MOBILEHCIDS

TUTORIAL: DEVELOPMENT OF INTERACTIVE APPLICATIONS FOR MOBILE DEVICES

7th International Conference on Human Computer Interaction with Mobile Devices and Services
(Mobile HCI 2005)

Enrico Rukzio (Media Informatics Group, University of Munich)
Michael Rohs (Deutsche Telekom Laboratories)
Daniel Wagner (Graz University of Technology)
John Hamard (DoCoMo Communications Laboratories Europe)

MOBILEHCIDS

APPLICATION DEVELOPMENT WITH J2ME

Enrico Rukzio

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Outline

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

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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

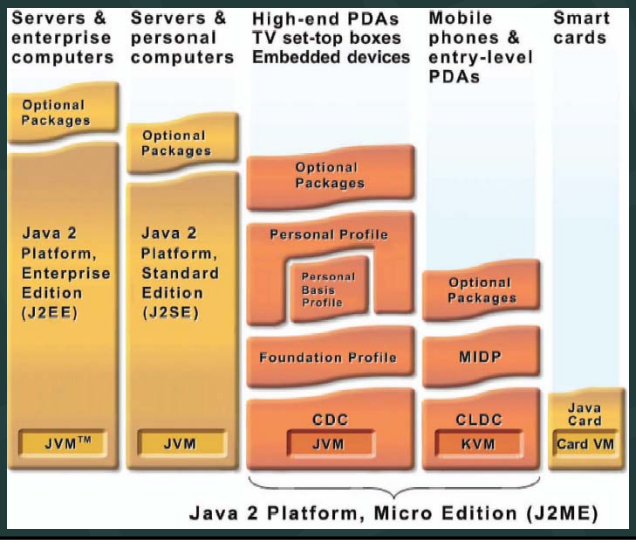
Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

The Java universe [3]



Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

J2ME: Basics

J2ME: Java 2 Platform, Micro Edition
- "Java for small devices"
- Divided in configurations, profiles and optional APIs

Stack
- Configuration + profile + optional APIs

Configuration: for a specific kind of devices
- Specifies a Java Virtual Machine (JVM)
- Subset of J2SE (Standard Edition)
- Additional APIs

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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)

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

The J2ME universe [1,9]

Smaller ⟷ Larger

Pagers | Mobile Phones | PDAs | Car Navigation Systems | Internet Appliances | Set-Top Boxes

MIDP (Mobile Information Device Profile)	**PDAP** (Personal Digital Assistant Profile)	Personal Profile
		Personal Basis Profile
		Foundation Profile
CLDC (Connected, Limited Device Configuration)		**CDC** (Connected Device Configuration)
J2ME (Java 2, Micro Edition)		

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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)

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

J2ME: 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)

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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

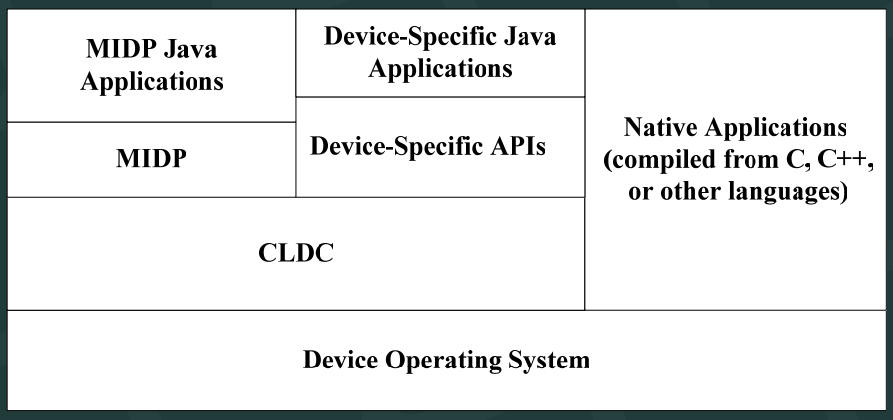
Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Device configurations: some 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 (the only limitation is the amount of free memory)	10 MByte	29 MByte
JAR size	Unlimited	Unlimited	Unlimited

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

J2ME: Compatibility [1, 9]



Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

MIDlet

MIDP applications are called MIDlets

Every MIDlet is instance of javax.microedition.midlet.MIDlet

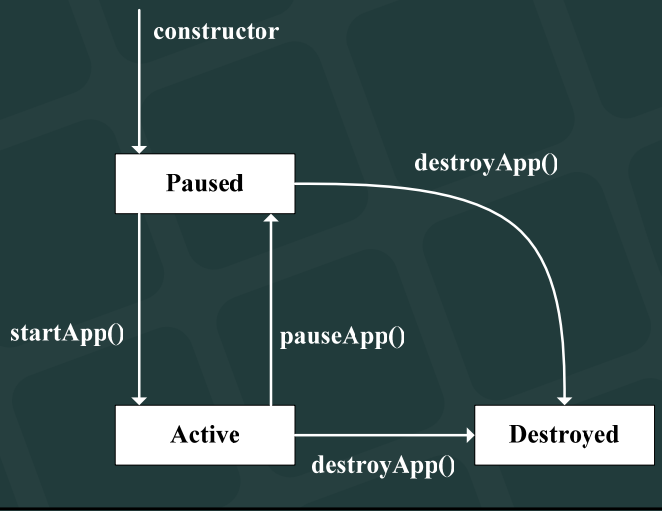
- No argument constructor
- Implements lifecycle methods

Conceptually similar to Applets

- Can be downloaded
- Executed in host environment

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

MIDlet (MIDP Application): Life Cycle



Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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())

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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

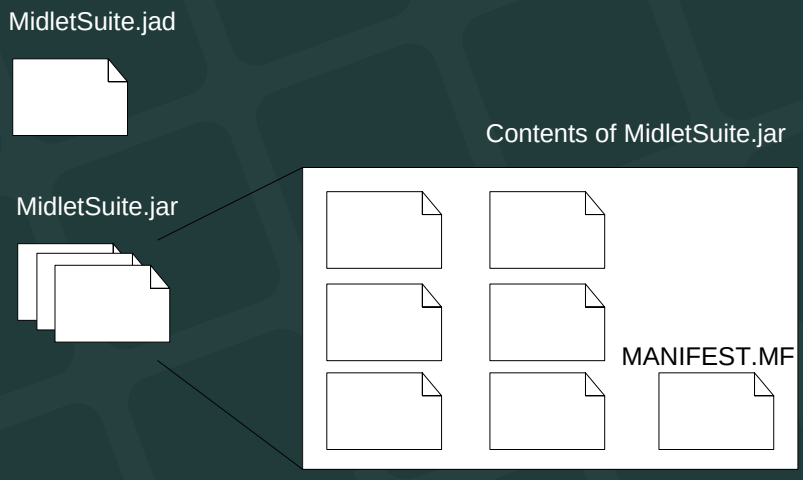
Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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
    - Sometime used for installation
- Test or Deploy

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Anatomy of a MIDlet suite



Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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.)

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

User Interface: View from the Top

User-interface classes *javax.microedition.lcdui*

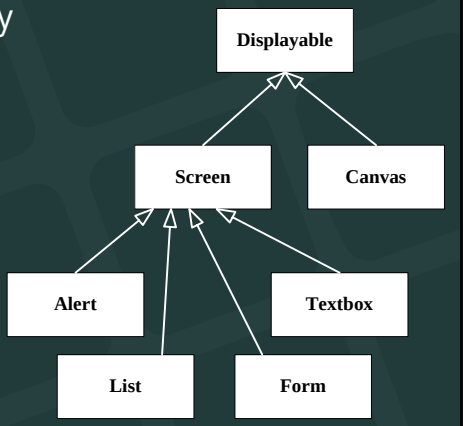
Device display represent by *Display (getDisplay())*

