

モジュール設計書（ver1）

モジュールNo	ファイル名	機能	備考
001	LoginServlet.java	ログイン・ログアウト処理	Servletクラス
002	ItemManageServlet.java	商品管理処理を受け付ける	Servletクラス
003	OrderManageServlet.java	注文管理処理を受け付ける	Servletクラス
004	ResultServlet.java	処理結果を表示する	Servletクラス
005	UserAuth.java	ログイン処理共通	共通クラス
006	ItemManage.java	商品管理共通	共通クラス
007	OrderManage.java	注文管理共通	共通クラス
008	UserBean.java	ユーザー情報を格納	Beanクラス
009	ItemBean.java	商品情報を格納	Beanクラス
0010	OrderBean.java	注文情報を格納	Beanクラス
0011	LoginDao.java	ユーザー情報DB処理用	Daoクラス
0012	ItemDao.java	商品情報DB処理用	Daoクラス
0013	OrderDao.java	注文情報DB処理用	Daoクラス
0014	StockDao.java	在庫情報DB処理用	Daoクラス

モジュールNo	001
クラス名	LoginServlet
目的	ログイン・ログアウトの処理等の処理を受け付ける
アクセス修飾子	public
そのほかの修飾子	なし
継承/実装	HttpServletを継承
パッケージ	パッケージ: kensyu
インポート	import java.io.IOException; import java.util.ArrayList; import javax.servlet.RequestDispatcher; import javax.servlet.ServletException; import javax.servlet.annotation.WebServlet; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; import javax.servlet.http.HttpSession;

メンバ変数定義

名称	内容	アクセス修飾子	その他の修飾子	データ型	初期値
serialVersionUID	シリアライズ	private	static, final	long	1L

コンストラクタ定義
なし

メソッド定義

名称	doGet
目的	HTTPリクエストのGETメソッドに対する処理を実行する
アクセス修飾子	public
そのほかの修飾子	#NAME?
戻り値	void
引数	HttpServletRequest request: HTTPリクエスト HttpServletResponse response: HTTPレスポンス
例外	ServletException, IOException
動作	*1. doPostメソッドを呼び出し、GET、POSTどちらのメソッドでも同じ処理が実行されるようにする。 引数: request ... 引数で渡されたrequest response ... 引数で渡されたresponse*

メソッド定義

名称	doPost
目的	HTTPリクエストのPOSTメソッドに対する処理を実行する
アクセス修飾子	public
そのほかの修飾子	#NAME?
戻り値	void
引数	HttpServletRequest request: HTTPリクエスト HttpServletResponse response: HTTPレスポンス
例外	ServletException, IOException
動作	1.requestのsetCharacterEncodingメソッドを使用してHTTPリクエストの文字コードを設定する。 引数: env ... "UTF-8" 2.requestのgetParameterメソッドを利用して押されたボタンの情報を取得し、ローカル変数 btn へ格納する。 引数: name ... "submit" ローカル変数名 ... btn 3. 遷移のためのリクエスト・ディスパッチャーを用意する。 ローカル変数名 ... rd 初期値 ... null 4. セッションスコープを用意する。 ローカル変数名 ... session 初期値 ... HttpSession オブジェクト (requestのgetSessionメソッドの戻り値) 1.入力処理 1.1 Inputクラスを実行し、「user_id」と「pwd」を変数へ代入する 2 ログインチェック処理 1.1 ログインチェックオブジェクトに入力値「user_id」と入力値「pwd」を渡し、ログインチェック結果を取得する 1.1-a ログインチェック結果が「真」の場合 1.1.1 ログイン認証処理を行う。 1.1-b ログインチェック結果が「偽」の場合 1.1.1 エラー処理 1 を行う 3 ログイン認証処理 1.1 Daoオブジェクトに入力値「user_id」と入力値「pwd」を渡し、DBを参照する。そして結果をEntityオブジェクトで取得する。 1.1-a Entityオブジェクトが存在する場合 1.1.1 操作選択画面処理を実行する。 1.1-b Entityオブジェクトが「null」の場合 1.1.1 エラー処理 1 を行う 4 操作選択画面処理 1.1 アクセス権限確認 1.1-a アクセス権限確認の値が「1」の場合 1.1.1 操作選択画面(在庫管理専用)を表示する 1.1-b アクセス権限確認の値が「2」の場合 1.1.1 操作選択画面(販売管理専用)を表示する 5.ログアウト処理 1.1 セッションを削除する 1.2 ホーム画面に遷移

