

JavaFX: JFX 業務システム・フレームワーク 使用ガイド

2013年2月20日
佐藤システム設計事務所

目次

1. はじめに.....	2
2. 開発環境.....	5
2.1. 開発環境の設定.....	5
2.1.1. NetBean IDE 7.2.....	5
3. 「JFX 業務システム・フレームワーク」の準備.....	6
4. アプリケーションの作成手順.....	7
4.1. 新しいプロジェクトを作成.....	7
4.2. 新しい処理画面を作成.....	7
5. 「JFX 業務システム・フレームワーク」の構成.....	8
5.1. データベース.....	10
5.2. 共通事項.....	13
5.2.1. グローバル変数.....	13
5.2.2. メッセージ・ボックス／インプット・ボックス.....	13
5.2.3. テーブル・ロック.....	14
5.2.4. レコード・ロック.....	14
5.2.5. データの暗号化・復号化.....	14
5.2.6. 数字入力テキスト・フィールド.....	15
5.2.7. PHPプログラムを実行するメソッド.....	15
5.2.8. 帳票印刷(売上傳票).....	16
5.2.9. ユーザ・ランク.....	16
5.3. ログイン／メニュー.....	17
5.4. 顧客管理.....	18
5.5. 売上管理.....	19
5.6. ユーザ設定・登録.....	20
5.7. 暗号化・復号化ツール.....	20
5.8. テンプレート.....	21
5.9. PHP プログラム.....	21

1. はじめに

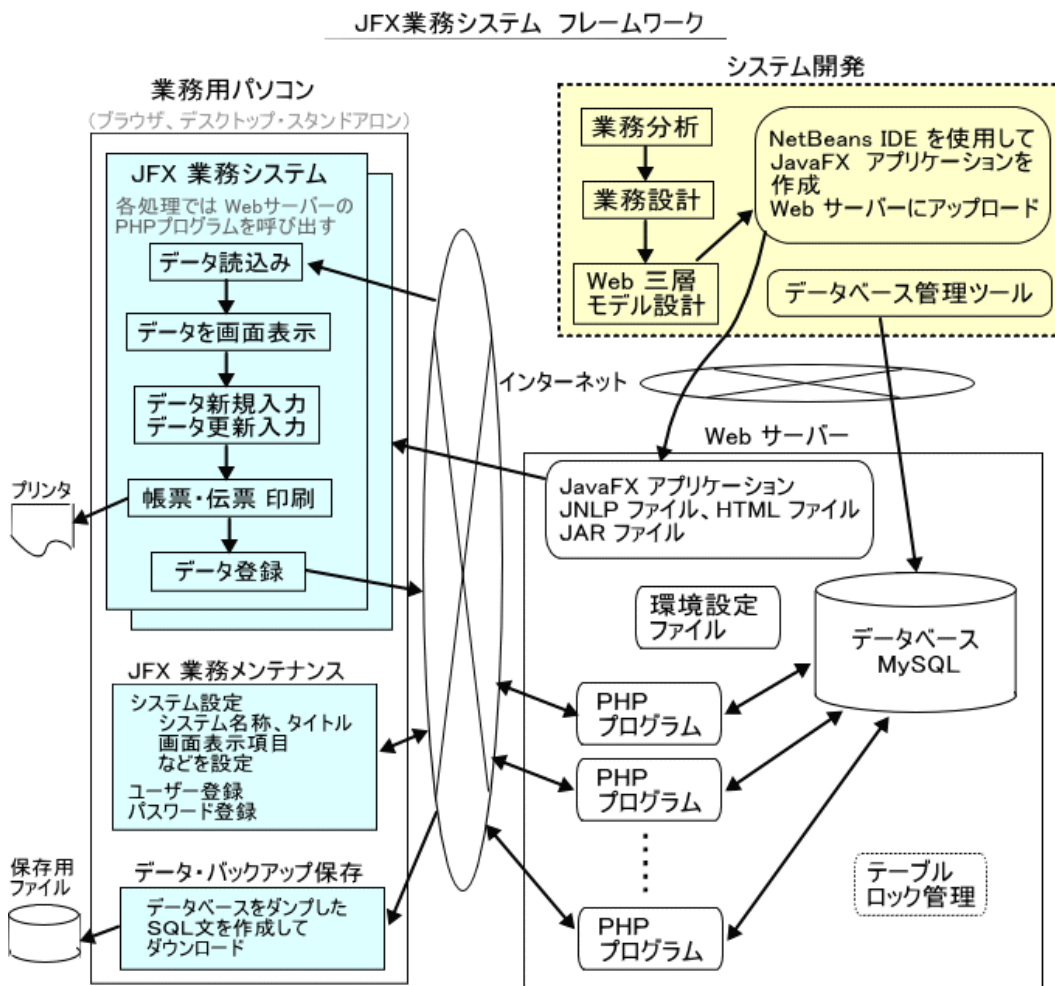
JavaFX とは Java 上で動作する リッチインターネット・アプリケーションの GUI ライブラリです。Java プログラムと fxml および css を併用して 業務システム を開発することができます。Java が使用できる環境 (Java SE 7 Update 7 以降がインストールされている Windows、Mac および Linux) で動作します。

「JavaFX: JFX 業務システム・フレームワーク」は JavaFX を使用した Web ベースの業務システムを開発するためのフレームワークです。

JavaFX を使用することにより、リッチクライアントの画面構成と操作性により、ユーザーの業務に適応した Web 業務システムを構築することができます。また、既存のクライアント・サーバ方式の業務システムをインターネット・ベースの業務システムに改訂することもできます。

