

0から始めるiアプリ(4)

iアプリをプログラミングする際に必要なのはJavaというプログラミング言語の基礎知識です。

独自の命令や駆使してプログラミングをするわけですが、Javaというベースになっている言語を知らないでプログラムを記述することはあり得ません。

本節では、新規プロジェクトを続けて作る際の注意事項と、簡単なJavaプログラミングの作法を体得しましょう。

1. 新規プロジェクトを作成する

新規にプロジェクトを作成する場合は、注意が必要です。



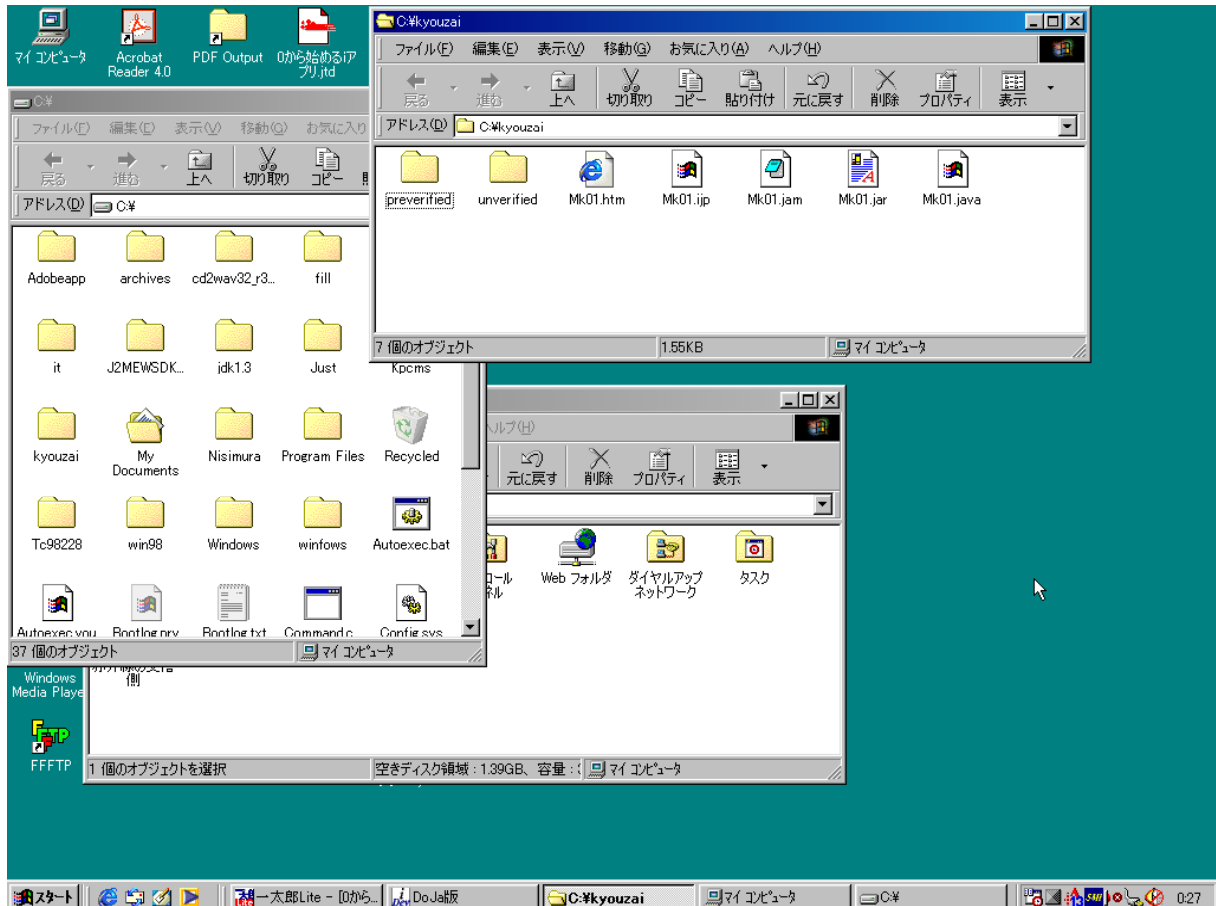
継続して、プロジェクトを作成する場合には、左記のようになっているはず
です。

ここには、メインクラス名と使用されている、~.javaファイル名が列挙されて
います。

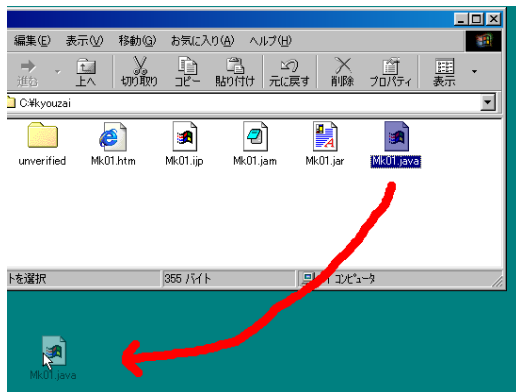
新規にプロジェクトを作成する場合には、このMk01.javaというソースファイ
ルを一旦移動して、新規に作成しなければなりません。

ここにそのファイルを登録したまま新規に作成してしまうと、そのファイルを
削除せざるを得なくなってしまうからです。

