

# Vorlesung

## Advanced Topics in HCI

### (Mensch-Maschine-Interaktion 2)

Ludwig-Maximilians-Universität München

LFE Medieninformatik

Albrecht Schmidt & Andreas Butz

WS2003/2004

<http://www.medien.informatik.uni-muenchen.de/>

# Developing mobile UI

- J2ME
- C++

# Introduction to Java Programming on Mobile Phones

**Enrico Rukzio**  
enrico@hcilab.org

# Outline

- History / Java Universe
- J2ME Basics
- J2ME: CLDC/MIDP
- Midlets
- Developing a user interface / storing data
- Resources / Documents / Tools (IDEs)
- Implementing “Hello World”
- Experiences

# Developing of Applications for mobile Devices

- Devices: Basic Phone, Extended Phone, Smartphone, PDA, Notebook
- Platforms (Mobile Phone, Smartphone)
  - Platform specific: Symbian OS (C++, OPL), Palm OS (C++), Pocket PC, Vendor-specific
  - Platform independent: J2ME (Java 2 Platform, Micro Edition)
    - Supported by Motorola, Nokia, Panasonic, Samsung, Sharp, Siemens, Sony Ericsson, Toshiba, etc.

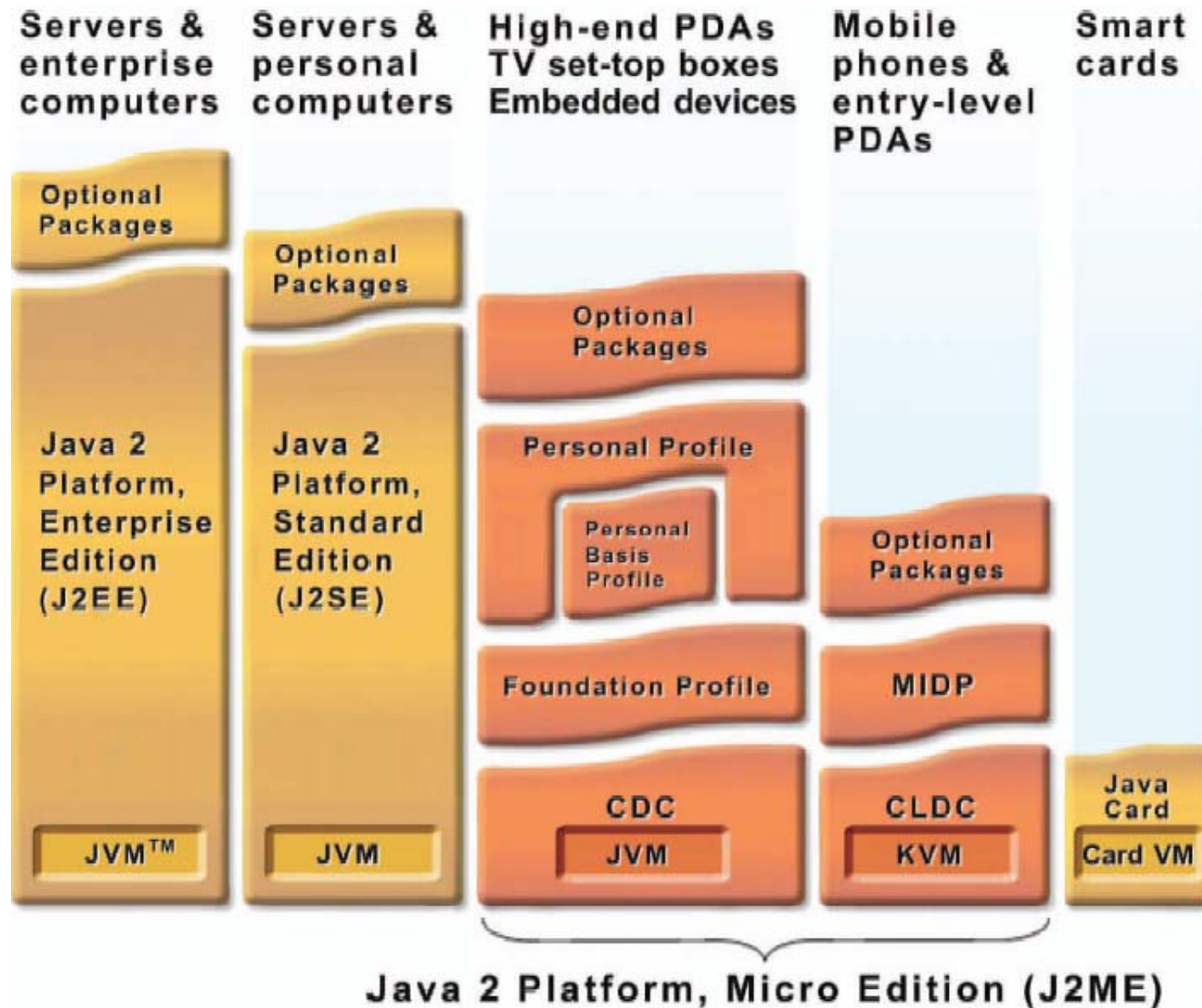
# Java on mobile devices: History [1,4,9]

- 1990: Java started as an internal project at Sun Microsystems
- 1995: Initial release of JDK 1.0 (applets → servlets)
- 1999: JavaOne conference
  - Subdivision of Java in
    - Java 2 Enterprise Edition (J2EE)
    - Java 2 Standard Edition (J2SE)
    - Java 2 Micro Edition (J2ME) (successor of Personal Java and Embedded Java)
- 2000/01 First mobile phones with support for J2ME

# Java on mobile devices: History [1,4,9]

- 2002: Second version of Mobile Information Device Profile (MIDP 2.0)
- April 2004: 250 Million mobile phones support J2ME [4]
- June 2005: 700 Million mobile phones support J2ME [10, 4] - more mobile phones with Java support than desktop PCs with Java support
- Now: most vendors of mobile phones (Motorola, Nokia, Panasonic, Samsung, Sharp, Siemens, Sony Ericsson, Toshiba, etc.) provide mobile phones that support J2ME