Display: easel

Displayable: canvas on easel

Canvas: Discovery

Screen: Abstraction



Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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) {}
}
```

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

MIDP: Persistent Storage [8]

Goal: Write Once, Run Anywhere

Anywhere?

- Device with Flash ROM
- Battery-backed RAM
- Small Hard Disk

Abstraction is needed

Record stores (small databases)

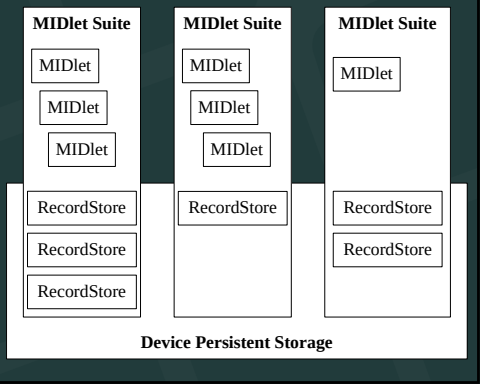
Min. 8KByte (Nokia 6600: ‘ the only limitation is the amount of free memory’)

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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 boarders possible !!!



Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

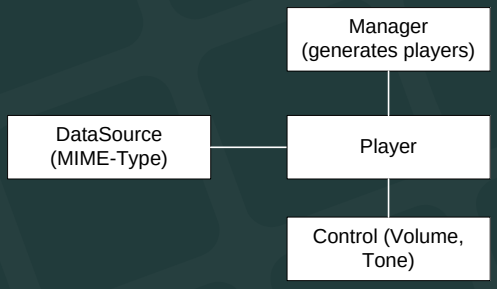
Connecting to the World

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

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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



Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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.

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Tool Support / Development Kits

Sun's MIDP reference Implementation
Sun J2ME Wireless Toolkit (Javadoc)
IDE
- Netbeans 4.1 + 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

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Netbeans 4.1

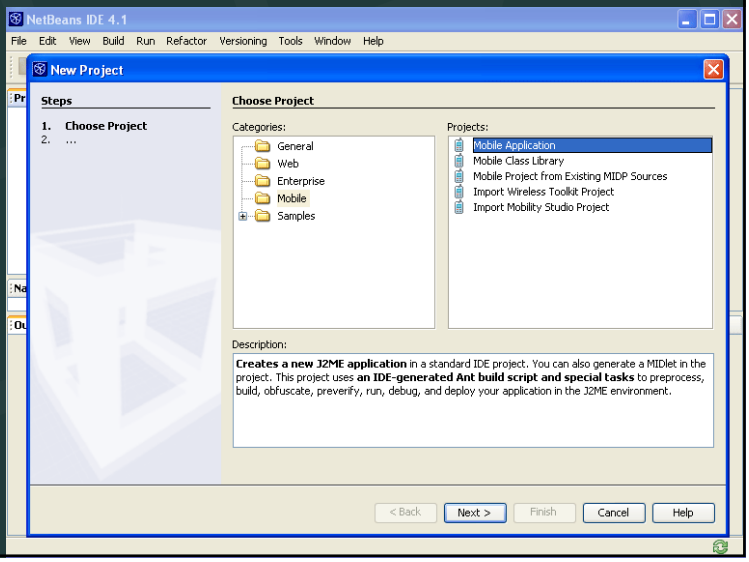
Cross-platform Java IDE

Available for free for non-commercial use.

Download and install Netbeans 4.1 (requires a J2SE JDK, version 1.4.2 or higher + Sun's Wireless Toolkit) + Mobility Pack at www.netbeans.org

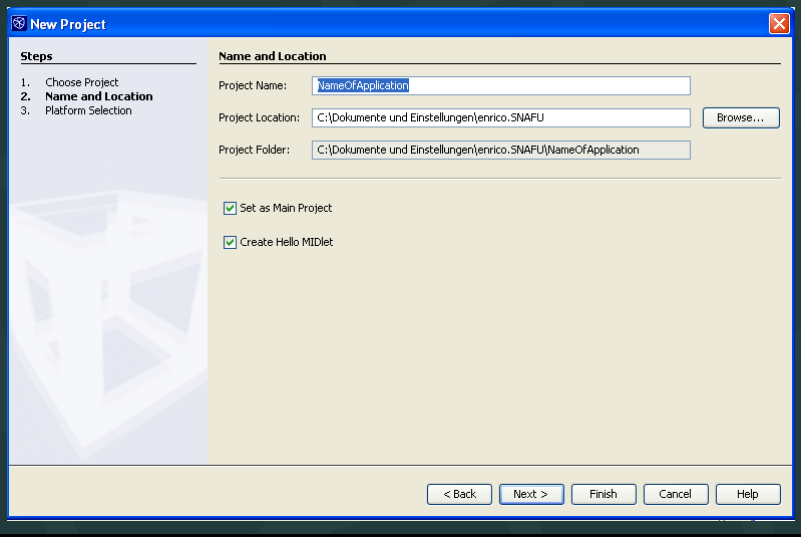
Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

New Project



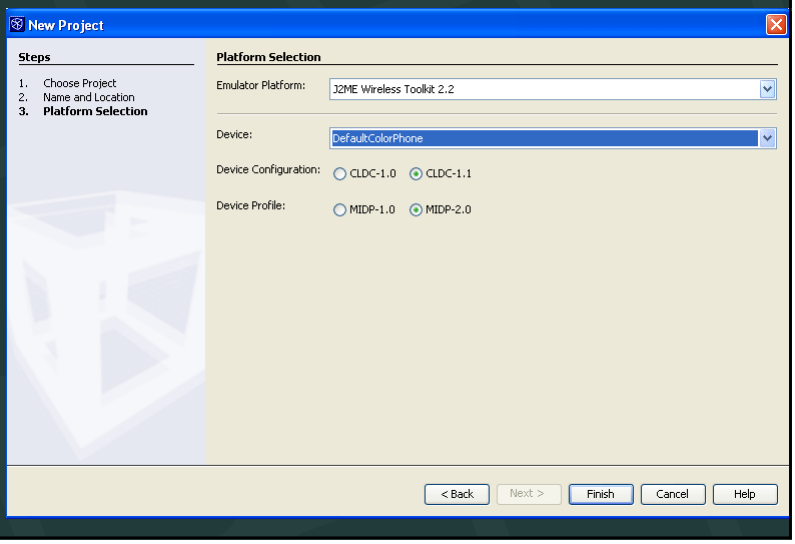
Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

New Project: Name and Location



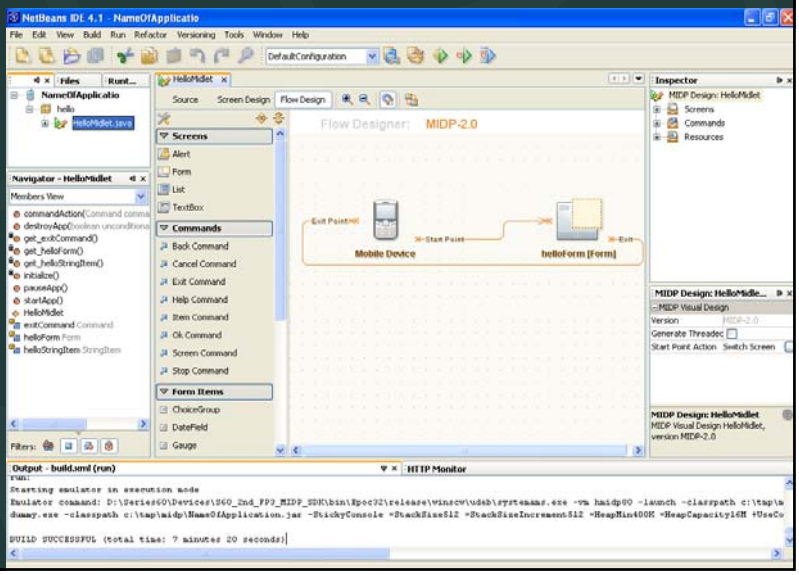
Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

New Project: Platform Selection



Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

IDE



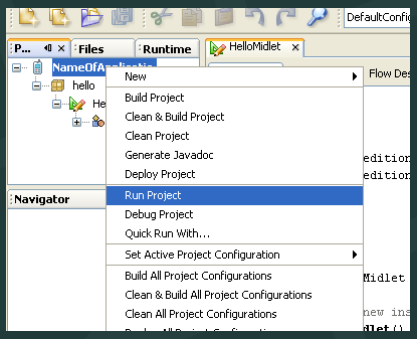
Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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) {
    }

}
```