モジュールNo	002
クラス名	ItemManageServlet
目的	商品管理処理を受け付ける
アクセス修飾子	public
その他の修飾子	なし
継承/実装	HttpServletを継承
パッケージ	kensyu
インポート	java.io.IOException java.util.ArrayList javax.servlet.RequestDispatcher javax.servlet.ServletException javax.servlet.annotation.WebServlet javax.servlet.http.HttpServlet javax.servlet.http.HttpServletRequest javax.servlet.http.HttpServletResponse

メンバ変数定義

名称	内容	アクセス修飾子	その他の修飾子	データ型	初期値
serialVersionUID	シリアライズ	private	static、final	long	1L

コンストラクタ定義

なし

メソッド定義

名称	doPost
目的	HTTPリクエストのPOSTメソッドに対する処理を実行する
アクセス修飾子	public
その他の修飾子	@Override
戻り値	void
引数	HttpServletRequest request : HTTPリクエスト HttpServletResponse response : HTTPレスポンス
例外	ServletException IOException
動作	リクエストの文字コードをUTF-8 itemManageクラスのインスタンスを作成 ローカル変数名・・・itemManage 1.1「item_id」、「item_name」、「item_img」、「order_quant」、「price」を変数に代入 1.2「登録」ボタンを押した場合、商品登録処理を実行 2商品登録処理 1.1商品チェックオブジェクトに「item_id」の値を渡し、DBを参照する。 1.1-a「item_id」が存在しない場合 1.1.1商品登録処理を行う 1.1-b「item_id」が使用されている場合 1.1.1エラー処理1を行う 1.2item_infoテーブルに「item_id」、「item_name」、「item_img」、「price」の数値を登録 1.3stock_infoテーブルに「item_id」、「stock_quant」の数値を登録 1.4処理が完了次第「登録完了」とポップアップ表示 3画面変遷処理 1.1「戻る」ボタンを押した場合、各管理者操作画面へ変遷

モジュールNo	003
クラス名	OrderManageServlet
目的	注文管理処理を受け付ける
アクセス修飾子	public
そのほかの修飾子	なし
継承/実装	HttpServletを継承
パッケージ	kensyu
インポート	java.io.IOException java.util.ArrayList javax.servlet.RequestDispatcher javax.servlet.ServletException javax.servlet.annotation.WebServlet javax.servlet.http.HttpServlet javax.servlet.http.HttpServletRequest javax.servlet.http.HttpServletResponse

メンバ変数定義

名称	内容	アクセス修飾子	その他の修飾子	データ型	初期値
serialVersionUID	シリアライズ	private	static、final	long	1L

コンストラクタ定義

なし

メソッド定義

名称	doPost
目的	HTTPリクエストのPOSTメソッドに対する処理を実行する
アクセス修飾子	public
そのほかの修飾子	@Override
戻り値	void
引数	HttpServletRequest request : HTTPリクエスト HttpServletResponse response : HTTPレスポンス
例外	ServletException IOException
動作	1.入力処理 1.1「order_id」、「user_id」、「item_id」、「order_quant」を変数に代入 2.IDチェック処理 1.1 注文チェックオブジェクトに「order_id」の値を渡し、DBを参照する。 1.1-a「order_id」が重複する場合 1.1.1 エラー処理1を行う 1.1-b「order_id」が重複場合 1.1.1 処理を続行 1.2 商品チェックオブジェクトに「item_id」の値を渡し、DBを参照する。 1.1-a「item_id」が登録されている場合 1.1.1 注文登録処理を行う 1.1-b「item_id」が登録されていない場合 1.1.1 エラー処理を行う 3. 注文登録処理 1.1 order_infoテーブルに「order_id」、「user_id」、「item_id」、「order_quant」の値を追加 1.2 stock_infoのstock_quantの数値を更新 1.2.1 stock_quantの値を取得 1.2.2 取得した値から入力された「order_quant」の数を引く 1.2.2-a 結果がマイナスになった場合 1.2.2.1 エラー処理を行う 1.2.2-b 結果が整数の場合 1.2.2.2 処理を続行 1.2.3 結果の値をstock_quantに挿入 1.3 処理が完了後、「登録完了」とポップアップ表示

