

# 卒業制作レポート

「java を使った携帯電話向けアプリケーション制作」

法政大学国際文化学部国際文化学科  
4年G組 01G0623 山口祐一郎

## 目次

- 1 . I アプリについて
- 2 . 開発環境
- 3 . 作品概要
- 4 . 使用方法
- 5 . ファイル解説
- 6 . プログラム解説
- 7 . 反省点

## 1. I アプリについて

### A. i アプリとは

I アプリは、NTT ドコモの携帯電話で実行可能な java アプリケーションのことです。2001年1月に発売された503i シリーズから、世界初の java 対応携帯電話として I アプリの実行環境が実装されました。

I アプリは HTTP により Web サーバーからダウンロードする形で、I モード java 携帯電話に保存され、実行可能となります。I アプリをダウンロードする Web サーバーは、NTT ドコモの公式サイトである必要はなく、自前のサーバーやレンタルサーバーでも I アプリを公開することが出来ます。

### B. i アプリの特徴

- ・ I モード対応 java は J2ME をベースに構成されている。

Java アプリケーションが動作する環境は J2EE, J2SE, J2ME の3つあります。J2EE は PC などのクライアントで Java を使用する場合の環境。J2SE は組み込み機器向けの環境。J2ME は PDA(携帯情報端末)や携帯電話などさまざまです。

- ・ プログラムサイズに上限がある。

アプリケーションサイズは JAR(java プログラム配布用の圧縮形式)で 503i シリーズで 10K バイト以下、504i シリーズで 30KB までと制限されている。

- ・ 機種によって異なる GUI。

使用する機種によって、GUI が異なる。SO503i は 120×130 ピクセル、D503I は 132×126 ピクセルのように機種により描ける画素が違って来るので、制作時に考慮する必要がある。

- ・ HTTP によるネットワークが可能。

HTTP によりネットワーク通信が可能である。

## 2. 開発環境

### A. 制作準備

I アプリを作成するにあたり、以下の3つのソフトウェアをインストールする必要があります。

- ・ JDK

JDK はエミュレータを動作させるのにも、I アプリをビルドさせるためにも必要です。Sun Microsystems の Web サイトからダウンロードが可能。

- ・ I モード java SDK

I モード java SDK は NTT ドコモお Web サイトからダウンロード可能。SDK には I アプリを作成するためのクラスライブラリとエミュレータ、ツールが含まれています。エミュレータとは、PC 上で I アプリを開発して、実際に動かしてテストできるようになります。そのために携帯電話と同じクラスライブラリを搭載しています。また、このクラスライブラリを使ったプログラムをコンパイルしてビルドするための最小限の支援機能も SDK に組み込まれています。

- ・ テキストエディタ

I アプリのソースコードをかく NOTEPAD

## B、作成手順

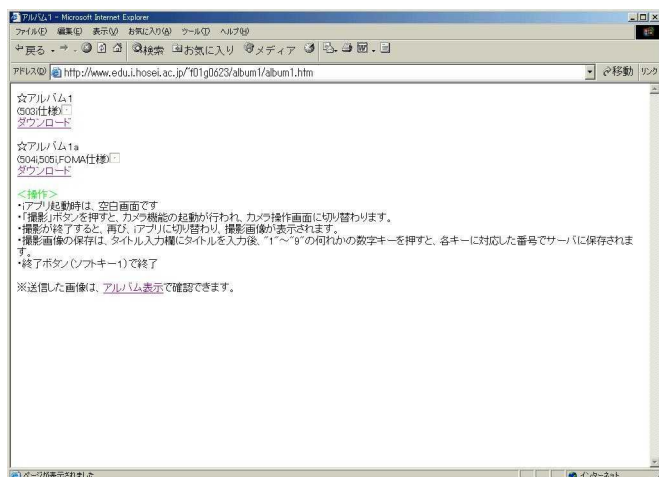
- 1、テキストエディタでソースコードの作成
- 2、コンパイラでソースコードをビルド
- 3、エミュレータでオブジェクトコードを動作確認
- 4、完成した java アプリケーションと ADF を Web サーバにアップ
- 5、実機でダウンロード & 動作確認