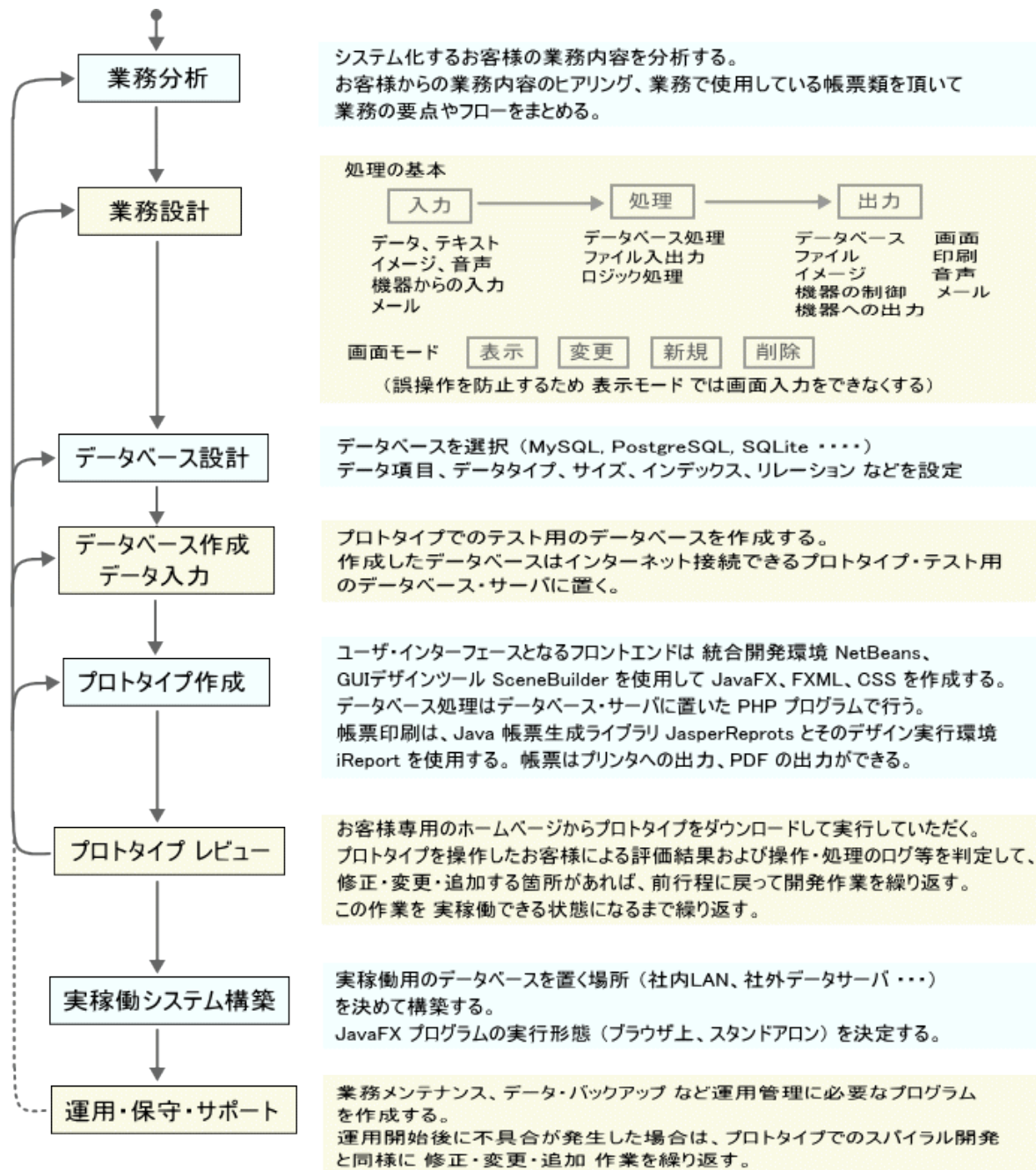
「JavaFX: JFX 業務システム・フレームワーク」は以下の要素から構成されています。

1. ユーザーが使用する業務用パソコンで動作 (ブラウザ あるいは スタンドアロンで動作) する JavaFX アプリケーションを構成する Java ファイル、fxml ファイル、css ファイルを作成する基準となるひな型。
2. Web サーバー側の PHP プログラムのひな型、データベース (MySQL) のテーブルのひな型。環境設定、テーブルロック方式、通信するデータの暗号化・復号化メソッドなど。



「JavaFX: JFX 業務システム・フレームワーク」を使用した業務アプリケーションの開発は、プロトタイプを使用したスパイラル開発に適しています。

開発したプロトタイプをお客様・開発者がレビューした結果に基づいて、必要な工程に戻って機能の修正・変更・追加を行います。この開発工程を繰り返すことにより、プロトタイプを最終的なシステムに仕上げて行くことができます。また、実稼働システムを運用開始後もプロトタイプを開発・テストした環境は残しておき、不具合の修正や機能の追加が必要になった場合に利用します。



使用する開発ツール

「JavaFX: JFX 業務システム・フレームワーク」を構成するプログラム・ファイルは、その種類により異なる開発ツールを使用して編集することになります。

主に使用する開発ツールはJavaの統合開発環境「Net Beans IDE」です。

「Net Beans IDE」で使用されるファイルは java、FXML、css ファイルです。

java ファイル …… JavaFX の API を使用して処理画面の構成や処理を記述する

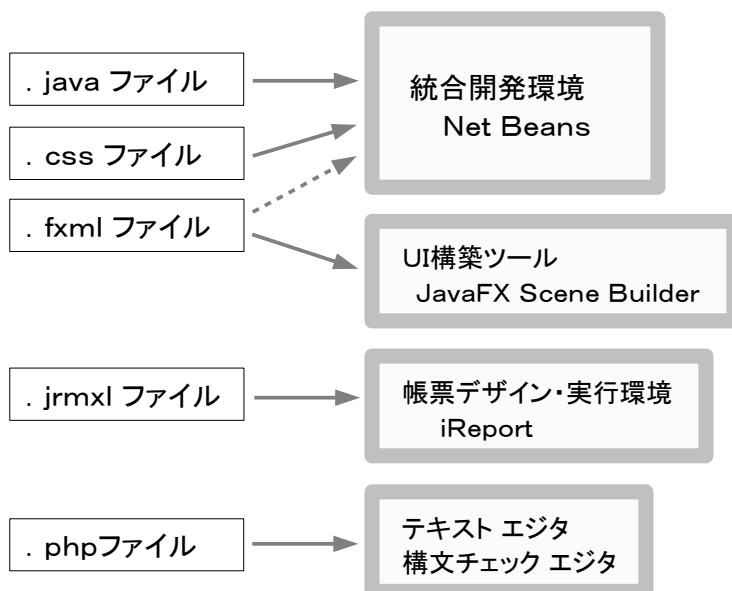
FXML ファイル …… 処理画面の定義を記述する XML ファイル

css ファイル …… 処理画面のボタンやテキストフィールドなどのコントロールのスタイルを定義する

各ファイルは「Net Beans IDE」で直接編集することができますが、FXML ファイルはUI構築ツールの「JavaFX Scene Builder」でUIコンポーネントを画面上に表示して編集していきます。

売上傳票などの帳票 jrmxl ファイルについては帳票デザイン・実行環境の「iReport」を使用して編集します。

PHP プログラムファイルはテキストエジタで直接編集します。PHPの構文チェックができるエジタも使用します。



2. 開発環境

「JFX 業務システム・フレームワーク」を開発したシステム環境

Windows 7 Professional SP1	OS
Java JDK 7 update 7	Java 開発用ツール (Java 2.2 が含まれている)
Net Beans IDE 7.2	Java 統合開発環境、JavaFX のプロジェクトを作成する
JavaFX Scene Builder 1.0	JavaFX のUI構築ツール
iReport 5.0.0	帳票デザイン・実行環境
PHP 5.4.7	スクリプト言語
MySQL 5.5	データベース
phpMyAdmin 3.5.4	データベース MySQL の管理ツール
Apache 2.2.22	Web サーバ
MKEditor for Windows 6.1	テキストエジタ
PHP エジタ standalone 1.33	PHP専用のエジタ、PHP の構文チェックに使用する
LibreOffice Writer 3.6.3.2	ワープロ、このドキュメントを作成、PDF ファイルに出力する

2.1. 開発環境の設定

2.1.1. NetBean IDE 7.2

● Java プラットフォームを追加して JavaFX を使えるようにする

[ツール]→[Java プラットフォーム]→「Java プラットフォーム・マネージャ」が開く。

[プラットフォームの追加] で「C:\Program Files\Java\jdk1.7.0_07」を選択する。新しいJava プラットフォームが追加されるので、「JavaFX」タブを開く。「JavaFX を有効にする」をチェックして、「JavaFX SDK:」を「参照」で指定する。

● プロジェクトで使用する JDK を設定する

プロジェクト名を右クリックして「プロパティ」を選択する。「プロジェクト・プロパティ」が開く。

「ライブラリ」を選択して、「Java プラットフォーム」を指定する。「プラットフォームの管理」→[ツール]→[Java プラットフォーム]→「Java プラットフォーム・マネージャ」が開くので、上記で追加した「Java プラットフォーム」を選択する。