# The Java universe





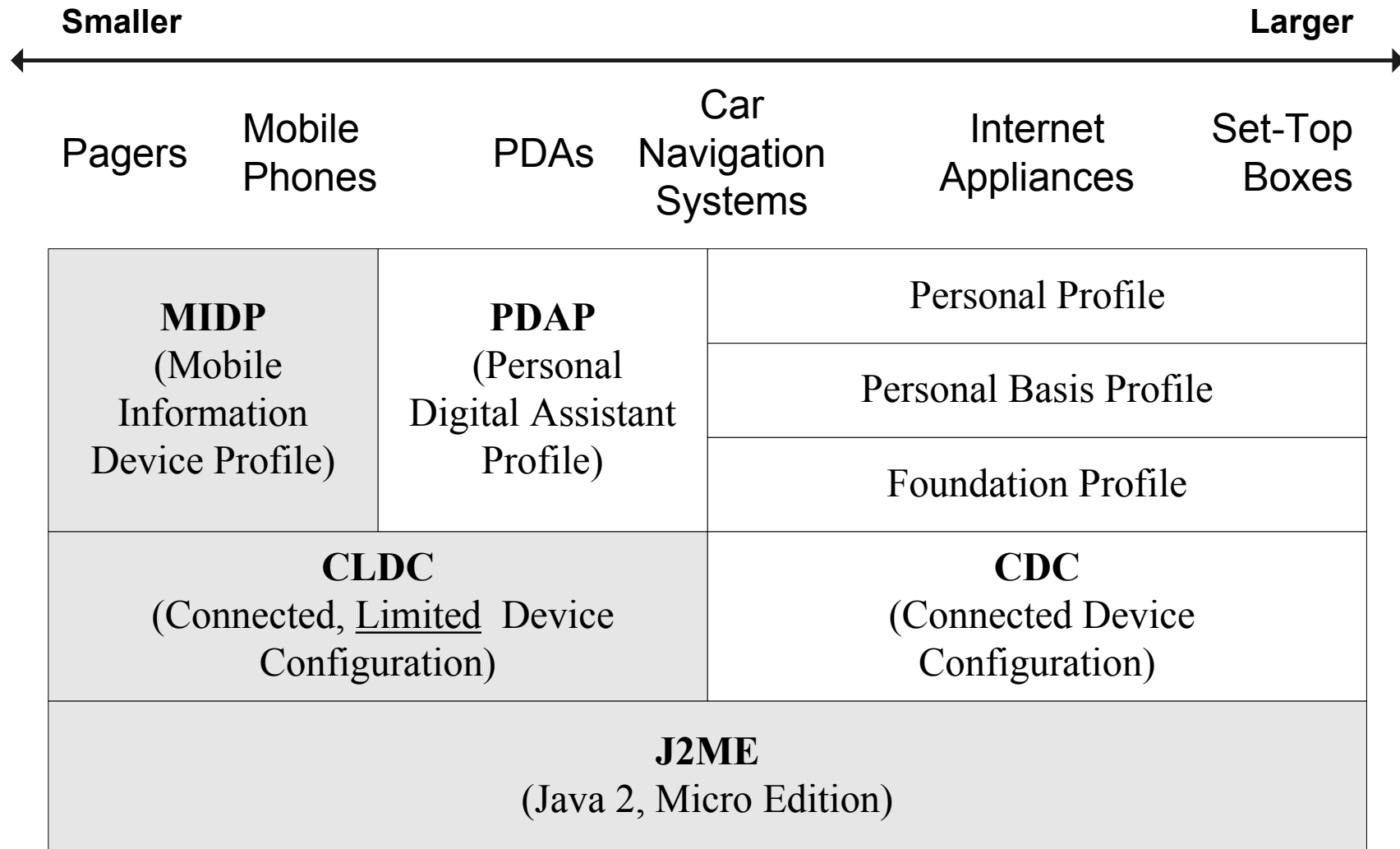
# J2ME: Basics

- J2ME: Java 2 Platform, Micro Edition
  - “Java for small devices”
- Stack
  - Configuration + profile + optional APIs
- Configuration: specific kind of device
  - Specifies a Java Virtual Machine (JVM)
  - Subset of J2SE (Standard Edition)
  - Additional APIs

# J2ME: Basics

- Profile: more specific than configuration
  - based on a configuration
  - adds APIs for user interface, persistent storage, etc.
- Optional APIs: additional functionality (Bluetooth, Multimedia, Mobile 3D, etc.)
- Everything is specified by a JSR (Java Specification Requests)

# The J2ME universe [1,9]



# J2ME: CLDC [JSR 30, 139]

- Connected, Limited Device Configuration
- For small devices (e.g. mobile phone, pager, PDA) with small screen size, limited memory, slow network connection
- For devices with 160 to 512KB (according to the specification) of memory for Java Platform
- JVM: KVM (“Kilobyte Virtual Machine”)
  - Not a full standard bytecode verifier
  - Adding native methods not allowed → not possible to access platform-specific functionality
- CLDC 1.0 / CLDC 1.1. (Floating point data types)

# J2ME: MIDP 2.0

- MIDP 2.0 (JSR 118, based on CLDC) [9]
- Mobile Information Device Profile for mobile phones and pagers
- Device characteristics (according to the specification):
  - Min. 128KB RAM (Java Runtime Heap)
  - 8KB for persistent data
  - Screen: > 94\*54 pixel
  - Input capacity, Network connection
- Advantages:
  - WORA (Write Once, Run Anywhere)
  - Security (Sandbox KVM)

# J2ME: APIs in CLDC 1.1 + MIDP 2.0

## **MIDP 2.0**

javax.microedition.lcdui  
javax.microedition.lcdui.game  
javax.microedition.media  
javax.microedition.media.control  
javax.microedition.midlet  
javax.microedition.pki  
javax.microedition.rms

## **CLDC 1.1**

java.lang  
java.lang.ref  
java.io  
java.util  
java.microedition.io

APIs are restricted  
when compared with  
J2SE

# Device configurations: examples

Type	Nokia 6600 (June 2003)	Nokia 6630 (June 2004)	Nokia N91 (End 2005)
Configuration	CLDC 1.0	CLDC 1.1	CLDC 1.1
Profile	MIDP 2.0	MIDP 2.0	MIDP 2.0
Optional APIs	Nokia UI, Wireless Messaging, Mobile Media, Bluetooth	APIs of 6600 + FileConnection and PIM, Mobile 3D Graphics	APIs of 6630 + Web Services, Security and Trust, Location, SIP, Scalable 2D Vector Graphics, Advanced Multimedia Supplements, JTWI
Heap Size	3 MByte	Unlimited	Unlimited
Shared Memory for Storage	6 MByte	10 MByte	29 MByte
JAR size	Unlimited	Unlimited	Unlimited

