
WILLCOM端末におけるアプリケーションの作成方法、その他

S1130006 阿部

背景 (1)

- WILLCOMはPHS事業を行う電気通信事業者。
(2005年にDDIポケットから改称)
- ポケベルに代わり、中高生の間でヒットする。
- 携帯電話の低価格化による普及により、契約者数が減少する。
- 2001年に定額制のデータ通信サービスを開始
(AirH)契約者数減少に歯止めがかかる。
- 海外投資ファンドによる買収

背景 (2)

- 音声通話定額サービスの開始(PHSどうし)
- 高速データ通信サービスの運用開始
(最大408Kbps)

SCCPで所有している端末

- W-ZERO3, W-ZERO3[es]
 - 解像度 640x480 (VGA)
 - OS Windows Mobile 5.0
 - Java VMはJBlend
 - MiniSDカードが使用可能

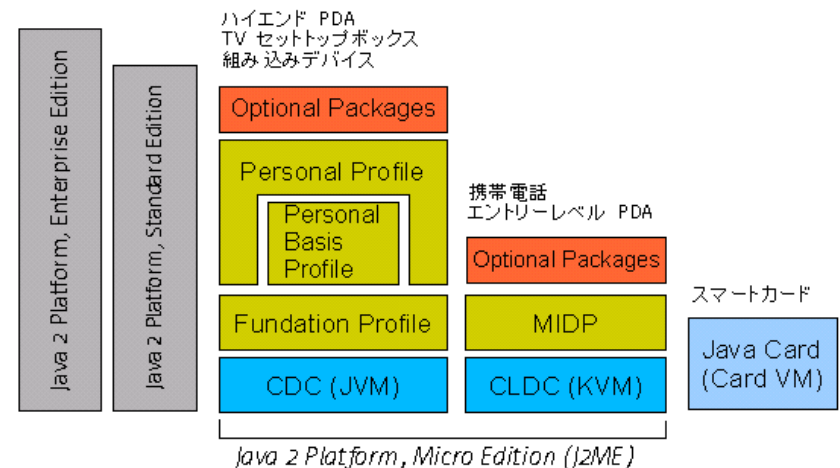


開発のアプローチ

- Sun の Java 2 Platform Micro Edition (J2ME) 準拠のアプリケーションを作成する。
 - 簡単
 - できることが少ない
- Window Mobile プラットフォーム向けネイティブアプリケーションを作成する。
 - システムやハードウェアの制御なども可能

J2MEについて

- J2MEは、携帯電話やPDA向けに最適化されたプラットフォーム
 - プラットフォーム(CLDC) 最低限のライブラリとVMの定義
 - プロファイル(MIDP) 特定の実行環境向けのAPIセット
 - オプションパッケージ



MIDPとは

- Mobile Information Device Profile の略
ミッドピーと読む(エムアイデバイスとも読む)
- 携帯電話向けに定義されたJava実行環境
- WILLCOM,AU(オープンアプリ),SoftBankの携帯電話などで採用されている。
- 現在のバージョンは2.0 MIDP3.0はJSR271として現在開発中

アプリ作成のステップ(J2ME)

1. 作成環境の準備
 2. 仕様の決定、設計と実装
 3. エミュレータにおいて動作の確認
 4. デバッグ
 5. 実機での実行
-

作成環境の準備(J2ME) (1)

- 主な作成環境
 - テキストエディタ + J2ME Wireless Toolkit
 - Eclipse + EclipseME + J2ME Wireless Toolkit
 - NetBeens + J2ME Wireless Toolkit
-

作成環境の準備(2)

- J2SE Development Kit のダウンロードとインストール
 - <http://java.sun.com/j2se/1.5.0/ja/download.html> から**JDK 5.0 Update 11**ダウンロード
(JDK 6.0でも良い)
 - 適当なドライブにインストールし、Pathの設定を行う。