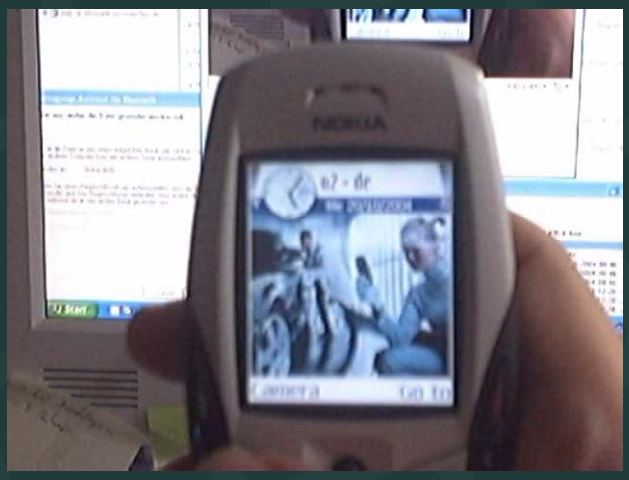
Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

IDE: Run Project



Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Install MIDlet on Mobile Phone



Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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.

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Experiences

- Different mobile devices have different KVMs (with different bugs)
- 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.)

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Experiences

Implementation on an abstract level. Not so much possibilities like in Symbian.

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Wishes for the next CLDC/MIPD Generation

New security model for J2ME

- Accessing data (record stores)
- Accessing the camera, microphone, network, Bluetooth

Quality of service

- Uncertain behavior when recording (quality, encoding) and playing (Which player?) media

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Market

Mobile devices

- are an exploding market
- because of increasing processing power, available memory and internet connectivity → attractive platform

Most supported platform on mobile devices: J2ME

- Write once – run everywhere
- Safety features for downloadable code

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

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
[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
[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

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

Deleted Slides

Enrico Rukzio
Enrico.Rukzio@ifi.lmu.de

J2ME: CDC (JSR 36, 218) [5, 9]

- CDC: Connected Device Configuration
- For set-top boxes, car navigation systems, smart communicators, high end PDAs, etc.
  - 32bit microprocessor
  - 256 KB RAM and 512 KB ROM for Java Runtime Environment
- Full Java Virtual Machine
  - CDC 1.01 (JSR 36) based on J2SE 1.3.1
  - CDC 1.1 (JSR 218) based on J2SE 1.4
- Supports three profiles: Foundation Profile, Personal Basis Profile, Personal Profile
- Outdated Personal Java is still used