● プロジェクトで使用する iReport を設定する

プロジェクト名を右クリックして「プロパティ」を選択する。「プロジェクト・プロパティ」が開く。「ライブラリ」を選択する。「コンパイル」タブで「JAR/フォルダの追加」ボタンをクリックする。以下の Jar ファイルを追加する。

```
C:\Program Files (x86)\Jaspersoft\iReport\ireport\modules
    ¥ext¥commons-beanutils-18. 2. jar
    ¥ext¥commons-collection-3. 2. 1.jar
    ¥ext¥commons-degester-2. 1.jar
    ¥ext¥commons-logging-1. 1.jar
    ¥ext¥commons-logging-1. 1.jar
    ¥ext¥iText-2. 1. 7. jar
    ¥ext¥iTextAsian. jar
    ¥ext¥jasperreports. jar
    ¥com-jaspersoft-ireport. jar
```

3. 「JFX 業務システム・フレームワーク」の準備

1. プロジェクトを置くフォルダを、「エクスプローラ」で作成してください。ここでは、フォルダ「D: ¥JavaFX」として説明します。
2. 「JFX 業務システム・フレームワーク」を以下のお客様サポート専用のホームページよりダウンロードしてください。
(接続するためにはユーザアカウントとパスワードの入力が必要です。)
http://www.satosys.com_jfx-gyom/user_download/

ダウンロードするファイル名は JFX_gyom_framework.zip です。ZIP形式の圧縮ファイルになっていますので、プロジェクトを置くフォルダ「D: ¥JavaFX」に展開してください。

展開したフォルダには フォルダ

JFX_gyom_framework JavzFX のプロジェクト

JFX_gyom_framework_PHP PHP プログラム

JFX_gyom_framework_iReport 帳票デザイン・ファイル

JFX_gyom_framework_Help ヘルプ・ファイル

がありますので、確認してください。

3. 「Net Beans IDE」を起動して、以下のように新しいプロジェクトを作成します。
[ファイル]→[プロジェクトを開く]→「D: ¥JavaFx¥JFX_gyom_framework」を選択して開く。
プロジェクト「JFX_gyom_framework」が作成されます。
4. 「iReport」を起動して、以下の 売上傳票帳票ファイル を開いてください。
D: ¥JavaFX¥JFX_gyom_framework_iReport¥uriage_denpyo¥uri_denpyo. jrxml
レポート全体の「プロパティ」で「Language」を「Java」に変更してください。
日本語を表示できるようにするため、以下のように フォントと クラスパス を設定してください。
日本語を表示したい「Static Text」、「Text Field」コンポーネントを選択して、そのプロパティで
「Pdf Font name」に「Heisei Kaku Go-W5」または「HeiseiMin-W3」を指定する。
「Pdf Encode」に「UniJIS-UCS2-H(Japanese)」を指定する。

4. アプリケーションの作成手順

4.1. 新しいプロジェクトを作成

「Net Beans IDE」の新しいプロジェクトを作成して、ここに新しいアプリケーションを作成します。
新しいプロジェクトを置くためのフォルダは、「D: ¥JavaFX ¥JFX_new_gyom」として説明します。

フォルダ「D: ¥JavaFX ¥JFX_gyom_framework」をコピーして、フォルダ「D: ¥JavaFX」に貼り付けてください。このフォルダのフォルダ名を「JFX_new_gyom」に変更してください。

「Net Beans IDE」を起動して、以下のように新しいプロジェクトを作成してください。

[ファイル]→[プロジェクトを開く]→「D: ¥JavaFx ¥JFX_new_gyom」を選択して開く。

プロジェクト「JFX_new_gyom」が作成されますが、プロジェクト名は「JFX_gyom_framework」となっていますので、以下のようにしてこのプロジェクト名を変更します。

開いたプロジェクトのプロジェクト名を右クリックして「プロパティ」を選択する。[プロジェクト・プロパティ]が開くので、「プロジェクト・フォルダ」が「D: ¥JavaFx ¥JFX_new_gyom」になっていることを確認。

[プロジェクト・プロパティ]を閉じる。

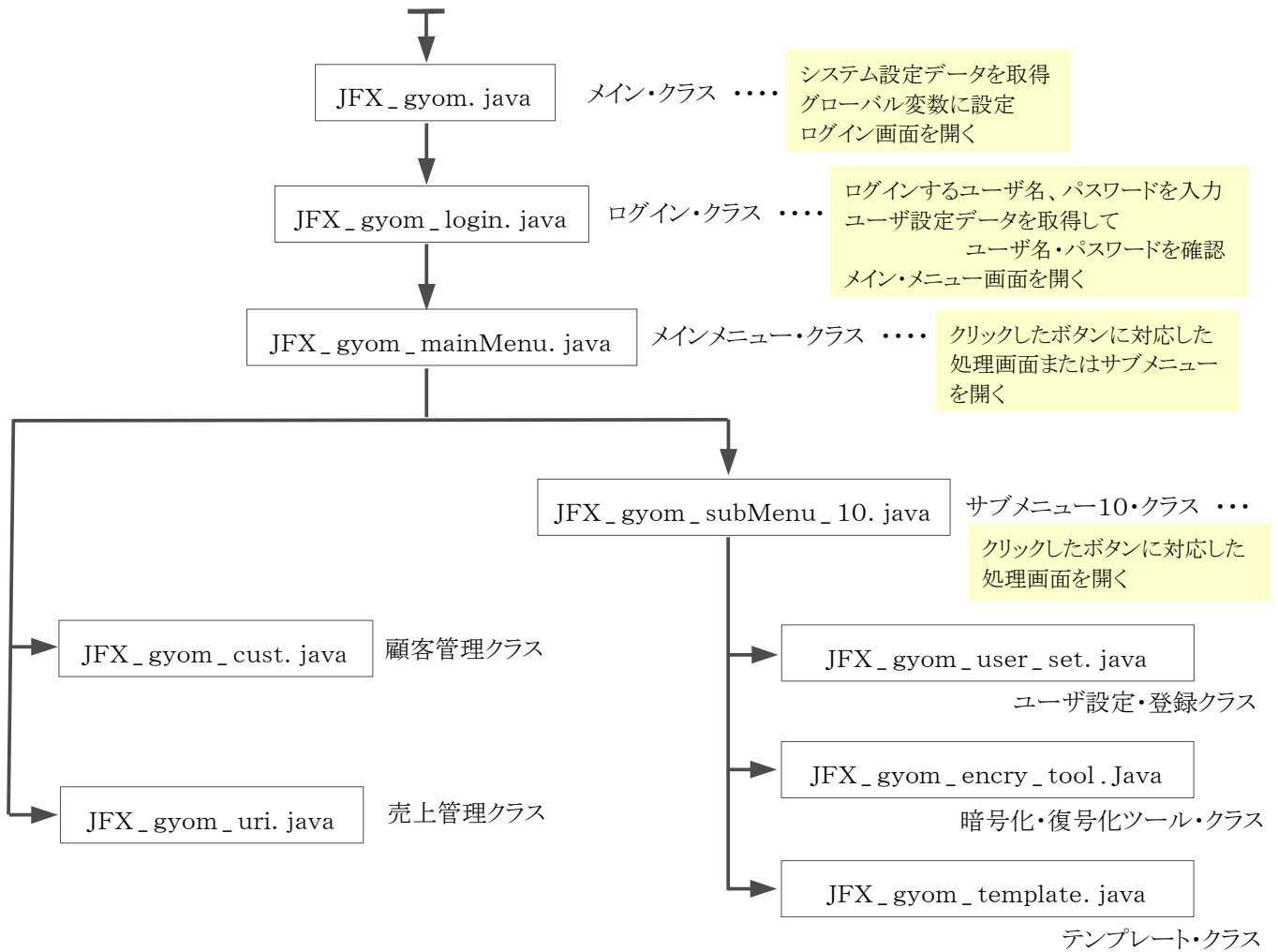
プロジェクト名を右クリックして「名前の変更」を選択して、プロジェクト名を「JFX_new_gyom」に変更する。

