【B5】PHPテクニカルセッション

EMBARCADERO TECHNOLOGIES®

DEVELOPER CAMP

Inside VCL for PHP / Delphi for PHPでPHPビジュアル開発

アナハイムテクノロジー株式会社・代表取締役
はやし　つとむ

アジェンダ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- Delphi for PHP ってなんだ？
- Inside VCL for PHP / VCL4PHPをいじり倒す
- Component指向でGo!

Delphi for PHP ってなんだ？

Delphi for PHPのファーストインプレッション

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- Delphi for PHPが最初に発表され時に

えっ、Delphiで書くとPHPにコンパイルされるの？

そいつはすげ～～～

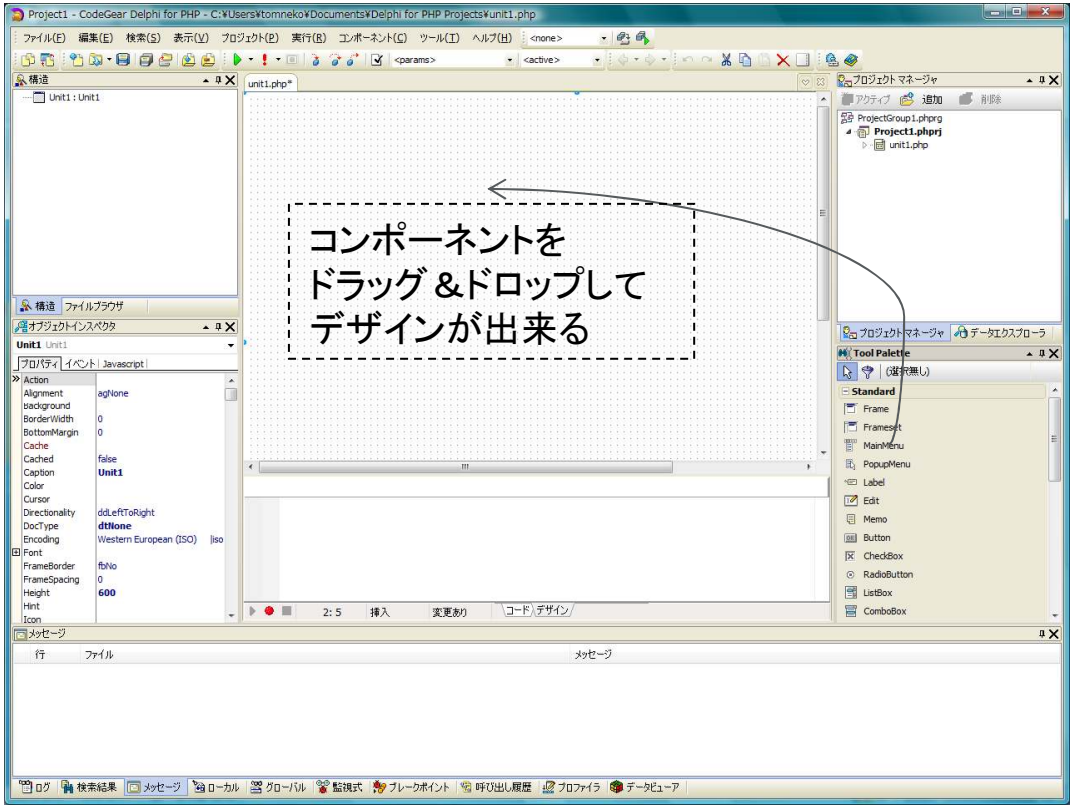
さすがにそうじゃなかったけど

- Delphi IDEでビジュアル開発
- Delphi同様のVCL for PHP

Delphi for PHPってなんだ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

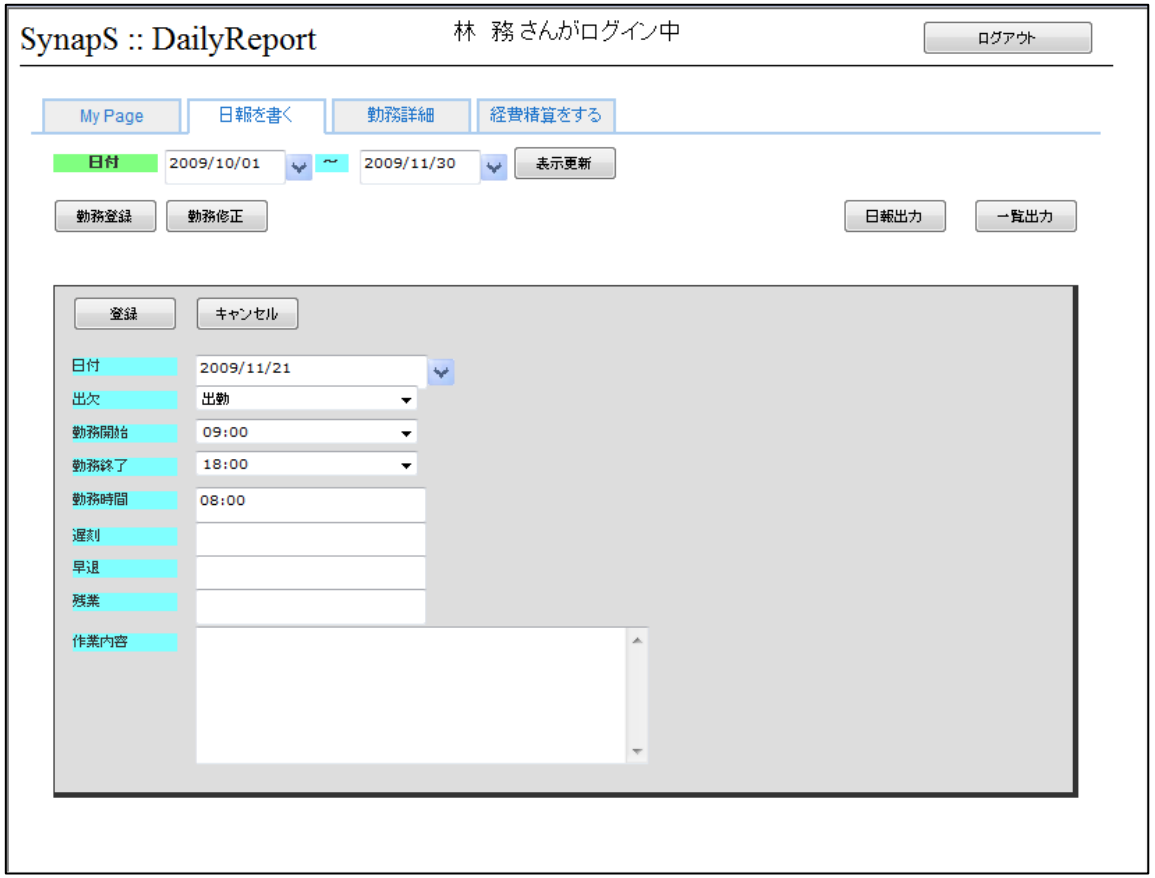
- PHPでもRubyでもPythonでも、LL言語と言われているものにはビジュアル開発の出来る開発環境がない。
- Delphi for PHPが唯一のソリューション。



業務系アプリのWebポーティングに最適

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- これまでC/Sで利用してきた業務系アプリをWebにポーティングしなくてはいけない場合などに最適。