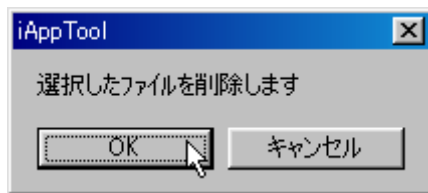
① デスクトップ上の [マイコンピュータ] → (C:) → [kyouzai] を開きます。



② Mk01.javaをデスクトップに移動します。



③ iアプリ開発 TOOL をアクティブにし、[削除] ボタンをクリックします。

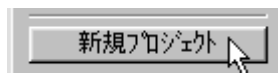


④ [OK] ボタンをクリックします。(メインクラスが削除されます)

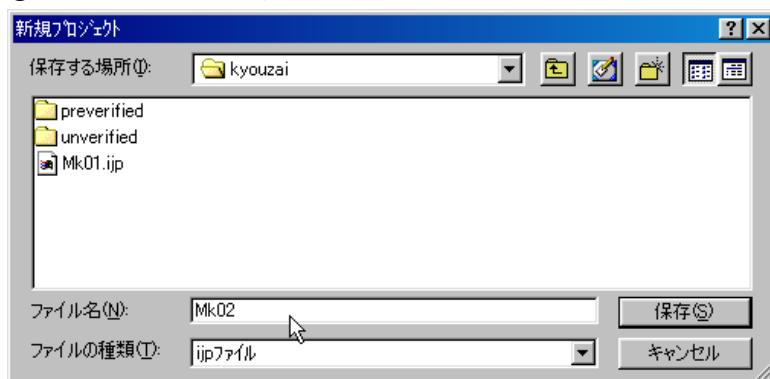
⑤ メインクラス欄をクリックし、[BACKSPACE] キーを押して、Mk01 という名前を削除します。



⑥ [新規プロジェクト] ボタンをクリックします。



⑦ Mk02 と入力します。

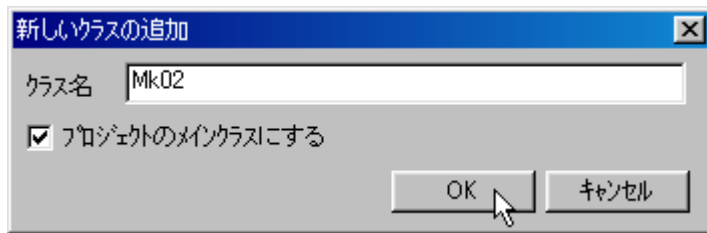


⑧ [保存] ボタンをクリックします。

⑨ [追加] ボタンをクリックします。



⑩ Mk02と入力し、[OK]ボタンをクリックします。



(メインクラスにMk02が追加されます)



