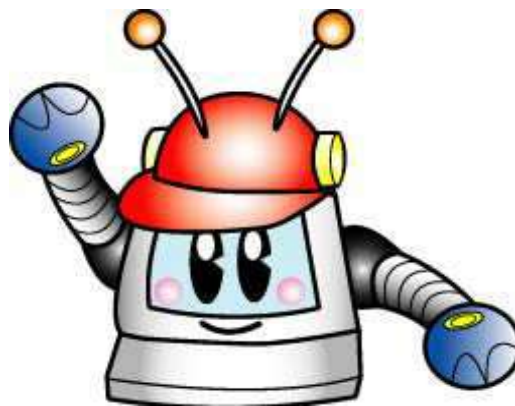


Webロボパターン別概要書

いい仕事をするには、いい道具が必要！



- Javaソースを自動生成
- JSP、アプレット、pdf作表パターン
- 高い生産性と保守性

会計開発をユーチューブに動画掲載中

<https://www.youtube.com/watch?v=SOXCGQ-bQe4>

(株) ユーサイドシステム

<http://www.webrobo.jp>

1 : はじめに

多くのシステム開発がW e b ベースにシフトしています。

しかしながらその技術変遷も激しく、どのような言語、環境で開発すべきか、開発ポリシーを確定することが困難です。言語ひとつとっても、P e r l、P H P、R u b y、J a v a、. N E T等種々存在します。更に、J a v a s c r i p tもA j a x、N o d e . J s等々新たなものが出てきて、何を選択すべきか混乱します。又、J a v aと言ってもj s pなのか、アプレット、アプリケーションかで全く様相が異なるのですが、J a v a言語としてひとくくりで語られています。基本的にはh t m lベースのサーバーサイド言語か、クライアント言語で大きな違いがあるのですが。

また、オープンな環境が大手独自のフレームワークによりクローズドになり、一層複雑さを増しています。更に追い討ちをかけているのが、際限のないセキュリィ対策での格闘、まさに受難と言わざるを得ません。

その上で、開発ツールの良し悪しを評価をするなんて、いい加減にして欲しいという声が聞こえてきそうです。しかし、I T業はグローバル化し、エンドレスに技術進歩する宿命を背負っておることを覚悟し、現在のコピー、ペースト作業での開発から脱却し、ルーチンワーク的な作業はコンピューターに任せて、新しい課題に挑戦できる余裕を確保しなくてはなりません。日本が物作りで成功を収めるロボット技術を、ソフトウェア開発にも適用すべきです。生産性の向上、高品質、高い保守性を担保し、社会の期待に応え、夢ある産業として生まれ変わるには、ソフトウェアにおけるロボット化への取り組みは欠かせません。社会的インフラとして益々重要な役割を担わなければならないI Tですが、開発以上にシステムの保守が重要になってきておりますが、一方では部品化が進んでおります。こうした流れを考える時、今のような人手に依存する開発手法では限界になってきています。開発における自動化とそれに伴う部品のデータベース化されたシステム管理は必須です。

弊社では10年来「W e b ロボ」の開発に取り組んでおり、どうにか期待を裏切らないレベルの自動生成システムが出来上がって参りました。それをご理解いただくことは紙面では極めて困難なことですが、実際のトライに踏み込む判断の一助になるよう当説明書を用意しました。少しのご辛抱をいただいておりますことをお願いいたします。必ずや「W e b ロボ」での開発が、極めて効果的手法であることの認識と、次のステップとして用意している入門編トライへの動機付けをすると確信します。

2 : W e b ロボの特徴

- ① S t r u t s フレームワークでの開発
- ② データベースとして P o s t g r e s 、 O r a c l e 、 M y s q l
- ③ 2 8 パターンに及ぶプログラムソースを生成
- ④ J s p 、アプレット、アプリケーション、p d f 作表の J a v a 生成
- ⑤ メインプログラムは自動生成ないし J A R ファイルで提供され、プログラム上の個人差が生じにくく、保守性が極めて高い
- ⑥ 最小限のプログラ定義作業にとどめ、極力推論でのプログラム自動生成
- ⑦ テーブルの変更等仕様の変更があっても、プログラム生成により関係部品はすべてメンテされ、後戻り作業が発生しない
- ⑧ クラウド上に「w e b ロボ」を配置し、利用者はクライアントプログラムをダウンロードするだけで、複雑な開発、実行環境構築をせずに即開発には入ることが可能。一週間程度の試用サイトを用意
開発者が互いに遠隔でも問題ない開発が可能
- ⑨ 開発ボリュームの過多に合わせて増減が可能な低額な月額使用料による利用方式で、買取の場合のリスクを回避が可能
- ⑩ W e b ロボは開発ツールとしての役割だけでなく、自動生成したソースの学習により教育ツールとして機能
- ⑪ I o T を想定する時、センサー等端末とのインターフェイスがとることと W e b 連携できることが条件になるが、W e b ロボはこれらが可能