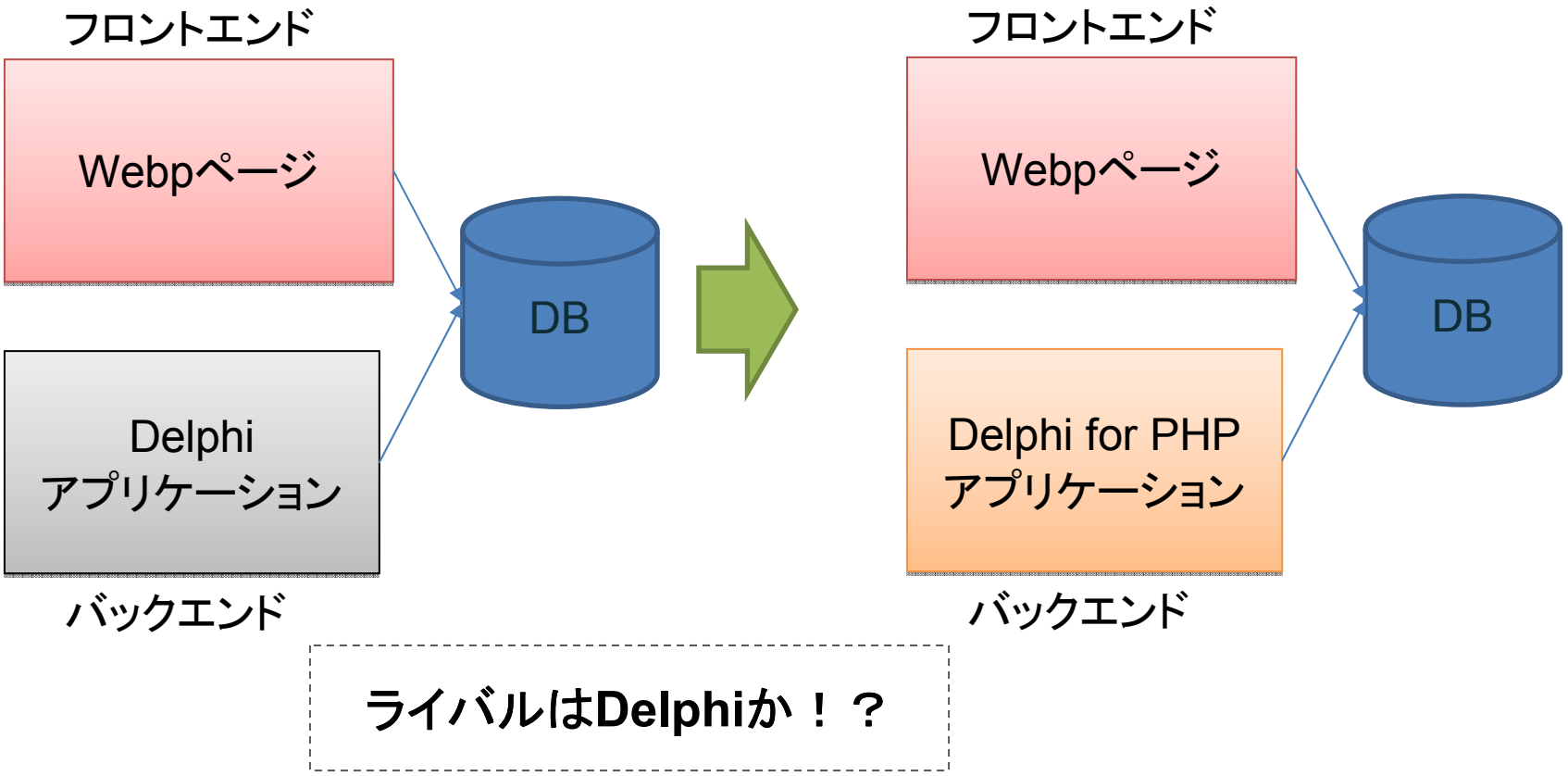


アナハイムで開発中の労務管理＋プロジェクト管理システムの画面

WebシステムのバックエンドをRAD開発

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

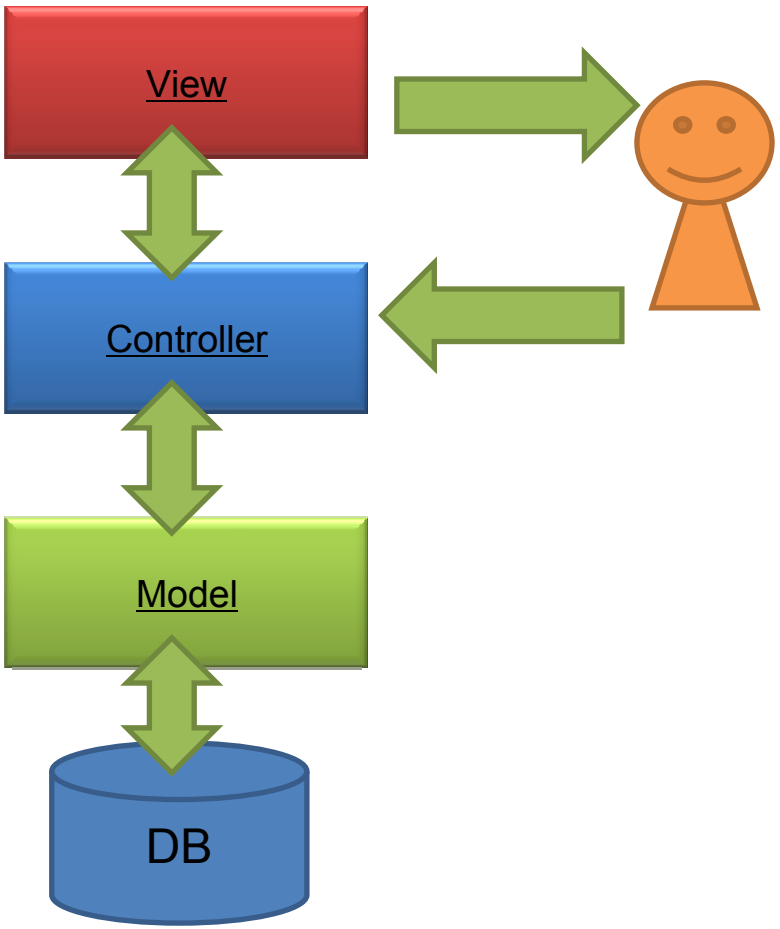
- Webのバックエンドで使われている業務系システムなどを作るのにも向いている。

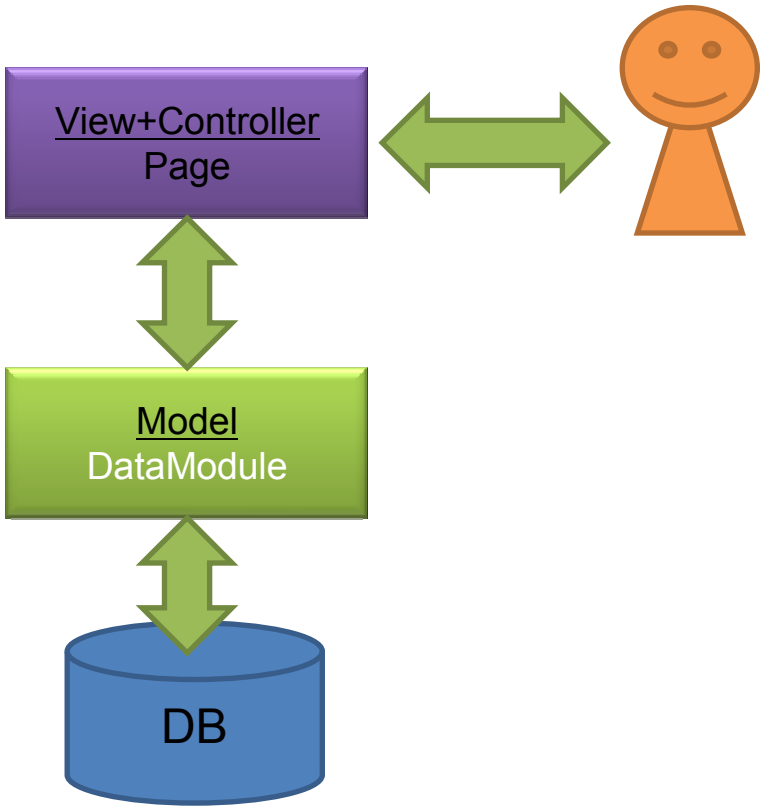


Delphi for PHPでのMVC

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- MVC(Model View Controler)でDelphi for PHPを考えると。





Ajaxの統合

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- Delphi for PHPは、サーバーサイドAjaxライブラリのxajaxをVCL for PHPに統合しています。そのため、PHPコードとJavaScriptコードの分散が防げます。

- Xajaxのバージョンは、0.2.5 (stable release)ですが、このバージョンでは非同期呼び出しのみサポートしています。同期呼び出しは最新版の0.5でサポートされているので、対応したいところです。