「メイン・クラス」、「ログイン」、「メインメニュー」、「サブメニュー」、「ユーザ設定・登録」、「暗号化・復号化ツール」は「JFX 業務システム・フレームワーク」のものをそのまま使用し、画面の表記とメニューボタンのクリックイベントの内容を変更するのみとします。

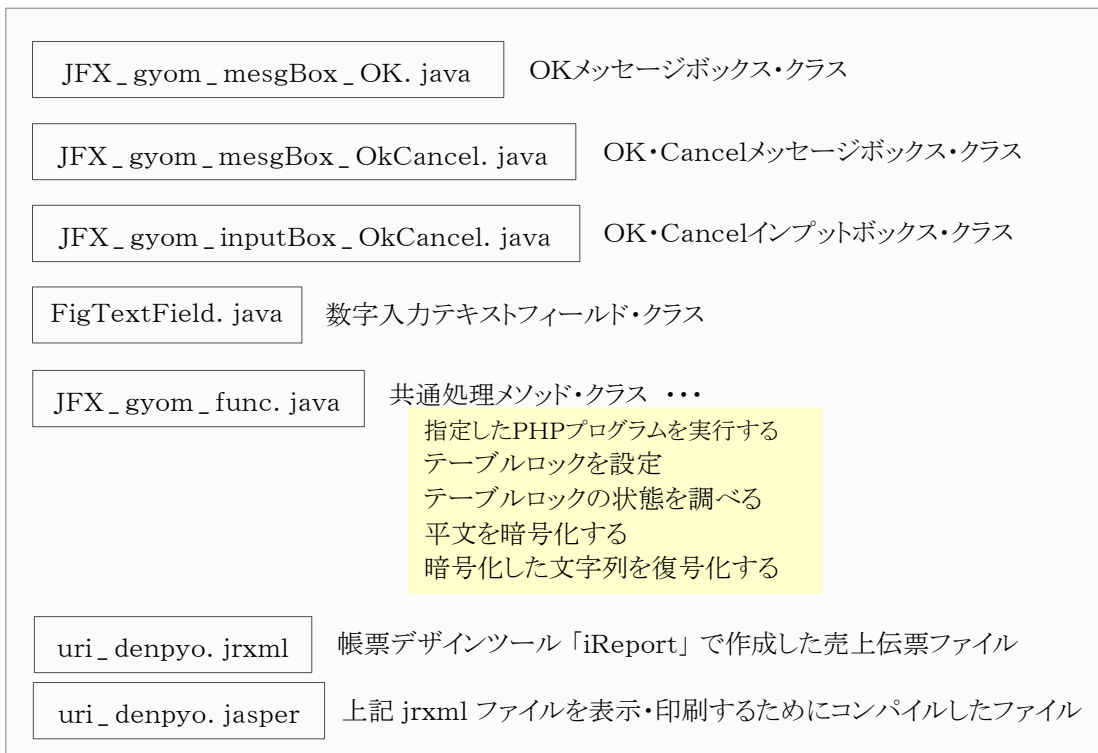
4.2. 新しい処理画面を作成

新しく作成する処理画面は、「顧客マスター」をひな型として「商品マスター」を作成するとして説明していきます。

5. 「JFX 業務システム・フレームワーク」の構成



共通画面・処理



PHP プログラム

mysql_server_data.txt	MySQLデータ初期設定ファイル。各PHPプログラムで起動時に読み込む
JFX_check_lock_data.php	指定したテーブルがロックされているか調べる
JFX_del_cust_data.php	指定した顧客データを顧客マスターテーブルから削除する
JFX_del_uri_data.php	指定した売上データを売上テーブルと売上明細テーブルから削除する
JFX_get_cust_data.php	指定した顧客データを顧客マスターテーブルから取得する
JFX_get_serch_cust.php	顧客検索画面で指定した顧客データを顧客マスターテーブルから取得する
JFX_get_system_data.php	システム設定データをシステム設定テーブルから取得する
JFX_get_uri_data.php	指定した売上データを売上テーブルと売上明細テーブルから取得する
JFX_get_user_data.php	指定したユーザ設定データをユーザ設定テーブルから取得する
JFX_get_user_password.php	指定したユーザのパスワードとユーザランクをユーザ設定テーブルから取得する
JFX_set_cust_data.php	顧客データを顧客マスターテーブルに書き込む
JFX_set_lock_data.php	指定したテーブルのロック年月日時分秒をテーブルロック管理テーブルに書き込む
JFX_set_login_log.php	ログインしたユーザ情報をログインログ・テーブルに書き込む
JFX_set_system_data.php	システム設定データをシステム設定テーブルに書き込む
JFX_set_unlock_data.php	指定したテーブルのロックを解除する
JFX_set_uri_data.php	売上データと売上明細データを売上テーブルと売上明細テーブルに書き込む
JFX_set_user_data.php	ユーザ設定データをユーザ設定テーブルに書き込む

5.1. データベース

RDBMS: MySQL 5.5

データベース名: **jfx_gyom_db**

文字セット: UTF-8

ストレージエンジン: InnoDB

テーブル: **システム設定** (先頭のレコードのみを使用する。ログインする直前に読み込む)**fx_system_set**

		カラム名	データ型	
●	▲	id	int (11)	
		system_title	varchar (50)	システムのタイトル
		help_url	varchar (100)	ヘルプファイルのURL
		ver_no	varchar (10)	バージョン番号
		exe_mode	varchar (10)	実行モード (“DEVELOP”、“PRODUCT”、“DEMO”)
		zeiritsu	tinyint (4)	消費税率 [%]
		zei_hasu_syori	varchar (10)	”切上”、“切捨”、“四捨五入”
		memo	varchar (100)	メモ

記号の説明: ●プライマリーキー ◎インデックス(ユニーク) ○インデックス ▲オートインクリメント
 データを更新: JFX_set_system_data.php を編集して書き込む内容を変更する

実行はコマンドプロンプトで

C: ¥php¥php.exe D: ¥JavaFX¥JFX_gyom_framework_php¥JFX_set_system_data.php

テーブル: **ユーザ設定** (ログインするユーザの ユーザ名、ユーザランク、パスワード を設定する)**fx_user_set**

		カラム名	データ型	
●	▲	id	int (11)	
		gyo_no	.int (11)	行番号
◎		user_nam	varchar (20)	ユーザ名
		user_rank	tinyint (4)	ユーザランク (0=管理者、1=読み書き可、2=表示のみ)
		password	varchar (50)	パスワード (暗号化)
		memo	varchar (100)	メモ