2. 基本的なJavaのソースを書きましょう

【課題2】画面を黒く塗りつぶし、縦方向中央に、「通話中」と白い文字で表示させなさい。



① [編集]ボタンをクリックします。



② 次のソースを入力します。

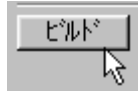
```
import com.nttdocomo.ui.*;
public class Mk02 extends IApplication {
    public void start(){
        Display.setCurrent(new Mk02d());
    }
}
class Mk02d extends Canvas{
    int w,h;
    public Mk02d(){
        w=getWidth();
        h=getHeight();
    }
    public void paint(Graphics g){
        g.lock();
        g.setColor(g.getColorOfRGB(0,0,0));
        g.fillRect(0,0,w,h);
        g.setColor(g.getColorOfRGB(255,255,255));
        g.drawString("通話中",0,h/2);
        g.unlock(true);
    }
}
```

③メモ帳画面で、[ファイル]→[上書き保存]を選びます。

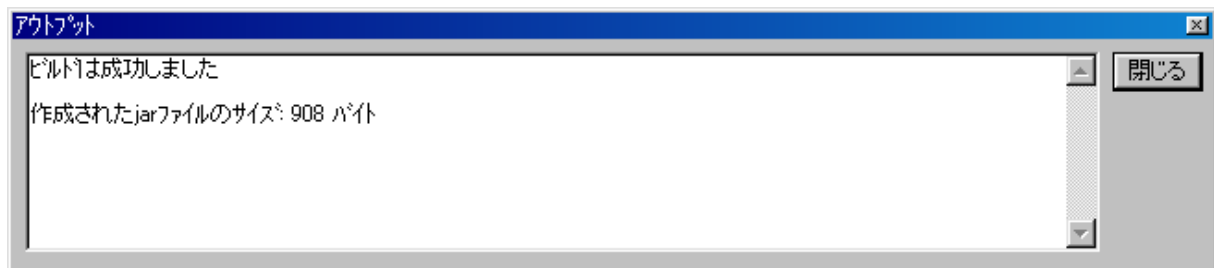
(画面上、とくに変化はありません)