モジュールNo	004
クラス名	ResultServlet
目的	処理結果を表示する
アクセス修飾子	public
その他の修飾子	なし
継承/実装	HttpServletを継承
パッケージ	kensyu
インポート	<pre>import java.io.IOException; import javax.servlet.ServletException; import javax.servlet.annotation.WebServlet; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse;</pre>

メンバ変数定義

名称	内容	アクセス修飾子	その他の修飾子	データ型	初期値
serialVersionUID	シリアライズ	private	static、final	long	1L

コンストラクタ定義

なし

メソッド定義

名称	doPost
目的	HTTPリクエストのPOSTメソッドに対する処理を実行する
アクセス修飾子	public
その他の修飾子	@Override
戻り値	void
引数	<pre>HttpServletRequest request : HTTPリクエスト HttpServletResponse response : HTTPレスポンス</pre>
例外	<pre>ServletException IOException</pre>
動作	処理結果を取得し、適切な結果ページに遷移

モジュールNo	005
クラス名	Login
目的	ユーザー認証処理を行う
アクセス修飾子	public
その他の修飾子	なし
継承/実装	なし
パッケージ	kensyu
インポート	import java.sql.SQLException;

メンバ変数定義

コンストラクタ定義
なし

メソッド定義

名称	loginCheck
目的	ユーザーIDとパスワードを検証する
アクセス修飾子	public
その他の修飾子	static
戻り値	LoginUserBean
引数	String userId, String password
例外	SQLException
動作	<div>1. LoginUserBean型のローカル変数beanを宣言し、nullを代入する。(戻り値用の変数を宣言する。)</div> <div>2. ログインID及びパスワード文字列をチェックする。</div> <div>条件1:[userId.length() > 20] --- true (userId が 20文字を超える場合)--- 2-1. beanを戻り値として返す。</div> <div>条件2:[password.matches("[a-zA-Z0-9_\$]{8,20}\$")] --- true (password が 半角英数字記号(\$)8文字以上-20文字以内では無い場合)--- 2-2. beanを戻り値として返す。</div> <div>3. LoginDaoオブジェクトを生成し、ローカル変数daoへ代入する。</div> <div>4. daoのselectUserメソッドを利用してユーザ情報を取得し、beanへ格納する。 引数:userId : userId 引数:password : password</div> <div>5. beanを戻り値として返す。</div>

モジュールNo	006
クラス名	ItemManage
目的	商品管理処理を行う
アクセス修飾子	public
そのほかの修飾子	なし
継承/実装	なし
パッケージ	kensyu
インポート	import java.sql.SQLException; import java.util.ArrayList;

メンバ変数定義

コンストラクタ定義
public ItemManage()

メソッド定義	
名称	registerItem
目的	新しい商品を登録する
アクセス修飾子	public
そのほかの修飾子	static
戻り値	boolean
引数	ItemBean item
例外	SQLException
動作	1.1 商品チェックオブジェクトに「item_id」の値を渡し、DBを参照する。 1.1-a 「item_id」が存在しない場合 1.1.1 商品登録処理を行う 1.1-b 「item_id」が使用されている場合 1.1.1 エラー処理を行う 1.2 item_infoテーブルに「item_id」、「item_name」、「item_img」、「price」の取値を登録 1.3 stock_infoテーブルに「item_id」、「stock_quant」の取値を登録 1.4 処理の完了次第、「登録完了」とポップアップ表示