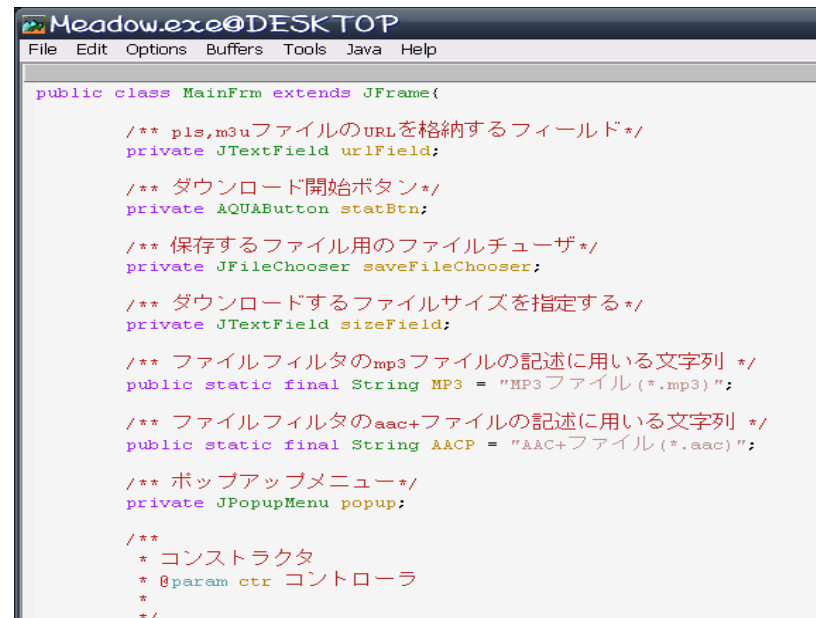
作成環境の準備(3)

- J2ME Wirelessツールキットのダウンロードのダウンロードとインストール
 - J2ME Wireless Toolkit 2.2 とパッチのダウンロードとインストール
 - 必要な端末のエミュレータ用スキンをダウンロードし追加する

作成環境の準備(4)

~ テキストエディタ を使用して作成 ~

- テキストエディタを準備
- 便利なテキストエディタ
 - Meadow
 - EmEditor
 - Xyzzy
 - teraPad



```
Meadow.exe@DESKTOP
File Edit Options Buffers Tools Java Help

public class MainFrm extends JFrame{

    /** pls,m3uファイルのURLを格納するフィールド*/
    private JTextField urlField;

    /** ダウンロード開始ボタン*/
    private AQUAButton statBtn;

    /** 保存するファイル用のファイルチューザ*/
    private JFileChooser saveFileChooser;

    /** ダウンロードするファイルサイズを指定する*/
    private JTextField sizeField;

    /** ファイルフィルタのmp3ファイルの記述に用いる文字列 */
    public static final String MP3 = "MP3ファイル (*.mp3)";

    /** ファイルフィルタのaac+ファイルの記述に用いる文字列 */
    public static final String AACPlus = "AAC+ファイル (*.aac)";

    /** ポップアップメニュー*/
    private JPopupMenu popup;

    /**
     * コンストラクタ
     * @param ctr コントローラ
     * */
}
```

作成環境の準備(5)

Eclipseを使用する場合

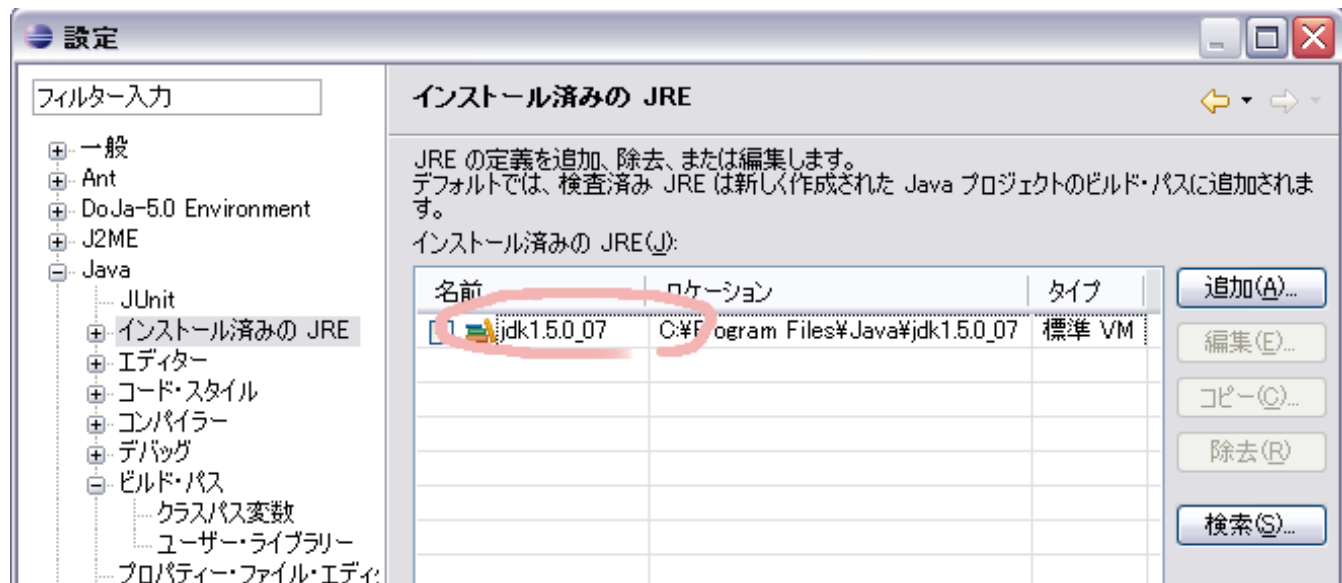
- Eclipse と 言語パッケージのダウンロード
 - <http://download.eclipse.org/eclipse/downloads/index>から最新版のEclipseとLanguageをダウンロード。
 - 好きなディレクトリに解凍し、Eclipse.exeから実行する。(ショートカットを作っておくと良い)
 - 設定を行う