## 3、作品概要

携帯電話アルバム。携帯電話のカメラで撮影した写真と文章をサーバにアップロードしていきアルバムを作成していく。その画像はパソコン、携帯電話から閲覧することが出来る。

## 4、使用方法

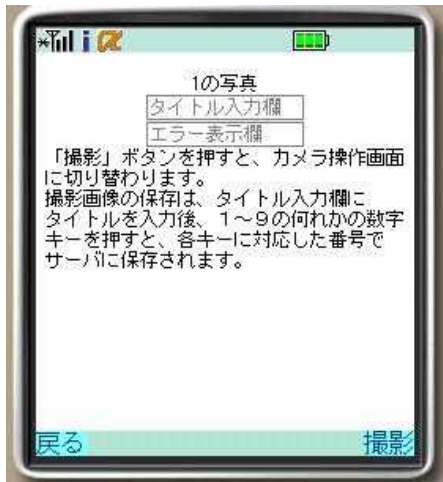
1. アプリケーションをアップロードした URL(<http://www.edu.i.hosei.c.jp/album2/album2.htm>)にいき、ソフトをダウンロードする。



2. 最初の画面で、これから撮影する画像の分類を選択して下さい。  
画像の分類 : "1", "2", "3", "4", "5", "6", "7", "8"



3. 像の分類を選択すると、「撮影」のための画面が表示されます。

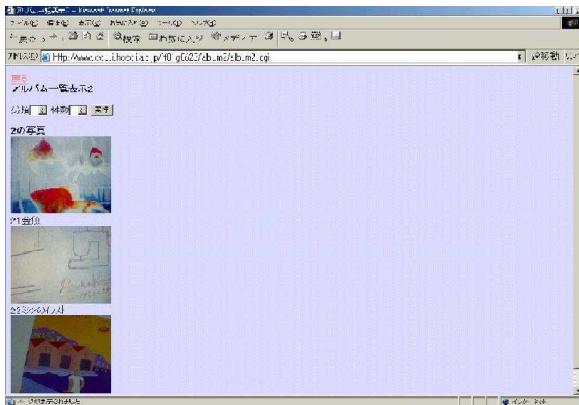


4. 「撮影」ボタンを押すと、カメラ機能の起動が行われ、カメラ操作画面に切り替わります。
5. 撮影が終了すると、再び、i アプリに切り替わり、撮影画像が表示されます。



6. 撮影画像の保存は、タイトル入力欄にタイトルを入力後、1～9の何れかの数字キーを押すと、各キーに対応した番号でサーバに保存されます。

7. アップロードしたアルバムをサーバ (<http://www.edu.i.hosei.c.jp/album2/album2.cgi>) で見る。



8 . 終了ボタン (ソフトキー 1) で終了

## 5 . ファイル解説

album2.java : Java プログラムソース

album2.jam : ADF(jam ファイル) i アプリ実行に必要な情報を記載したテキストファイル

album2.jar : JAR ファイル (クラスファイルを 1 つのアーカイブにまとめたもの)

album2t.cgi : i アプリから送信された画像タイトルデータを、指定された番号のエリアに上書き。

album2g.cgi : i アプリから送信された画像データを、指定された番号のファイル名で上書き。

album2.dat : 画像タイトルデータ

album2num.dat : 番号管理ファイル

album2.htm : i アプリダウンロード用 HTML ( i モード用)