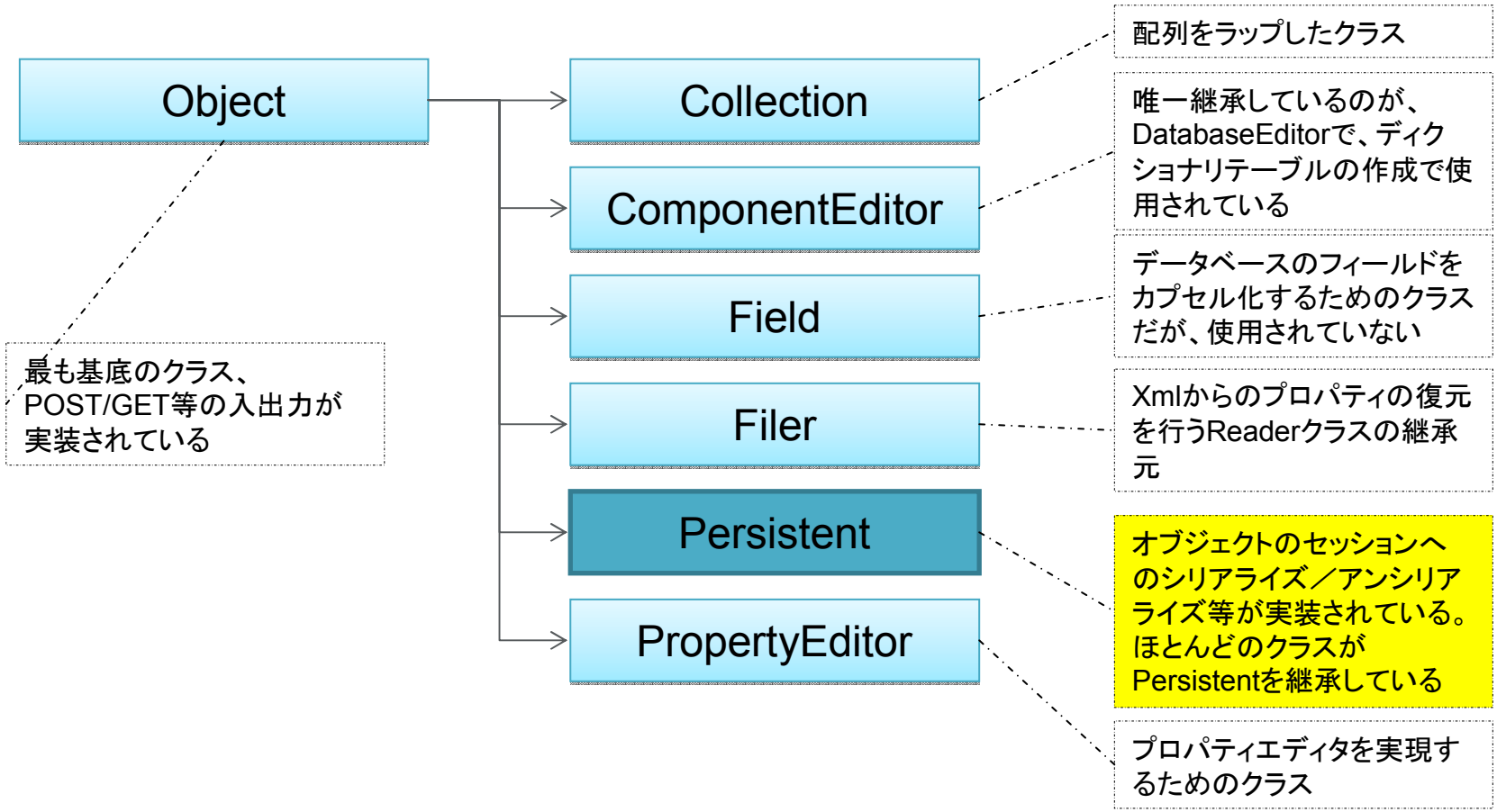
- http://www.xajaxproject.org/en/home/

Inside VCL for PHP / VCL4PHPをいじり倒す

VCL for PHPのクラス階層

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

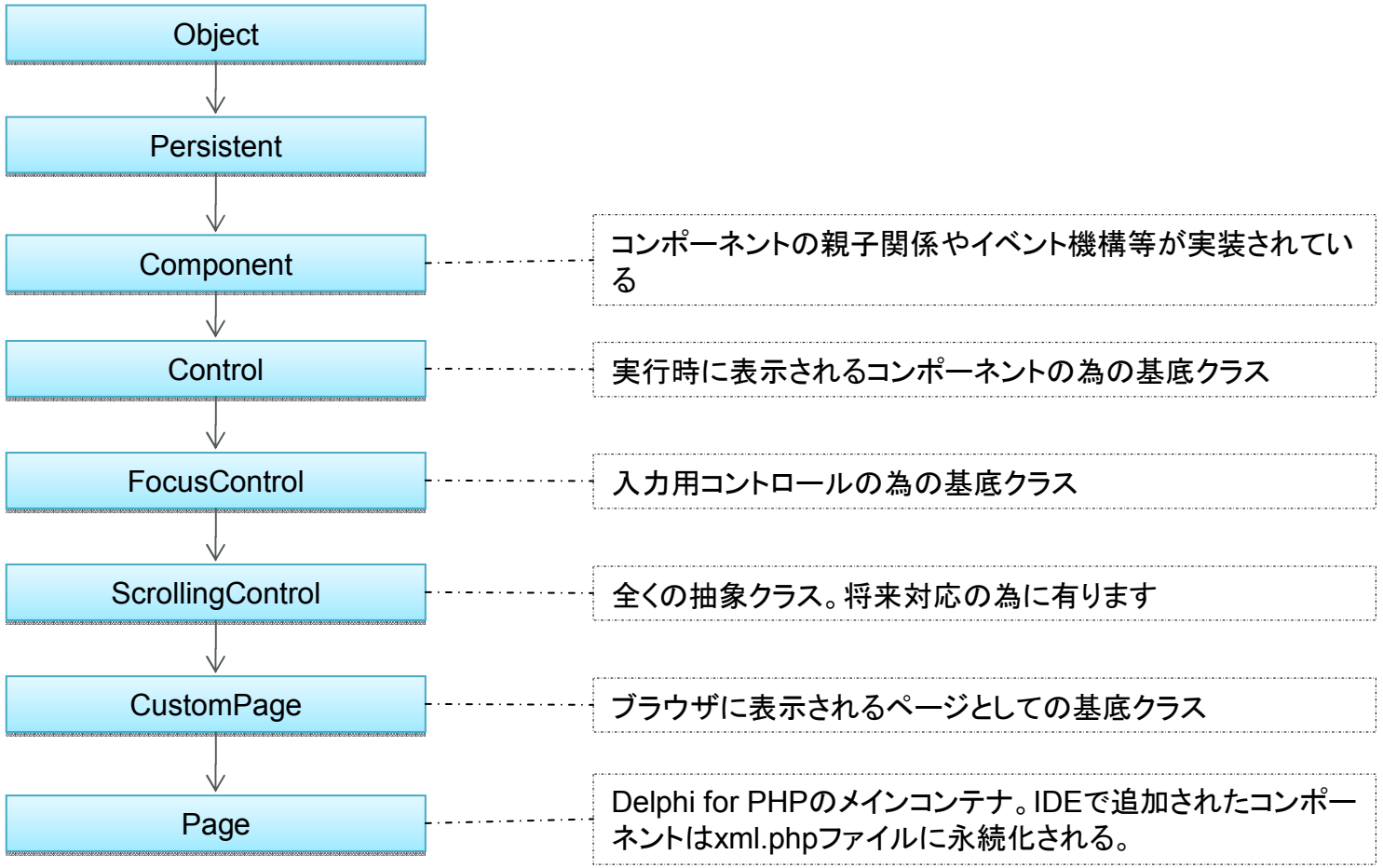
- Delphi for PHPを極めるには、VCL for PHPを極めるべし