記号の説明: ●プライマリーキー ◎インデックス(ユニーク) ○インデックス ▲オートインクリメント

5. 「JFX 業務システム・フレームワーク」の構成

テーブル: **テーブルロック管理** (各データベース処理で主に使用するテーブル名でロックを管理する)

fx_table_lock

		カラム名	データ型	
●	▲	id	int (11)	
◎		table_name	varchar (20)	主に使用するテーブル名
		lock_date_time	varchar (14)	ロックした年月日時分秒 “yyyymmddhhMMss”

記号の説明: ●プライマリーキー ◎インデックス(ユニーク) ○インデックス ▲オートインクリメント

テーブル: **ログインログ** (ログインしたユーザを記録する)

fx_login_log

		カラム名	データ型	
●	▲	id	int (11)	
		user_name	varchar (20)	ログインしたユーザ名
		login_date_time	varchar (14)	ログインした年月日時分秒 “yyyymmddhhMMss”

記号の説明: ●プライマリーキー ◎インデックス(ユニーク) ○インデックス ▲オートインクリメント

テーブル: **顧客マスタ** (顧客データを登録する)

fx_cust_master

		カラム名	データ型	
●	▲	id	int (11)	
◎		cust_no	int (11)	顧客番号
○		cust_name	varchar (150)	顧客名 (暗号化)
○		cust_name_kana	varchar (150)	顧客名カナ (暗号化)
○		tel	varchar (90)	電話番号 (暗号化)
		fax	varchar (90)	FAX 番号 (暗号化)
		zip_code	varchar (8)	郵便番号
		address_1	varchar (150)	住所1 (暗号化)
		address_2	varchar (150)	住所2 (暗号化)
		seikyū_flag	tinyint (4)	請求フラグ
		kakeuri_flag	tinyint (4)	掛売フラグ
		uri_ruikai_kin	Decimal (10,0)	売上累計金額
		cust_kubun	varchar (10)	顧客区分
		reg_date	varchar (8)	登録年月日 (“yyyymmdd”)
		memo	varchar (300)	メモ (暗号化)
		lock_date_time	varchar (14)	ロックした年月日時分秒 “yyyymmddhhMMss”

記号の説明: ●プライマリーキー ◎インデックス(ユニーク) ○インデックス ▲オートインクリメント

テーブル: **売上データ** (売上データを登録する)

fx_uri_data

		カラム名	データ型	
●	▲	id	int (11)	
◎		rec_no	int (11)	レコード番号
◎		uri_no	int (11)	売上番号
○		cust_no	int (11)	顧客番号
○		uri_date	varchar (8)	売上年月日 (“yyyymmdd”)
		zeinuki_gokei_kin	Decimal (12,0)	税抜合計金額
		zei_gokei_kin	Decimal (8,2)	消費税合計金額
		zeikomi_gokei_kin	Decimal (12,0)	税込合計金額
		tekiyou	varchar (50)	摘要
		lock_date_time	varchar (14)	ロックした年月日時分秒 “yyyymmddhhMMss”

記号の説明: ●プライマリーキー ◎インデックス(ユニーク) ○インデックス ▲オートインクリメント

テーブル: **売上明細データ** (売上明細データを登録する)

fx_uri_meisai_data

		カラム名	データ型	
●	▲	id	int (11)	
○		uri_no	int (11)	売上番号
○		gyo_no	tinyint (4)	行番号
		uri_kubun_code	varchar (10)	売上区分コード
		product_code	varchar (10)	商品コード
		suu	Decimal (8,0)	数量
		tanka	Decimal (8,0)	単価
		kin	Decimal (10,0)	金額
		bikou	varchar (20)	備考

記号の説明: ●プライマリーキー ◎インデックス(ユニーク) ○インデックス ▲オートインクリメント

5.2. 共通事項

5.2.1. グローバル変数

「JavaFX: JFX 業務システム・フレームワーク」全体で使用する 定数・変数 はグローバル変数として JFX_gyom.java で定義されています。これらは、定数として初期化されるかログイン前またはログイン時にデータベースより取得した内容がセットされます。それ以外の場所で内容を変更しないようにしてください。

[定数]

```
static final String PHP_URL_ = "http://localhost/JavaFX/JFX_gyom_framework_PHP/"; //PHPプログラムの URL
static final String ENCRYPT_KEY_ = "4uWe3Q#@95"; //暗号化キー
static final String DELIMITER_ = "#%"; //送受信するデータのフィールドを区切るデリミタ
```

[ログインする前にテーブル fx_system_set より取得してセットする]

```
static String System_TITLE_; //システム・タイトル
static String Help_URL_; //ヘルプ・ファイルの URL
// "file:///D:/JavaFX/JFX_gyom_framework_Help/jfx-gyom_help.html"

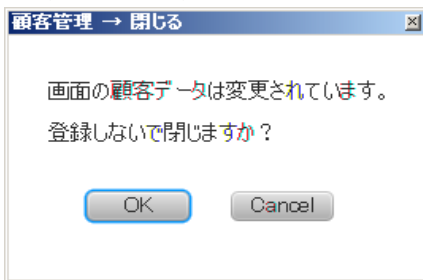
static String Ver_NO_; //バージョン番号("1.000")
static String Exe_MODE_; //実行モード("DEVELOP", "PRODUCT", "DEMO")
static double ZeiRITSU_; //消費税率 (0.05)
static String Zei_HASU_SYORI_; //消費税端数処理("切捨", "切上", "四捨五入")
static int Lock_Hold_Time_; //ロック保持時間 [分]
```

[ログイン時にテーブル fx_user_set より取得してセットする]

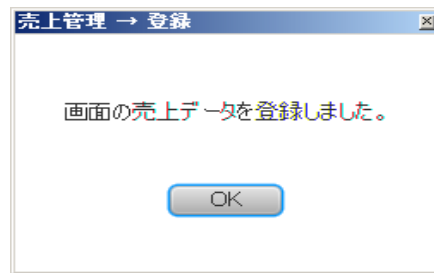
```
static String User_NAME_; //ログインしたユーザー名
static int User_RANK_; //ユーザーランク (0=管理者, 1=読み書き可, 2=表示のみ)
```

5.2.2. メッセージ・ボックス／インプット・ボックス

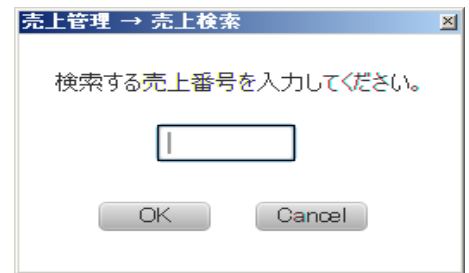
メッセージ・ボックス OK・Cancel



メッセージ・ボックス OK



インプット・ボックス OK・Cancel



「OK・Cancel メッセージボックス」の場合について説明します。

メッセージ・ボックスを開くには「OK・Cancel メッセージボックス・クラス」のオブジェクトを取得してそのオブジェクトの `msgBox_OkCancel_show()` メソッドを実行します。

引数は「タイトル、メッセージ、「OK」ボタンをクリックした時に実行する処理の指定、X位置、Y位置」となります。

(例)

```
JFX_gyom_msgBox_OkCancel obj = new JFX_gyom_msgBox_OkCancel();
obj.msgBox_OkCancel_show("顧客管理 → 登録", "画面の顧客データを登録しますか?", "cust_reg", 520, 400);
```