## 6 . プログラム解説

• album2.htm

```
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<TITLE>アルバム</TITLE>
</HEAD>
<BODY>
<P align="left"> i アプリ <P> アルバム <OBJECT declare id="imode01"
data="http://www.edu.i.hosei.ac.jp/~f01g0623/album2/album2.jam"
type="application/x-jam"></OBJECT><BR>
<A ijam="#imode01" href="imode9.htm">ダウンロード</A><BR>
</P>
<P>< 操作 > <BR>
・最初の画面で、これから撮影する画像の分類を選択して下さい。 <BR>
画 像 の 分 類 :
&quot;1&quot;,&quot;2&quot;,&quot;3&quot;,&quot;4&quot;,&quot;5&quot;,&quot;6&
quot;,&quot;7&quot;,&quot;8&quot;<BR>
・画像の分類を選択すると、「撮影」のための画面が表示されます。 <BR>
・「撮影」ボタンを押すと、カメラ機能の起動が行われ、カメラ操作画面に切り替わります。
<BR>
・撮影が終了すると、再び、i アプリに切り替わり、撮影画像が表示されます。 <BR>
・撮影画像の保存は、タイトル入力欄にタイトルを入力後、&quot;1&quot; ~ &quot;9&quot;
の何れかの数字キーを押すと、各キーに対応した番号でサーバに保存されます。 <BR>
・終了ボタン（ソフトキー 1）で終了</P>
<P> 送信した画像は、ブラウザから<A href="album2.cgi">アルバム表示 2 </A>で確認で
きます。 <BR>
操作：画像の分類と件数を選択して実行ボタンを押して下さい。 <BR>
<FONT color="#ff0000">注意</FONT>：<FONT color="#ff0000">送信された画像は即イ
ンターネット上に公開されるので、公開されて困る画像は送信しないで下さい。
</FONT><BR>
</P>
</BODY>
</HTML>
```



・ album2.java

```
// パッケージのインポート

import java.io.*;
import javax.microedition.io.*;
import com.nttdocomo.io.*;
import com.nttdocomo.ui.*;
import com.nttdocomo.device.Camera;

// メイン処理

// IApplication のサブクラスとして"album2"を宣言

public class album2 extends IApplication {

// 初期化処理
static mymenu          w_menu;
static mypanel         w_panel;
static int             w_syurui;
// i アプリ起動時に、この"start"メソッドが呼ばれる
public void start() {
// オブジェクトを生成
w_menu = new mymenu();
w_panel = new mypanel();
// ソフトキー 1 のラベルに"終了"を設定
w_menu.setSoftLabel(w_menu.SOFT_KEY_1,"終了");
// ソフトキー 1 のラベルに"終了"を設定
w_panel.setSoftLabel(w_panel.SOFT_KEY_1,"戻る");
// ソフトキー 2 のラベルに"撮影"を設定
w_panel.setSoftLabel(w_panel.SOFT_KEY_2,"撮影");
// 現在の画面に設定
```

```

Display.setCurrent(w_menu);
}
// Panel のサブクラスとして"mymenu"を宣言
class mymenu extends Panel implements ComponentListener,SoftKeyListener {
// 変数の宣言
private ImageLabel          w_imagelabel;
private Ticker              w_ticker;
private Button[]           w_button;
private TextBox             w_title,w_textbox;
private Label               w_label,w_label2;
private String[]           w_bunrui = {"1","2","3","4","5","6","7","8"};
HTMLLayout                 w_layout1;

public mymenu(){
//
w_layout1 = new HTMLLayout();
setLayoutManager(w_layout1);
w_layout1.begin(HTMLLayout.CENTER);

// ティッカー作成
w_ticker = new Ticker("写真を投稿してください。");
w_ticker.setBackground(Graphics.getColorOfName(Graphics.WHITE));
w_ticker.setForeground(Graphics.getColorOfName(Graphics.BLUE));
// パネルに追加
add(w_ticker);

// ラベル定義
w_label = new Label("写真の分類を下記の中から選択して下さい");
w_label.setForeground(Graphics.getColorOfName(Graphics.BLACK));
// パネルに追加
add(w_label);
w_layout1.br();

// 操作ボタン定義

