

0から始めるiアプリ(2)

環境を整えることは難しくなくなりました。

前章ではインストールまでを体験しましたが、この章ではビルドとよばれるJavaプログラミングの一般的な作法について学習しましょう。

また、iアプリ開発ツールを使えば、Javaのプログラミング開発は初心者にも容易になりますが、そのかわり、それならではの作法を覚えなくてはなりません。

ただし、その行程は一度覚えてしまえばあとは繰り返しです。

この単元で開発工程を体得しましょう。

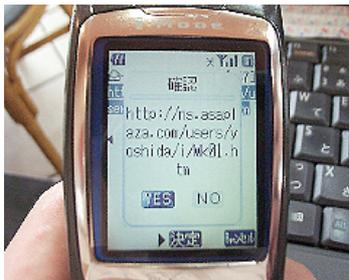
1. ユーザーがiアプリを利用する流れ

一般ユーザーがiアプリを利用するにはホームページからダウンロードして、自分の携帯電話で実行します。

セキュリティ的には現在はキツメの設定になっていますので、実機で実行といってもできることは限られています(ユーザーのアドレス帳をiアプリでは利用できない等)。

ですから、iアプリでもダウンロード型のアプリケーションも開発できるのですが、それはダウンロードしたサイトのみデータをダウンロードできるような使用になっています。

さて、ユーザーのiアプリ利用は次のような手順をふむことになります。



① 503iシリーズでiアプリが掲載されているサイトへユーザーがアクセスする。

(iモードを使ってiアプリ掲載サイトへアクセスし始める)



② iアプリをダウンロードするハイパーリンクをクリックする。

(ダウンロードのハイパーリンクが表示される)



③ [決定]ボタンを押す。

(ダウンロードが始まる)



④ 実機にiアプリが格納される。
(iアプリがリストアップされる)



⑤ ダウンロードされたiアプリが選ばれているのを確認の上、[決定]ボタンを押す。
(JavaがVMが起動する)



⑥ iアプリが実行される。
(このときは通信費はかかっていない。実機のみでの実行。)

②の行程ではユーザーはiモードを使ってホームページにアクセスしています。

ただ、サイト側では、ホームページのハイパーリンクには、ADFと呼ばれるiアプリのファイル名や作成年月日などがかかれた設定ファイルを読みに行くようになっています。

実機の方ではその設定ファイルの中身を読み、実際のiアプリの実体を読み込みます。

つまり、iアプリを利用してもらうために、開発者は次のファイルを用意することになります。

- (1) iアプリ ~ .jar
- (2) ADFファイル ~ .jam
- (3) ホームページファイル ~ .html

(2)と(3)はテキストファイルですが、(1)はJavaで書かれたバイナリー形式です。ですから、開発ツールなどがなくとも(2)と(3)はテキストエディタ等で開発が可能です。

ADFファイルには、iアプリ実行体のファイル名、最終更新日、アプリケーションサイズなどiアプリ実行体に関する情報が記述されています。

(iアプリ開発TOOLを使用すれば、自動的に生成されます)。

3. はじめてのiアプリ作成～作業フォルダの作成～

それでは、実際にiアプリを開発してみましょう。
プログラミングの勉強で一番最初に知る、画面上に文字を表示させる最も伝統的な課題です。
仕様ソフトは、SeiichiNishimura氏のiアプリ開発Tool DoJa版です(前節参照)。

【課題1】画面上に「Hello!PC」と赤い色で表示させなさい。

【仕様】 表示文字 : Hello!PC
表示色 : getColorOfRGB(255,0,0)・・・赤です(詳しくは後章)。
表示座標値 : x座標 30 y座標 60
ソース名 : Mk01