作成環境の準備(6)

Eclipseを使用する場合

■ Eclipseの設定

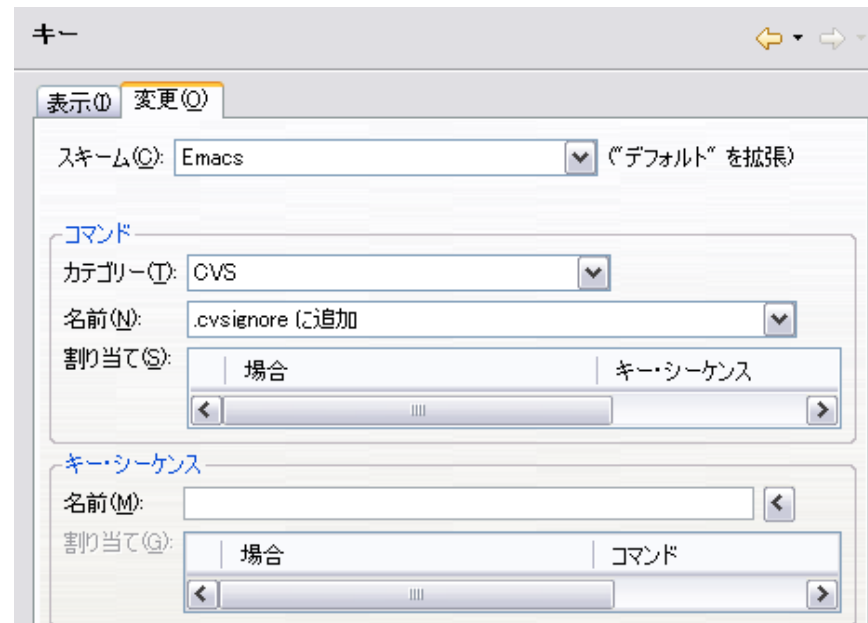
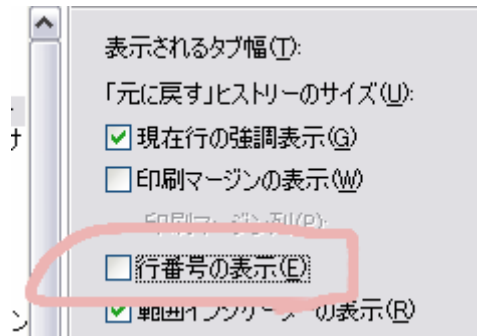
- インストール済みのJREの項目で、VMがJREではなく、JDKになっていることを確認



作成環境の準備(7)

Eclipseを使用する場合

- Eclipseの設定(他にやっておくと良い項目)
 - 行番号の表示(設定 → 一般 → テキストエディタ)
 - ショートカットやキーバインドの設定



作成環境の準備(8)