```

```

w_button = (new Button[]{
new Button(" 1 "),
new Button(" 2 "),
new Button(" 3 "),
new Button(" 4 "),
new Button(" 5 "),
new Button(" 6 "),
new Button(" 7 "),
new Button(" 8 ")
});
for(int i = 0; i < 8; i++){
w_button[i].setBackground(Graphics.getColorOfName(Graphics.AQUA));
add(w_button[i]);
}
w_layout1.br();
// ラベル定義
w_label2 = new Label("それぞれ 9 枚まで保存可能");
w_label2.setForeground(Graphics.getColorOfName(Graphics.BLACK));
add(w_label2);
w_layout1.br();

// テキストボックス (エラー表示)
w_textbox = new TextBox("エラー表示欄", 16, 1, TextBox.DISPLAY_ANY);
// パネルに追加
add(w_textbox);
// テキストボックス (エラー表示) を、入力不可に設定
w_textbox.setEnabled(false);

w_layout1.end();

// イベントリスナーのインスタンス登録
// コンポーネントリスナー
setComponentListener(this);
// ソフトキーリスナー
setSoftKeyListener(this);
}

```

```

// コンポーネントイベント発生時に実行
public void componentAction(Component component, int type, int param){
// "グラフ表示"ボタンが押された時に実行
for(int i = 0; i < 8; i++){
if((component == w_button[i] && (type == BUTTON_PRESSED))){
//
w_syurui = i;
}
}
// カメラ撮影に遷移
w_panel.w_label.setText(w_bunrui[w_syurui]+"の写真");
Display.setCurrent(w_panel);
}

// ソフトキーが離されたときに実行
public void softKeyReleased(int param){
// ソフトキー 1 が離された時に実行
if(param == this.SOFT_KEY_1){
// i アプリ終了
terminate();
}
// ソフトキー 2 が離された時に実行
if(param == this.SOFT_KEY_2){
}
}
// ソフトキーが押されたときに実行

public void softKeyPressed(int param){
}
}

// Panel のサブクラスとして"mypanel"を宣言
class mypanel extends Panel implements SoftKeyListener,KeyListener {
// 変数の宣言
private Camera w_camera;

```

```

private ImageLabel          w_imagelabel;
private Label              w_label;
private Label[]           w_help;
private TextBox            w_title,w_textbox;
HTMLLayout                w_layout;

public mypanel(){
//
w_layout = new HTMLLayout();
setLayoutManager(w_layout);
w_layout.begin(HTMLLayout.CENTER);

// カメラオブジェクトの取得
w_camera = Camera.getCamera(0);
// イメージオブジェクトをもつイメージラベルを作成
w_imagelabel = new ImageLabel();
// パネルに追加
add(w_imagelabel);
w_layout.br();

// ラベル (分類を表示)
w_label = new Label("空");
// パネルに追加
add(w_label);
w_layout.br();
// テキストボックス (画像タイトル)
w_title = new TextBox("タイトル入力欄", 16, 1, TextBox.DISPLAY_ANY);
// パネルに追加
add(w_title);
w_layout.br();
// テキストボックス (画像タイトル) を、入力不可に設定
w_title.setEnabled(false);

// テキストボックス (エラー表示)
w_textbox = new TextBox("エラー表示欄", 16, 1, TextBox.DISPLAY_ANY);
// パネルに追加

```

```

add(w_textbox);
// テキストボックス (エラー表示) を、入力不可に設定
w_textbox.setEnabled(false);

// ラベル (操作説明)
w_help = (new Label[]{
new Label("「撮影」ボタンを押すと、カメラ操作画面"),
new Label("に切り替わります。"),
new Label("撮影画像の保存は、タイトル入力欄に"),
new Label("タイトルを入力後、1～9の何れかの数字"),
new Label("キーを押すと、各キーに対応した番号で"),
new Label("サーバに保存されます。")
});
for(int i = 0; i < 6; i++){
add(w_help[i]);
w_layout.br();
}

w_layout.end();

// イベントリスナーのインスタンス登録
// ソフトキーリスナー
setSoftKeyListener(this);
// キーリスナー
setKeyListener(this);
}

// ソフトキーが離されたときに実行
public void softKeyReleased(int param){
// ソフトキー 1 が離された時に実行
if(param == this.SOFT_KEY_1){
// メニューに遷移
Display.setCurrent(w_menu);
}
// ソフトキー 2 が離された時に実行
if(param == this.SOFT_KEY_2){