「OK」ボタンをクリックした時に実行する処理は、「OK・Cancel メッセージボックス・クラス」の「OK」ボタンをクリックした時のイベント `handleOKbtnAction()` メソッド内で、指定した引数毎に記述します。

「Cancel」ボタンをクリックした時は、何もしないでメッセージ・ボックスを閉じます。

5.2.3. テーブル・ロック

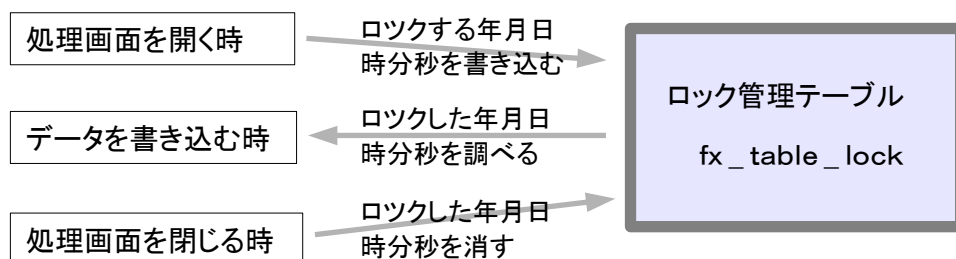
処理画面において、データを表示・変更・新規作成・削除 するためにデータを取得する時に、その処理画面で主に使用するテーブル名で、ロック管理テーブル「fx_table_lock」のカラム「lock_date_time」にロックする年月日時分秒を書き込みます。書式は”yyyymmddhhmmss”です。(例: 20130108153025)

すでに他のユーザによってロックがかけられている場合は、ロック保持時間 Lock_Hold_Time_ [デフォルトで5分] 以上経過していれば上書きします。ロック保持時間 が経過していない場合は、「テーブルが他のユーザによってロックされている。」と表示して、[変更]、[削除]、[登録] ボタンを表示しないようにしています。従って、他のユーザによってロックされている場合はデータの更新ができません。(表示はできます。)

自分がロックした場合でも、ロック保持時間 が経過していると、他のユーザによってロックが上書きされてしまう場合があります。このため、データを更新する時にロックが上書きされているか調べて、上書きされている場合はデータ更新ができないようにしています。

処理画面を閉じる時に、ロックした年月日時分秒 が自分がロックしたものであれば、消しています。

レコード・ロックを行う場合は、テーブル・ロックする必要はありません。



5.2.4. レコード・ロック

処理画面において、[変更]、[削除] ボタンをクリックした時に、その処理画面に表示しているレコードに対してロックをかけます。その処理画面で主に使用するテーブルのレコードの カラム「lock_date_time」にロックする年月日時分秒を書き込みます。書式は”yyyymmddhhmmss”です。(例: 20130108153025)

すでに他のユーザによってロックがかけられている場合は、ロック保持時間 Lock_Hold_Time_ [デフォルトで5分] 以上経過していれば上書きします。ロック保持時間 が経過していない場合は、「レコードが他のユーザによってロックされている。」と表示して、変更・削除モードにすることはできません。

ロックをかけることができた場合でも、データの 登録・削除 を実行する時に、ロックが上書きされているか調べて、上書きされている場合はデータ登録・削除 ができないようにしています。

データ登録・削除 が完了した時に、ロックした年月日時分秒 を消しています。

5.2.5. データの暗号化・復号化

テーブルのカラム毎に暗号化するか否かを選択することができます。暗号化するデータは暗号化してからデータベースに書き込みます。暗号化してあるデータはデータベースからの読み込み時に復号化してから画面表示します。データベース上で暗号化されているためセキュリティ上も安心です。

暗号化／復号化は共通処理メソッド・クラス JFX_gyom_func. java のメソッド encrypt_text() / decrypt_text() を使用します。

暗号化したデータをデータベースにPOST 送信すると、それを受信したPHP プログラムで自動的に URLデコード されるため、URLエンコードしてから POST 送信する必要があります。

(例)

```

JFX_gyom_func func_obj = new JFX_gyom_func();
String wstr = func_obj.encrypt_text("データ"); //データを暗号化する
  
```

5. 「JFX 業務システム・フレームワーク」の構成

```
wstr = URLEncoder.encode(wstr, "UTF-8") + JFX_gyom.DELIMITER; //URL エンコードする  
wstr = func_obj.decrypt_text("暗号化されたデータ"); //暗号化されたデータを復号化する
```

暗号化／復号化 処理は Java 標準の `javax.crypto` と `javax.crypto.spec` パッケージを使用します。暗号化アルゴリズムは「Blowfish」を使用します。

暗号化キーは メイン・クラス `JFX_gyom.java` で 定数 `ENCRYPT_KEY_` として宣言しています。

暗号化キーを紛失すると、データベース内の暗号化したデータを復号化して読み取ることが一切できなくなります。従って、暗号化キーは通常のバックアップ保存をするとともに、さらに別途保存(管理者が筆記でメモするなど)しておいてください。

5.2.6. 数字入力テキスト・フィールド

数字のみ入力できるテキスト・フィールドです。クラス `FigTextField.java` で定義されています。

数字入力テキスト・フィールド は売上管理クラス `JFX_gyom_uri.java` の売上年月日 テキスト・フィールド および 売上明細テーブルビューの 数量、単価 フィールドに使用されています。

数字入力テキスト・フィールド は UI構築ツール「JavaFX Scene Builder」の標準コントロールにはないので作成できません。従って、`java` ファイルに直接コーディングします。

(例)

```
@FXML private static AnchorPane anchor_pane;  
private static FigTextField fig_txt = new FigTextField();  
  
anchor_pane.getChildren().add(fig_txt);  
fig_txt.setPrefSize(42,24); //テキスト・フィールドのサイズ(長さ、高さ)  
fig_txt.setLayoutX(657); //X位置  
fig_txt.setLayoutY(176); //Y位置
```

5.2.7. PHPプログラムを実行するメソッド

所定の URL にある PHPプログラムを実行します。URL は メイン・クラス `JFX_gyom.java` で定義されている定数 `String PHP_URL_` に設定します。デフォルトで `"http://localhost/JavaFX/JFX_gyom_framework_PHP/"` となっています。

PHPプログラムを実行するメソッドは、共通処理メソッド・クラス `JFX_gyom_func.java` で定義されているメソッド `exe_PHP_prog()` を使用します。

```
public String exe_PHP_prog(String PHP_prog_name, String arg_content)
```

引数として 実行するPHPプログラムのファイル名、PHPプログラムに POST送信するデータを指定してください。戻り値は、PHPプログラムから POST受信したデータ文字列です。

(例)

