



7
street | media

Välkommen!

- ✓ **Vem är jag?**
 - Magnus Bladh
 - Pluggat data och matematik i Uppsala. Därefter jobbat ett par år som programmerare på bl.a RSA Security och Vattenfall.
 - Jobbar nu mer med spelutveckling på Streetmedia 7.
- ✓ **Streetmedia 7:**
 - Uppsala företag
 - Mobil spel och applikations utveckling
 - Kontor i glunten

2

7
street | media

Marknaden

World wide

- ✓ 2003: 4 miljarder kr.
- ✓ 2005: 15 miljarder kr (prognos)

Europa

- ✓ 2003: 500 miljoner kr
- ✓ 2005: 2 miljarder kr (prognos)
- ✓ 2006: 4,5 miljarder kr (prognos)

- ✓ I Sverige säljs 2005 150 000 spel / månad

3

7
street | media

Marknaden

World wide

- ✓ 2003: 4 miljarder kr.
- ✓ 2005: 15 miljarder kr (prognos)

Europa

- ✓ 2003: 500 miljoner kr
- ✓ 2005: 2 miljarder kr (prognos)
- ✓ 2006: 4,5 miljarder kr (prognos)

- ✓ I Sverige säljs 2005 150 000 spel / månad.

4

7
street | media

Ny teknik

- ✓ GPRS / UMTS
- ✓ Bluetooth.
- ✓ 3D grafik.
- ✓ GPS.
- ✓ Bild / film.
- ✓ Ljud.
- ✓ Nya Apier.

7
street | media

Hur blir det roligare

- ✓ **Nya spelkoncept.**
- ✓ **Ny teknik.**
- ✓ **Mobilitet.**
- ✓ **Populärkulturella fenomen.**
- ✓ **Inte bara spel (böcker, toys...).**



Vad är problemen då? – telefonen

- ✓ Lite minne (64 kB).
- ✓ Processor kapacitet. (100 – 400 Mhz)
- ✓ Liten skärm (128x128).
- ✓ Knappsats.
- ✓ Olika API:er.
- ✓ Massor av telefoner.
- ✓ Telefon buggar.

7



Hur funkar det?

- ✓ Java, BREW, C/C++, Mobile VB?
- ✓ J2ME är idag standard.
- ✓ Java är förinstallerat på telefon.
- ✓ "Bara" att programmera.



Vad behöver man

Måste ha:

- Dator med Windows / Linux
- JDK (Java Development Kit)
- WTK (Wireless toolkit)
- Texteditor || IDE (Eclipse, JBuilder, ...)
- "Javatelefon"

Bra att ha:

- ANT
- Antenna
- Emulatorer (Nokia, SonyEricsson, Samsung, Motorola)

Underlättat att ha:

- Bluetooth Dongle / kort / ...
- SVN / CVS
- J2ME Polish

9



street | media



The lifestyle entertainment specialist.



Vad är J2ME?

- ✓ **Plattform** (bestående av teknologier /spec avsedda för "small devices").
- ✓ **Består av Configurations, profiles & packages.**
- ✓ **Configuration** (specificationer för JVM samt bas API:er).
- ✓ **Profiles** (Adderar API:er för att kunna skapa applikationer. Ex: UI, storage, app-lifecycle).
- ✓ **Optional Packages** (Tilläggs API:er t.ex. Bluetooth API, MMAPi, 3D-API)

11



J2ME stack



12

- ✓ Configuration för "small devices". Skall ha:
 - >128 kB persistent memory.
 - >32 kB volatile memory.
 Kräver uppkoppling.
- ✓ CLDC stödjer klasser från följande J2SE packages:
 - java.lang, java.io, java.util .
 - starkt reducerade.
- ✓ Utöver dessa finns endast stöd för GCF (generic connection framework).

- ✓ Profile that will enable application development within mobile and wireless connected devices.
- ✓ Finns två versioner MIDP1.0 & MIDP 2.0.
- ✓ Innehåller API:er för:
 - User Input, persistent storage, messaging, networking,...
- ✓ Innehåller specifikation om MIDlets och application lifecycles.
- ✓ Undvik missförstånd:
 - MIDP != J2ME
 - MIDP != All wireless java technology

- ✓ En MIDlet är ett program avsett att köras på ett "device" med MIDP.
- ✓ Mobile devices interagerar med MIDletar via AMS (Application Management software).
- ✓ En MIDlet kan befinna sig i 3 olika tillstånd.
 - MIDlet active.
 - MIDlet paused.
 - MIDlet destroyed.
- ✓ En MIDlet eller flera MIDlets kan packas ihop till en MIDlet suite.

- ✓ Jad och Jar filer är grundstenar i en s. k. MIDlet suite.
- ✓ JAD = Java Application Descriptor.
 - beskriver: jar storlek, placering, .
 - configuration & profile requirements.
 - Kan innehålla s.k. attributes.
- ✓ JAR = Java Archive.
 - Innehåller en eller flera MIDlets. Samt "resources" t.ex bilder, ljud, text, ...
- ✓ .mf = Manifest.
 - Innehåller samma data som Jad-filen. Är placerad i Jar-filen



```

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

public class HelloMIDlet extends MIDlet implements CommandListener {
    private Form mMainForm;

    public HelloMIDlet() {
        mMainForm = new Form("HelloMIDlet");
        mMainForm.append(new StringItem(null, "Hello, MIDP!"));
        mMainForm.addCommand(new Command("Exit", Command.EXIT, 0));
        mMainForm.setCommandListener(this);
    }

    public void startApp() {
        Display.getDisplay(this).setCurrent(mMainForm);
    }

    public void pauseApp() {}
    public void destroyApp(boolean unconditional) {}
    public void commandAction(Command c, Displayable s) { notifyDestroyed(); }
  }
  
```

Hur blir det en MIDlet då?

1. Skapa ett projekt.



2. Ange projektnamn och namn på MIDlet:en.



3. Placera Källkod och bilder rätt map i projektet.

4. Tryck på Build.

5. Tryck på Run.

Programmerings Knep 1.

- ✓ Tänk **ALLTID** på telefonens begränsningar (minne och CPU).
- ✓ Flytta om möjligt beräkningsintensiva funktioner till en server.
- ✓ Bygg små applikationer.
 - Använd obfuscators.
 - Minimera grafikstorlek (olika bildbehandlingsprogram jfr photoshop och gimp).
- ✓ Använd så lite runtime memory som möjligt.
 - Hjälp din garbage collector: Sätt färdig använda objekt till null.
 - Lazy instantiation: Skapa endast objekt då de behövs.
 - Släpp "resources" så snabbt som möjligt.
 - Undvik "exceptions"... eller inte.

Programmerings Knep 2.

- ✓ Tänk alltid "performance".
 - Optimera på rätt ställe. Använd profiler.
 - Använd lokala variabler i största möjlighet.
 - Undvik "String" konkatenering. StringBuffer är mycket snabbare.
 - Se upp med *synchronized*. Mycket overhead.
 - Kolla dina loopar.
 - Undersök compilatorns *-optimization options*.
- ✓ Testa rätt.
 - Lita inte alltför mycket på emulatorn.
 - Lita inte på telefonen (vad som funkar på en telefon funkar inte på alla).