```

```

try {
// カメラ機能の起動
w_camera.takePicture();
} catch (Exception ce) {}
// カメラオブジェクトに保持されている画像を、メディアイメージの形式で取得
MediaImage w_media = w_camera.getImage(0);
try {
w_media.use();
} catch (Exception ce) {}
// メディアイメージからイメージオブジェクトを取得
Image w_image = w_media.getImage();
// ラベル画像を設定・変更
w_imagelabel.setImage(w_image);
// 結果表示
w_textbox.setText("撮影完了");
// テキストボックス (画像タイトル) を、入力可に設定
w_title.setEnabled(true);
// テキストボックス (画像タイトル) に、フォーカスを設定
w_title.requestFocus();
}
}
// ソフトキーが押されたときに実行

public void softKeyPressed(int param){
}

// キーが離されたときに実行
public void keyReleased(Panel panel,int param){
if (param == Display.KEY_1) {
// タイトルと画像を 1 番目のエリアに出力
titleOut(1);
}
if (param == Display.KEY_2) {
// タイトルを 2 番目のエリアに出力
titleOut(2);
}
}

```

```
if (param == Display.KEY_3) {
// タイトルを 3 番目のエリアに出力
titleOut(3);
}
if (param == Display.KEY_4) {
// タイトルを 4 番目のエリアに出力
titleOut(4);
}
if (param == Display.KEY_5) {
// タイトルを 5 番目のエリアに出力
titleOut(5);
}
if (param == Display.KEY_6) {
// タイトルを 6 番目のエリアに出力
titleOut(6);
}
if (param == Display.KEY_7) {
// タイトルを 7 番目のエリアに出力
titleOut(7);
}
if (param == Display.KEY_8) {
// タイトルを 8 番目のエリアに出力
titleOut(8);
}
if (param == Display.KEY_9) {
// タイトルを 9 番目のエリアに出力
titleOut(9);
}
// 画像を出力
imageOut();
}
// キーが押されたときに実行    今回は無し
public void keyPressed(Panel panel,int param){
}
```



```

// 画像タイトルデータ書込処理
// サーバ上の"album2.dat"ファイルにデータを書込

void titleOut(int w_num) {
    HttpURLConnection          wo_connect;
    // テキストフィールドから"タイトル"を受取
    // CGI に渡すためのデータ (w_num : タイトル) を編集
    w_num = (w_syurui + 1) * 10 + w_num;
    String w_data = w_num + ":" + w_title.getText();
    try {
        // 接続を設定
        String wt_url = new String(getSourceURL()+"album2t.cgi");
        wo_connect = (HttpURLConnection)Connector.open(wt_url,Connector.READ_WRITE,true);
        // 要求メソッドとコンテンツタイプを設定
        wo_connect.setRequestMethod(HttpURLConnection.POST);
        wo_connect.setRequestProperty("Content-Type", "text/plain");
        // 出力ストリームを取得
        OutputStream wo_stream = wo_connect.getOutputStream();
        // OutputStreamWriter インスタンスを作成
        //   Unicode を SJIS に変換
        OutputStreamWriter wo_buffer = new OutputStreamWriter(wo_stream);
        // w_data (w_num : タイトル) を書込
        wo_buffer.write(w_data);
        // 出力ストリームをクローズ
        wo_buffer.close();
        wo_stream.close();

        // 実際に接続
        wo_connect.connect();
        // 入力ストリームを取得
        InputStream wi_stream = wo_connect.getInputStream();
        // InputStreamReader インスタンスを作成
        //   SJIS を Unicode に変換
        InputStreamReader wi_buffer = new InputStreamReader(wi_stream);
        // StringBuffer インスタンスを作成