メソッド定義	
名称	getOrderList
目的	注文一覧を取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	ArrayList<OrderBean>
引数	
例外	SQLException
動作	1.商品一覧表示処理 1.1 item_infoテーブルの各カラム値を取得 1.1-a DBにデータがない場合 1.1.1 エラー処理を実行 1.1-b DBにデータがある場合 1.1.1 処理を終行 1.2 item_infoテーブルのitem_idとFKしているstock_infoテーブルのstock_quantの値を取得 1.3 表のそれぞれ対応する欄に取得した値を挿入

メソッド定義	
名称	updateOrder
目的	注文情報を更新する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	boolean
引数	OrderBean order
例外	SQLException
動作	指定された商品の情報をデータベースで更新する 1.1「変更」ボタンが押された場合、編集対象のデータを表から選択 1.2 選択された値に付随するテーブルデータを変数へ代入 1.3 商品編集画面へ遷移 1.3 商品編集削除機能の商品編集処理を実行

メソッド定義	
名称	deleteOrder
目的	注文を削除する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	boolean
引数	String orderId
例外	SQLException
動作	1.1「削除」ボタンが押された場合、削除対象のデータを表から選択 1.2 選択された値に付随するテーブルデータを変数へ代入 1.3 商品編集削除機能の商品削除処理を実行

モジュール名	007
クラス名	OrderManage
目的	注文管理処理を行う
アクセス修飾	public
そのほかの修	なし
開発者	なし
パッケージ	hensyu
インポート	java.sql*, java.util.ArrayList

メンバ変数定義

コンストラクタ定義
public OrderManage()

メソッド定義	
名称	registerOrder
目的	注文を登録する
アクセス修飾	public
そのほかの修	なし
戻り値	boolean
引数	OrderBean order
例外	SQLException
動作	1.1 order_infoテーブルに「order_id」、「user_id」、「item_id」、「order_quant」の値を挿加 1.2 stock_infoのstock_quantの値を変更 1.2.1 stock_quantの値も取得 1.2.2 取得した値から入力された「order_quant」の取を引く 1.2.2-a 結果がマイナスになった場合 1.2.2.1 エラー処理を行う 1.2.2-b 結果が整数の場合 1.2.2.2 処理を続行 1.2.3 結果の値もstock_quantに挿入 1.3 処理が完了後、「登録完了」とポップアップ表示

メソッド定義	
名称	getOrderList
目的	注文一覧を取得する
アクセス修飾	public
そのほかの修	なし
戻り値	ArrayList<OrderBean>
引数	
例外	SQLException
動作	1.注文一覧表示処理 1.1 order_infoテーブルの各カラム値を取得 1.1-a DBにデータがない場合 1.1.1 エラー処理を続行 1.1-b DBにデータがある場合 1.1.1 処理を続行 1.2 表のそれぞれ対応する欄に取得した値を挿入

メソッド定義	
名称	updateOrder
目的	注文情報を更新する
アクセス修飾	public
そのほかの修	なし
戻り値	boolean
引数	OrderBean order
例外	SQLException
動作	指定された注文の情報をデータベースで更新する 1.1 各販売機能の編集処理で取得した変数の一覧を表示 1.2 編集対象の値を変更 1.3 編集した値を変数へ代入、DBの対応したデータの値を上書き 1.3-a 編集データが商品情報の場合 1.3.1 処理の続行 1.3-b 編集データが注文情報の場合 1.3.1 編集前のorder_quantから編集後のorder_quantの値を引き変数に代入 1.3.2 stock_infoのstock_quantに 変数の値を加算し、結果の値をstock_quantに上書き 1.4 「編集完了」ポップアップ表示 1.5 各一覧画面へ遷移