# J2ME: Compatibility [1,9]

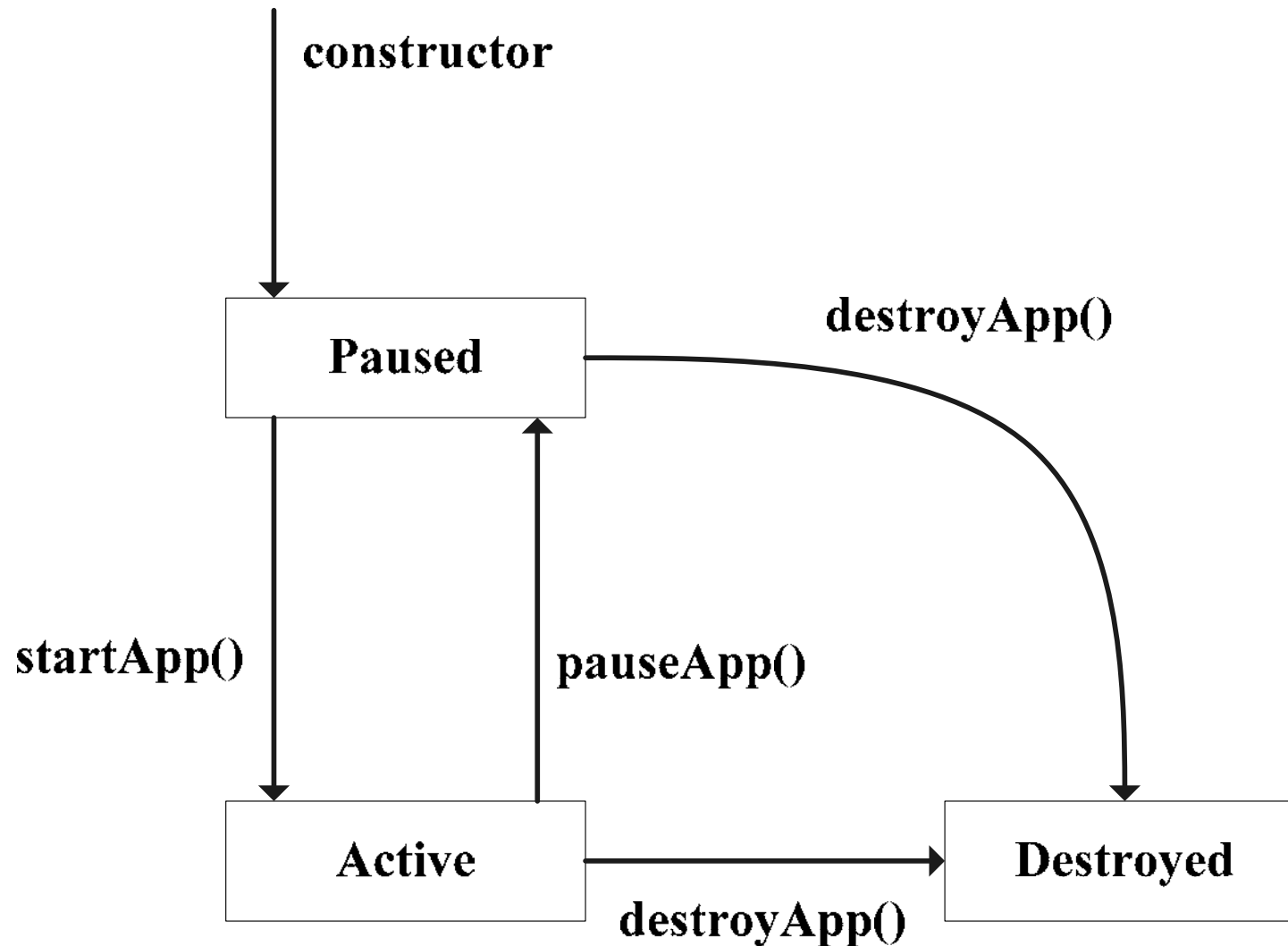
<b>MIDP Java Applications</b>	<b>Device-Specific Java Applications</b>	<b>Native Applications (compiled from C, C++, or other languages)</b>
<b>MIDP</b>	<b>Device-Specific APIs</b>	
<b>CLDC</b>		
<b>Device Operating System</b>		



# MIDlet

- MIDP applications are called MIDlets
- Every MIDlet is instance of `javax.microedition.midlet.MIDlet`
  - No argument constructor
  - Implements lifecycle methods
- Conceptually similar to Applets
  - Can be downloaded
  - Executed in host environment

# MIDlet (MIDP Application): Life Cycle



# MIDlet (MIDP Application): Life Cycle

- Application Manager: controls the installation and execution of MIDlets
- Start of a MIDlet: constructor + startApp (done by Application Manager)
- MIDlet
  - place itself in Paused state (notifyPaused())
  - destroy itself (notifyDestroyed())

# MIDlet (MIDP Application): Life Cycle

- Application Manager
  - pauseApp() and destroyApp() could be triggered by Application Manager
- 'active' Paused state
  - resumeRequest() – MIDlet wants to become Active
- Methods for every state transition

# MIDlet Build Cycle

- Edit source code
- Compile
- (Application) Package, MIDlet Suite
  - MIDlets + Classes + Ressources + Manifest Information => Java Archive (JAR)
  - Manifest: describes content of archive (versions of CLDC and MIDP, name, version, vendor)
  - Application Descriptor (\*.jad)
    - same information like manifest (+ MIDlet-Jar-Size, MIDlet-Jar-URL), but a external file
    - Sometimes used for installation
- Test or Deploy

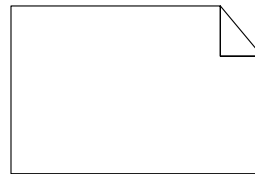
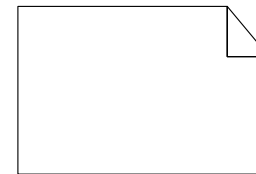
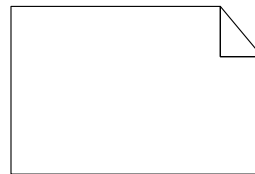
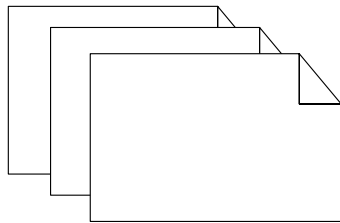
# Anatomy of a MIDlet suite

MidletSuite.jad



Contents of MidletSuite.jar

MidletSuite.jar



**MANIFEST.MF**



# MIDP: User Interface

- Goal: Write Once, Run Anywhere
- Anywhere?
  - different screen sizes
  - resolution of screen
  - color or grayscale screen
  - different input capabilities (numeric keypad, alphabetical keyboards, soft keys, touch screens, etc.)