2 : W e b ロボの機能

1 : メニュー

W e b ロ ボ http://www.webrobo.jp

プロジェクトフォルダー Webロボバージョン

環境設定 | テーブル定義 | プログラム定義 | プログラム生成実行 | 翻 訳 | 仕様書作成 | 修正プログラム管理 | 回避外部入出力 | 終 了

プログラム生成	実行	関連部品生成	Javaプログラム閲覧	form.actionxml閲覧
プログラムパラメータ生成	Tomcatの起動	Struts.XML生成	JSPプログラム閲覧	Form.xml閲覧
Action,Jsp生成	Tomcatの終了	Validation作成	プログラム編集クラス閲覧	Action.xml閲覧
アプレット定義生成	TomcatManagerの起動	メッセージ作成	アプレットプログラム閲覧	Valid.xml閲覧
JNA生成	プログラム実行	Htmljsp変換	Action閲覧	validation.xml閲覧
アプレットファイル名チェック	単体実行環境コピー	struts-config選択作成		メッセージ閲覧
JSPヘルプ処理生成	単体プログラム実行			struts-config閲覧
アプレットヘルプ処理生成				

by YousideSystem

- ①プロジェクト環境の設定
- ②テーブル定義とテーブル関係部品の生成
- ③プログラムの定義とプログラム生成
- ④S T R U T S 関係部品生成
- ⑤各種仕様書作成

2 : テーブル定義

①以下画面でテーブル項目を定義

テーブル項目定義保守

システム識別子 テーブル項目定義保守

テーブル名 DB名

項目番号	テーブル項目(1)	画面項目名(36)	プライマリ	キー1	キー2	キー3	キー4	データ型	データ型名(18)	項目長	桁区分	桁区切り	空値	UNIQUE	チェック	チェック項目	文字
1	kaincd	会員コード	1	2				1	smallint	5.0	1	桁区切なし	NO	YES	1	数字	
2	kainmei	会員名						3	varchar	40.0			NO	NO	0		
3	kainkubun	会員区分		1				1	smallint	1.0	1	桁区切なし	YES	NO	1	数字	
4	jyusyo	住所						3	varchar	80.0			NO	NO	0		
5	denrwa	電話番号						3	varchar	12.0			NO	NO	0	電話番号	
6	syumi	趣味						3	varchar	4.0			YES	NO	0		
7	tourokubi	登録日						3	varchar	10.0			NO	NO	7	年月日	

登録 | 取消終了 | 行挿入 | F3 参照 | 行削除 | 操作説明

画面に表示する項目名を入力します。なるべく漢字でわかりやすい表現にします。

②テーブル仕様書の作成

テーブル名 DB名 テーブル定義書 作成日付 2015/10/11 ページ 1 コンテキスト
kaiin nyumon

番号	項目名	p	キー	項目タイプ	項目長	桁区切文字キー	空値	チェック	最小値	最大値	初期値
	画面項目名			展開区分				UNIQUE			備考
1	kaiincd 会員コード	1	2	smallint	5.0	桁区切な	NO	数字 YES			
2	kaiinmei 会員名			varchar	40.0	全角かな	NO	NO			
3	kaiinkubun 会員区分		1	smallint	1.0	桁区切な	YES	数字 NO			1
4	jyusyo 住所			varchar	80.0		NO	NO			
5	dennwa 電話番号			varchar	12.0		NO	電話番号 NO			
6	syumi 趣味			varchar	4.0		YES	NO			
7	tourokubi 登録日			varchar	10.0		NO	年月日 NO			

③テーブルクリエートSQL文を生成

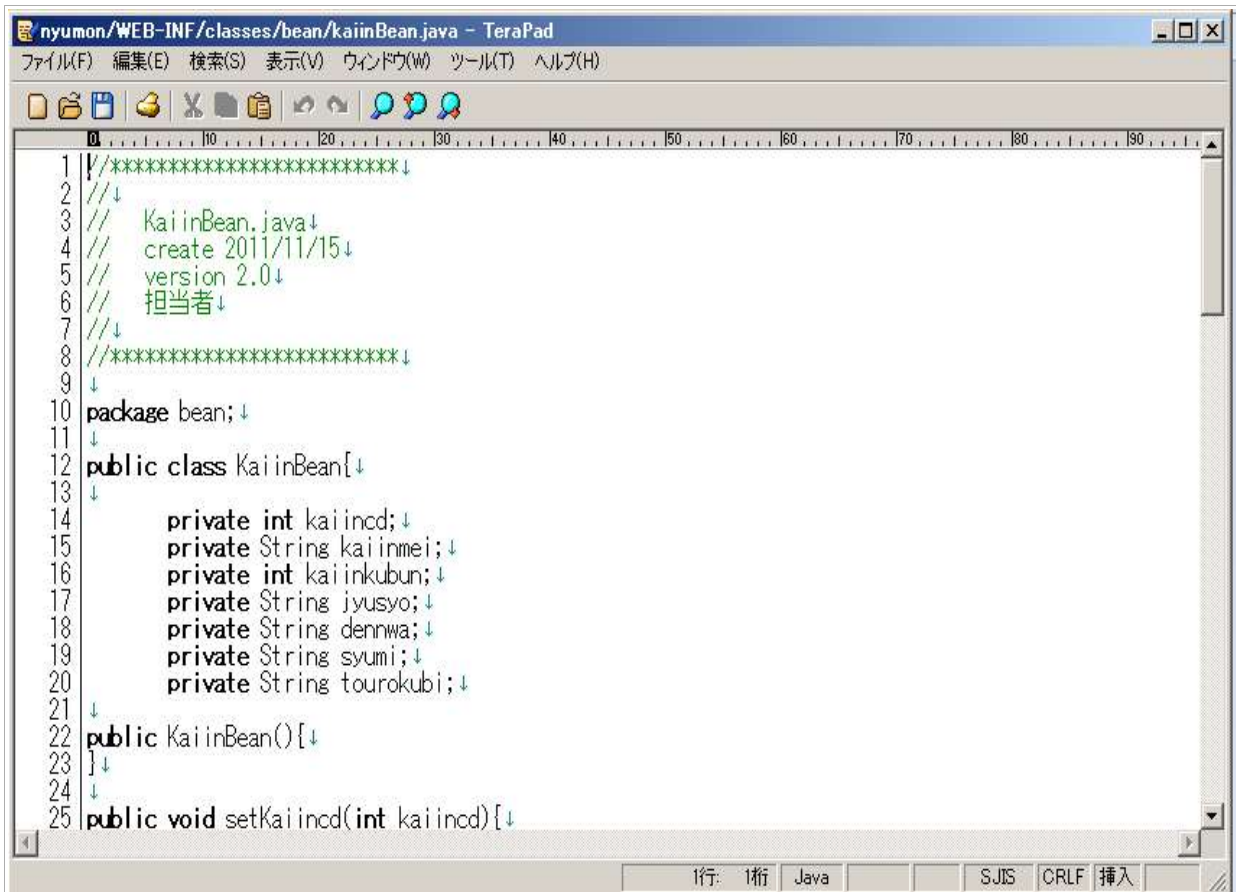
当SQL文を実行することで、データベースのテーブルを容易に生成

```
nyumon/sqlscript/kaiin.txt - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)

1 CREATE TABLE kaiin (
2   kaiincd smallint NOT NULL UNIQUE,
3   kaiinmei varchar( 40) NOT NULL,
4   kaiinkubun smallint,
5   jyusyo varchar( 80) NOT NULL,
6   dennwa varchar( 12) NOT NULL,
7   syumi varchar( 4),
8   tourokubi varchar( 10) NOT NULL,
9   PRIMARY KEY (kaiincd)
10 );
11 CREATE INDEX kaiin1 ON kaiin (kaiinkubun,kaiincd);
12
13 [EOF]
```