メソッド定義	
名称	deleteOrder
目的	注文を削除する
アクセス修飾	public
そのほかの修	なし
戻り値	boolean
引数	String orderId
例外	SQLException
動作	指定された注文をデータベースから削除する 1.1 各販売機能の削除処理で取得した変数の一覧をポップアップ表示 1.2 ポップアップ上で削除、キャンセルの確認 1.2-a 削除が選択された場合 1.2.1 対象のテーブルのカラムを削除 1.2.2 「削除完了」ポップアップ表示 1.2.3 削除処理のポップアップを閉じる 1.2-b キャンセルが選択された場合 1.2.1 キャンセルしましたとポップアップ表示 1.2.2 削除処理のポップアップを閉じる

モジュール名	008
クラス名	ILoginUserBean
説明	ログインユーザー情報を保持する
アクセス修飾子	public
そのほかの修飾子	なし
継承関係	java.io.Serializableを継承
パッケージ	com.hoge
インポート	java.io.Serializable

メンバー変数 定義			
変数	内容	アクセス修飾	その他の修飾
user_id	ユーザID	private	String
pwd	パスワード	private	String
name	ユーザ名	private	String
auth	権限	private	int

コンストラクタ定義	
アクセス修飾子	public
説明	String user_id, String pwd, String username, int Auth
動作	1.メンバー変数 userID に値を設定する。 2.メンバー変数 password に値を設定する。 3.メンバー変数 name に値を設定する。

メソッド定義	
変数	getuser_id
説明	ユーザIDを取得するメソッド
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	String
例外	なし
動作	1.メンバー変数user_idを戻り値として返す。

メソッドの定義	
変数	setuser_id
説明	ユーザIDを設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
例外	String user_id
動作	1.メンバー変数user_idへ引数のuser_idを代入する。

メソッドの定義	
変数	getpwd
説明	パスワードを取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	String
例外	なし
動作	1.メンバー変数pwdを戻り値として返す。

メソッドの定義	
変数	setpwd
説明	パスワードを設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
例外	String password
動作	1.メンバー変数pwdへ引数のpwdを代入する。

メソッドの定義	
変数	getuser_name
説明	ユーザ名を取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	String
例外	なし
動作	1.メンバー変数user_nameを戻り値として返す。

メソッドの定義	
変数	setuser_name
説明	ユーザ名を設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
例外	String userName
動作	1.メンバー変数user_nameへ引数のuser_nameを代入する。

メソッドの定義	
変数	getauth
説明	ユーザ権限を取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	int
例外	なし
動作	1.メンバー変数authを戻り値として返す。

メソッドの定義	
変数	setauth
説明	ユーザ権限を設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	int userAuth
例外	なし
動作	1.メンバー変数authへ引数のauthを代入する。

モジュールNo	009
クラス名	ItemBean
目的	商品情報を格納する
アクセス修飾子	public
そのほかの修飾子	なし
継承関係	java.io.Serializableも実装
パッケージ	kenyu
インポート	java.io.Serializable

メンバー変数定義

変数	内容	アクセス修飾子	その他の修飾子	データ型	初期値
item_id	商品のID	private		String	
item_name	商品名	private		String	
price	単価	private		int	
stock_quant	在庫数	private		int	

コンストラクタ定義

アクセス修飾子	public
引数	String item_id, String item_name, int price, int stock
動作	1. メンバ変数 item_id に空白を設定する。 2. メンバ変数 item_name に空白を設定する。 3. メンバ変数 price に 0 を設定する。 4. メンバ変数 stock_quant に 0 を設定する。

メソッド定義

変数	getItem_id
目的	商品のIDを取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	String
引数	なし
例外	なし
動作	1. メンバ変数 item_id を戻り値として返す。

メソッドの定義

変数	setItem_id
目的	商品のIDを設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
引数	String itemId
例外	なし
動作	1. メンバ変数 item_id へ引数の item_id を代入する。

メソッドの定義

変数	getItem_name
目的	商品名を取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	String
引数	なし
例外	なし
動作	1. メンバ変数 item_name を戻り値として返す。

メソッドの定義

変数	setItem_name
目的	商品名を設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
引数	String itemName
例外	なし
動作	1. メンバ変数 item_name へ引数の item_name を代入する。

