwendungen. Werden die MIDlet-Quellcodes im (gewohnten) JCreator erstellt, so müssen sie dann in den Arbeitsbereich des Handy-Emulators KToolbar übertragen werden. Der Festplattenschutz (HD-Guard) der

Schulrechner erfordert von den SchülerInnen ein verlässliches Abspeichern der erstellten Dateien in deren persönlichen Ordner(bereich).



Langsam wuchs eine Sammlung von JAVA-Midlets. Zunächst und immer wieder waren es kleine Programmsequenzen, die jeweils ein wichtiges Element der Datenverarbeitung, z. B. eine Datenstruktur, einen Algorithmus oder eine spezielle BenutzerInneninteraktion zum Thema hatten. Im Weiteren entstanden bereits (einfache) Anwendungen. Z. B. ein textbasierter Zufallszahlengenerator oder ein grafikbasierter Funktionsplotter (Abb. links).

Nicht zu unterschätzen ist der Aufwand, betreffend die Einarbeitung in die Bedienung der (JAVA-) Komponenten eines Mobiltelefons (Speicherkarte, Methoden des Transfers der Daten (OTA (= Over The Air), Bluetooth, Kabel, Cardreader), Speichern von Daten auf der Speicherkarte bzw. im Handy, Installieren der Anwendungen auf dem Gerät. Funktionstests der Anwendungen müssen oft produkt(gruppen)spezifisch durchgeführt werden, SchülerInnenfragen in diesen Angelegenheiten konnten aus diesem Grund ebenfalls oft nur das jeweilige Handy betreffend (und mühsam) geklärt werden.

### 4.3 Seitenblicke

Besonders interessant für uns ist, dass man gerade bei der Recherche im Internet immer wieder über Tools stolpert, die einerseits Arbeiten, die man wegen ihrer Komplexität bisher gescheut hat, sehr erleichtern, andererseits den Blickwinkel auf völlig neue Konzepte lenken. Drei Beispiele: Videora iPod Converter (<http://www.videora.com>): ein Freeware-Tool, das es erlaubt, mp4-Filmchen für das Handy zu erstellen. MaxDox (<http://www.Maxdox.org>): ein Mobile-Designer-Tool, ebenfalls Freeware. Mikro-Lern-Technologien durch Seibersdorf Research (<http://www.knowledgepulse.com>).

Sehr informativ erscheint uns in diesem Zusammenhang der Artikel "Mobile Technologien machen Schule", Ch. Dorninger in PCNEWS 102, Feb. 2007, Seite 9 ff. (siehe 6. Literatur)

Die Zusammenarbeit mit einem Mobilfunkbetreiber – geeignete Verbindungen (ev. via IMST-Netzwerk) zu Entscheidungsträgern vorausgesetzt – könnte helfen, das Projekt in der Zukunft weiter zu entwickeln.

## 4.4 Lehrerfortbildung

Im Rahmen der Informatikwochen Wien (IWW) des PI-Wien <http://www.pi-wien.at> wurde der LehrerInnen-Fortbildungskurs "JAVA & Mobile Devices - Programmieren von JAVA-Anwendungen für Mobiltelefone" angeboten und in der Zeit vom 4. bis 5. Juli 2007 (8:30 - 16:00) in den Informatikräumlichkeiten der Neulandschule abgehalten.

Bei dieser Veranstaltung konnte die in ihrer Grundstruktur fertiggestellte Internetplattform <http://www.nls.at/~jam> einem Fachpublikum präsentiert und von diesem eingehend getestet werden.

Im Zuge des Fortbildungskurses wurden vor allem jene Unterrichtssequenzen und JAVA-MIDlets vorgestellt, die wir bereits an unseren SchülerInnen erprobt bzw. mit ihnen thematisiert hatten.

## 4.5 Finanzielles

Was die Finanzen des Projekts betrifft, wurden folgende Mittel aufgewendet:

Ein JAVA-fähiges Mobiltelefon (Sony-Ericsson K750i) wurde im November 2006 angeschafft. Der Vertrag für 24 Monate, 5 € pro Monat Grundgebühr, ergibt Gesamtkosten von 120 €, die in dankenswerter Weise von IMST3 übernommen wurden. (Diese Vorgehensweise ist wirtschaftlicher als der Kauf des "vertragslosen" Mobiltelefons um ca. 500 €.)

Die Bücher "Lehrer erforschen ihren Unterricht", H. Altrichter, P. Posch, 20,40 €, und "Web Usability", J. Nielsen und H. Loranger, 41,10 €, sowie „JAVA lernen mit BlueJ“, David J. Barnes und Michael Kölling, 41,10 € und „JAVA-ME“, Ulrich Breyman und Heiko Mosemann, 41,10 € wurden gekauft. (siehe 6. Literatur)

Um 69,70 € wurde eine 320-GB-Festplatte für den Internetserver besorgt.

Somit ergaben sich Gesamtkosten in der Höhe von 333,40 €.