PageからObjectまでの継承関係の図

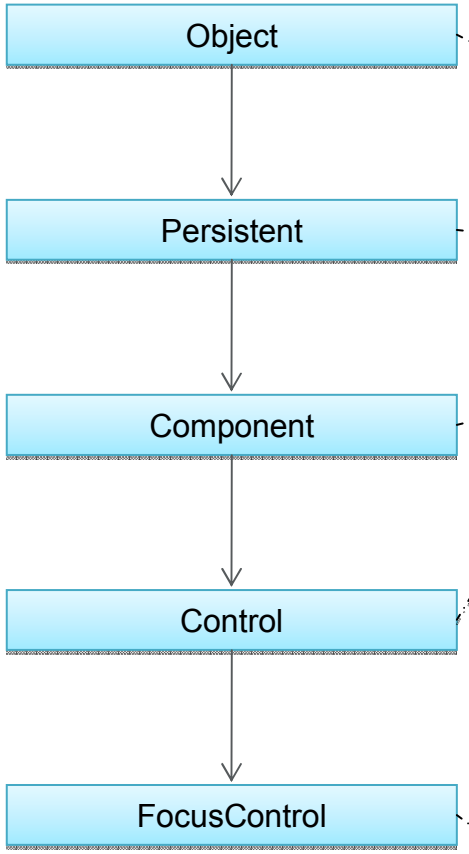
EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- フォームとして表示されるPageコンポーネントまでの継承関係



FocusControlからObjectまでのコンストラクタ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP



system.in.php(98)

```
function __construct()
    {
        global $input;

        //Assign the global input object so it can be used from inside
        $this->input=$input;
    }
```

無し

controls.inc.php(796)

```
function __construct($aowner=null)
    {
        $this->_font=new Font();
        $this->_font->_control=$this;

        //Calls inherited constructor
        parent::__construct($aowner);
    }
```

controls.inc.php(2283)

```
function __construct($aowner=null)
    {
        //Creates the controls list.
        $this->controls=new Collection();

        //Calls inherited constructor.
        parent::__construct($aowner);

        $this->_layout=new Layout();
        $this->_layout->_control=$this;
    }
```

classes.inc.php(1195)

```
function __construct($aowner=null)
    {
        //Calls the inherited constructor
        parent::__construct($aowner);

        //List of children
        $this->components=new Collection();

        //Initializes the owner
        $this->owner=null;

        //Initializes the name
        $this->_name="";

        $this->_controlstate=0;

        if ($aowner!=null)
        {
            //If there is an owner
            if (is_object($aowner))
            {
                //Stores it
                $this->owner=$aowner;

                //Adds itself to the list of components from the
owner
                $this->owner->insertComponent($this);
            }
            else
            {
                throw new Exception("Owner must be an
object");
            }
        }

    }
```

ページが表示される流れ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- ページが表示される際には、オブジェクトの階層を辿ってコンストラクタが呼ばれた後で、loadResource(__FILE__)でリソースの読み込み、PageのShow()メソッド呼び出しが行われます。

```
<?php
require_once("vcl/vcl.inc.php");
//Includes
use_unit("forms.inc.php");
use_unit("extctrls.inc.php");
use_unit("stdctrls.inc.php");

//Class definition
class Unit1 extends Page
{
	public $Button1 = null;
	public $Label1 = null;
	function Button1Click($sender, $params)
	{



	}

}
```

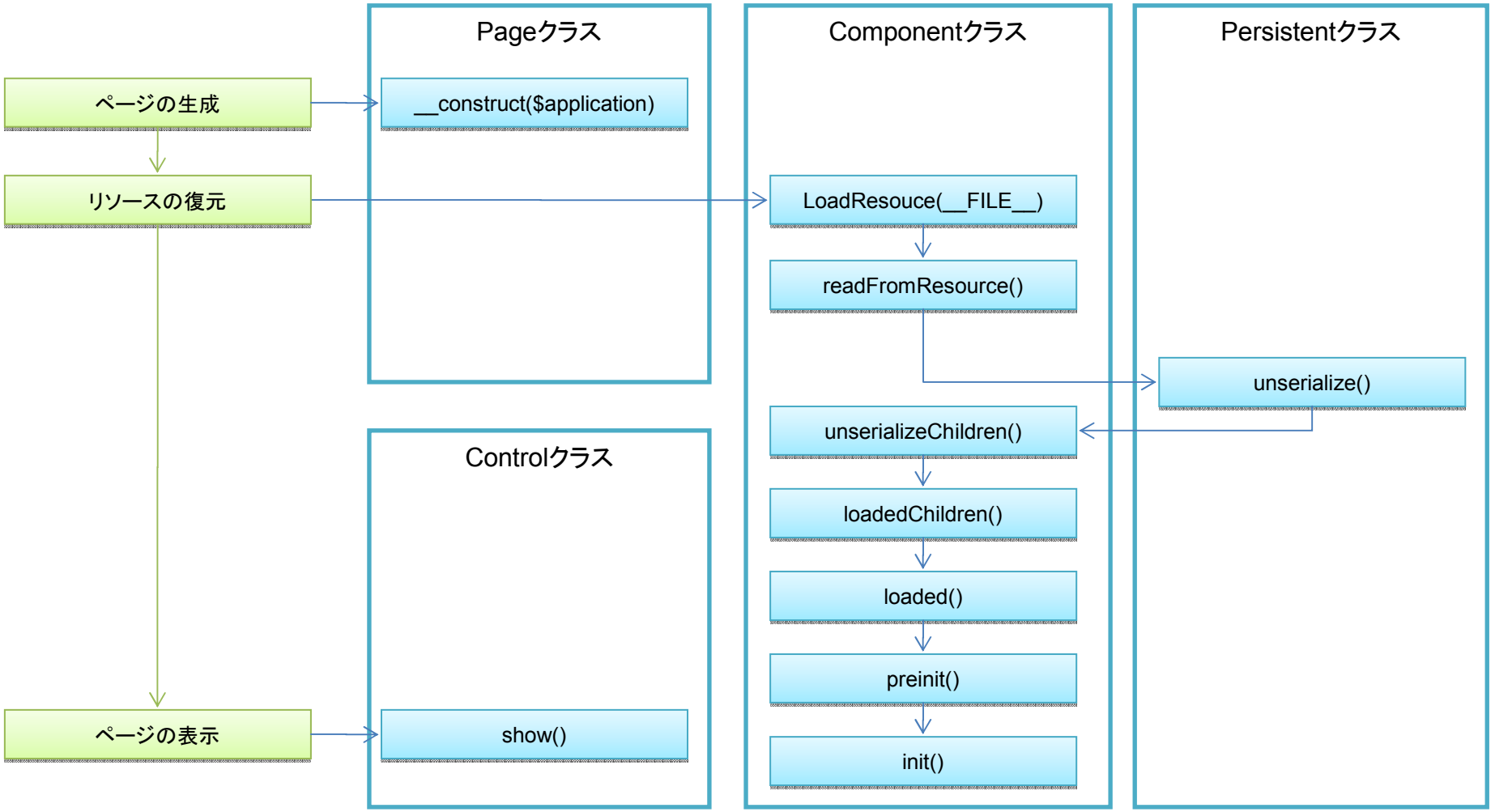
（右へ続く）

```
global $application;

global $Unit1;

//Creates the form
$Unit1=new Unit1($application);

//Read from resource file
$Unit1->loadResource(__FILE__);

//Shows the form
$Unit1->show();

?>
```

※フォームにボタンを1個とラベルを1個置いた状態のスケルトン

ページが表示される流れ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

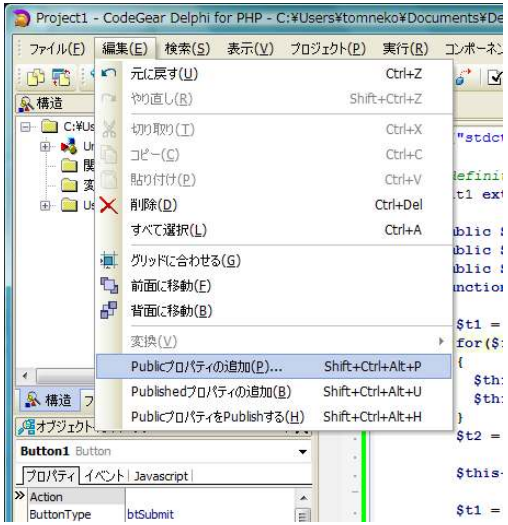


※実際にはPageからObjectまでの間の全てのコンストラクタが呼ばれています

PublishedプロパティとPublicプロパティ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- Delphi for PHPの「編集」メニューから、二つのプロパティが追加出来る。
  - Publicプロパティの追加
  - Publishedプロパティの追加
  - PublicプロパティをPublishする