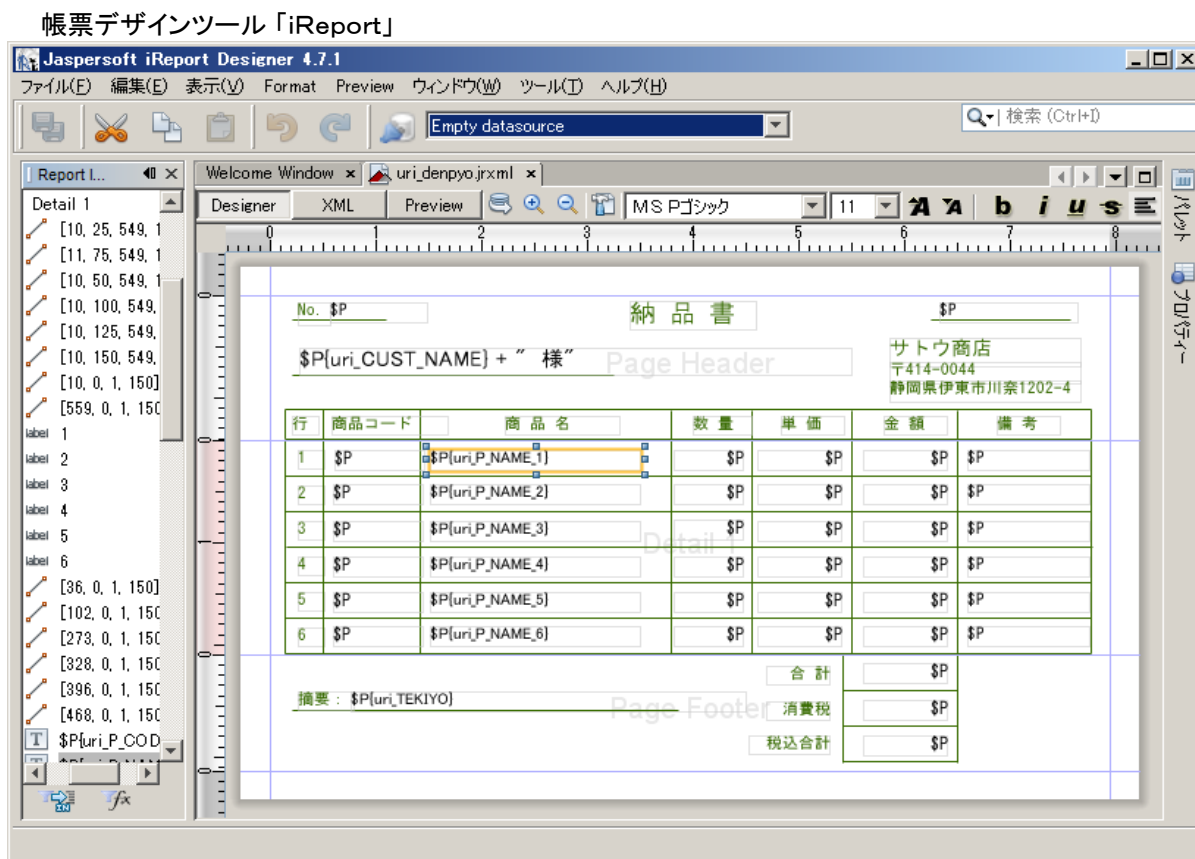
```
String content = "cust_no=" + cust_no + "&order_by=" + order_by + "&os_name=" + JFX_gyom.Os_name;  
  
JFX_gyom_func func_obj = new JFX_gyom_func();  
String wstr = func_obj.exe_PHP_prog("JFX_get_cust_data.php", content); ////指定した PHP プログラムを実行する  
  
if (wstr.startsWith("exe_PHP_prog(): ERROR")) { //エラー処理  
  
}  
else { //正常処理  
  
}
```

5.2.8. 帳票印刷(売上傳票)

帳票デザインツール「iReport」で帳票のレイアウト設計を行います。設計した帳票は jrxml ファイルに出力されます。帳票生成の Java ライブラリ「JasperReports」を使用して、jrxml ファイルをコンパイルして jasper ファイルを作成します。

売上管理クラス JFX_gyom_uri.ava の画面の売上傳票を印刷出力するメソッド prn_uri_data() 内で、「iReport」で作成した帳票記述ファイル uri_denpyo.jrxml をコンパイルして、帳票ファイル uri_denpyo.jasper を作成しています。この jasper ファイルを、ビューワで開きプレビューして、さらにプリンタに印刷出力するダイアログを表示しています。jasper ファイルを PDF に出力することもできます。

売上傳票へのデータの受け渡しは、画面のデータを連想配列にセットして、jasper ファイルにコンパイルする時にその連想配列をパラメータとして渡しています。



5.2.9. ユーザ・ランク

ログインしたユーザに対して以下のようなユーザ・ランクを設定しています。

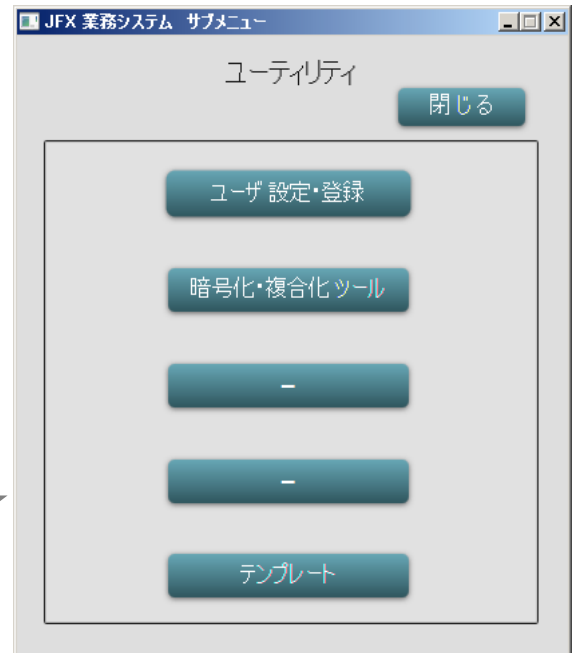
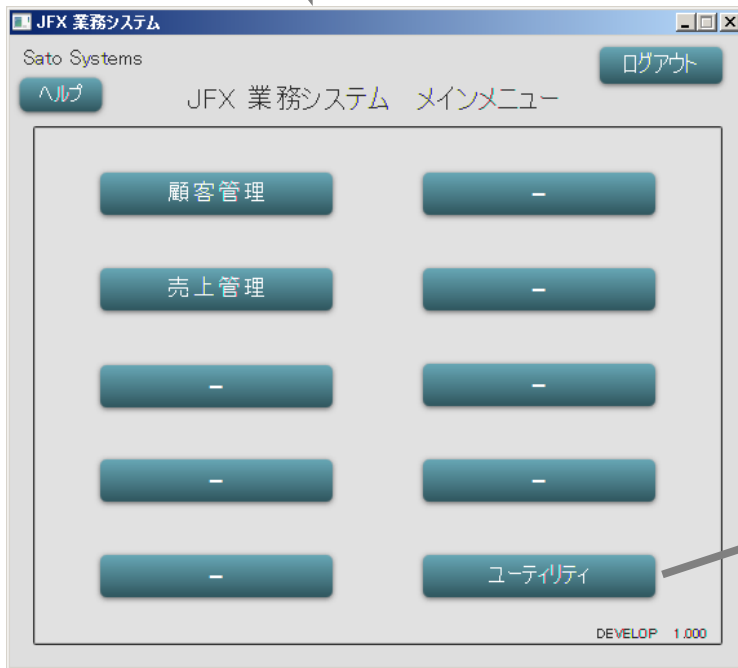
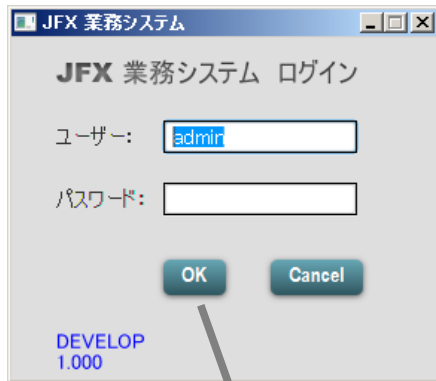
0=管理者、 1=読み書き可、 2=表示のみ

