Praktikum Mobile Productivity
Outline:

• Basic Information
• Organizational Stuff
• Technology
  • SVN
  • Java ME
  • IDEs
• Exercise 1
Basic Information

Basic Conditions:

- Two phases: single and group work phase
- Software development in a team

Time plan

- Phase 1 – Single Work:
  - Exercise 1 – 4
  - Exercise 5 partially

- Phase 2 – Project Work:
  - Exercise 5 – 8
  - Project Implementation
Organizational Stuff

- **4 SWS**
- **Weekly Meeting**
  - Tuesday 10:00 – 12:00
  - Room 107, Amalienstraße 17
- **Room for the practical parts:**
  - Medienlabor 103, Amalienstraße 17
  - Special accounts required
  - Open during normal working times (8:00 – 17:00)
  - 1 key for each group
- **Homepage:**
  - [http://www.medien.ifi.lmu.de/lehre/ws0607/pmp/](http://www.medien.ifi.lmu.de/lehre/ws0607/pmp/)
- **SVN**
  - `svn://murx.medien.ifi.lmu.de/ws0607/pmp/team[number]`
    (e.g. `svn://murx.medien.ifi.lmu.de/ws0607/pmp/team1`)
Team 1:
- Ragutt
- Möller
- Moog
- Reithmeier
- Kaczkowski

Team 2:
- Gassner
- Wiesner
- Berwein
- Dingler
- Jacobi
- Konda

Team 3:
- Pollner
- Sauerwein
- Weinand
- Kahl
- Rabinowitsch
Organizational Stuff

**Needed Accounts**

- SVN username
- Medienlabor-Kennung
- Belegungsplan Medienlabor
SVN - General

- Version control system
- Enables collective editing of shared source code
- Data stored in a „Repository“ which is accessed over the network
- Editing on local copies of the files
- Old version available on the server
- When possible, files will be merged automatically when edited by multiple users at the same time
- Similar to CVS
SVN – First Steps (using Tortoise SVN)

1. Download a SVN Client like Tortoise SVN for Windows
   http://tortoisesvn.net/
2. Checkout your team repository (creates a local copy of the repository)
   Create an empty folder, open it, right-click and choose „Checkout“.
SVN – First Steps (using Tortoise SVN)

3. Each time you start working perform the “Update” command.
4. Each time you’re done working perform a “Commit”. Both commands are located in the right-click menu.
5. Further functionalities are available in the right-click menu like “delete”, “rename” and more.
   **Attention:** Do not use the OS-functionalities for this functions.

For further Information read the German SVN introduction by Richard Atterer, which can be found here:
http://www.medien.ifi.lmu.de/fileadmin/mimuc/mmp_ss04/Projektaufgabe/mmp-subversion.pdf
Java ME

- slim Java for mobile devices
- Java ME stack
  - configuration + profile + additional APIs
- Configuration
  - JVM + minimal amount of functionality
  - subset of Java SE
  - e.g. CLDC 1.1
- Profiles
  - enhance the configuration with functionality
  - APIs for user interface, persistent storage, etc.
  - e.g. MIDP 2.0
- Additional APIs for Bluetooth connections, Multimedia and more
The Java ME Universe

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

The diagram shows a spectrum from smaller devices (Pagers, Mobile Phones, PDAs, Car Navigation Systems) to larger devices (Internet Appliances, Set-Top Boxes). Each category has its own set of profiles and specifications, catering to different levels of functionality and resource availability.
MIDlets

- MIDP applications are called MIDlets
- Every MIDlet is an instance of javax.microedition.midlet.MIDlet
  - Constructor
  - Implements lifecycle methods
- Conceptually similar to Applets
  - Can be downloaded
  - Executed in host environment
MIDlet Life Cycle 1/2

- **constructor**
- **Paused**
  - `startApp()`
  - `resumeRequest()`
  - `pauseApp()`
  - `notifyPaused()`
- **Active**
  - `destroyApp()`
- **Destroyed**
  - `destroyApp()`
  - `notifyDestroyed()`
MIDlet Life Cycle 2/2

- Application Manager controls the installation and execution of MIDlets
- Start of a MIDlet: constructor + startApp() are executed by the Application Manager
- MIDlet
  - Place itself in paused state (notifyPaused())
  - Destroy itself (notifyDestroyed())
- One method for every state transition
MIDlet Build Cycle 1/2

1. Edit source code
2. Compile (like compiling normal java)
3. Preverify
   • Bytecode verification (makes sure it behaves well + won’t do nasty things) is split into two steps
   • lightweight second verification on the mobile device (standard verification too memory intensive)
   • special class format (adds 5% to normal class file size)
   • Normally not visible for the programmer
MIDlet Build Cycle 2/2

4. (Application) Package, MIDlet Suite
   - MIDlets + Classes + Resources + 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
     - Normally used for installation

5. Test or Deploy
Anatomy of a MIDlet Suite

Diagram:
- MidletSuite.jad
- MidletSuite.jar
- Contents of 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 $\Rightarrow$ Scaling
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’)
- New Mobile Phone contain the File API, which allows direct access to the file system
Persistent Storage: Record Stores

- **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 !!!
Recommended IDEs

  - Much better Java ME support then EclipseME (e.g. graphical interface editor)

  - May be better for developers experienced in using eclipse
Exercise 1

- Basics of Java ME
- Deadline October 23rd, 12 p.m.
- Check your solution to your personal SVN folder inside your team’s SVN repository
Have fun!