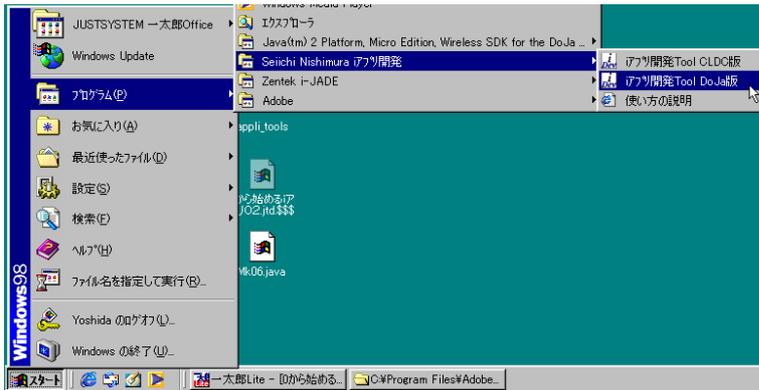
【参考画面】



【入カソース】-----

```
import com.nttdocomo.ui.*;
public class Mk01 extends IApplication {
    public void start(){
        Display.setCurrent(new Mk01d());
    }
}
class Mk01d extends Canvas{
    public void paint(Graphics g){
        g.lock();
        g.clearRect(0,0,getWidth(),getHeight());
        g.setColor(g.getColorOfRGB(255,0,0));
        g.drawString("Hello!PC",30,60);
        g.unlock(true);
    }
}
}
-----
```

- ① [スタート]→[プログラム]→[Seiichi Nishimura iアプリ開発]→[iアプリ開発 Tool DoJa版]をクリックします。



(iアプリ開発 Tool DoJa版が起動します。2001年04月27日現在はバージョンが0.7になっています)



【ワンポイント】iアプリ開発 Tool DoJa版の画面

① iアプリを新規に作成する際に、プロジェクトというiアプリ専用フォルダのようなものを作るのですが、それを作成します。

② プロジェクトを作成後、実際のソースを入力していくのですが、そのソースファイルがリストアップされる欄です。

③ プロジェクトを作成後、この[追加]ボタンをクリックし、ソースを記述する実際のファイル名(メインクラス名と表記されます)を入力します。

④ メインクラスを追加した後、この[編集]ボタンをクリックし、ソースを入力していきます。クリックするとデフォルトではメモ帳が起動するようになっています。

⑤ 新規にプロジェクトを作成する際には、メインクラス欄にリストアップされているファイルをこのボタンを使って削除します。

その際には、実際のフォルダからも削除されてしまうので、注意が必要です。必要なソースファイルは必ず別な場所に移動しておきましょう。

⑥ 後述の画像ファイルを扱いには、この欄に、[追加]ボタンをクリックし、そのファイルをリストアップしておかなくてはなりません。

⑦ NTTドコモ社から提供されているDoJaにて実行確認する際には、このボタンをクリックして、確認します。

すぐ上に、プリーベリファイなどのメッセージも表示されます。

⑧ i-JADEを使って実行確認する際にクリックします。

下にあるメニューで、p503i、n503i、f503i用のi-JADEを起動できます。

⑦、⑧の動作テストボタンはプリーベリファイが成功しても、すぐに消えてしまうことがあります。その際には、ソース内正しくない記述がある場合がほとんどですので、プリーベリファイが成功してもソースを再確認するようにしてみましょう。

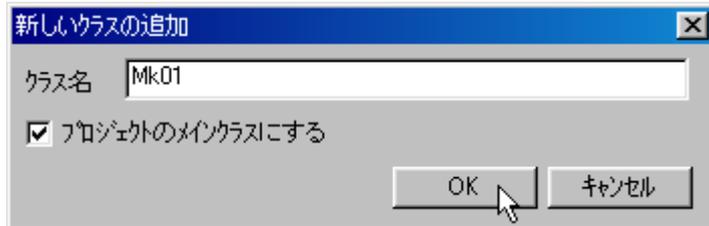
①の上部にある[環境設定]ボタンを使えば、ソースを入力するときのエディタを変更することができます。使い慣れたものがあれば、それを設定しておくとう便利です。