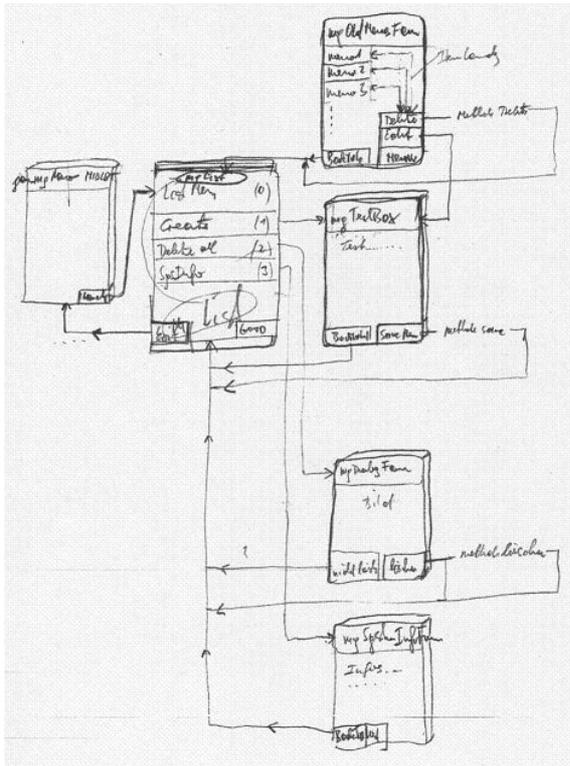
An dieser Stelle sei nochmals erwähnt, dass für die JAVA-Programmierarbeit ausschließlich frei, d. h. kostenlos über das Internet downloadbare Software verwendet wurde. Damit wird die Open-Source-Initiative des Bundesministerium für Unterricht, Kunst und Kultur (BMUKK) unterstützt, die den Einsatz von freier Software in Schulen fördern will. Des weiteren gibt es durch den Einsatz von (kosten)freier Software zumindest theoretisch keine Hürde, die SchülerInnen davon abhält, die im Schulunterricht verwendeten Programme auch zu Hause zu installieren und an ihren Projekten weiterzuarbeiten.

# 5 FEEDBACK – EVALUATION & REFLEXION

## 5.1 Feedback

Feedback von unseren SchülerInnen erhielten wir sowohl in vielen Einzelgesprächen als auch durch Gespräche im Plenum. Förderlich dafür war zweifelsohne die kleinen Gruppengröße (4 Mädchen / 10 Knaben).

Inhaltlich bot das SchülerInnen-Feedback folgendes:



Einige SchülerInnen fühlten sich überfordert (vergl. Abb. links), einige meinten "es war so schwierig", verstanden die Aussage aber auch in jenem positiven Sinn, dass es „eine Herausforderung war“.

"Vieles hat nicht auf Anhieb funktioniert." – Eine Tatsache, mit der wir Lehrer durchaus leben konnten, hatten wir es doch mit ProgrammieranfängerInnen zu tun, die mit für sie ganz neuen und durchaus komplexen Werkzeugen arbeiten mussten.

"Es hat manchmal so lange gedauert." Damit sollte dem Faktum Rechnung getragen werden, dass gerade die interessanteren Programme unmöglich in einer Doppelstunde erstellt und zum Laufen gebracht werden können/konnten, den SchülerInnen also ein ordentliches Maß an Durchhaltevermögen abverlangt.

Abb. oben: Skizze betreffend Überlegungen zur Ablaufsteuerung des MIDlets „my-memo“.

Die "große Hilfsbereitschaft der Lehrer" wurde gelobt. Unsere SchülerInnen wussten unseren großen Einsatz erfreulicher Weise zu schätzen.

"Die Handys waren cool." – Das hatten wir erhofft (bzw. insgeheim angenommen?).

Als das schönste Programm wurde die Analog-Uhr gekürt. (Gerade die wirklich schon etwas schwierigeren Programme begeistern (natürlich) am meisten.)

Gender-sensitiv(e) ist Programmieren auch. Mädchen möchten eher Programme, die ästhetisch ansprechend sind und z. B. interessante Farb- und Musterkombinationen haben, Knaben bevorzugen Programme, "in denen sich etwas tut" (Animationen, Spiele) bzw. Programme, die "versteckte Tricks" wie z. B. Geheimschalter und spezielle, nicht allgemein bekannte Tastenkombinationen haben.

Feedback erhielten wir natürlich auch von LehrerInnenkollegInnen im Zuge des PI-Fortbildungskurses "JAVA & Mobile Devices - Programmieren von JAVA-Anwendungen für Mobiltelefone" (siehe 4.4 Lehrerfortbildung). Die Idee, sich dieser Thematik

angenommen zu haben wurde gut geheißen und begrüßt. Die Tatsache, dass eine den Unterricht begleitende Didaktik sowie im Internet verfügbare Unterlagen (siehe <http://www.nls.at/~jam>) geboten wurden, fand sehr positives Echo und gibt Mut, auf dieser Schiene weiter zu arbeiten. Die Einbindung der JAVA-Handy-Programmierung in den Unterricht wurde von den KollegInnen als sehr wahrscheinlich (für das nächste Schuljahr) in Erwägung gezogen.