④テーブルビーンの生成

以下のようにテーブルビーンを自動生成

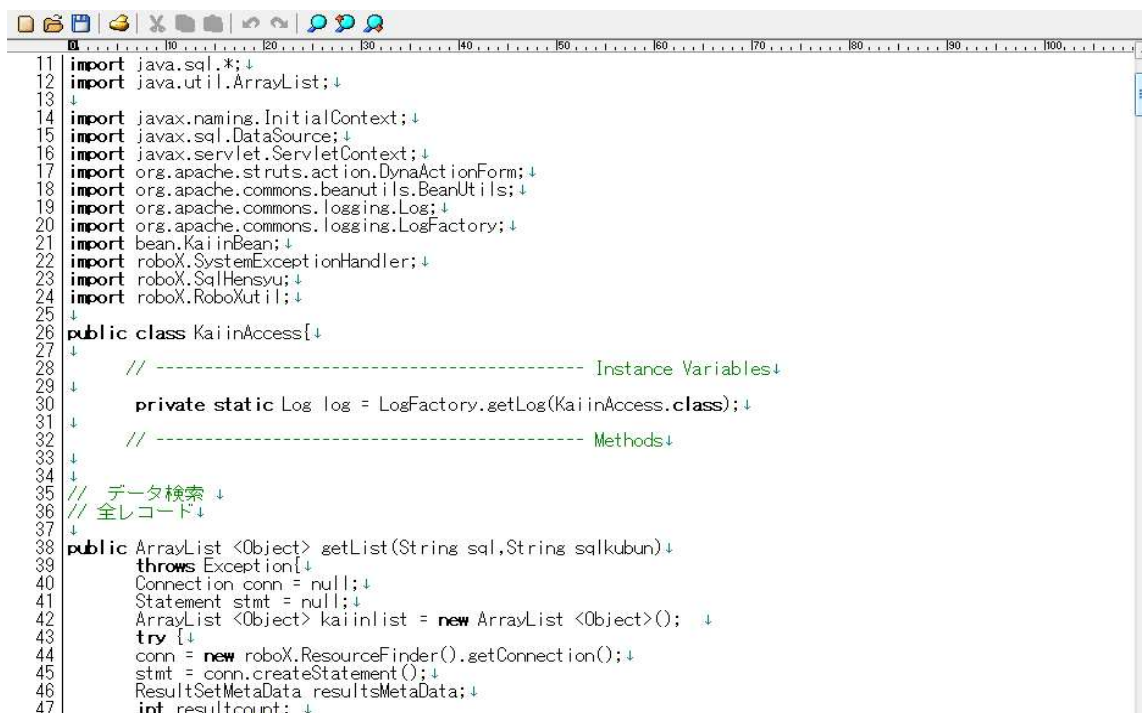


```
nyumon/WEB-INF/classes/bean/kaiinBean.java - TeraPad
ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)

1  //*****↓
2  //↓
3  //  KaiinBean.java↓
4  //  create 2011/11/15↓
5  //  version 2.0↓
6  //  担当者↓
7  //↓
8  //*****↓
9  ↓
10 package bean;↓
11 ↓
12 public class KaiinBean{↓
13 ↓
14     private int kaiinod;↓
15     private String kaiinmei;↓
16     private int kaiinkubun;↓
17     private String jyusyo;↓
18     private String dennwa;↓
19     private String syumi;↓
20     private String tourokubi;↓
21 ↓
22 public KaiinBean(){↓
23 }↓
24 ↓
25 public void setKaiinod(int kaiinod){↓
```