```

```

StringBuffer w_StringBuffer = new StringBuffer();
// CGI から送信されたストリームを読取
String                w_remsg;
int                   w_char;
while ((w_char = wi_buffer.read()) != -1) {
w_StringBuffer.append((char)w_char);
}
w_remsg = (String)w_StringBuffer.toString();
// メッセージ表示領域に、CGI からの返答メッセージを設定
w_textbox.setText(w_remsg);
// 入力ストリームをクローズ
wi_stream.close();
// 接続をクローズ
wo_connect.close();
}
// 例外処理
catch(Throwable e) {w_textbox.setText("タイトル書込エラー");}
}

// 画像を送信

public void imageOut() {

// 変数の宣言
HttpConnection        w_connect;
InputStream            w_in;
OutputStream          wo_stream;
byte[]                w_data;
int                   w_length;

try {
// 入力ストリームを取得
w_in=w_camera.getInputStream(0);

// 出力先への接続を設定

```

```

String w_url = new String(getSourceURL()+"album2g.cgi");
w_connect = (HttpConnection)Connector.open(w_url,Connector.READ_WRITE,true);
// 要求メソッドとコンテンツタイプを設定
w_connect.setRequestMethod(HttpConnection.POST);
w_connect.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
// 出力ストリームを取得
wo_stream = w_connect.openOutputStream();

// 入力ストリームから出力ストリームへ、順次、書き込む
w_data=new byte[128];
while ((w_length=w_in.read(w_data)) != -1 )      {
wo_stream.write(w_data,0,w_length);
}
// 入力ストリームをクローズ
w_in.close();
// 出力ストリームをクローズ
wo_stream.close();
// 実際に接続
w_connect.connect();

// 入力ストリームを取得
InputStream wi_stream = w_connect.openInputStream();
// InputStreamReader インスタンスを作成
// SJIS を Unicode に変換
InputStreamReader wi_buffer = new InputStreamReader(wi_stream);
// StringBuffer インスタンスを作成
StringBuffer w_StringBuffer = new StringBuffer();
// CGI から送信されたストリームを読取
String          w_remsg;
int             w_char;
while ((w_char = wi_buffer.read()) != -1) {
w_StringBuffer.append((char)w_char);
}
w_remsg = (String)w_StringBuffer.toString();
// メッセージ表示領域に、CGI からの返答メッセージを設定
w_textbox.setText(w_remsg);

```

```
// 入力ストリームをクローズ
wi_stream.close();
// 接続をクローズ
w_connect.close();
}
catch(Exception e) {w_textbox.setText("画像書込エラー");}
}
}
```

・ album2.cgi

// [ 1 ] 初期処理

// 初期値設定

// この CGI のファイル名

\$cgi\_file = './album2.cgi';

// この CGI のタイトル名

\$cgi\_title = 'アルバム一覧表示 2';

// ログを格納するファイル

\$title\_file = './album2.dat';

// トップの URL

\$top\_url = 'index.html';

// トップのタイトル

\$top\_title = '戻る';

// ゲストブックの背景：水色

\$w\_bgcolor = '#ddddff';

// ゲストブックのテキスト：黒

\$w\_text='#000000';

// ゲストブックのテキスト：青

\$w\_text3='#0000ff';

// ゲストブックのリンク：薄赤

\$w\_link='#ff7f7f';

// ゲストブックのリンク済：薄青

\$w\_vlink='#7f7fff';

// ゲストブックの投稿表題欄の背景：薄黄

\$w\_bgcolor2 = '#ffffb0';

// ゲストブックの投稿表題欄テキスト：緑

\$w\_text2='#00ff00';

// ゲストブックの投稿内容欄の背景：白

\$w\_bgcolor3 = '#ffffff';

```

// META 文
                                $meta_html = '<meta http-equiv="Content-Type"
content="text/html; charset=x-euc-jp">';
// BODY 文
                                $body_html   =   "<body   bgcolor=$w_bgcolor
text=$w_text link=$w_link vlink=$w_vlink>";

// これから HTML 文を出力する宣言

print "Content-type: text/html¥n¥n";

// 今日の日付を取得
// 今日の日付を、「9999 年 99 月 99 日 ( 曜日 )」の形式で、$w_today に設定

($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$flag)=(localtime(time));
if($year < 2000) {
                                $year += 1900;
}
$mon++;
@WEEK=("日","月","火","水","木","金","土");
$wday=$WEEK[$wday];
$w_today = $year ."年" .$mon ."月" .$mday ."日" ." (" . $wday . ") ";
$w_date = "$year/$mon/$mday $wday $hour:$min";

// 投稿者の ip アドレスを取得
//   この CGI にアクセスしたユーザ側のリモート IP アドレスを、環境変数
'REMOTE_ADDR'から、$w_ip に設定

$w_ip=ENV{'REMOTE_ADDR'};

// [ 2 ] メイン処理