メソッドの定義

変数	getPrice
目的	単価を取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	int
引数	なし
例外	なし
動作	1. メンバ変数 item_price を戻り値として返す。

メソッドの定義

変数	setPrice
目的	単価を設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
引数	int price
例外	なし
動作	1. メンバ変数 item_price へ引数の price を代入する。

メソッドの定義

変数	getstock
目的	在庫数を取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
引数	int
例外	なし
動作	1. メンバ変数 item_stock を戻り値として返す。

メソッドの定義

変数	setstock
目的	在庫数を設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
引数	int stock
例外	なし
動作	1. メンバ変数 item_stock へ引数の stock を代入する。

モジュールNo	010
クラス名	HistoryBean
目的	購入履歴情報を格納する
アクセス修飾子	public
そのほかの修飾子	なし
継承/実装	java.io.Serializableを実装
パッケージ	kenyuu
インポート	java.io.Serializable

メンバ変数定義

変名	内容	アクセス修飾子	その他の修飾子	データ型	初期値
order_id	注文ID	private		String	
user_id	ユーザーID	private		String	
item_id	商品ID	private		String	
order_quant	数量	private		int	

コンストラクタ定義

アクセス修飾子	public
引数	String order_id, String user_id, String item_id, int order_quant,
動作	1. メンバ変数 order_id に空白を設定する。 2. メンバ変数 user_id に空白を設定する。 3. メンバ変数 item_id に空白を設定する。 4. メンバ変数 order_quant に空白を設定する。

メソッド定義

変名	getOrder_id
目的	注文IDを取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	String
引数	なし
例外	なし
動作	1. メンバー変数order_idを戻り値として返す。

メソッドの定義

変名	setOrder_id
目的	注文IDを設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
引数	String orderId
例外	なし
動作	1. メンバー変数 order_idへ引数のorder_idを代入する。

メソッドの定義

変名	getUser_id
目的	ユーザーIDを取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	String
引数	なし
例外	なし
動作	1. メンバー変数user_idを戻り値として返す。

メソッドの定義

変名	setUser_id
目的	ユーザーIDを設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
引数	String userId
例外	なし
動作	1. メンバー変数 user_idへ引数のuser_idを代入する。

メソッドの定義

変名	getItem_id
目的	商品IDを取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	String
引数	なし
例外	なし
動作	1. メンバー変数item_idを戻り値として返す。

メソッドの定義

変名	setItem_id
目的	商品IDを設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
引数	String itemId
例外	なし
動作	1. メンバー変数 item_idへ引数のitem_idを代入する。

メソッドの定義

変名	getOrder_quant
目的	数量を取得する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
引数	int
例外	なし
動作	1. メンバー変数order_quantを戻り値として返す。

メソッドの定義

変名	setOrder_quant
目的	数量を設定する
アクセス修飾子	public
そのほかの修飾子	なし
戻り値	void
引数	int quantity
例外	なし
動作	1. メンバー変数 order_quantへ引数のorder_quantを代入する。

モジュールNo	011
クラス名	LoginDao
目的	ユーザー情報のDB処理を行う
アクセス修飾子	public
その他の修飾子	なし
継承/実装	HttpServletを継承
パッケージ	kensyu
インポート	java.sql.Connection java.sql.DriverManager java.sql.PreparedStatement java.sql.ResultSet java.sql.SQLException

メンバ変数定義					
名称	内容	アクセス修飾子	その他の修飾子	データ型	初期値
con	DB接続用	private		Connection	null
ps	SQL実行用	private		PreparedStatement	null
rs	SQL実行結果用	private		ResultSet	null
bean	ユーザ情報用	private		LoginUserBean	null