<Publicプロパティのスケルトン>

```
protected $_test=null;

function readtest() { return $this->_test; }
function writetest($value) { $this->_test=$value; }
function defaulttest() { return null; }
```

<Publishedプロパティのスケルトン>

```
protected $_test=null;

function gettest() { return $this->_test; }
function settest($value) { $this->_test=$value; }
function defaulttest() { return null; }
```

Defaultメソッドは、シリアライズで利用される

※このスケルトンの中には、プロパティ名が有りません！

プロパティの実現

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- VCL for PHP では、DelphiのVCLと同様に各クラスのプロパティに対して、set/get出来るようになっています。
- これは、PHPの特殊メソッド __get() と __set() を利用することで実現しています。このメカニズムはObjectクラスで実装されています。
- get + hogehoge / set + hogehoge がPublishedプロパティで、これはシリアライズ＋アンシリアライズされる。
- read + hogehoge / write + hogehogte はPublicプロパティで、これはシリアライズされない。

Publishedプロパティのシリアライズ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- PersistentクラスのSerialize()で実装されている。
  - PHP5で実装されたリフレクションクラスの機能を使ってset付のメソッドの存在をチェックしています。
  - プロパティがコンポーネントの場合は、最後に SerializeChildren メソッドで一括してシリアライズするので、ここではオーナー名+コンポーネント名を値として保存するように変換しています。
  - デフォルト値と現在の値を比較して同じであればシリアライズをスキップします。セッションへの負荷を減らす工夫。
  - 最後に、子のコンポーネントに対して全てシリアライズする指令を発して終わります。

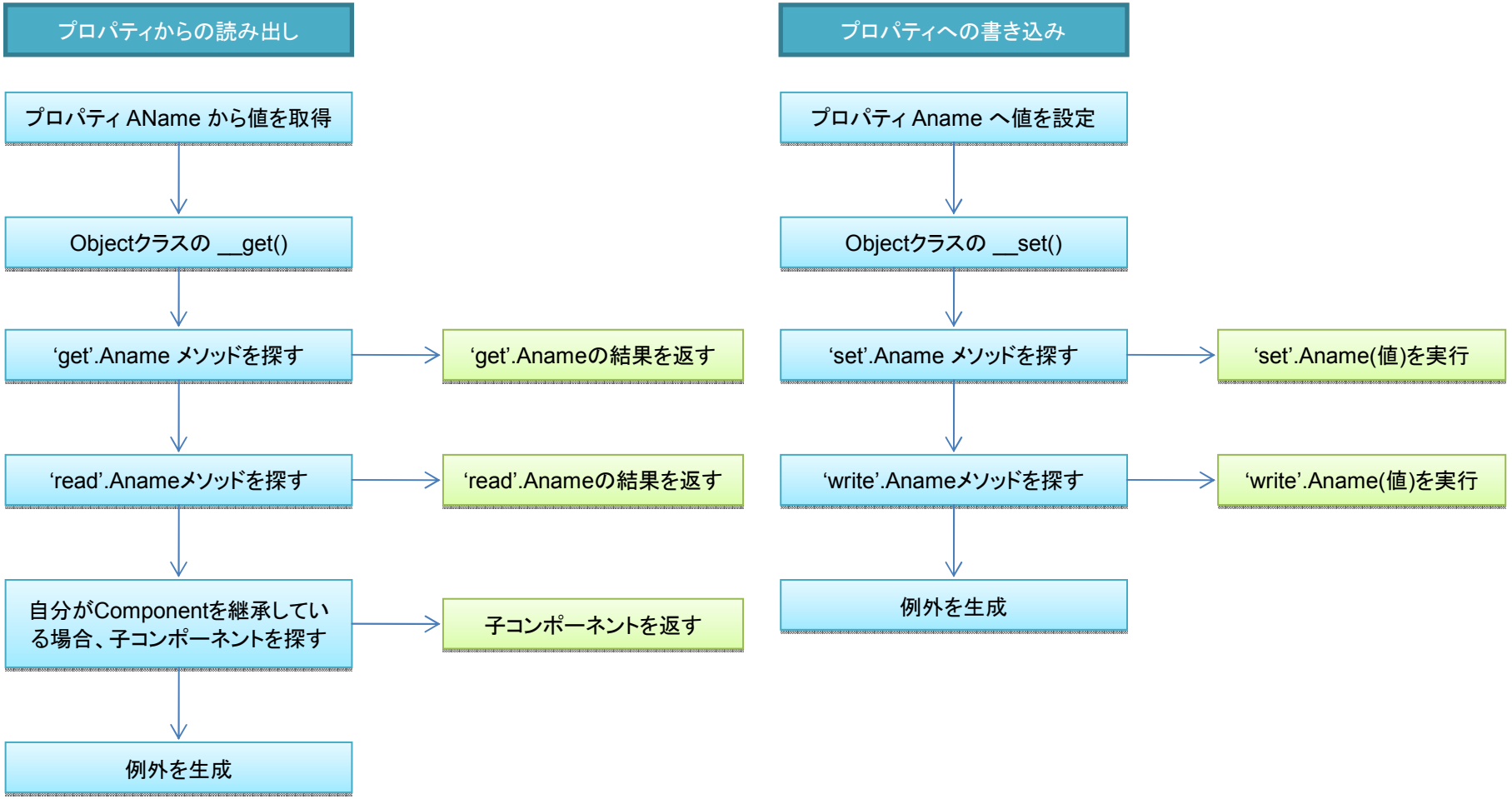
シリアライズが実行されるタイミング

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- オブジェクトの永続化は、PersistentオブジェクトのSerialize / UnSerialize メソッドを利用して実現しています。SerializeはPHPのファイナライザであるshutdown_functionで実行されています。

```
forms.inc.php(58)
function VCLShutdown()
{
	global $application;

	//This is the moment to store all properties in the session to retrieve them later
	$application->serializeChildren();

	//Uncomment this to get what is stored on the session at the last step of your scripts
	/*
	echo "<pre>";
	print_r($_SESSION);
	echo "<pre>";
	*/
}

register_shutdown_function("VCLShutdown");
```

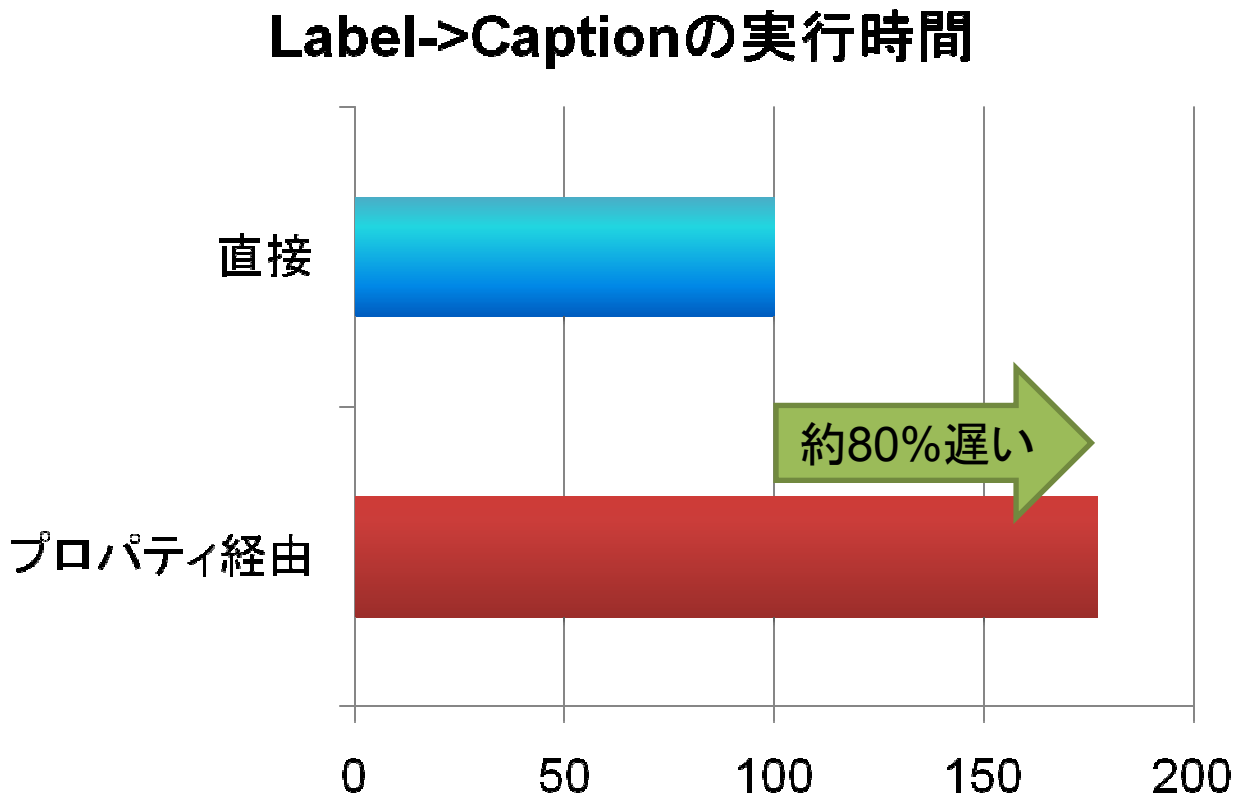
プロパティの実現



プロパティのオーバーヘッド

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- プロパティを使用するオーバーヘッドはどれくらいあるのか？