Eclipseを使用する場合

- プラグインとしてEclipseMEを追加する
 - Eclipseのヘルプ→ソフトウェア更新→検索およびインストールを選択
 - インストールする新規フューチャーを選択し、次へ。
 - 新規リモートサイトに<http://eclipseme.org/updates/>を追加し、終了を選択。インストールするフューチャーとしてEclipseMEが現れるので選択し、インストール。
 - 再起動後、新規プロジェクトからJ2MEのプロジェクトが作成できることを確認。

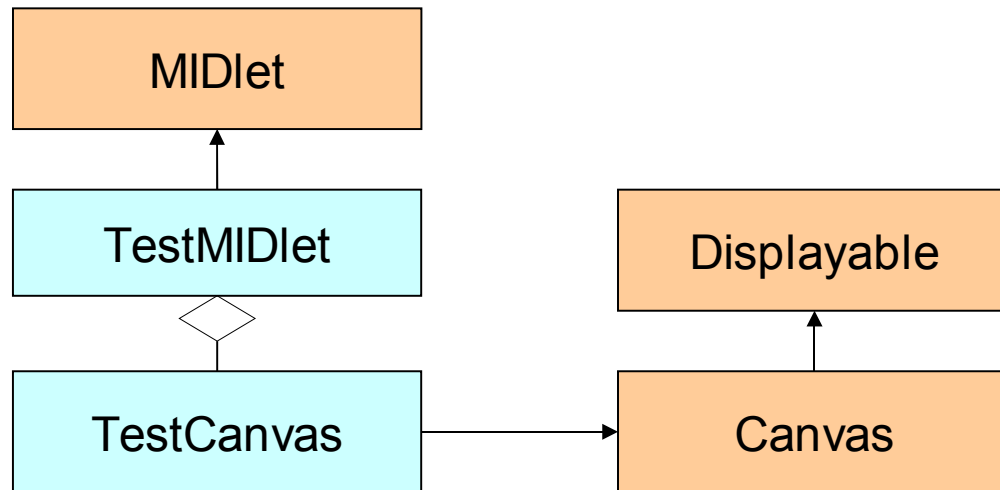
作成環境の準備(9)

Eclipseを使用する場合

- EclipseとJ2ME Wireless Toolkitの関連付け
 - ウィンドウ→設定を選択。
 - J2ME→Device Managerへ進みImportボタンを押す
 - J2ME Wireless Toolkitのあるフォルダを選択
(例 C:/WTK25/)
 - Refreshボタンを押す。使用したいエミュレータのスキンを選択し、終了

実装(1)

- 簡単なMIDPアプリケーションの構成
 - ここでは、画面に四角を描くだけの処理をするアプリケーションを作成する。
 - TestMIDletクラスとTestCanvasクラスを作成



実装(2)

- MIDletクラス(`javax.microedition.midlet.MIDlet`) は、MIDPアプリケーションの抽象クラス。すべてのMIDPアプリケーションはMIDletクラスを継承したクラスから開始する。
- Canvas(`javax.microedition.lcdui.Canvas`)クラスは、携帯電話の画面への描画処理を行うためのメソッド(関数)をもつ抽象クラス。
 - このクラスを継承する際は必ず、`paint()`メソッドを実装しなければならない。

実装(3)

- TestMIDletクラスのソースコード(1)

```
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Canvas;
/**
TestMIDlet クラス
*/
public class TestMIDlet extends MIDlet{

    /** コンストラクタ */
    public TestMIDlet(){
        TestCanvas canvas = new Canvas();
        Display.getCurrent(this).setCurrent(canvas);
    }

    public void startApp(){}
    public void pauseApp(){}
    public void destroyApp(boolean flag){}
}
```

実装(4)

- TestCanvasのソースコード(1)

```
import javax.microedition.lcdui.Canvas;  
import javax.microedition.lcdui.Graphics;
```

```
/** TestCanvasクラス*/
```

```
public class TestCanvas extends Canvas{
```

```
/**
```

```
  描画処理を行う
```

```
  @param g レンダリングに必要なGraphicsオブジェクト
```

```
*/
```

```
public void paint(Graphics g){
```

```
    final int width = 50;
```

```
    final int height = 50;
```

```
    //黒色に変更
```

```
    g.setColor(0,0,0);
```

```
    //画面中央に50x50の正方形を描画
```

```
    g.fillRect(getWidth()/2 - width/2,getHeight()/2 - height/2,width,height);
```