⑤テーブルアクセスメソッドの生成

テーブルとのアクセスに必要な全てのメソッドクラスを自動生成



```
11 import java.sql.*;↓
12 import java.util.ArrayList;↓
13 ↓
14 import javax.naming.InitialContext;↓
15 import javax.sql.DataSource;↓
16 import javax.servlet.ServletContext;↓
17 import org.apache.struts.action.DynaActionForm;↓
18 import org.apache.commons.beanutils.BeanUtils;↓
19 import org.apache.commons.logging.Log;↓
20 import org.apache.commons.logging.LogFactory;↓
21 import bean.KaiinBean;↓
22 import roboX.SystemExceptionHandler;↓
23 import roboX.SqlHensyu;↓
24 import roboX.RoboXutil;↓
25 ↓
26 public class KaiinAccess{↓
27 ↓
28     // ----- Instance Variables↓
29 ↓
30     private static Log log = LogFactory.getLog(KaiinAccess.class);↓
31 ↓
32     // ----- Methods↓
33 ↓
34 ↓
35 // データ検索 ↓
36 // 全レコード↓
37 ↓
38 public ArrayList <Object> getList(String sql,String sqlkubun)↓
39     throws Exception{↓
40     Connection conn = null;↓
41     Statement stmt = null;↓
42     ArrayList <Object> kaiinlist = new ArrayList <Object>(); ↓
43     try {↓
44         conn = new roboX.ResourceFinder().getConnection();↓
45         stmt = conn.createStatement();↓
46         ResultSetMetaData resultsMetaData;↓
47         int resultcount; ↓
```

⑥テーブル定義での項目チェック指定

テーブル項目定義で項目チェック指定があり、下記のようなチェック指定ができます。この指定により、テーブルへのインサート、アップデートするプログラム生成時に自動的に指定の項目チェックルーチンを挿入する機能があります。

テーブル定義時に以下の項目チェック指定が可能

選択肢

選択画面

	管理コード	管理名
選択	0	
選択	1	数字
選択	2	urlアドレス
選択	3	mailアドレス
選択	4	タグ可
選択	5	日
選択	6	月
選択	7	年月日
選択	8	郵便番号
選択	9	電話番号
選択	10	英数のみ
選択	11	英のみ
選択	12	半角片かな
選択	13	全角片カナ
選択	14	全角平かな
選択	15	電話又は携帯
選択	16	携帯電話
選択		
選択		
選択		
選択		
選択		
選択		
選択		
選択		
選択		
終了	前頁	次頁

⑦暗号化項目指定

情報漏洩についての対応は極めて大きな課題になってきています。

情報が外部に出ても重要な特に個人情報等の項目は、暗号化処理が施されておれば最悪の事態は回避されます。項目単位の暗号化、復号化の処理をプログラム単位に施すことには、大きな開発負荷が伴います。

しかし、Web ロボではテーブル定義で暗号化したい項目にフラグを立てるだけで、その項目にアクセスするプログラムに暗号化、復号化の処理を施す機能があります。

⑧アプレットによるマスタメンテプログラムの自動生成

http://localhost:8080/nyumon/Nyu00010.do 会員マスタ保守

＊ ＊ 会員マスタ保守 ＊ ＊

会員コード

会員名

会員区分

住所

電話番号

趣味

登録日

F1登録 F3削除 F4cs出 F6後退 F7前進 F8取消 F9参照 F10HELP

Webロボの大きな機能として、テーブル定義のみからアプレットでマスタメンテプログラムを自動生成します。

アプレットないしアプリケーションの場合、画面作成には座標値が要求されますが、座標値を自動計算します。

登録、修正、削除、レコード前進、後退、CSV出力の機能を有し更に、テーブル項目で項目チェックをした内容で、項目チェックルーチンが施されたプログラムが生成されます。

又、複数のキーがあった場合、どのキーによりレコード前進、後退させるかのそのキーを指定すれば、自動的にそのキーによるレコード読み込みのSQL文を生成します。

3：表題定義

プログラムタイトル、プログラムID、プログラムパターン等プログラム作成全体定義をします。

定義に当たって以下の画面でパターンを選択します。

①表題定義登録

プログラム番号	1	展開区分	表題	処理パターン	
プログラム名	nyu00010.apl	プログラム名は「システム識別子」		1 jsp通常登録	apl) とします。
タイトル名	会員マスタ保守			2 jsp通常保守	
パターン	44 アプレットマスタ登録保守	パターン詳細		3 jspメニュー	1:アプリケーション
				4 jsp一覧閲覧	
				5 jsp明細表示	
				6 jspログイン	
				7 jsp検索画面	
				8 jsp座標登録	
				9 jspマップリンク	
				10 jspアップロード	
				11 アプレットJTABLE入力	
				12 アプレットJTABLE保守	
				13 jsp入力	
				14 jsp予定登録	
				15 jspメール送信	
				16 jsp更新処理	
				17 jspインポート	
				18 jspエクスポート	
				19 jspファイル整理	
				24 アプレット一覧表示	
				27 jspファイル出力	
				28 jspレコード削除	
				31 Pdf一覧印刷	
				32 Pdf一覧表印刷	
				33 アプレット伝票登録	
				34 アプレット伝票保守	
				35 アプレット通常登録	
				36 アプレット通常保守	
				37 アプレット通常登録保守	
				38 アプレットメニュー	
				39 アプレット検索	
				40 直打詳細印刷	
				41 直打一覧印刷	
				42 アプレット通常閲覧	
				43 アプレット伝票閲覧	
				44 アプレットマスタ登録保守	
				45 ログイン	
WIDTH		HEIGHT	画面の場合		
主テーブル名	kaiin				テーブルとテーブルの間を 切ります。
主DB名	nyumon				
前プログラム					
HTMLバラ名					
URLチェック					
ログインチェック					