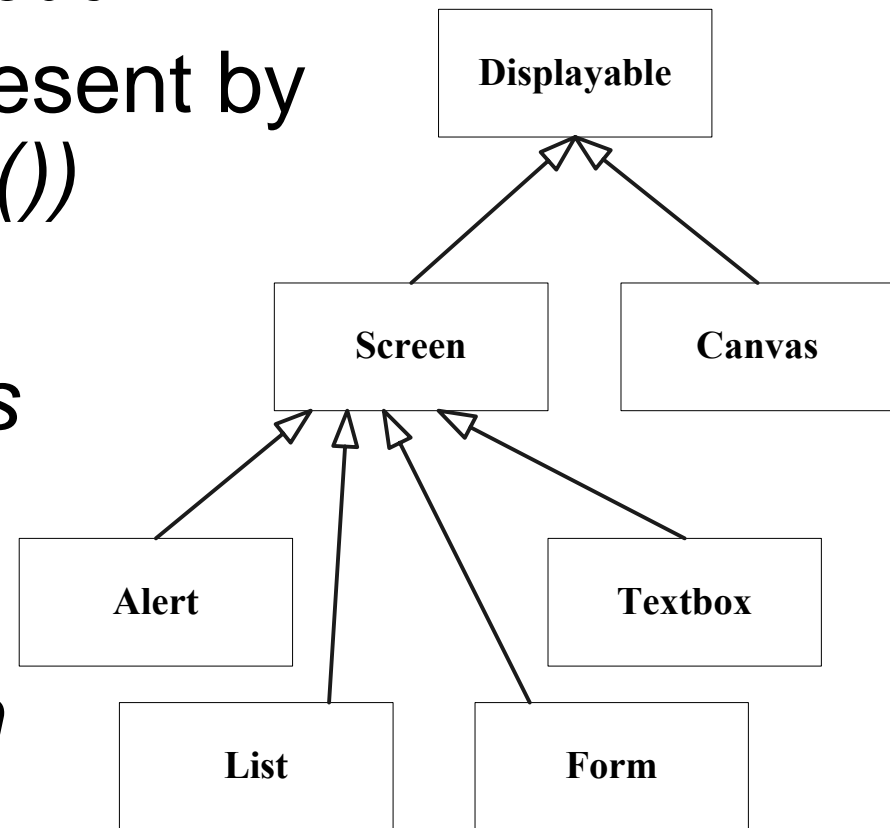
# User Interface: Methodology

- Abstraction (→ Preferred Method)
  - specifying a user interface abstract terms
  - (Not:) “Display the word ‘Next’ on the screen above the soft button.”
  - Rather: “Give me a Next command somewhere in this interface”
- Discovery (→ Games)
  - Application learns about the device + tailors the user interface programmatically
  - Screen size → Scaling



# User Interface: View from the Top

- User-interface classes  
*javax.microedition.lcdui*
- Device display represent by  
*Display (getDisplay())*
- *Display: easel*
- *Displayable: canvas on easel*
- *Canvas: Discovery*
- *Screen: Abstraction*



# User Interface: View from the Top

- Changes the contents of the display:  
passing *Displayable* instances to  
*Display's setCurrent()*
- Typical Sequence
  - Show a *Displayable*
  - Wait for input
  - Decide what *Displayable* should next
  - Repeat

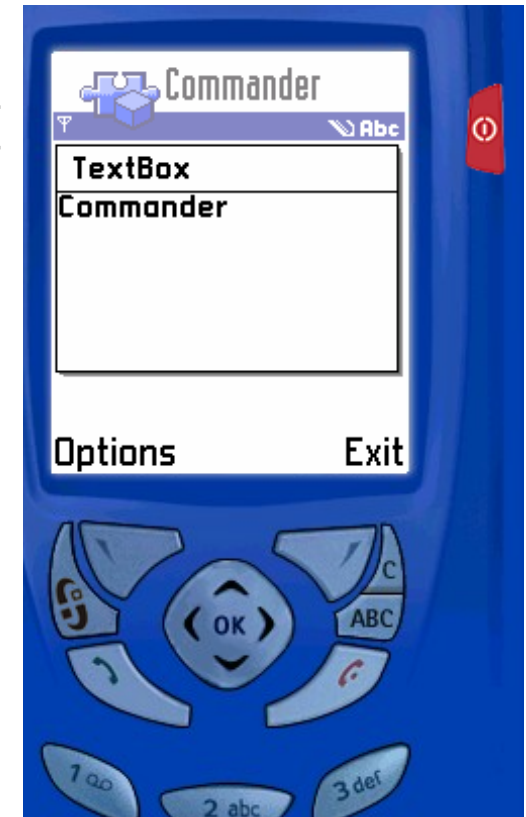
# User Interface: Simple Example

```
public class Commander extends MIDlet {
    public void startApp() {
        Displayable d = new TextBox("TextBox", "Commander", 20, TextField.ANY);
        Command c = new Command("Exit", Command.EXIT, 0);
        d.addCommand(c);
        d.setCommandListener(new CommandListener() {
            public void commandAction(Command c, Displayable s) {
                notifyDestroyed();
            }
        });

        Display.getDisplay(this).setCurrent(d);
    }

    public void pauseApp() {}

    public void destroyApp(boolean unconditional) {}
}
```

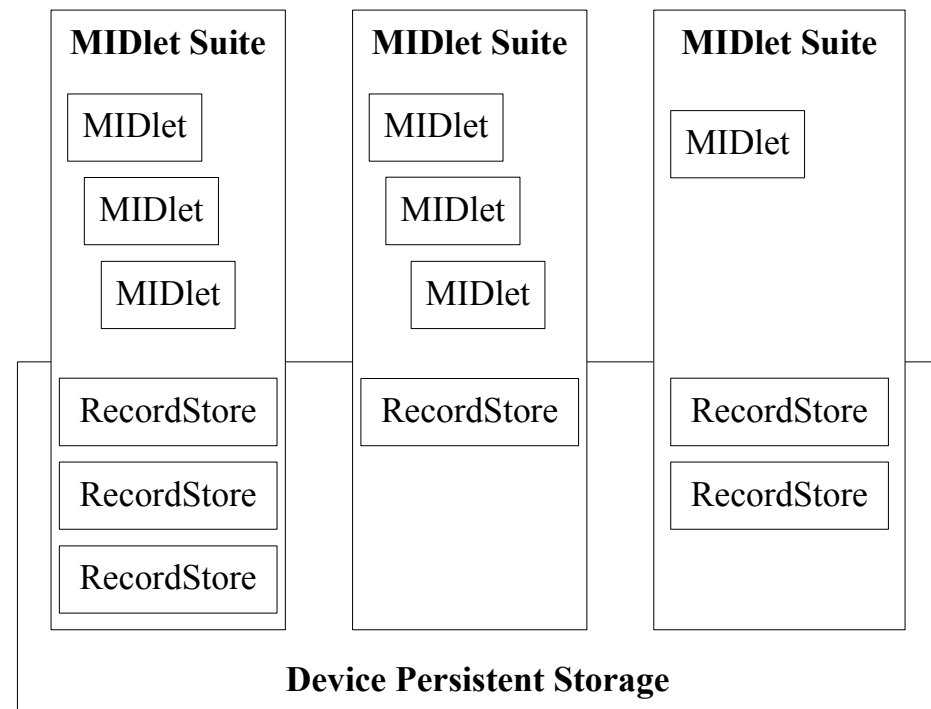


# MIDP: Persistent Storage

- Goal: Write Once, Run Anywhere
- Anywhere?
  - Device with Flash ROM
  - Battery-backed RAM
  - Small Hard Disk
- Abstraction is needed
- Record stores (small databases)
- Min. 8KByte (Nokia 6600: ‘the only limitation is the amount of free memory’)

# Persistent Storage: Records

- *Record store*
  - contains *records* (pieces of data)
  - instance of `javax.microedition.rms.RecordStore`
- Every MIDlet in a MIDlet Suite can access every Record Store
- Since MIDP 2.0: Access across Suite borders possible !!!