※測定環境：Core2Quad9550+Mem4G+WindowsXP SP2のマシン上で
Delphi for PHP IDEからデバッグモードで実行した場合の
相対値です。

オブジェクトの動的生成と削除

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- ページ上に表示されるオブジェクトを動的に生成して、永続化することが出来ます。

```
//Labelを動的に生成する
    function Button1Click($sender, $params)
    {
      $labelCount = 0;
      foreach($this->controls->items as $key => $obj)
      {
        if ($obj->ClassName() == 'Label')
        {
          $lastLabel = $obj;
          $labelCount++;
        }
      }
      $newLabel = new Label($this);
      $newLabel->Name = 'Label'.($labelCount+1);
      $newLabel->Top = $lastLabel->Top + 16;
      $newLabel->Left = $lastLabel->Left;
      $newLabel->Caption = 'Label'.($labelCount+1);
      $newLabel->Parent = $this;
    }
```

又は、以下のようにする。

```
$newLabel = new Label();
$this->insertComponent($newLabel);
```

ユニークな名前を必ず付けること。シリアライズで利用されるため、忘れると正常に動作しない。ていうか、表示されません。

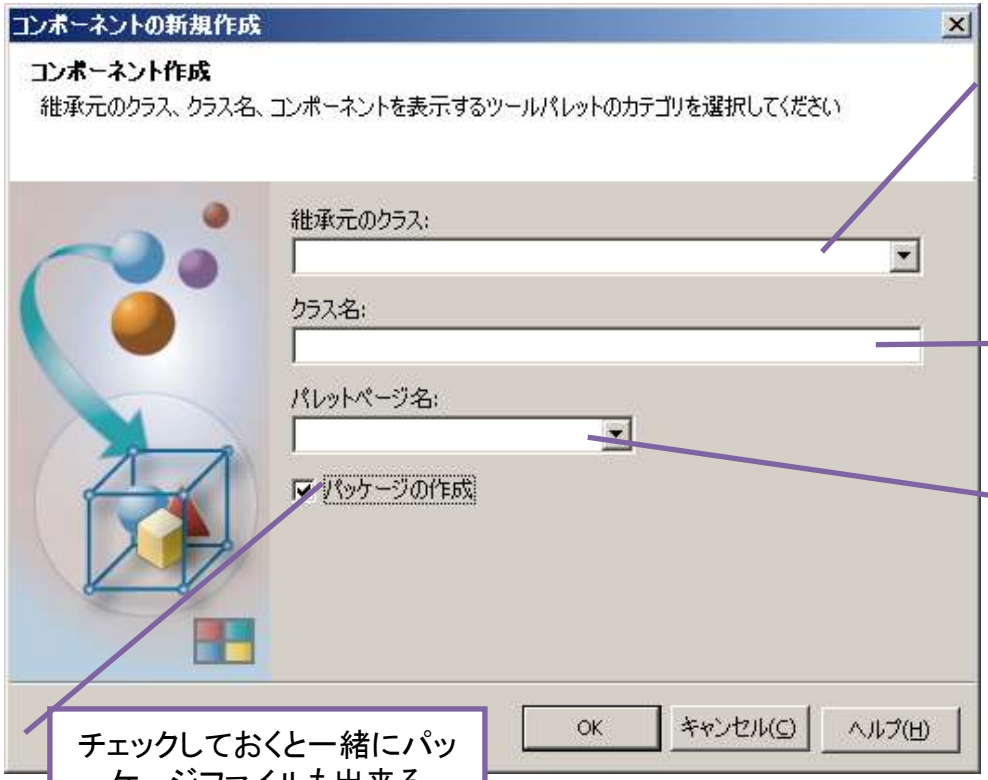
これを忘れると、controls配列に追加されないため、表示されない。

Component指向でGo!

Componentを作成する

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- メインメニューの「コンポーネント」ー「新規コンポーネント」からダイアログを表示させる。



継承元のクラスを指定する

作成するクラス名を指定する

コンポーネントを登録するパレットを指定する

チェックしておくと一緒にパッケージファイルも出来る

後で変更出来るので深く考えないで作成する。

生成されるスケルトン

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- 左がコンポーネント、右がパッケージのスケルトン。

```
<?php
require_once("vcl/vcl.inc.php");
//Includes

//Class definition
class SampleCompo extends Component
{
    function __construct($aowner = null)
    {
        parent::__construct($aowner);
    }

    function dumpContents()
    {
        parent::dumpContents();
    }
}

?>
```

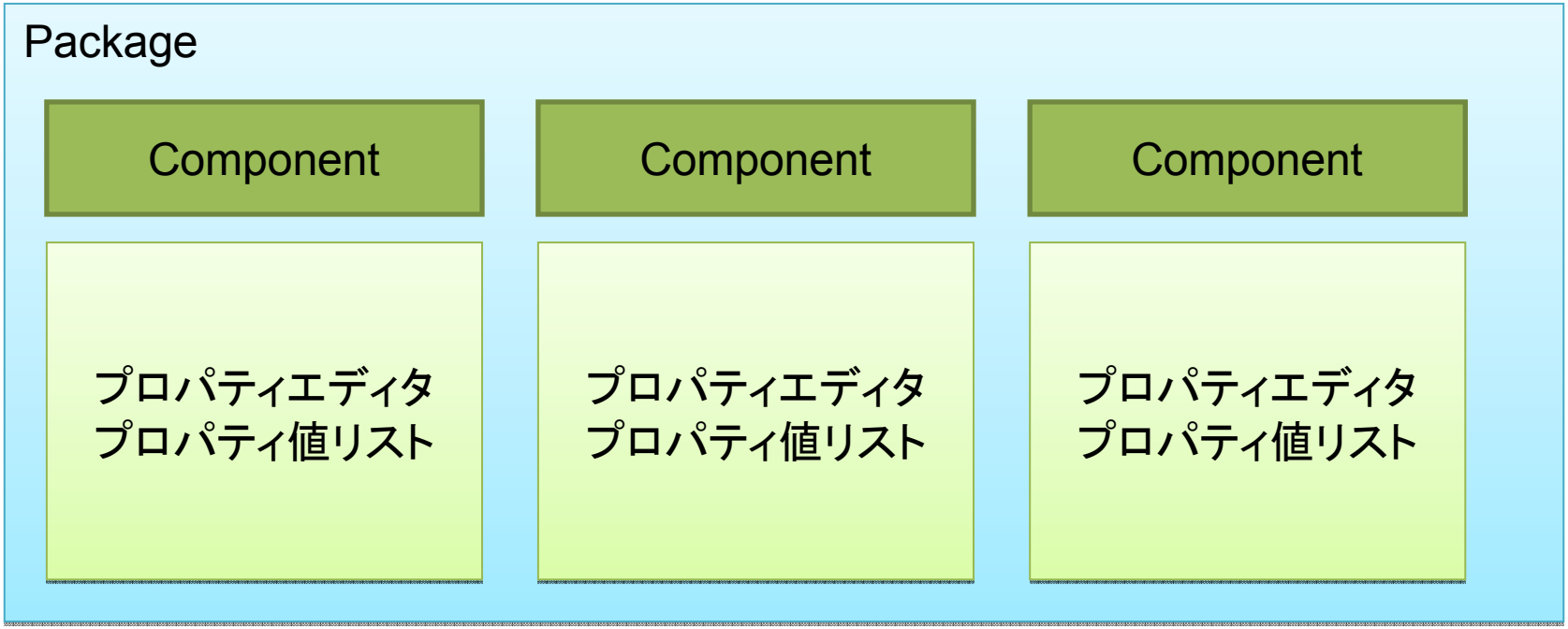
```
<?php
  require_once("vcl/vcl.inc.php");
  use_unit("designide.inc.php");

    setPackageTitle("Put the title of your package
here");
  //Change this setting to the path where the icons
for the components reside
  setIconPath("./icons");

  //Change yourunit.inc.php to the php file which
contains the component code

registerComponents("Samples",array("SampleCo
mpo"),"unit1.inc.php");
?>
```