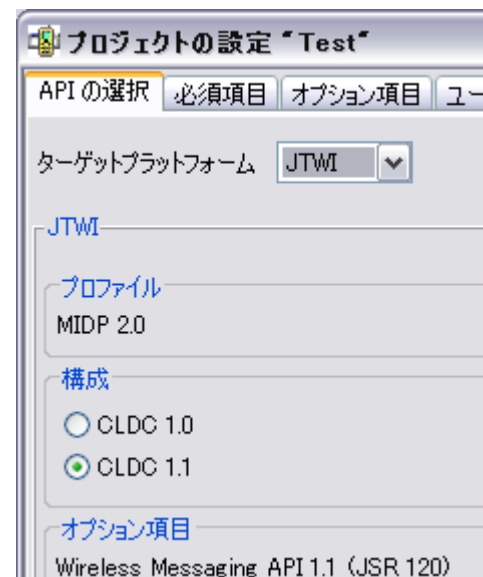
```
}
```

```
}
```

実装(5)

テキストエディタ を使用する場合

- すべてのプログラム→J2ME Wireless Toolkit →KToolbarを選択
- 新規作成をクリックし、プロジェクト名とクラス名(MIDletを継承するクラス)を入力。ここでは、Test,TestMIDletと入力
- プロジェクトの作成を選択、プロジェクトの設定を行う
 - CLDC1.1を選び、その他は必要に応じて設定する



実装(6)

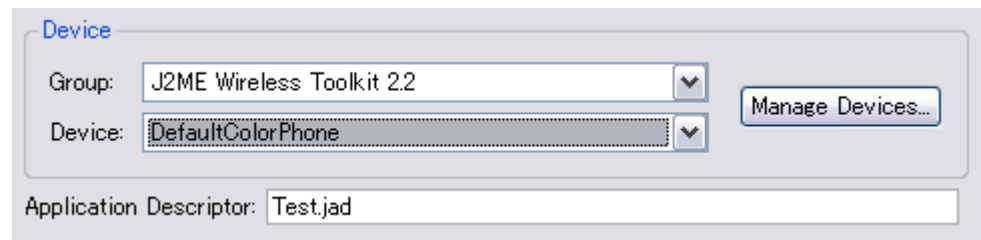
テキストエディタ を使用する 場合

- Wireless Toolkitをインストールしたディレクトリ以下
\\WTK22(WTK25)\apps\Test\src にソースコードを配置
- ビルドボタンを押してビルドを行う。
- ビルドは完了しました。 という文がコンソールに表示されれば、ビルドが成功。

実装(7)

Eclipseを使用する場合

- 新規→プロジェクトを選択
- ウィザードの選択で、J2ME→J2ME MIDlet Suiteを選び、次へ。
- プロジェクト名を入力
- 使用するエミュレータのスキンを選択
- Javaのビルド設定はそのままで完了

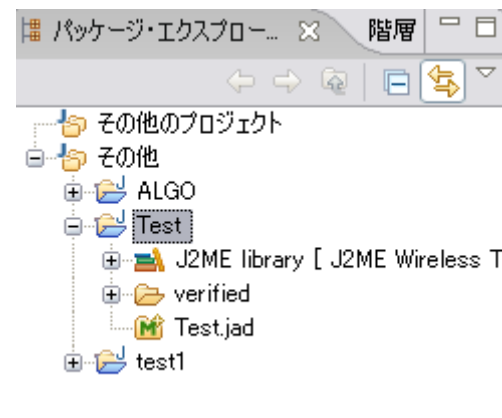


実装(8)

Eclipseを使用する場合

- 画面左側にパッケージエクスプローラが表示されていない場合は、ウィンドウ→ビューの表示からパッケージエクスプローラを選択
- プロジェクト上で右クリックし、新規→その他→J2ME→J2ME MIDletを選択。(新規→クラスでも、MIDletをスーパークラスに指定すれば可)
- プロジェクトの設定を行う。(プロジェクト名にTestMIDlet)