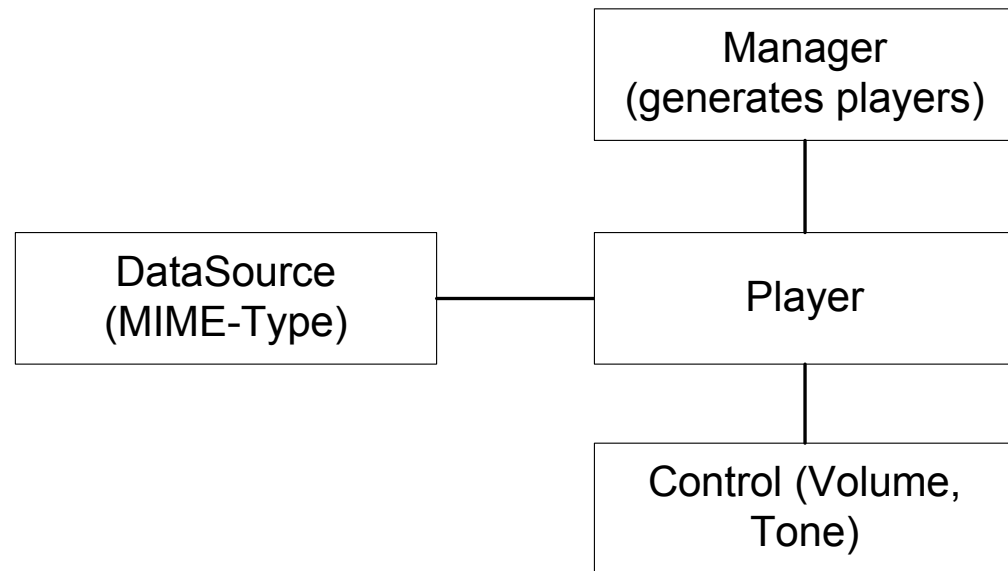


# Connecting to the World

- Generic Connection Framework
- Extremely flexible API for network connections
- Contained in `javax.microedition.io`
- Classes based on *connection* interface
  - `URLConnection` (Get / Post) / `HttpsURLConnection`
  - `URLConnection`
  - `ServerSocketConnection` (Responding to incoming connections)
  - `SecureURLConnection` (TLS or SSL socket)
  - `CommConnection` (SerialPort)
  - `DatagramConnection` (UDP `DatagramConnection`)

# MMAPI (Sound, Music, Video)

- Mobile Media API
- General API for multimedia rendering and recording
- ABB (Audio Building Block) – play simple tones (MIDI – note, duration, volume) and sampled audio (wav, mp3)
- Player lifecycle
  - UNREALIZED
  - REALIZED
  - PREFETCHED
  - STARED
  - CLOSED



# Further APIs

- Wireless Messaging API (JSR-120)
- Mobile Media API (JSR-135)
- Bluetooth API (JSR-82 no OBEX)
- FileConnection and PIM API (JSR-75)
- Mobile 3D Graphics API (JSR-184)
- Location API (JSR-179)
- Web Services API (JSR-172)
- Advanced Multimedia Supplements (JSR-234)
  
- Further APIs (not JSRs): kXML, kSOAP, Parsing of GPS data, etc.



# Material

- Jonathan B. Knudsen, Sing Li. Beginning J2ME: From Novice to Professional. ISBN: 1590594797. 2005.
- Java.Sun.Com (Documentation, Code samples & Articles, FAQs, white papers, technical articles, etc.)
  - <http://java.sun.com/products/cldc/>
- Forum.nokia.com (Documents, Code & examples, tools, forum)
  - <http://www.forum.nokia.com>
- Links to documentations and tutorials at hcilab.org
  - <http://www.hcilab.org/documents/tutorials/DocuTuto/index.html>
- Sun Wireless Toolkit: JavaDoc