ログイン時に ユーザ設定テーブル fx_user_sest より ユーザ・ランク を取得し、グローバル変数 User_RANK_ にセットしています。

メイン・メニュー画面で、ユーザ・ランクが「0=管理者」でなければ、[ユーティリティ] サブメニューは表示しません。

顧客管理画面、売上管理画面を開く時に、ユーザ・ランクが「2=表示のみ」の場合は [変更] [新規] [削除] [登録] ボタン を非表示にして、データ変更ができないようにしています。

5.3. ログイン／メニュー



5.4. 顧客管理

JFX 業務システム

表示 表示 変更 新規 削除 顧客管理 顧客検索 登録 閉じる

顧客番号 10002 顧客区分 顧客区分BB 登録年月日 2012年06月13日

顧客名 顧客BBBBB TEL 0123-456-7890 請求書

フリガナ コキヤクBBBBB FAX 0123-456-7891 掛売

〒 100-0002 住所1 東京都台東区 売上累計 ¥ 200,655

住所2 6-45-8

メモ テスト用データ(2)xxx

2 / 9

顧客管理 → 顧客検索

顧客番号 10002 ⇒ 検索

顧客名 ⇒ 検索

顧客名カナ ⇒ 検索

電話番号 ⇒ 検索

閉じる

顧客管理画面に表示する顧客の行を選択して、[OK] ボタンをクリックしてください。

顧客番号	顧客名	顧客名カナ	電話番号	住所
10002	顧客BBBBB	コキヤクBBBBB	0123-456-7890	東京都台東区

OK

5.5. 売上管理

JFX 業務システム

表示 表示 変更 新規 削除 売上管理 売上検索 顧客検索 伝票印刷 登録 閉じる

顧客番号 10005 顧客区分 顧客区分AA 請求書
 顧客名 顧客EEEEExxx TEL 0123-666-7788 掛売 売上累計 ¥ 4,319,175

〒 100-0256 住所1 東京都千代田区xxxxxx メモ テスト用データ(5)xxxx
 住所2 5-9-12xx

売上番号 1027 取引区分 01 現金 行追加 行挿入 行削除 売上年月日 2009 年 01 月 13 日

行	売上区分	商品コード	商品名	数量	単価	金額	備考
1	01 区分A	00001	商品AAAAAA ¥3000	1	2,000	2,000	aaa
2	02 区分B	00002	商品BBBBBB ¥4000	2	2,500	5,000	bbb
3	03 区分C	00003	商品CCCCCC ¥5000	3	3,000	9,000	ccc
						合計	121,100
						消費税	6,055 (税率 5%)
						税込合計	127,155

摘要 EEEEEEEEE777

検索項目 検索解除

売上伝票

No. 1027 納品書 2009年01月13日

顧客EEEEExxx 様 サトウ商店
 〒414-0044
 静岡県伊東市川奈1202-4

行	商品コード	商品名	数量	単価	金額	備考
1	00001 商	商品AAAAAA ¥3000	1	2,000	2,000	aaa
2	00002 商	商品BBBBBB ¥4000	2	2,500	5,000	bbb
3	00003 商	商品CCCCCC ¥5000	3	3,000	9,000	ccc
4						
5						
6						
					合計	121,100
					消費税	6,055
					税込合計	127,155

摘要: EEEEEEEEE777

Page 1 of 1

5.6. ユーザ設定・登録

行	ユーザー名	ユーザーランク	パスワード	メモ
1	admin	0 管理者		ああ AAAA-EEEE
2	test_0	0 管理者	test_0x	AA+BB+CC
3	test_1	1 読み書き	test_1x	メモ あああああああ *
4	test_2	2 読のみ		メモ AAAAAA *

ログインするユーザを、ユーザ設定テーブル「fx_user_set」に登録します。既存ユーザの変更・削除、新規ユーザの追加ができます。ユーザ名には日本語を使用しないでください。ユーザ・ランクは「0＝管理者、1＝読み書き可、2＝表示のみ」となります。パスワードは設定しないこともできます。[登録] ボタンをクリックすると、データベースに画面の内容を書き込みします。

5.7. 暗号化・復号化ツール

データベース「MySQL」に書き込んだ暗号化データは、データベース管理ツール「phpMyAdmin」などで表示しても暗号化されているため判読することができません。このため、本ツールを使用して復号化してください。暗号化されたデータをコピーして、本ツールに貼り付け [復号化] ボタンをクリックしてください。復号化されたデータが表示されます。また、暗号化したデータをデータベース管理ツールなどで直接データベースに書き込む場合は、本ツールで暗号化したデータをコピーして貼り付けしてください。

5.8. テンプレート



新しいアプリケーションを作成する場合、その処理画面のひな型となるテンプレートです。「顧客管理」、「売上管理」の処理画面と大きく異なる処理画面の場合、このテンプレートを使用してください。

以下の3個のファイルをコピーして、別のファイル名で貼り付けてください。

JFX_gyom_template.java、 JFX_gyom_template.fxml、 JFX_gyom_template.css

貼り付けしたファイルのファイル名を新しい処理画面のものに変更してください。拡張子はそのままにしておいてください。

5.9. PHP プログラム

[初期設定ファイル]

各PHPプログラムでは、起動時に初期設定ファイル `mysql_server_data.txt` を読み込んでいます。この初期設定ファイルは設定項目をカンマで区切ったテキストファイルで、PHPプログラムと同じ場所に置きます。書式は以下のとおりです。

データベースのIPアドレス、 データベースのユーザ名、 パスワード、 データベース名、 文字コード、 POST送受信するデータのデリミタ

(例) `localhost, root, xxxxx, jfx_gyom_db, UTF8, #%`

データベースのIPアドレス: データベース MySQL を置いているサーバの IPアドレス。PHPプログラムと同じパソコン内にある場合は「localhost」とします。(例) 192.168.1.20

データベースのユーザ名: データベース MySQL を使用できるユーザ・アカウント

パスワード: データベースのユーザが MySQL を使用するためのパスワード

文字コード: MySQL に設定されている文字コード (UTF8)

デリミタ: JavaFX側から あるいは PHPプログラム側から POST送受信するデータのフィールドを区切るための文字列。通常データとして使われない記号などを文字列として設定します。もし、データ内に同じ文字列が存在した場合は、POST送受信後の処理でエラーになります。そのおそれがある場合は、送受信前にデリミタと同じ文字列を他の文字列に変換し、送受信後に元に戻すことが必要になります。

(例) “AAAA#%_BBBB#%_123456#%_”

[PHPプログラムの戻り値]

5. 「JFX 業務システム・フレームワーク」の構成

正常に処理した場合：処理した結果を示す文字列を返す場合とデータベースから取得したデータを返す場合があります。
エラーの場合：「PHPプログラム名 + ” NG ” + エラーの内容」を返します。

PHPプログラムを呼び出した JavaFX 側では、この戻り値を取得して、正常時の処理／エラー時の処理を行ってください。