④iアプリ開発 TOOLの画面上で、[ビルド]ボタンをクリックします。

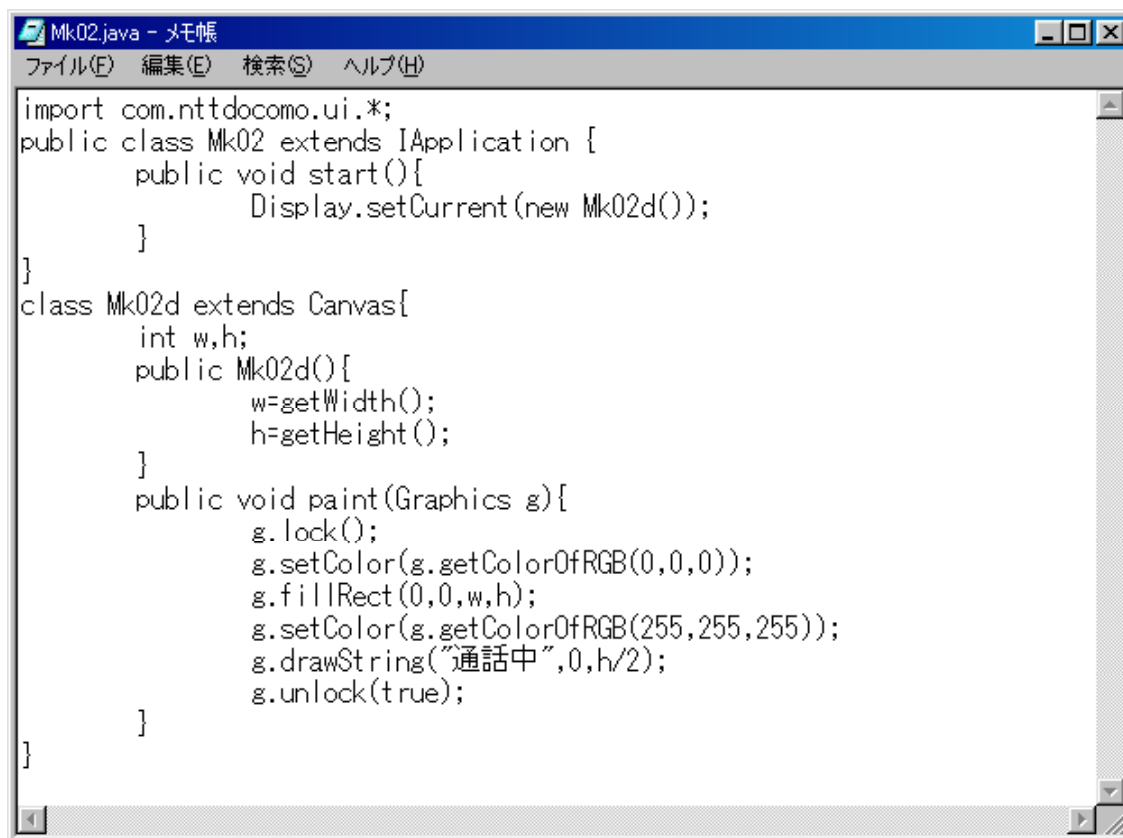
⑤[ビルド]ボタンをクリックします。



([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メッセージを参考に、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、再ビルドを試みます。)

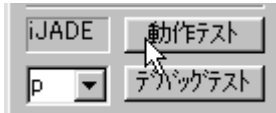


⑥ DoJaのほうの、[動作テスト]ボタンをクリックし、動作確認します。



⑦ エミュレータ画面の[×]ボタンをクリックし、DoJaエミュレータ画面を閉じます。

⑧ いちおう、i-JADEのほうでも確認します。i-JADEのほうの、[動作テスト]ボタンをクリックします。



(若干、画面サイズが違うことを確認します。しかし、いちおう縦方向の位置は両方ともほぼ中央にあります。)



⑨ エミュレータ画面の[×]ボタンをクリックし、i-JADEエミュレータ画面を閉じます。

【ワンポイント】メインクラスはいったん移動する

iアプリで開発TOOLでは新規にプロジェクトを作成する際、以前の～.javaファイルを削除しなくてはなりません。

せっかく入力してビルドしたソースはとっておきたいものですよ。

そこで、一旦作業フォルダを開き、その中からつくった、～.javaファイルを移動し、そして、新たに新規メインクラスを作成します。このときに、メインクラス名はファイルを削除しただけでは、消えませんが手動で消してください。

そうしないと、次に作るメインクラスでビルドの際にエラーになってしまいます。

かならず、メインクラス名はソース内のそれと一致してはいけません。

3. Javaのソースの基本構造を理解しましょう

本節のソースではあえて、Javaの基本的な構造を理解してもらうために、多めのソースを入力してもらいました。

```
-----Mk02.java-----
01 : import com.nttdocomo.ui.*;
02 : public class Mk02 extends IApplication {
03 :     public void start(){
04 :         Display.setCurrent(new Mk02d());
05 :     }
06 : }
07 : class Mk02d extends Canvas{
08 :     int w,h;
09 :     public Mk02d(){
10 :         w=getWidth();
11 :         h=getHeight();
12 :     }
13 :     public void paint(Graphics g){
14 :         g.lock();
15 :         g.setColor(g.getColorOfRGB(0,0,0));
16 :         g.fillRect(0,0,w,h);
17 :         g.setColor(g.getColorOfRGB(255,255,255));
18 :         g.drawString("通話中",0,h/2);
19 :         g.unlock(true);
20 :     }
21 : }
```