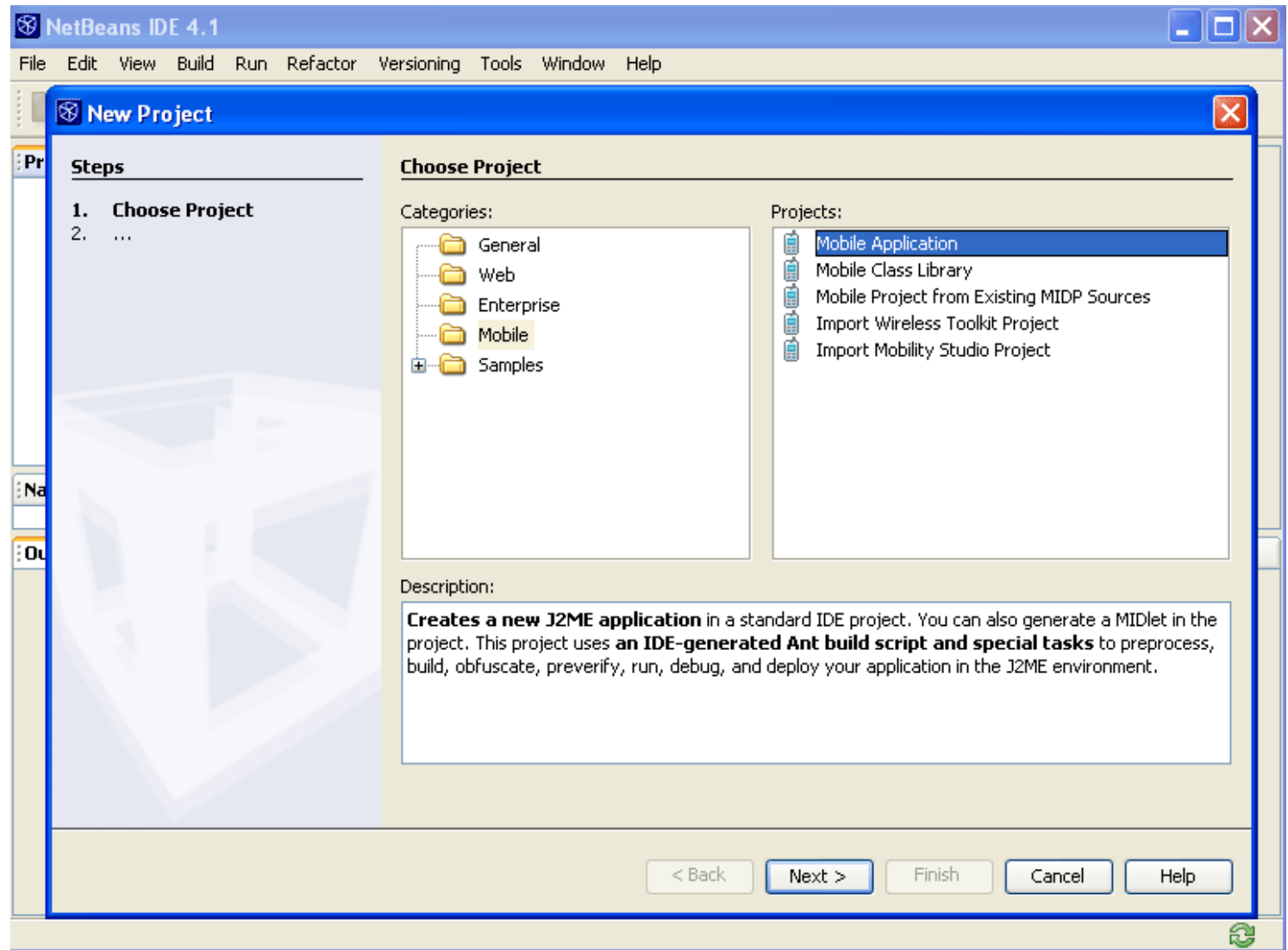
# Tool Support / Development Kits

- Sun's MIDP reference Implementation (do not use it)
- Sun J2ME Wireless Toolkit (Javadoc)
- IDE
  - Netbeans 5.5 + Mobility Pack
  - Eclipse (with Plug-In EclipseME)
  - Borland JBuilder MobileSet
  - IBM WebSphere Studio Device Developer
  - Metrowerks Code Warrior Wireless Studio
  - Sun ONE Studio, Mobile Edition
- Vendor Specific Toolkits & Documentation

# Netbeans

- Cross-platform Java IDE
- Available for free for non-commercial use.
- Download and install Netbeans 5.0 / 5.5 Beta (requires a J2SE JDK, version 1.4.2 or higher and Sun J2ME Wireless Toolkit) + Mobility Pack at [www.netbeans.org](http://www.netbeans.org)

# New Project



# New Project: Name and Location

**New Project**

**Steps**

1. Choose Project
- 2. Name and Location**
3. Platform Selection

**Name and Location**

Project Name:

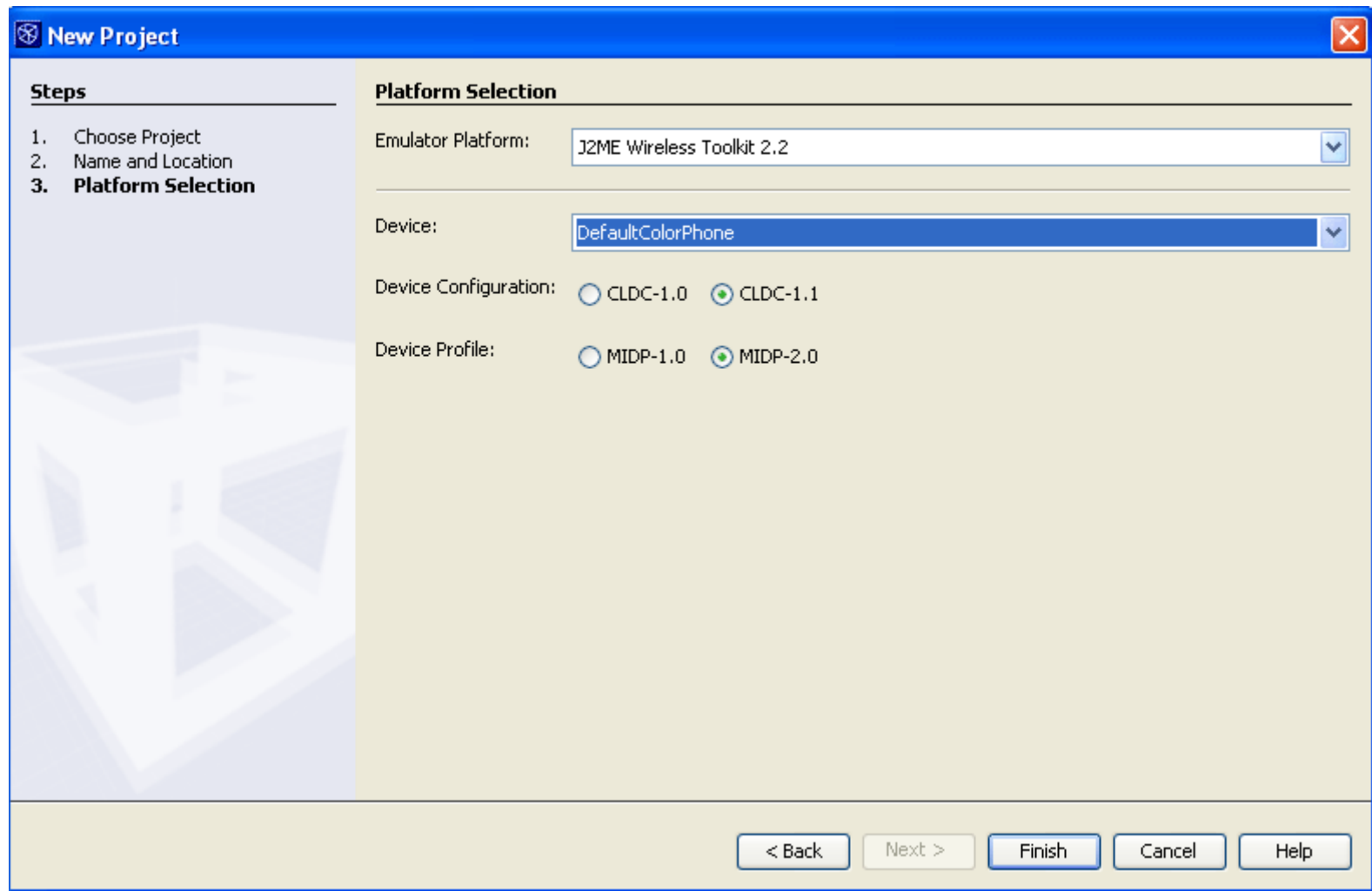
Project Location:

Project Folder:

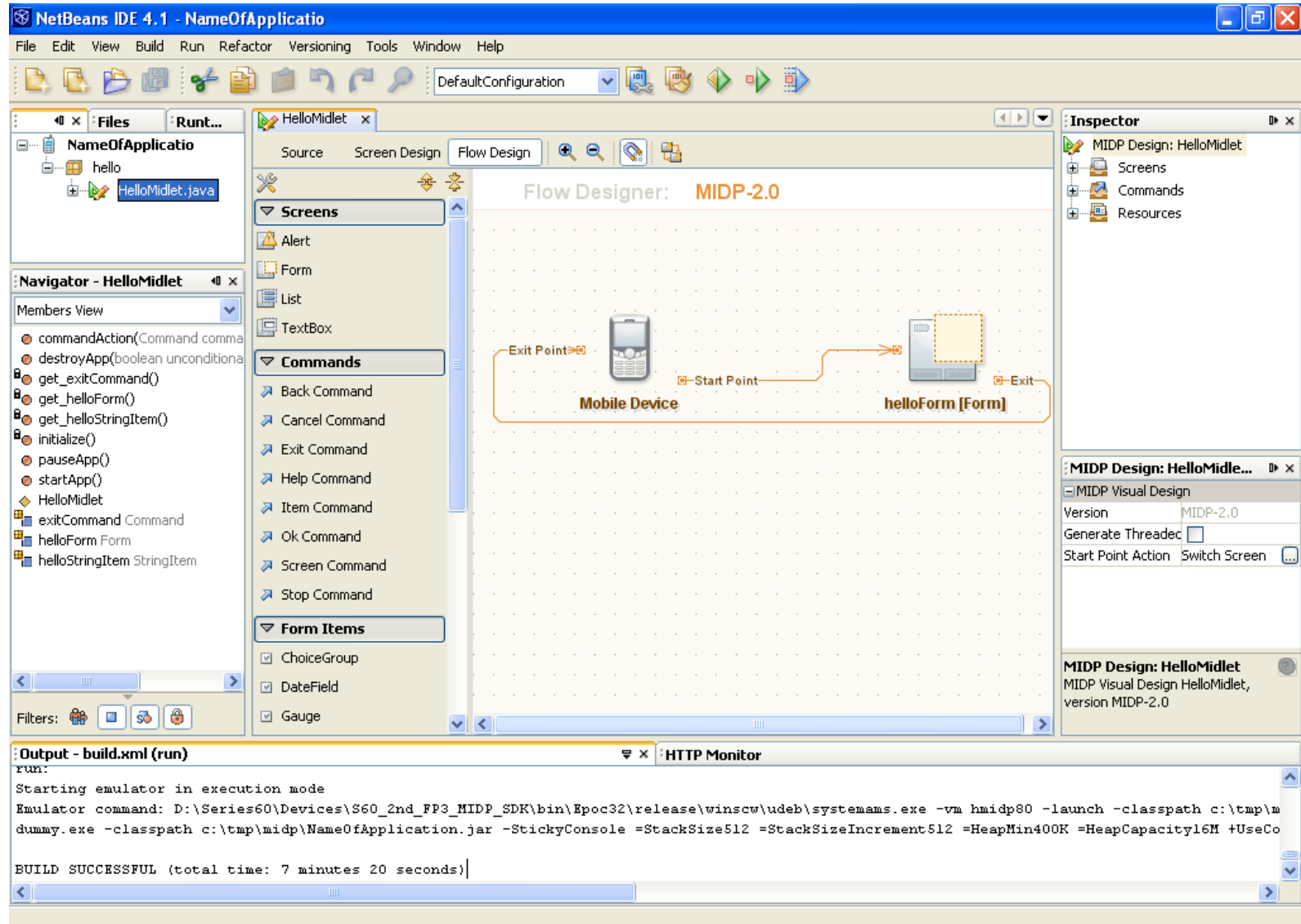
Set as Main Project

Create Hello MIDlet

# New Project: Platform Selection



# IDE



# IDE: Source

```
package hello;

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

/**
 *
 * @author enrico
 */
public class HelloMidlet extends MIDlet implements javax.microedition.lcdui.CommandListener {

    /** Creates a new instance of HelloMidlet */
    public HelloMidlet() {
    }

    This section is auto-generated by NetBeans IDE.

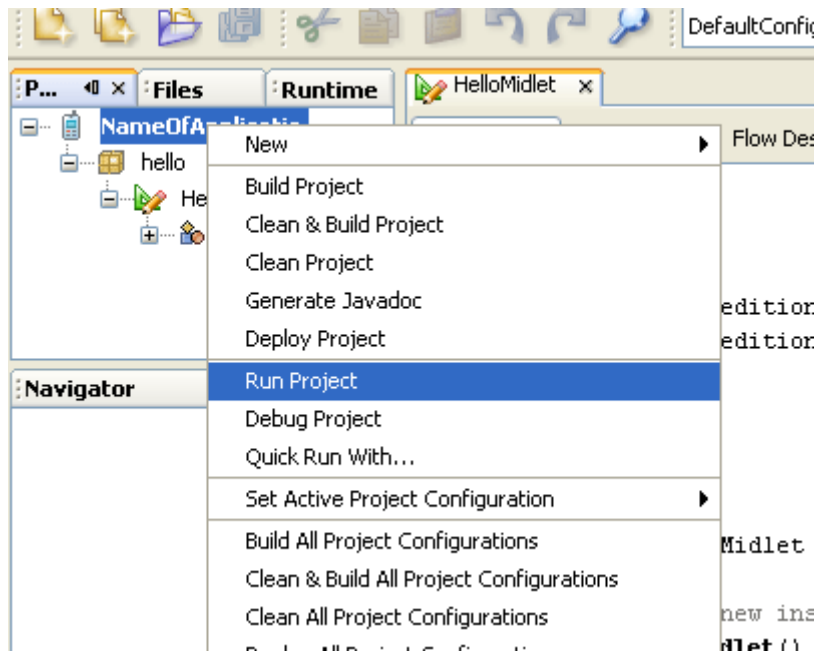
    public void startApp() {
        initialize();
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}
```



# IDE: Run Project



# Übungsaufgabe 7

- Prototyping a mobile phone application, (Per-group homework, 2 weeks)
- Application
  - Select one of the following applications, develop a paper prototype and implement the application using J2ME, Flash lite or Python.
  - Vocab trainer: An application which realizes a vocab trainer with the following functions:
    - add a new dictionary entry
    - show German word - request English word
    - show English word - request German word
  - Who wants to be a millionaire?: An application which realizes a “Who wants to be a millionaire?” application with the following functions:
    - add a new question including 4 answers of which just one is correct
    - show an arbitrarily selected question including the possible answers
    - selection of one answer and indication whether the answer was correct or not

# Übungsaufgabe 7

- Task 1: Paper prototype of one selected application
  - Develop a paper prototype of your application including all key screens. This prototype should then be tested by two other persons. These persons have to fulfil tasks described in “0 Application”.
  
- Task 2: Implementation of one selected application
  - The prototype supporting the described functions should be implemented using J2ME, Flash Lite or Python.
  - J2ME
    - J2ME Sources for HelloWorld, Address book and vocable management will be presented in the MMI-2 lecture
    - [http://www.medien.informatik.uni-muenchen.de/fileadmin/mimuc/mmi2\\_ss06/uebung/Examples.zip](http://www.medien.informatik.uni-muenchen.de/fileadmin/mimuc/mmi2_ss06/uebung/Examples.zip)

# Übungsaufgabe 7

- Praktikumsraum: Amalienstraße 17, Raum 103/101
- User: mmi2 / Passwort: mmi2 (will be deactivated in 2 weeks)
- Raum 103: J2ME Bücher
- Jobs



# Experiences

- CLDC/MIDP is a powerful platform for building novel application on mobile devices
- Everything (phones, APIs, tools, books, documentation, etc) is getting better in a very fast way
- Programming with J2ME It is still a novelty for most people.
- New APIs (Mobile Media, Bluetooth, etc.) have new bugs. “Old” APIs (storage, UI) are already in a matured state.
- Different mobile devices have different KVMs (with different bugs)

# Experiences

- Testing of applications on the mobile phone (!!!) is very important.
- Big differences between the emulators and the real phone.
- Lack of memory and processing power is still a problem.
- Debugging on the mobile phone is a big problem. (No meaningful error messages.)
- Implementation on an abstract level. Not so much possibilities like in Symbian.