ここに自分でuse_unitすること。

パッケージとコンポーネントの関係

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- Delphi for PHPのコンポーネントはパッケージに含まれ、パッケージ内でプロパティエディタや値リストなどの設定を行います。



Packageファイル

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- パッケージファイルでやるべき事。

コンポーネントのアイコンは、ここで指定したディレクトリにある、同名のbmpが使用される

パッケージのタイトルを設定する

コンポーネントを登録する

```
<?php
  require_once("vcl/vcl.inc.php");
  use_unit("designide.inc.php");

    setPackageTitle("Anaheim Technology Components");
  //Change this setting to the path where the icons for the components reside
  setIconPath("./icons");

  //Change yourunit.inc.php to the php file which contains the component code
  registerComponents("Anaheim-Tech",array("PHPExcelSimpleReport"),"phpexcel.inc.php");

  registerPropertyValues("PHPExcelSimpleReport","Orientation",array('Default','Portrait','Landscape'));
  registerPropertyValues("PHPExcelSimpleReport","PaperSize",array('A4','A3'));
  registerPropertyValues("PHPExcelSimpleReport","DataSet",array('DataSet'));
  registerPropertyEditor("PHPExcelSimpleReport","TitleRowColor","TSamplePropertyEditor","native");

?>
```

プロパティエディタを設定する

プロパティで利用出来る値を設定する

プロパティエディタ

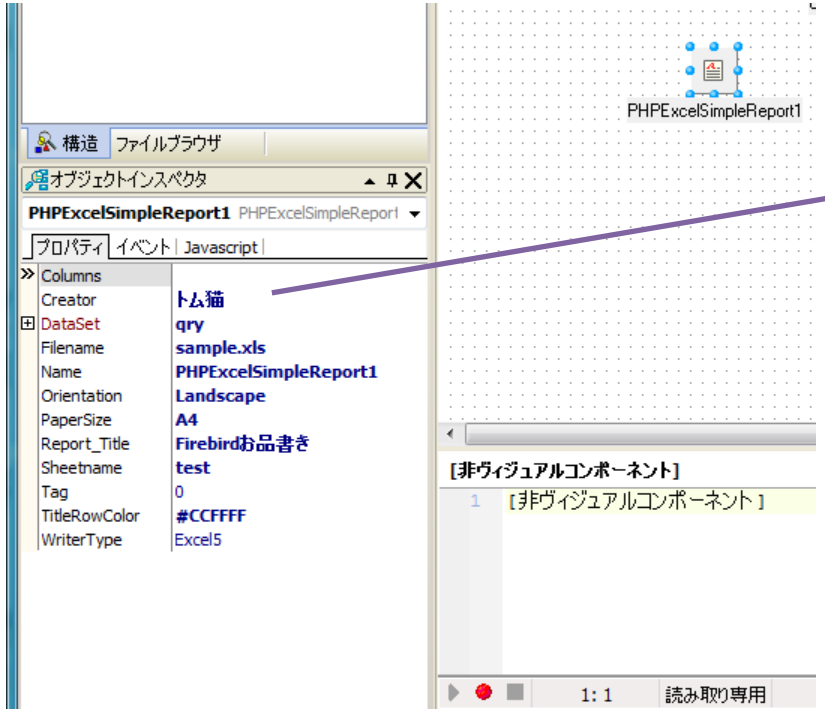
EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- プロパティエディタは以下のものが用意されている、Delphiで自作することも可能。
  - <標準プロパティエディタ>
    - TStringListPropertyEditor　SQL等の複数行の文字列プロパティ用
    - TItemsPropertyEditor　メニューやツールバーで使用されているItemsプロパティ用
    - TGridColumnsPropertyEditor　DBGridのColumnsプロパティ用
    - TFormValidatorRulesPropertyEditor　FormValidatorのRulesプロパティ用
    - TImagePropertyEditor　Imageプロパティ用
    - TImageListPropertyEditor　ImageListプロパティ用
    - THTMLPropertyEditor　Caption等のHTMLで設定するプロパティ用
    - TValueListPropertyEditor　値リストのプロパティ用
    - TFilenamePropertyEditor　ファイル名の指定用（相対位置）
    - TAbsolutePathPropertyEditor　ファイル名の指定用（絶対位置）
    - TSamplePropertyEditor　名前はサンプルですが、色の設定用

コンポーネントとプロパティ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- コンポーネントのPublishedプロパティはシリアライズされると同時に、オブジェクトインスペクタに表示される。



SetメソッドのあるPublishedプロパティはここに表示される

```
protected $_creator = "";
//Creator
  public function setCreator($pValue)
  {
    $this->_creator = $pValue;
  }
  public function getCreator()
  {
    return $this->_creator;
  }
  public function defaultCreator()
  {
    return null;
  }
```

PHPExcelコンポーネント

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

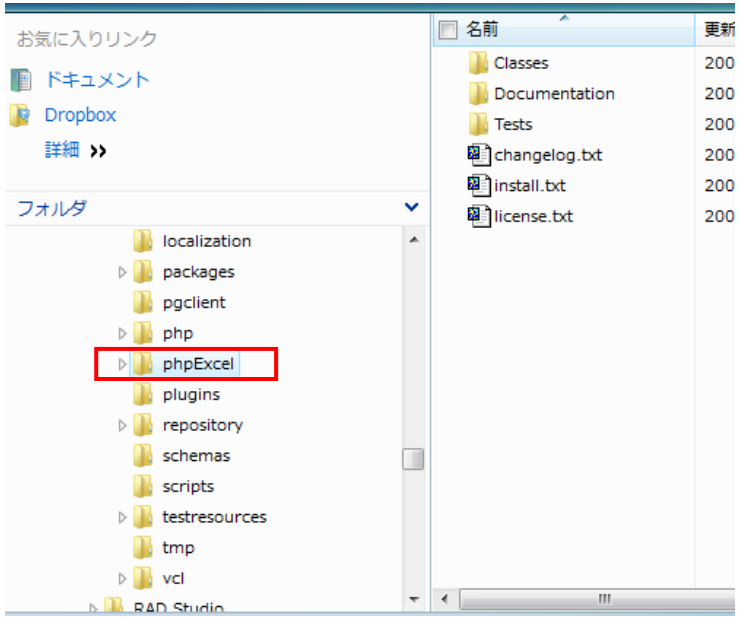
- PHPExcelはExcel2007/Excel2000形式等のファイルを読み書き出来る大変便利なクラスライブラリ。
  - 出力に対応しているファイル形式
    - Excel 2007 (spreadsheetML)
    - BIFF8 (Excel 97 and higher)
    - PHPExcel Serialized Spreadsheet
    - CSV (Comma Separated Values)
    - HTML
    - PDF
  - 読み取りに対応しているファイル形式
    - Excel 2007 (spreadsheetML)
    - BIFF5 (Excel 5.0 / Excel 95), BIFF8 (Excel 97 and higher)
    - PHPExcel Serialized Spreadsheet
    - Excel 2003 XML format
    - Symbolic Link (SYLK)
    - CSV (Comma Separated Values)

PHPExcelコンポーネント

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- PHPExcelのインストールは、ソースコードのzipアーカイブをダウンロードして、PHPのインクルードパスに展開するだけです。Delphi for PHPのIDEからアプリケーションを起動する場合は、以下の位置に配置する。