```

```

// 受取データを、$buffer に取込
// 環境変数'REQUEST_METHOD' (フォームの分類) を判定して、
// フォーム(POST)の場合：標準入力(STDIN)から、$buffer にデータ取込

if ($ENV{'REQUEST_METHOD'} eq "POST"){
    read(STDIN,                                     $buffer,
    $ENV{'CONTENT_LENGTH'});
} else {
    $buffer = $ENV{'QUERY_STRING'};
}

// 受取データをデコード
// $buffer のデータを、'&'で切り分けて、配列@form-data に入れる。

@form_data = split(/&/,$buffer);
foreach (@form_data) {
    ($param, $value) = split(/=/, $_);
    $value =~ tr/+//;
    $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",
hex($1))/eg;
    $value =~ s/</&lt;/g;
    $value =~ s/>/&gt;/g;
    $value =~ s/¥r/g;
    $FORM{$param} = $value;
}

// 画像タイトルデータ取込

&album2_in;

// 入力フォームからのメッセージ取込 & ログファイルに追加

if($FORM{'action'} eq "regist"){

```





プログラム)により

```
# コードを EUC に変換する
if($FORM{'bunrui'}){
    $w_bunrui = $FORM{'bunrui'};
    $w_bunrui =~ s/¥n//;
}
if($FORM{'kensu'}){
    $w_kensu = $FORM{'kensu'};
    $w_kensu =~ s/¥n//;
}
}

// [ 2 . 3 ] 画像タイトルデータ表示

sub album2_edit {

    # HTML 文の最初の部分を表示
    print <<"EOL";
    <html>
    <head>
    <title>$cgi_title</title>
    </head>
    $body_html

    <a href="$Stop_url"
target="_top">Stop_title</a><br>

    <b>$cgi_title</b>

EOL

    # [ 2 . 3 . 1 ] 入力フォーム編集
    &form_out;

    # 入力フォームからのデータの場合、[ 2 . 3 . 2 ]
    明細編集を行う
```

```
        if ($FORM{'action'} eq "regist"){
            &album2_edit_meisai;
        }
    }
```

# [ 2 . 3 . 1 ] 入力フォーム編集

```
sub form_out {

    print <<"EOL";
    <FORM ACTION="$cgi_file" METHOD="post">
    <INPUT TYPE="hidden" NAME="action" VALUE="regist">
    分類<SELECT NAME="bunrui">
    <OPTION VALUE=0>
    <OPTION VALUE=1>1
    <OPTION VALUE=2>2
    <OPTION VALUE=3>3
    <OPTION VALUE=4>4
    <OPTION VALUE=5>5
    <OPTION VALUE=6>6
    <OPTION VALUE=7>7
    <OPTION VALUE=8>8
    </SELECT>
    件数<SELECT NAME="kensu">
    <OPTION VALUE=0>
    <OPTION VALUE=1>1
    <OPTION VALUE=3>3
    <OPTION VALUE=5>5
    <OPTION VALUE=9>9
    </SELECT>
    <INPUT TYPE="submit" VALUE="実行">
    </FORM>

    EOL
```

```

}

// [ 2 . 3 . 2 ] アルバム編集

sub album2_edit_meisai {
#
@bunrui=("1","2","3","4","5","6","7","8");

print <<"EOL";
<b>@bunrui[$w_bunrui - 1]の写真</b><br>
EOL

# アルバムを編集
for ($i=1;$i<=$w_kensu;$i++){

    $k = $w_bunrui * 10 + $i;
    $w_title = @w_title_rec[$k];
    chop($w_title);
    ($i_seq,$i_title)=split(/:/,$w_title);

# 1 件分のアルバムを編集
print <<"EOL";
    
    <br>
    <font
color=$w_text>$w_title</font>

    <br>

EOL
}
}