上記欄はアプレットの
代入する親画面項目を
サブミット指定は指定値に代入した後、サブミットするかどうかの指定です。
詳細は操作説明をご覧ください。

検索プログラムで検索値を

Webロボの際立った特徴は、Web開発で使われる各種プログラムのロジックをパターン化し、プログラムのメインロジックは自動生成ないしJARファイルとして提供します。パターンの指定のみで検証済みのメインロジックが取り込まれ、信頼性の高いプログラムが生成できます。また、メインロジックをプログラマーに任せることで生じる、プログラム保守性の低下を回避します。

アプレット、Pdfのプログラムでは絶対座標値が要求されますので、デザインツールなしでプログラム作成をすることは極めて大きな負担になりますが、Webロボでは画面、帳票定義定義より定義を取り込み、簡単に難度の高いプログラムを作成します。特に、項目数の多いデータ入力プログラムでは、カーソル制御された細やかに項目移動されるアプレット、アプリケーション開発に威力を発揮します。また、例えば多くの商品マスタから商品コードを選択してコード入力するプログラムの場合、別画面で複数の条件指定して該当データを一覧表示して選択入力をするプログラムは厄介ですが、Webロボは極めて容易に実現できます。

Webロボはhtmlベースのパターンはもとより、開発負担の大きいアプレット、アプリケーション、Pdfでの作表パターンでの自動生成により、操作性の良いエントリープログラムと見やすい野線入り作表プログラム作成を容易にしております。更に、最近とみにIoTが叫ばれていますが、IoTを考える時、htmlベースプログラムではセンサー等のデータとのインターフェイスをとることはできません。Webロボでは、Cソースを読んでJavaアプリケーションへのJNAによる

Web ロボは、IoT 開発をも視野においた各種自動生成を志向しています。

プログラム表題定義書		作成日付: 2016/10/11	ページ: 1
プログラム番号	1 nyu00010.a01	会員マスタ保守	
パターン	アプレットマスタ登録保守 画面区分	担当	1 nyu0001
画面指定	FULLSCREEN WIDTH HEIGHT 実行数	実行確認	
テーブル名	main		
ID名	nyu0001		
WHERE句			
プログラム	氏	前	
EDW/ラネ			
チェック	URLチェック	ログインチェック	
コメント			
送込関数	初期データ定義 main record_read CSV出力 CSV戻込 store delete		
関連部品			

プログラムパターンとして29に及ぶパターンがあり、パターンに対応した定義プログラムと自動生成プログラムを用意しております。

[illegible]

j s p パターンの場合、基本的に上図の行番号の 1 0 1、1 0 2、2 0 1 のように画面項目の行列番号を指定することで、j s p による h t m l 画面を生成するとともにテーブルへのインサートロジックを生成します。

会員登録 (J S P)

会員コード	
会員名	
会員区分	
住所	
電話番号	
趣味	

2. j s p 保守

⑦の検索画面による保守レコード指定により、レコードを表示するとともにレコードの削除ないしアップデートの処理ロジックを生成します。

3. j s p メニュー

j s p による h t m l のメニュー画面とリンク先を生成します。

J S P メニュー

会員登録	会員保守検索	
駅名マスタ保守	駅座標登録	ファイル出力
会員メール送信	会員登録HTML	

4. j s p 一覧閲覧

7ないし21の検索画面の条件指定より、j s p による h t m l でのレコード一覧表示画面を生成します。画面内レコード数の設定により、前頁、次頁のボタンを自動生成します。また、リンク先を指定した場合、詳細レコード表示等のプログラムへ容易にリンクします。

会員保守 覧 (page 1)

会員コード	会員名	kn_name	会員保守	会員閲覧	源泉徴収票	会員区分	ホームページ
5	野崎 T R 3	福島	☐	☐	☐	1	
11	ああああああああああ	東京	☐	☐	☐	2	http://www.yyyt.jp
12	ああああ	東京	☐	☐	☐	2	
20	あああ	福島	☐	☐	☐	1	http://www.yuu.jp
21	あつくええ	福島	☐	☐	☐	2	http://www.yyyyyyyyyy.co.jp
22	ああああああああああ a a	福島	☐	☐	☐	2	http://www.yyyy.jp
24	あああああ	福島	☐	☐	☐	2	http://www.webrobo.jp
25	あああああ 1	福島	☐	☐	☐	2	http://www.yyy.jp
26	あああああああ	福島	☐	☐	☐	2	http://www.ttt.jp
34	s s s s s s s s s s	東京	☐	☐	☐	2	

5. j s p 明細表示

j s p による h t m l の 1 レコード詳細画面を表示を生成します。

会員詳細閲覧

会員コード	1
会員名	鈴木 竜
会員区分	0
駅名	勿来
県名	東京
住所	福島県いわき市藤島町下矢田字宿畑12F
電話番号	0246-29-1366
趣味	スポーツ/音楽/旅行/その他
メール	webrobo@mail.jp
ホームページ	http://www.frr.jp
登録日	2011/11/28