## 5.2 Evaluation & Reflexion

Ganz real und intensiv fand Evaluation und Reflexion einmal wöchentlich nach der Informatik-Wahlpflichtfach-Doppelstunde statt. Dabei besprachen wir, die Projektmitarbeiter, den Ablauf der (eben) abgehaltenen Unterrichtseinheit. Gab es Probleme bei Hard- und/oder Software, Eigenfehler, klassische und/oder schwer zu entdeckende Fehler (in Programmquellcodes) der SchülerInnen, Verständigungsschwierigkeiten? Wie waren die Rückmeldungen der SchülerInnen? Was kam an, was stieß auf Ablehnung? Wollten die SchülerInnen nur belehrt werden oder (bereits) selbstständig und problemorientiert arbeiten?

Meist folgt dann noch ein gegenseitiges Vorführen eigener, neu erstellter Programmsequenzen inklusive anschließender Besprechung von darin entdeckten Ungereimtheiten sowie das Überlegen möglicher Verbesserungen und die Grobplanung der nächsten Unterrichtseinheit.

Lauffähige, also fehlerfreie Programme stellen eo ipso ein gelungenes Produkt dar. Darüber hinaus bedeuten SchülerInnen, die im Rahmen des Unterrichts mit den einzusetzenden Werkzeugen zurechtkommen einen nicht zu unterschätzenden Erfolg. SchülerInnen, die die Initiative ergreifen und eigene, außerhalb(!) des Unterrichts erstellte Programme vorlegen, blieben (leider nur) unser Wunschdenken.

Der Ansatz, die SchülerInnen nur mit einfachen, nicht zu umfangreichen Programmen zu konfrontieren, um so ein Maximum an Erfolg(erlebnissen) zu initiieren, hat sich unserer Meinung nach bewährt. Vielfalt und "Highlights" hatten stets Vorrang vor Vollständigkeit und absoluter Fehlerfreiheit.

Was von unseren Vorhaben konnte nicht realisiert werden? Zum einen die technische und inhaltliche Einrichtung des MIDlet-File Up- und Download-Bereichs der Website; die Zeit war einfach zu kurz. Zum anderen der angedachte MIDlet-Programmier-Wettbewerb. Der von SchülerInnenseite kommende Rücklauf an Ideen, die mit den uns zur Verfügung stehenden Mitteln umsetzbar gewesen wären, war leider nicht befriedigend. Die angesprochenen Vorhaben sind aber nicht ad acta gelegt und ausbaufähig erscheint unsere Arbeit in viele Richtungen.

Abschließend sei noch auf eine vom Institut für Unterrichts- und Schulentwicklung (IUS) in Klagenfurt durchgeführte, groß angelegte, zweiteilige SchülerInnen- und LehrerInnenbefragung, hingewiesen, in die auch unsere Schule mit allen ProjektteilnehmerInnen eingebunden war. Teil eins fand im Jänner 2007 statt, Teil zwei im Juni 2007. Ein Ziel der Befragung war es, die allgemeine Einstellung der SchülerInnen und LehrerInnen zu mathematisch-naturwissenschaftlichen Fächern (also auch zur Informatik) zu erheben und herauszufinden, wie SchülerInnen und LehrerInnen den Unterricht in diesen Fächern erleben. Ein anderes Ziel war, Erfahrungen mit und Meinungen zu unserem Projekt zu erforschen. Die Evaluierung des gesammelten Datenergebnisses steht noch aus. Laut IUS ist eine Darstellung im Internet <http://ius.uniklu.ac.at/> geplant.

## 6 LITERATUR

DORNINGER, Ch. (2007). "Mobile Technologien machen Schule", in PCNEWS 102, Feb. 2007, Seite 9 ff. Wien: PCNEWS-Eigenverlag

ALTRICHTER, H., POSCH, P. (1998). Lehrer erforschen ihren Unterricht. Eine Einführung in die Aktionsforschung (3. Aufl.). Bad Heilbrunn: Klinkhardt, ISBN 978-3-7815-1414-0

NIELSEN, J., LORANGER, H. (2006). "Web Usability". München: Addison-Wesley, (Broschiert - September 2006), ISBN 978-3-8273-2448-1

BARNES, D. J., KÖLLING, M. (2006). " JAVA lernen mit BlueJ". München: Verlag Pearson Studium, ISBN 978-3-8273-7073-0

BREYMAN, U., MOSEMANN, H. (2006). "JAVA-ME". München: Verlag Hanser, ISBN 978-3-446-22997-6

KNUTH, D. E. (1997). "The Art of Computer Programming", Vol. 1: "Fundamental Algorithms". Addison-Wesley Pub. Co, ISBN 0-201-89683-4

Des Weiteren wurde Literatur im großen Maße aus dem Internet bezogen. Artikel zur Programmiertechnik, Online-Foren, Herstellerinformationen. Eine Zusammenstellung diverser (von uns verwendeter) Quellen gibt es auf der Internetplattform <http://www.nls.at/~jam>.