```

・ album2g.cgi

//データの受信

```
binmode(STDIN);
```

```
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
```

// サーバ上の番号管理ファイルを読み込む

```
$num_file = 'album2num.dat';
```

```
open(DATA, "$num_file") || die print "Can't read ¥n";
```

```
    # 番号管理ファイルを読み込む
```

```
    @numdata = <DATA>;
```

```
close(DATA);
```

// 番号管理ファイルの、SEQ を取り出す

```
$seq = $numdata[0];
```

```
chop($seq);
```

// 画像の出力

```
$file = "album" . $seq . ".jpg";
```

```
print "Content-type: image/jpg¥n¥n";
```

```
open(IMG,">$file");
```

```
binmode(IMG);
```

```
print IMG $buffer;
```

```
close(IMG);
```

// 書き込み完了メッセージ出力

```
print "画像$seq 書込完了";
```

・ album2t.cgi

```
//これから HTML 文を出力する宣言

print "Content-type: text/html¥n¥n";

// 画像タイトルデータのファイル名

$data_file = 'album2.dat';

// 番号管理ファイルのファイル名

$num_file = 'album2num.dat';

// 標準入力により、i アプリからの画像タイトルデータを受け取る

read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});

// 出力するエリアの番号を抽出

// SEQ を取り出す
($seq,$title) = split(/:/, $buffer);

// SEQ を番号管理ファイルに上書き

open(DATA, ">$num_file") || die print "Can't write ¥n";
                print DATA "$seq¥n";

close(DATA);
```

```
// サーバ上の画像タイトルデータを読み込む

open(DATA, "$data_file") || die print "Can't read ¥n";
    # 全ての画像タイトルデータを一度に読む
    @titledata = <DATA>;

close(DATA);

// SEQ の位置に、画像タイトルデータを書き出す
$i = 0;
foreach $line(@titledata) {
    ($data1,$data2) = split(/:/, $line);
    if ($data1 eq $seq) {
        $titledata[$i] = "$buffer¥n";
    }
    $i = $i + 1;
}

// 全ての画像タイトルデータをディスクに上書き

open(DATA, ">$data_file") || die print "Can't write ¥n";
    foreach $line(@titledata) {
        print DATA $line;
    }

close(DATA);

// 書き込み完了メッセージ出力
print "タイトル$seq 書込完了";

exit(0);
```

## 7. 反省

制作において、一番反省すべき点は時間配分です。はじめに作るものを決め、制作をし、動くようになるまでにほとんどの時間を費やしてしまった。携帯電話で動くようになったのは今年に入ってからだった。

制作において大事なことは、実際に動いているものを使用し、改善していく作業だと思いました。使用すると、その作品の問題点が見えてくるので、それを一つ一つ直していくことでたくさんの方が使える作品になっていくのだと思います。

その作業をする時間が多くとれずに、大きく3つの反省点があります。

1つは、デザインです。ユーザーの使いやすい、わかりやすいデザインをもう少し追及したかったです。アルバムの表示を見やすいように工夫が出来ればよかったと思います。

次はサーバの問題です。画像を保存するサーバを個人で設定することが出来る機能をつけようと思ったのですが、結局できずに不特定多数の人に配布をすることが出来ませんでした。

また、セキュリティーにも気を配って制作をしたかったです。複数で一つサーバを共有する際、許可していない人の使用を禁ずるための個人IDの設定など多くの方が使いやすい作品にしたかったです。

多くの反省点がありますが、一応形になり、使用できるようになったことは良かったあとだと思います。プログラムの大変さ、厳しさを身を持って体験し、我々が使用している様々なソフトの素晴らしさ、優秀さを感じる事が出来き、この作品制作が自分のためになったと思いました。