戻る

6. j s p ログイン

j s p による h t m l のログイン画面を生成します。

J S P ログイン

担当者コード	1	担当者コード
パスワード	****	パスワード
ログイン		

7. j s p 検索画面

j s p による h t m l の条件検索画面を生成します。

会員保守検索

開始会員コード	
終了会員コード	
実行 戻る 会員一覧印刷	

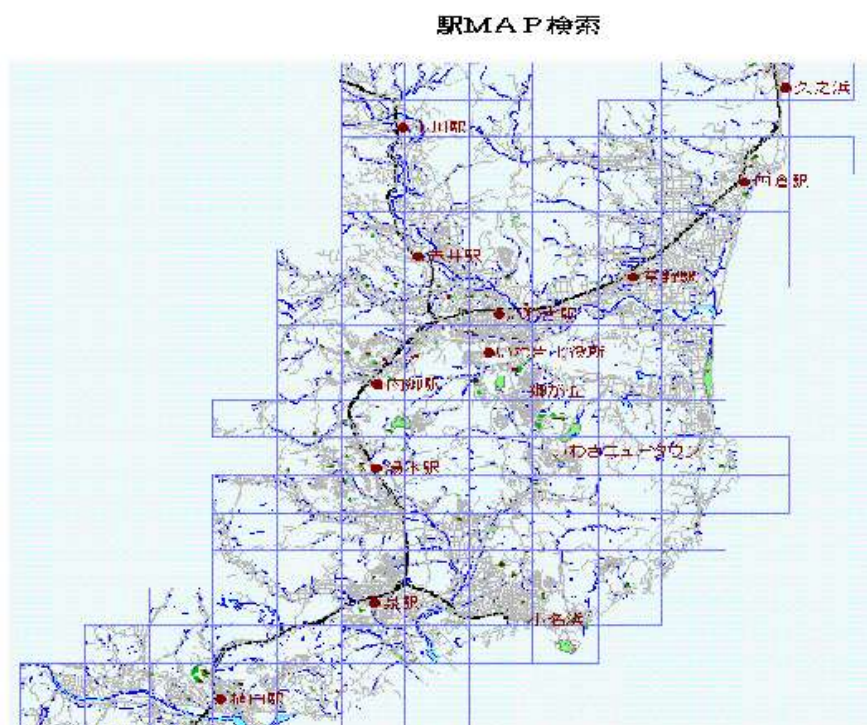
8. j s p 座標登録

j s p による h t m l のマップ画像の座標値登録プログラムを生成します。
利用画像を下敷きにして、2 点をクリックして座標登録をさせます。



9. j s p マップリンク

j s p による h t m l のマップ画像からのリンクプログラムを生成します。
閲覧したい画像周辺をクリックすることで、該当 h t m l へリンクします。



10. j s pアップロード

j s pによるh t m lのアップロードプログラムを生成します。

下図画面より、画像、C S Vファイル等をアップロードします。

アップロード前に、画像を表示確認することができます。

画像容量、画像タイプ、画像のリサイズしてが可能です。



11. メール送信

j s pによるメール送信プログラムを生成します。

12. j s pインポート

j s pによるテキスト、C S Vファイルのデータベースへのインポートプログラムを生成します。

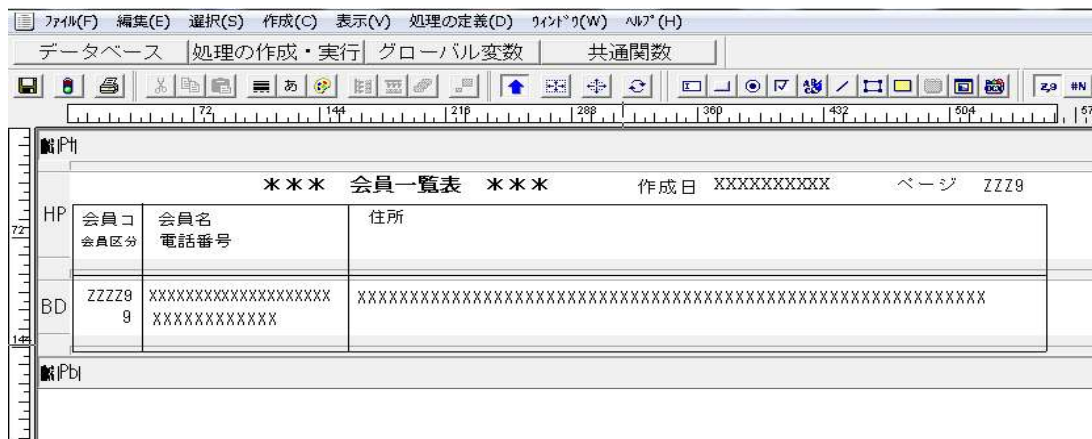
13. j s pエクスポート

j s pによるデータベースからテキスト、C S Vファイルへのエクスポートプログラムを生成します。

エクスポートファイルの指定、追加、上書きの指定、テーブル定義で指定した項目のチェック、ユニークチェック、テーブルのレコード存在チェック等をするプログラムを極めて簡単に生成します。

14. p d f一覧表印刷

下図のように、P d f作表ではW e bロボが利用するM R D Bのデザインツールで帳票定義し、その座標を取り込んでいます。



P d f l i b なるツールを使用することで、通常サーバーに生成される P d f ファイルを直接クライアントに生成します。また、プログラム生成時 P d f へのパスワード設定指示により、容易に P d f にクライアントパスワードが設定されます。セキュリティーの高い文書システム構築が可能です。