<Delphi for PHP2.0インストールパス>¥apache2
...
¥php
¥phpExcel
¥Classes
¥Documentation
¥Tests
...
¥vcl



PHPExcelコンポーネントのプロパティ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- PHPExcelコンポーネントには以下のプロパティを追加した。
  - Columns　配列型：レポートの各列の情報
  - Creator　文字列型：作成者
  - DataSet　データセット型：元になるデータセット
  - Filename　文字列型：生成されるファイル名
  - Orientation　文字列型：用紙のタテ・ヨコの指定
  - PaperSize　文字列型：A4とA3だけ限定で用紙サイズを選択
  - Report_Title　文字列型：レポートのタイトル
  - Sheetname　文字列型：シート名
  - TitleRowColor　文字列型：カラー指定
  - WriterType　文字列型：Excel5とExcel2007から選択

データセットなどのオブジェクト型プロパティ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

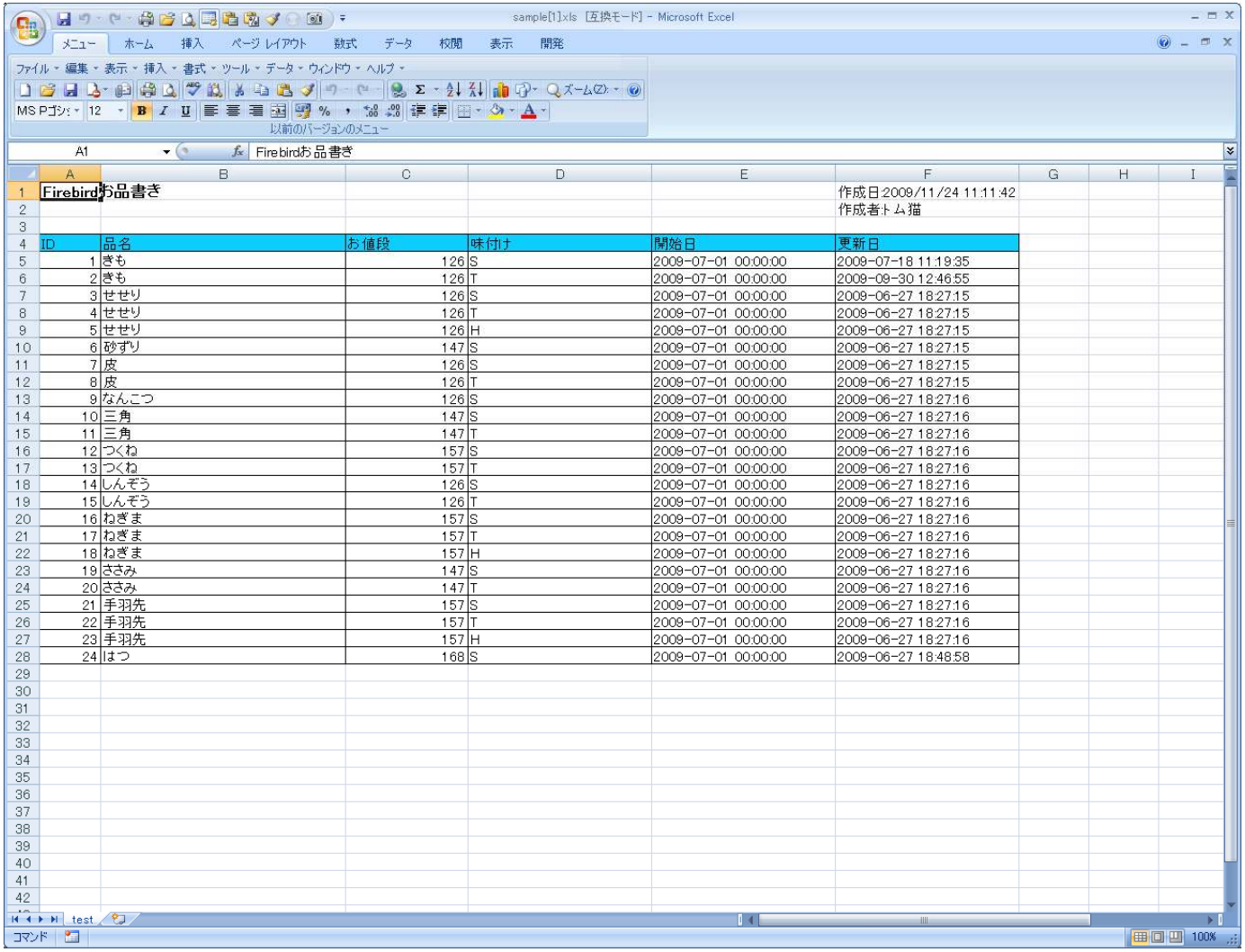
- データセットやデータソース等のオブジェクト型のプロパティには注意が必要。

- setメソッドでは Componentクラスの fixupProperty メソッドを利用します。

```
//DataSet
 protected $_dataset = null;
 public function setDataSet($pValue)
 {
   $this->_dataset = $this->fixupProperty($pValue);
 }
 public function getDataSet()
 {
   return $this->_dataset;
 }
 public function defaultDataSet()
 {
   return null;
 }
```

実行時の様子

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP



sample[1].xls [互換モード] - Microsoft Excel

	A	B	C	D	E	F
1	Firebirdお品書き					作成日 2009/11/24 11:11:42
2						作成者 トム猫
3						
4	ID	品名	お値段	味付け	開始日	更新日
5	1	きも	126	S	2009-07-01 00:00:00	2009-07-18 11:19:35
6	2	きも	126	T	2009-07-01 00:00:00	2009-09-30 12:46:55
7	3	せせり	126	S	2009-07-01 00:00:00	2009-06-27 18:27:15
8	4	せせり	126	T	2009-07-01 00:00:00	2009-06-27 18:27:15
9	5	せせり	126	H	2009-07-01 00:00:00	2009-06-27 18:27:15
10	6	砂ずり	147	S	2009-07-01 00:00:00	2009-06-27 18:27:15
11	7	皮	126	S	2009-07-01 00:00:00	2009-06-27 18:27:15
12	8	皮	126	T	2009-07-01 00:00:00	2009-06-27 18:27:15
13	9	なんこつ	126	S	2009-07-01 00:00:00	2009-06-27 18:27:16
14	10	三角	147	S	2009-07-01 00:00:00	2009-06-27 18:27:16
15	11	三角	147	T	2009-07-01 00:00:00	2009-06-27 18:27:16
16	12	つくね	157	S	2009-07-01 00:00:00	2009-06-27 18:27:16
17	13	つくね	157	T	2009-07-01 00:00:00	2009-06-27 18:27:16
18	14	しんぞう	126	S	2009-07-01 00:00:00	2009-06-27 18:27:16
19	15	しんぞう	126	T	2009-07-01 00:00:00	2009-06-27 18:27:16
20	16	ねぎま	157	S	2009-07-01 00:00:00	2009-06-27 18:27:16
21	17	ねぎま	157	T	2009-07-01 00:00:00	2009-06-27 18:27:16
22	18	ねぎま	157	H	2009-07-01 00:00:00	2009-06-27 18:27:16
23	19	ささみ	147	S	2009-07-01 00:00:00	2009-06-27 18:27:16
24	20	ささみ	147	T	2009-07-01 00:00:00	2009-06-27 18:27:16
25	21	手羽先	157	S	2009-07-01 00:00:00	2009-06-27 18:27:16
26	22	手羽先	157	T	2009-07-01 00:00:00	2009-06-27 18:27:16
27	23	手羽先	157	H	2009-07-01 00:00:00	2009-06-27 18:27:16
28	24	はつ	168	S	2009-07-01 00:00:00	2009-06-27 18:48:58

Firebird徹底入門　登場！

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- 今日の懇親会の景品です！

アナハイムテクノロジーについて

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- アナハイムテクノロジー株式会社
  - 〒157-0072　世田谷区祖師谷1-22-26-S-208
  - TEL 03-5787-7791　FAX 03-5787-7792
  - http://www.anaheim-tech.com/
- Delphiとオープンソースの技術支援を行っています
- Firebird/InterBaseのご相談もぜひ当社へ！
- Delphi for PHPの導入のご相談受付中です