4. はじめてのiアプリ作成～作業フォルダの作成～

①メインクラス側の[追加]ボタンをクリックします。



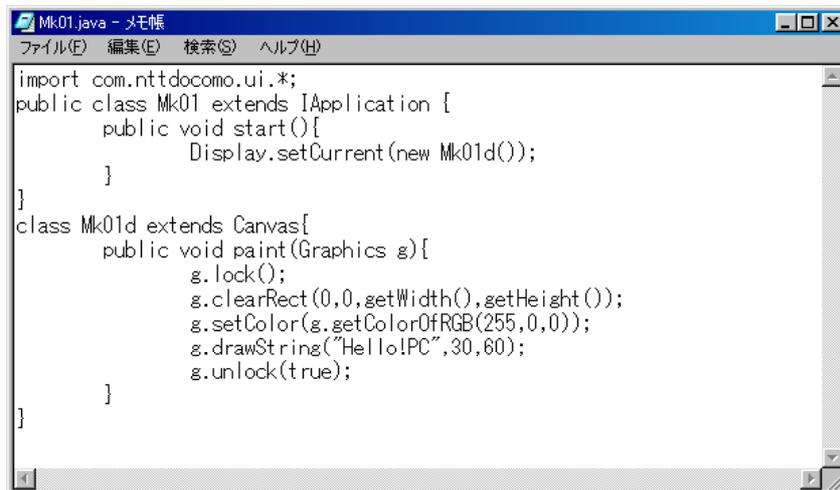
②直接入力モードで、Mk01と入力し、[OK]ボタンをクリックします。



③[編集]ボタンをクリックします。



④直接入力モードで正しく入力します。



【注意事項】

- 直接入力モードで。
- 大文字小文字を正確に。
- セミコロンやカッコの数に注意です。
- . はピリオドです。
- *はアスタリスクです。
- ;はセミコロンです。

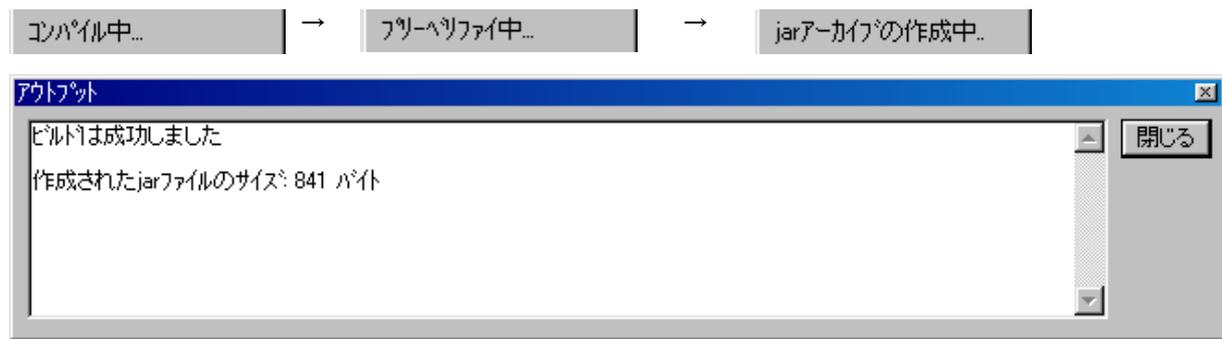
⑤[ファイル]→[上書き保存]を選びます。

(とくに画面上変化はありません。)

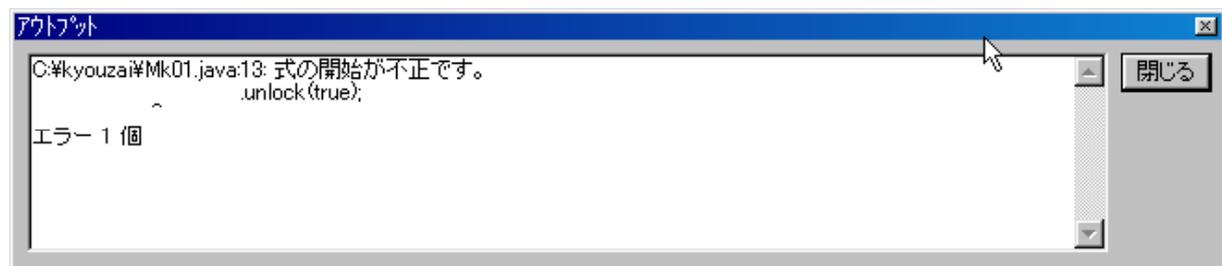
⑥[ビルド]ボタンをクリックします。



([ビルド]ボタンをクリックすると、ソースが正しければ以下のようにメッセージが表示されます。)



(ソースが不適切な場合は以下のように表示されますので、メモ帳画面で再確認し、再び上書き保存し、[ビルド]ボタンをクリックし、ビルドを試みます。)