ページ 1

[illegible]

pdf による 1 レコードの詳細印刷プログラムを生成します。

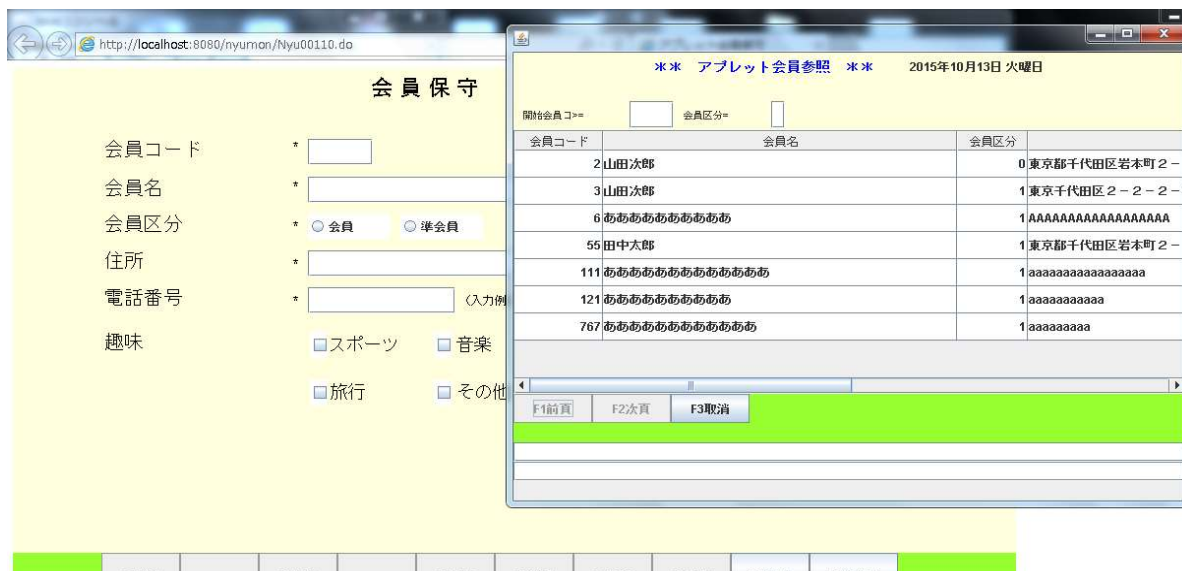
[illegible]

16. アプレット（アプリケーション）マスター登録保守

上記テーブル定義で説明したように、テーブル定義さえすればアプレットでのカーソル制御された操作性の高い、マスタ登録保守プログラムが完全に自動生成されます。

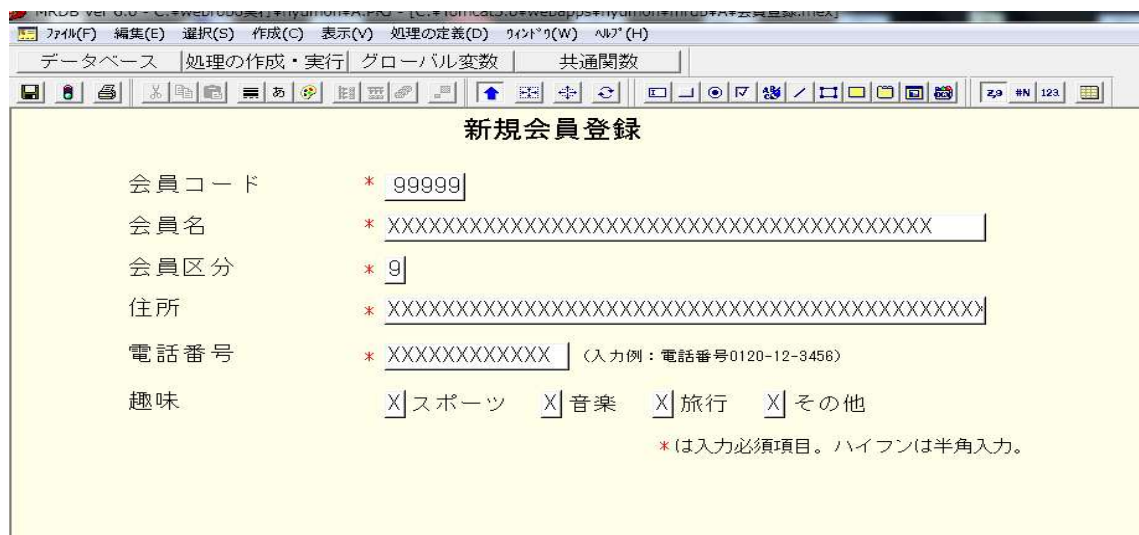
17. アプレット（アプリケーション）一覧表示

Webロボならではの機能として、アプレットプログラムで別ウインドウに参照プログラムを開いて、コード検索をするプログラムを極めて容易に作成することが出来ます。画面定義もせずに条件指定を行番号で指定し、条件指定でのSQL文のみを関数定義するだけでアプレット画面での前頁、次頁制御し、該当レコードをクリックすることでコードをメインプログラムに渡す機能の下図のようなプログラムを自動生成します。