# Praktikum WS06/07

- Entwicklung von Mediensystemen (Mobile Endgeräte)
- Development of a mobile application within a team (idea, concept, implementation, evaluation)
- Supervisors: Albrecht Schmidt, Gregor Broll, Alexander De Luca, (Enrico Rukzio)



# Hardware



- Mobile Phones: Nokia N90, Nokia N91, Nokia N70, Nokia 6630 (2x), Nokia 6600 (4x), Nokia 3220, Siemens S65 (2x), Samsung SGH-E760
- SIM-Cards (O2, T-Mobile, Vodafone)





# J2ME References

- [1] Jonathan B. Knudsen. Wireless Java: Developing with J2ME. Second Edition. ISBN: 1590590775.
- [2] Stephen Neal. Overview of J2ME and Nokia APIs. Sun Tech Days. [http://www.nokia.co.jp/forum/publish/documents/Tech\\_Days\\_Yokohama\\_Workshop\\_Session.pdf](http://www.nokia.co.jp/forum/publish/documents/Tech_Days_Yokohama_Workshop_Session.pdf)
- [3] J2ME datasheet <http://java.sun.com/j2me/j2me-ds.pdf>
- [4] Heise Newsticker, <http://www.heise.de/newsticker/>
- [5] CDC Data Sheet. [http://java.sun.com/j2me/docs/j2me\\_cdc.pdf](http://java.sun.com/j2me/docs/j2me_cdc.pdf)
- [6] What's in MIDP 2.0: A Guide for Java Developers <http://www.forum.nokia.com/ndsCookieBuilder?fileParamID=3632>
- [7] MIDP 2.0: An Introduction <http://www.forum.nokia.com/ndsCookieBuilder?fileParamID=3231>
- [8] Understanding the Record Management System <http://developers.sun.com/techtopics/mobility/midp/articles/databaserms/>
- [9] Jonathan B. Knudsen, Sing Li. Beginning J2ME: From Novice to Professional. ISBN: 1590594797.
- [10] Sun Takes Java App Server Open Source- <http://www.itjungle.com/tlb/tlb062805-story04.html>

# books

- What's in MIDP 2.0: A Guide for Java Developers

<http://www.forum.nokia.com/ndsCookieBuilder?fileParamID=3632>

- focus on section 3-3.23, without 3.3 (page 15-17)

- MIDP 2.0: An Introduction

<http://www.forum.nokia.com/ndsCookieBuilder?fileParamID=3231>

- without section 3-7, focus on section 2 (page 4-19)

- Understanding the Record Management System

<http://developers.sun.com/techttopics/mobility/midp/articles/databaserms/>

- Record store discovery; Creating + Opening + closing records stores, Adding + reading + updating records, Delete records + record stores

# Chapter 3: Mobile HCI

## Table of Content

- Input & Output Devices
- Input & Output Techniques
- Guidelines
- Example: Applications for Mobile Phones
- Mobile Gaming
- System Architectures for Mobile UIs

# Motivation

## It is expected that Mobile Gaming is becoming a huge market (1)

- “Visiongain predicts that by 2009, there will be around 2.420 billion Java handsets in the market”
- “Mobile gaming has changed the wireless market today... 200 million people will be playing Web games on mobile phones by 2005... While the number of new wireless users slows down as the markets reach saturation, wireless gaming represents a true growth area. In 2001 only about 22% ...which is expected to further rise to 50% in 2006.”

<http://www.ewirelessnews.com/PressCentre/81/24-billion-Java-handsets-in-the-market-by-2009.html>

# Phone Platforms offer many technology options

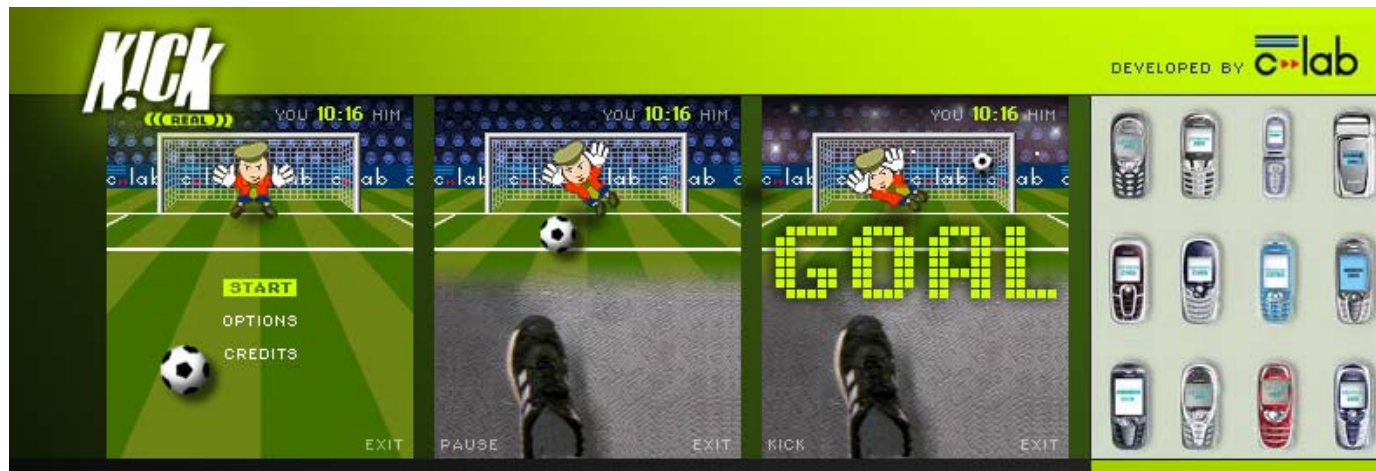
- Input
  - Keys
  - Microphone (voice/noises)
  - Camera
    - Pictures/photos as background,
    - motion as control mechanism
  - Location information
    - Cell-ID information
    - GPS phones
- Output
  - Graphical output
  - Audio output
  - Tactile output
- Networking
  - Short range (e.g. Bluetooth, IR)
  - Over the phone network (e.g. GSM/UMTS)
  - P2P
  - Server based

# Examples: Camera as Input (I)

- CamBlaster!  
<http://www.realeyes3d.com/pages/products/camblaster.php>



- Football game <http://www.kickreal.de/>



# Examples: Camera as Input Digitizer



- <http://www.realeyes3d.com/pages/products/digitizer.php>

# Examples: Camera as Input (2)

## W-Postcard

- Combing 2 pictures in a postcard



Before using w-Postcard



After using w-Postcard

Simple to use: Shoot background, shoot comment, and send!

Load or take picture of the background



Take picture of your handwritten message



Preview your personalized w-Postcard™



<http://www.realeyes3d.com/pages/products/postcard.php>