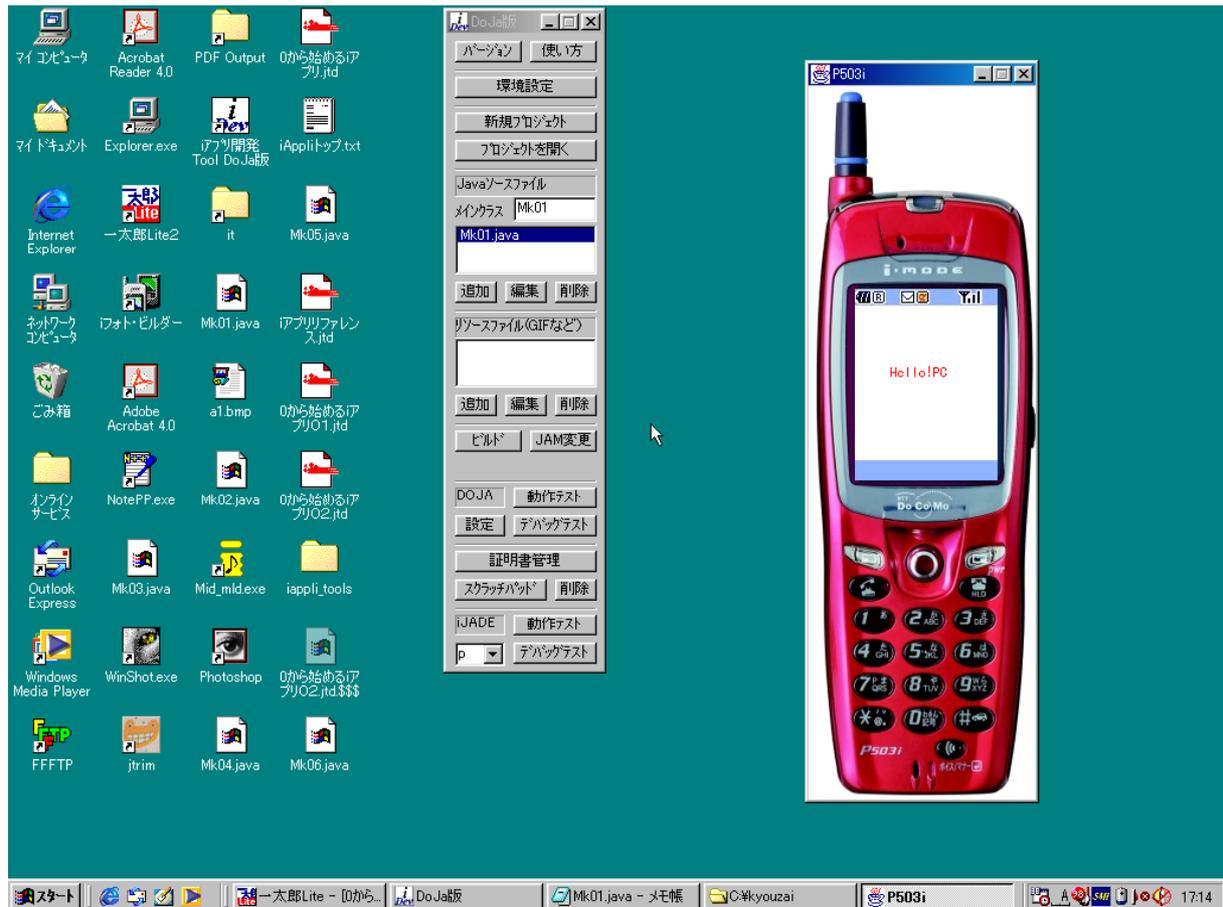
(正しくビルドされた場合には、(C:)ドライブのkyouzaiフォルダには以下のように保存されています。)



⑦ [動作テスト]ボタンをクリックします。



(i-JADEが起動し、実行画面が表示されます)



⑧ i-JADEの[×]ボタンをクリックし、i-JADEを終了させます。
(動作テストごとに新規のウィンドウで実行画面が表示されるため)

⑨ iアプリ開発 Tool DoJa版の[×]ボタンをクリックし、終了します。

【ワンポイント】開発はたいへん！？

iアプリ開発 Tool DoJa版を使えば、ビルドやjarファイルの生成が楽になったといっても、ソース記述のたいへんさは変わりません。

市販の開発ツールなどでは長い命令などは自動的に、つづりが表示されたり、カッコの数もチェックしてくれたりして、とても便利です。

メモ帳で記述する以上は、ミスは当たり前と考えましょう。回数をこなしてくるとメッセージから、だいたい場所を推察できるようになります。

初歩の段階でよくあるミスは次のようなものです。

- ・カッコの数があっていない。
- ・スペルミス × IApplication ○ IApplication (1と小文字のLを間違えている)
- ・セミコロンが抜けている。(これはJavaの文法として、命令の最後には必ずつけるものなので、しっかりと体に覚え込ませましょう。)
- ・修正をしているのに、上書き保存をせずにビルドしている。(上書き保存をしなければ、ビルドは前のソースを使って行われます。また、メモ帳では[CTRL]+[S]というショートカットも使えませんので、注意が必要です。)
- ・意外と、voidをvioidと記述するように順番違いというつづりミスも多いです。