

iアプリ Java ゲームプログラミング

第4回 メソッドの作成

メソッド

メソッドとは、クラスに備えられた機能のことをいいます。逆に言えば、クラスに機能を追加したいときはメソッドを定義する、ということです。

MyAppliクラスには、アプリケーションを開始するためのstartメソッド、MyCanvasクラスには、画面の描画を行うpaintメソッドがすでに作成されています(ただし、処理は記述されていません)。言い換えれば、MyAppliクラスには、アプリケーションを開始する機能、MyCanvasクラスには、画面の描画を行う機能があると言えます。

iアプリでゲームを作成する場合は、キャンバスクラスに機能を追加していきます。つまり、MyCanvasクラスに必要なメソッドを追加していきます。

ゲームを初期化する機能、メインループ、プレイヤーや敵キャラクターの処理、キャラクターを描画する機能などは、メソッドとしてキャンバスクラスに記述します。

課題

キャンバスクラスに、以下の機能を追加しましょう。

- ・実行.....runメソッド
- ・初期化.....initメソッド
- ・内部処理...procメソッド
- ・描画.....drawメソッド

(1) MyCanvasクラスのソースコードを以下のように変更しましょう。

- MyCanvas.java -

```
import com.nttdocomo.ui.*;

/*
 * 作成日： 2005/06/20
 *
 * TODO この生成されたファイルのテンプレートを変更するには次を参照。
 * ウィンドウ > 設定 > Java > コード・スタイル > コード・テンプレート
 */

/**
 * @author waisemen
 *
 * TODO この生成された型コメントのテンプレートを変更するには次を参照。
 * ウィンドウ > 設定 > Java > コード・スタイル > コード・テンプレート
 */
public final class MyCanvas extends Canvas {

    // --- 定数の定義(ゲームに必要な定数を以下に定義します)

    // --- 変数の定義(ゲームに必要な変数を以下に定義します)

    // --- メソッドの定義(ゲームに必要なメソッドを以下に定義します)
    // --- 実行
    void run() {
    }

    // --- 初期化
    void init() {
    }

    // --- 内部処理
    void proc() {
    }
}
```

```

// --- 描画
void draw() {

}

/* (Javadoc なし)
 * @see com.nttdocomo.ui.Frame#paint(com.nttdocomo.ui.Graphics)
 */
public void paint(Graphics arg0) {
    // TODO 自動生成したメソッド・スタブ

}
}

```

Eclipseでは、プログラムの入力中に補完候補が現れることがあります。一覧から選択して「Enter」キーを押すと、自動的に選んだ項目が入力され便利です。このような補完機能を活用することで、単純な入力ミスを防ぐことが可能です。補完候補は「Ctrl」+「Space」キーで手動表示させることもできます。

エラーが検出されたときには、画面左にエラーマーク表示されます。ここで、エラーマークをクリックすると、エラーの修正候補が表示されます。

このように、Eclipseは、非常に強力な入力支援機能を持っています。

各メソッドの役割は、以下のようになります。

- ・runメソッドは、ゲームを実行するメソッドです。このメソッドにゲームの中心となる処理を記述します。
- ・initメソッドは、ゲームを初期化するメソッドです。変数や画像など、ゲームを初期状態に設定するための処理を記述します。
- ・procメソッドは、ゲームの内部処理を行うメソッドです。キャラクターの座標や状態を変更するための処理を記述します。
- ・drawメソッドは、描画を行うメソッドです。内部処理で変更された状態を画面に表示するための処理を記述します。

(2) MyIAppクラスのソースコードを以下のように変更しましょう。

- MyIAppI.java -

```

import com.nttdocomo.ui.*;
...
コメント省略
...
public final class MyIAppI extends IApplication {

    /* (Javadoc なし)
     * @see com.nttdocomo.ui.IApplication#start()
     */
    public void start() {
        MyCanvas canvas = new MyCanvas(); // キャンバスの生成
        Display.setCurrent(canvas); // キャンバスをディスプレイに設定
        canvas.run(); // キャンバスのrunを呼び出し、ゲームを実行する
    }
}
}

```

startメソッドは、iアプリが実行されたときに最初に実行されるメソッドです。startメソッドで行うのは、キャンバスの作成、ディスプレイへの設定、キャンバス実行です。

iアプリでは、詳細な処理はキャンバスクラスに記述するのが一般的なので、アプリケーションクラスには、プログラムの起動と終了のみを記述します(今回のプログラムには、終了はありません)。

(3) プログラムが入力できたら、「Ctrl」+「s」キーを押して保存しましょう。この際、自動的にコンパイルされます。

(4) アイコン○をクリックし、実行しましょう。