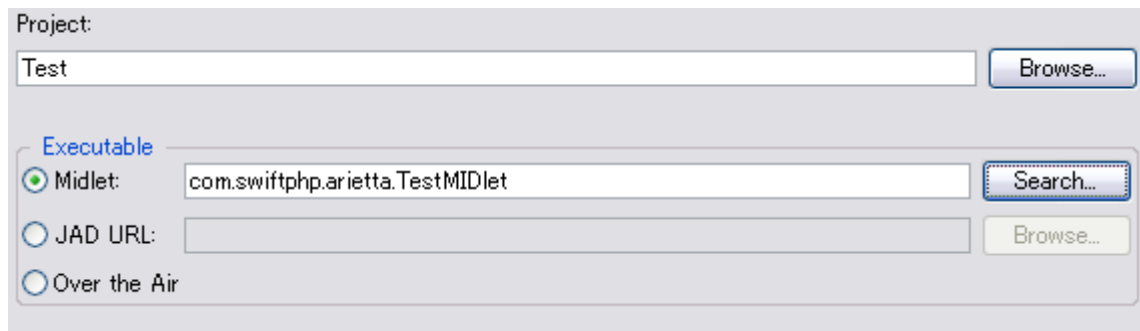
ソース・フォルダー(D):	Test
パッケージ(K):	com.swiftphp.arietta
<input type="checkbox"/> エンクロージング型(Y):	
名前(M):	TestMIDlet
修飾子:	<input checked="" type="radio"/> public(P) <input type="radio"/> default(L) <input type="checkbox"/> abstract(T) <input type="checkbox"/> final(L) <input type="checkbox"/>
スーパークラス(S):	javax.microedition.midlet.MIDlet
インターフェイス(I):	



実装(9)

Eclipseを使用する場合

- 新規→クラスを選び、クラス名にTestCanvas、スーパークラスにCanvas(Ctrl + spaceで自動補完、Emacsキーバインドのときは、Alt + /)を指定
- ソースコードとコメントを書く
- 実行→構成および実行を選択。Wireless Toolkit Emulatorをダブルクリックする。
- ExecutableはMidletにチェックを入れ、Searchボタンを押しMIDletクラスを継承したクラス(TestMIDlet)を選択



実装(10)

Eclipseを使用する場合

- コンソールにエラーが出ず、エミュレータが起動すればビルド成功

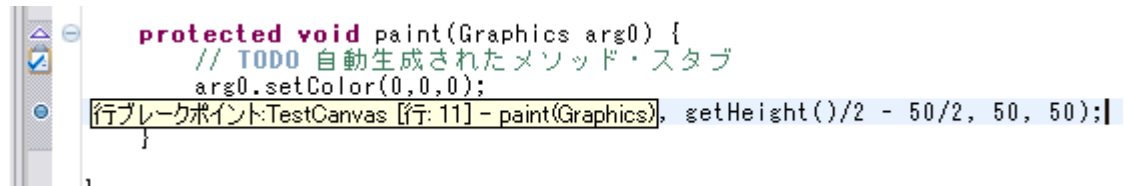
エミュレータで動作を確認

- J2ME Wireless Toolkitを用いる場合は、ビルド後に実行ボタンを押しエミュレータを起動
- Eclipseの場合は、実行→構成および実行→Wireless Toolkit Emulator から実行
- 図のような画面ができれば成功



デバッグ

- Eclipseの場合、動作を停止させたい場所にブレークポイントを置き、実行→構成およびデバッグから、デバッグを開始する



```
protected void paint(Graphics arg0) {  
    // TODO 自動生成されたメソッド・スタブ  
    arg0.setColor(0,0,0);  
    行ブレークポイント:TestCanvas [行: 11] - paint(Graphics), getHeight()/2 - 50/2, 50, 50);|  
}
```

- Wireless Toolkitを使用している場合は、`System.out.println()` 等を使用し、コンソールへの出力を確認する

実機への転送

- WILLCOM端末の場合は、USBケーブルを使って直接転送が可能
- binフォルダ内のjarファイルとjadファイルを転送し携帯端末上で実行する。(Wireless Toolkit を使う場合はプロジェクト→パッケージ→パッケージを生成でjarファイルを生成するひとつようがある。)

その他

- MIDPを学ぶ前にJava(J2SE)をある程度習得すると良い。
(Javaの基礎→よく使用するクラスライブラリを覚える→J2SEのGUI(AWT, Swing)でGUIアプリケーションの作り方を学ぶ)
- UMLやデザインパターンを必要に応じて学ぶ。