コンストラクタ定義

メソッド定義	
名称	checkUser
目的	ユーザー認証を行う
アクセス修飾子	public
その他の修飾子	
戻り値	LoginUserBean
引数	: String userId, Sting password
例外	SQLException
動作	<pre>--- <try> --- 1. ClassクラスのforNameメソッドを使用し、JDBCドライバをロードする。 forNameメソッドの引数 : com.mysql.jdbc.Driver 2. DriverManagerクラスのgetConnectionメソッドを使用し、Connectionオブジェクトを生成し、メンバ変数conへ代入する。 getConnectionメソッドの第1引数 : "jdbc:mysql://localhost:3306/abc" 接続文字列 getConnectionメソッドの第2引数 : "root" 接続ユーザ getConnectionメソッドの第3引数 : "root" 接続パスワード 3. conのprepareStatementメソッドを利用し、SQL文を生成する。そしてメンバ変数psへ代入する。 prepareStatementの引数 : "select name from user_info where id = ? and pass = ?" 4. psのsetStringメソッドを利用してユーザIDを設定する。 setStringメソッドの第1引数 : 1 ユーザID setStringメソッドの第2引数 : userId ユーザID 5. psのsetStringメソッドを利用してパスワードを設定する。 setStringメソッドの第1引数 : 2 パスワード setStringメソッドの第2引数 : password パスワード 6. psのexecuteQueryメソッドを使用し、SQLを実行する。実行結果をメンバ変数rsへ代入する。 7. rsから実行結果を取り出す。(rs.next()を実行し、falseになるまで繰り返し、) 繰り返し条件 : [rs.next()の実行結果がfalseになるまで] ---rs.next()の実行結果がtrueの場合--- 7-1. LoginUserBeanクラスのインスタンスを生成し、メンバ変数beanへ代入する。 7-2. beanのsetUserIdメソッドを使用し、ユーザIDを設定する。 setUserIdメソッドの引数 : userId 7-3. beanのsetNameメソッドを使用し、ユーザ名を設定する。 setNameメソッドの引数 : name --- < catch > --- 8. 例外処理(ClassNotFoundException) 8-1. 例外を表示する --- <finally> --- 9. 終了処理 9-1. データベースとの接続を解除する。(closeメソッドを実行させる。) 10. ユーザ情報 (bean) を返す。</pre>

モジュールNo	012
クラス名	ItemDao
目的	商品情報のDB操作を行う
アクセス修飾子	public
その他の修飾子	なし
継承/実装	なし
パッケージ	kensyu
インポート	import java.sql.*; import java.util.List; import java.util.ArrayList; import com.inventory.bean.HistoryBean; import com.inventory.util.DBConnection;

メンバ変数定義

コンストラクタ定義
public ItemDao() {}

メソッド定義	
名称	getItemList
目的	商品一覧を取得する
アクセス修飾子	public
その他の修飾子	@Override
戻り値	List<ItemBean>
引数	
例外	SQLException
動作	DBに接続 商品一覧を取得するSQLを実行 結果をリストに格納して返す

名称	insertItem
目的	商品一覧を登録する
アクセス修飾子	public
その他の修飾子	boolean
戻り値	ItemBean item
引数	
例外	SQLException
動作	DBに接続 商品を登録するSQLを実行 成功時はtrue、失敗時はfalseを返す

名称	getItem
目的	特定の商品情報を取得する
アクセス修飾子	public
その他の修飾子	@Override
戻り値	List<ItemBean>
引数	String itemId
例外	SQLException
動作	DBに接続 商品一覧を取得するSQLを実行 結果をリストに格納して返す

名称	updateItem
目的	商品情報を更新する
アクセス修飾子	public
その他の修飾子	boolean
戻り値	ItemBean item
引数	
例外	SQLException
動作	DBに接続 商品情報を更新するSQLを実行 成功時はtrue、失敗時はfalseを返す

名称	deleteItem
目的	商品を削除する
アクセス修飾子	public
その他の修飾子	boolean
戻り値	String itemId
引数	
例外	SQLException
動作	DBに接続 商品を削除するSQLを実行 成功時はtrue、失敗時はfalseを返す