Javaではプログラムをいくつかのブロックに分けて記述します。

上記のソースでも、よくみると、{ と } で区切られたいくつかの部分に分かれることに気づくはずで

す。
public class Mk02 extends IApplication という部分と、class Mk02d extends Canvas です。

iアプリでは、まず最初に、①の部分を実行します。

Mk02というIApplicationを実行しなさい、という意味になります。

そして、その内部の、start(){・・・}で囲われた部分が具体的には実行されることになります。

そこには、Mk02d()と書かれていますので、次のclass Mk02d extends Canvasが実行されるわけです。

iアプリにおける、Javaプログラムの基本構造は次のようになります。

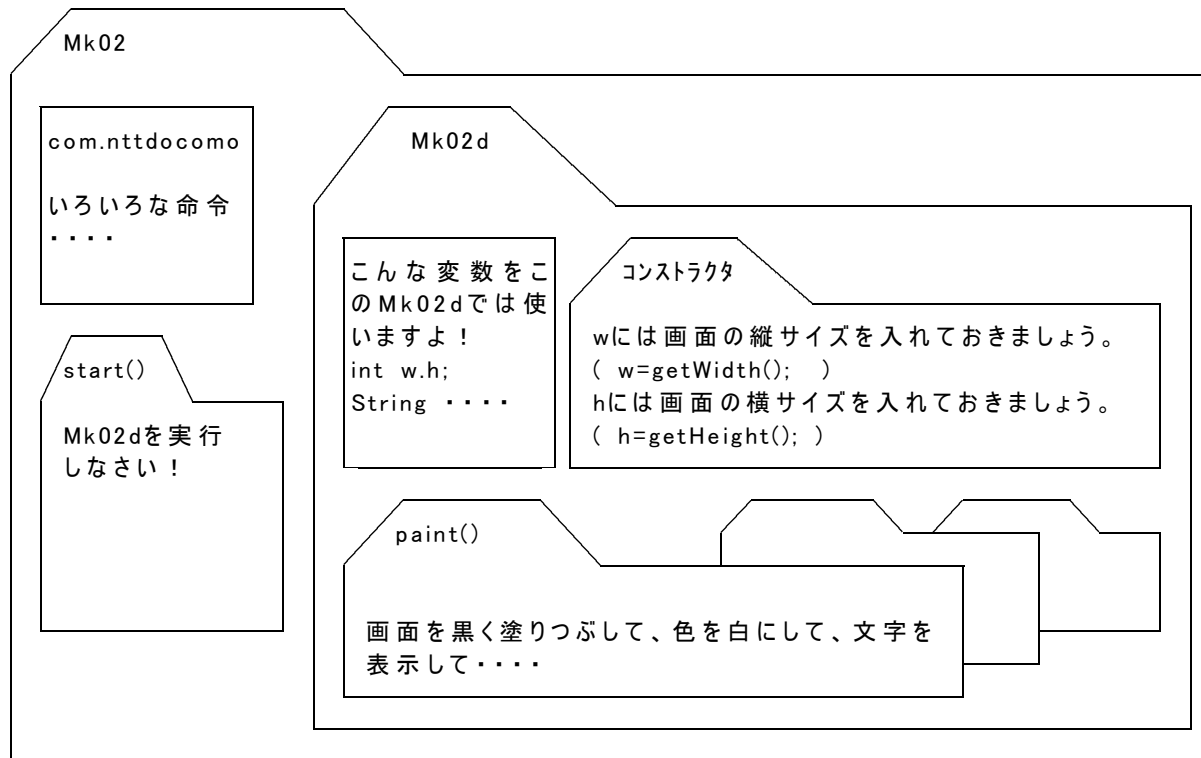
- 01 宣言部 : iアプリで使う命令を読み込む部分。
この場合は、命令セット「com.nttdocomo.ui」のすべて(*)を読み込みなさい、という意味。
- 02～06 メインクラス部 : この部分で実際に実行するクラスの名前を定義します。
- 07～21 実行部 : 実際にプログラムで動作する部分。

そして、実行部は次の部分に分類されます。

- 08 宣言部 : 変数(数値を入れる箱)など、そのクラス内で使用するすべての変数をここで記述しておきます。「クラス内で変数を宣言しておく」という際にはここを指します。
- 09～12 コンストラクタ : 宣言部で定義した変数の初期値を代入します。変数に限らずiアプリで利用する様々な設定値の初期化を行います。
- 13～20 実行部 : このソースではペイント・描画部がそれにあたります。このように実際の実行をする部分群を言います。これ以外には、アクションゲームで当たり判定をしたり、どのキーが押されたかなどを調べたりするなど、実際の処理をpublic void ...(){・・・}でくくった内部にそれらの処理を記述しておきます。