モジュールNo	013
クラス名	OrderDao
目的	注文情報のDB操作を行う
アクセス修飾子	public
そのほかの修飾子	なし
継承/実装	なし
パッケージ	kensyu
インポート	import java.sql.*; import java.util.List; import java.util.ArrayList; import com.inventory.bean.HistoryBean; import com.inventory.util.DBConnection;

メンバ変数定義

コンストラクタ定義
public OrderDao() {}

メソッド定義	
名称	getOrderList
目的	注文一覧を取得する
アクセス修飾子	public
そのほかの修飾子	
戻り値	List<HistoryBean>
引数	
例外	SQLException
動作	DBに接続 商品一覧を取得するSQLを実行 結果をリストに格納して返す

名称	getOrder
目的	特定の注文情報を取得する
アクセス修飾子	public
そのほかの修飾子	
戻り値	HistoryBean
引数	String orderId
例外	SQLException
動作	DBに接続 指定されたorderIdの注文情報を取得するSQLを実行 結果をHistoryBeanに格納して返す

名称	insertOrder
目的	注文を登録する
アクセス修飾子	public
そのほかの修飾子	
戻り値	boolean
引数	HistoryBean order
例外	SQLException
動作	DBに接続 注文を登録するSQLを実行 成功時はtrue、失敗時はfalseを返す

名称	updateOrder
目的	注文情報を更新する
アクセス修飾子	public
そのほかの修飾子	
戻り値	boolean
引数	HistoryBean order
例外	SQLException
動作	DBに接続 注文情報を更新するSQLを実行 成功時はtrue、失敗時はfalseを返す

名称	deleteOrder
目的	注文を削除する
アクセス修飾子	public
そのほかの修飾子	
戻り値	boolean
引数	HistoryBean order
例外	SQLException
動作	DBに接続 注文を削除するSQLを実行 成功時はtrue、失敗時はfalseを返す

モジュールNo	014
クラス名	StockDao
目的	在庫情報のDB操作を行う
アクセス修飾子	public
そのほかの修飾子	なし
継承/実装	なし
パッケージ	kensyu
インポート	import java.sql.*; import com.inventory.util.DBConnection;

メンバ変数定義

コンストラクタ定義

```
public StockDao() {}
```

メソッド定義

名称	updateStock
目的	在庫数を更新する
アクセス修飾子	public
そのほかの修飾子	
戻り値	boolean
引数	String itemId, int quantity
例外	SQLException
動作	DBに接続 在庫数を更新するSQLを実行 成功時はtrue、失敗時はfalseを返す

名称	getStock
目的	特定商品の在庫数を取得する
アクセス修飾子	public
そのほかの修飾子	
戻り値	int
引数	String itemId
例外	SQLException
動作	DBに接続 指定されたitemIdの在庫数を取得するSQLを実行 在庫数を返す

モジュールNo	013
クラス名	ErrorHandler
目的	エラー処理を行う
アクセス修飾子	public
そのほかの修飾子	なし
継承/実装	なし
パッケージ	kensyu
インポート	import java.util.logging.Logger; import java.util.logging.Level; import javax.servlet.http.HttpServletRequest;

メンバ変数定義

```
private static final Logger logger = Logger.getLogger(ErrorHandler.class.getName());
```

コンストラクタ定義

```
private ErrorHandler() {} // インスタンス化を防ぐため
```

メソッド定義

名称	displayErrorMessage
目的	エラーメッセージを表示する
アクセス修飾子	public
そのほかの修飾子	static
戻り値	void
引数	HttpServletRequest request, String message
例外	
動作	リクエスト属性にエラーメッセージを設定

名称	logError
目的	エラーログを記録する
アクセス修飾子	public
そのほかの修飾子	static
戻り値	void
引数	String message, Exception e
例外	
動作	ロガーを使用してエラーメッセージと例外スタックトレースを記録

名称	handleException
目的	例外を処理する
アクセス修飾子	public
そのほかの修飾子	static
戻り値	void
引数	HttpServletRequest request, Exception e
例外	
動作	エラーメッセージを表示 エラーログを記録