【ワンポイント】iアプリはフォルダにまとめるようなもの？！

前述のソースではいくつかのブロックにプログラムが分かれていることがわかりました。そこで、次のような図にしてみるとわかりやすいでしょう。



このようにいくつかのまとまりに分けられます。

後述のスレッドという処理では、機能を付加するようになるのですが、その際も、フォルダに特殊な設定をする、いうようにイメージするとわかりやすいでしょう。

【ワンポイント】オブジェクト指向

後述のソース解説内には、～に対して、という記述が多数みられます。

これはJavaに限らず、オブジェクト指向とよばれるプログラミング言語特有の考え方です。

ソースでは、gというグラフィックスを生成し、利用できるようにしています。

そして、そのgに対して、私たちはいろいろな描画処理を施すことになるのです。

ちょうど、リモコンを使い、テレビに対してチャンネルを変えさせるといったように、です。

このようにして、プログラムで扱える様々なパーツに対し、いろいろな指示・命令(これをメソッドといいます)を与えるようにして記述して書かれるプログラミング言語を、オブジェクト指向型のプログラムといいます。

具体的には、g.drawString()のように、ピリオドでオブジェクトとメソッドが区切られます。

そこで、そのピリオドを「～に対して」と読むようにすると、他の人が書いたプログラムでも容易に読めるようになるでしょう。

ちなみにメソッドではかならず()がつきます。

4. プログラムを理解してみましょう

ソースを全て理解しようとするのは、現時点ではあまり意味がありません。それはまずiアプリ開発の流れを体得する方が左記だからです。

ここでは、いちおうポイントに絞って、命令の解説をしたいと思います。

わかる範囲で、適宜覚えるようにしてください。

```
-----Mk02.java-----
01 : import com.nttdocomo.ui.**1;
02 : public class *2Mk02 extends IApplication*3 {
03 :     public void start(){
04 :         Display.setCurrent*4(new Mk02d*5());
05 :     }
06 : }
07 : class *6Mk02d extends Canvas*7{
08 :     int w,h*8;
09 :     public Mk02d()*9{
10 :         w=getWidth()*10;
11 :         h=getHeight()*11;
12 :     }
13 :     public void paint*12(*13Graphics g){
14 :         *14g.lock();
15 :         g.setColor*15(g.getColorOfRGB(0,0,0)*16);
16 :         *17g.fillRect(0,0,w,h);
17 :         g.setColor(g.getColorOfRGB(255,255,255)*18);
18 :         *19g.drawString("通話中",0,h/2);
19 :         g.unlock(true)*20;
20 :     }
21 : }
```

-
- *1 com.nttdocomo.uiという命令群すべて、という意味。～uiという命令セットはいくつもあるのです。
 - *2 このiアプリ名。
 - *3 このJavaプログラムが、iアプリであることを示しています。
 - *4 携帯の画面(Display)を使えるようにしなさい、という意味。
 - *5 07行目のMk02dと同一でなければなりません。
 - *6 04行目で使えるように宣言したクラス名です。
 - *7 キャンバス型というグラフィックに特化したiアプリを作るときに記述します。
 - *8 wとhという整数型の変数を使えるようにしなさい、という意味。
 - *9 Mk02d内で利用する変数等の初期値を設定しています。
 - *10 携帯の画面の横方向の数値を自動的に検知します。
 - *11 携帯の画面のたて方向の数値を自動的に検知します。
 - *12 このまとまりでは描画(paint)しなさい、という意味。
 - *13 gというグラフィックスを生成しなさい、という意味。このgに対して様々な命令をしていきます。
 - *14 gを使えるようにしなさい(ロックしている何か反対のような気はしますが)、という意味。
 - *15 gに対して色を設定しなさい、という意味。
 - *16 赤・緑・青それぞれの値を0にした色の値を求めています。光の3原色については後述。
 - *17 (0,0)という座標値から、画面の横サイズ、縦サイズいっぱいまで塗りつぶしています。
 - *18 赤・緑・青それぞれの値がすべて255というのは白です。
 - *19 gに対して文字を表示しなさい、という意味。縦の座標値はhの2分の1としています。
 - *20 ロックした画面を解除して、描画処理を終えます。