18. アプレット（アプリケーション）登録

下図のように、アプレットではWebロボが利用するMRDBのデザインツールで画面定義し、その座標を取り込んでいます。



取り込んだ座標、文字ポイント等よりアプレット画面と詳細項目定義で定義した処理ロジックよりアプレットによる登録プログラムを生成します。

19. アプレット（アプリケーション）保守

MRDBによる画面定義と詳細項目定義より、アプレットでのデータ保守プログラムを生成します。

20. アプレット（アプリケーション）登録保守

MRDBによる画面定義と詳細項目定義より、アプレットでのデータ登録保守併用プログラムを生成します。

21. アプレット（アプリケーション）メニュー

MRDBによる画面定義と詳細項目定義より、アプレットでのメニュープログラムを生成します。



22. アプレット（アプリケーション）検索

MRDBによる画面定義と詳細項目定義より、アプレットでの検索条件指定プログラムを生成します。

想定される検索条件指定の次のプログラムは、閲覧プログラムか印刷プログラムかCSV主力プログラムです。

当検索条件指定によるSQL文により、閲覧、印刷、CSV出力プログラムが実行されます。尚、CSV出力プログラムはプログラム定義なしで作成することができます。

ログイン

担当者コード 鈴木太郎

パスワード

28. 直打詳細印刷

MRDBによる帳票定義と詳細項目定義より、P d f によらないアプリケーションでのレコード詳細印刷プログラムを生成します。

29. 直打一覧表印刷

MRDBによる帳票定義と詳細項目定義より、P d f によらないアプリケーションでの一覧表印刷プログラムを生成します。

4：詳細項目定義

①詳細項目定義

MRDB伝票項目定義（登録、保守）プログラム番号										6		アプレット会員登録		テーブル名			
開放閲覧		主テーブル名				kaiin				項目名							
目番	項目区分	項目区分名	タイプ	目タイプ	テーブル名	テーブル項目	画面項目名(36)	半角項目名(24)	画面項目長	ファイル項目長	エツ	空値	データ型				
1	1	表題部	1	入力文字	kaiin	kaincd	#会員コード	kaincd	5	5.0	1	NO	smallint				
2	1	表題部	1	入力文字	kaiin	kainmei	#会員名	kainmei	40	40.0	0	NO	varchar				
3	1	表題部	5	radiobutton	kaiin	kainkubun	#会員 準会員 横	kainkubun	1	1.0	1	YES	smallint				
4	1	表題部	1	入力文字	kaiin	jyusyo	#住所	jyusyo	80	80.0	0	NO	varchar				
5	1	表題部	1	入力文字	kaiin	dennwa	#電話番号	dennwa	12	12.0	9	NO	varchar				
6	1	表題部	6	checkbox			#趣味スポーツ	syumi1	1	4.0							
7	1	表題部	6	checkbox			#趣味音楽	syumi2	1	4.0							
8	1	表題部	6	checkbox			#趣味旅行	syumi3	1	4.0							
9	1	表題部	6	checkbox			#趣味その他	syumi4	1	4.0							
10	6	作業部	7	出力	kaiin	syumi		syumi	4	4.0	0	YES	varchar				
11	6	作業部	7	出力	kaiin	tourokubi		tourokubi	10	10.0	7	NO	varchar				

入出力項目でどのような制御をするかを定義します。

J a v a 言語とW e b ロボの規約にのっとて記述した、関数と呼んでいる制御文を差込します。

プログラムパターンごとに登録内容がことなりますので、それに対応した項目定義プログラムを用意しています。

②詳細項目定義仕様書

プログラム nru00100.apl										項目詳細定義		作成日付 2015/10/12 ページ 1	
6 アプレット会員登録										主テーブル名 kain			
項目番号	項目区分	項目名	項目タイプ	項目長	ヘルプ	参照プログラム	D B 名	テーブル名	WHERE句	繰込項目	繰込項目番号		
		ファイル項目名			パブリック変数	入力前処理				ファイル出力式			
1	表題部	#会員コード	入力文字	5									
158		kaincd kaincd	FIELD		Myrsncl nrsnnc1:								
2	表題部	#会員名	入力文字	40									
158		kainmei kainmei	FIELD		int screenfile = 0;								
3	表題部	#会員_準会員_横	radiobutton	1									
158		kainkubun kainkubun	FIELD		int classfile = 0;								
4	表題部	#住所	入力文字	80									
158		jyusyo jyusyo	FIELD		String syumi = "";								
5	表題部	#電話番号	入力文字	12									
158		dennwa dennwa	FIELD										
6	表題部	#趣味スポーツ	checkbox	1									
158		syumi1	FIELD										
7	表題部	#趣味音楽	checkbox	1									
158		syumi2	FIELD										

4 : 関数定義

①関数登録

スクリプト定義		テーブル項目	プログラム項目
スクリプト名	@会員レコード後退	nyu	テーブル名
説明文	アプレット会員保守用レコード後退		プログラム名
			項目名

行番号	スクリプト定義
1	//@会員レコード後退
2	if (!get\$varh("kaiincd").equals("")){
3	\$db.sql="select * from kaiin where kaiincd<"+get\$varh("kaiincd")
4	+ " order by kaiincd desc limit 1";
5	\$db.host="/Applet.do";
6	dbreadflg=DbAccess.sqlRead(\$db);
7	if (dbreadflg==3){\$msg1="例外が発生しました！"+\$db.sql;return;}
8	if (dbreadflg==2 \$d.\$db.r_count==0){\$msg1="該当するレコードがありません！";
9	return;}
10	\$den_row=\$d.\$db.r_count;
11	denpyo_read(\$den_row);
12	}

登録 破棄終了 行挿入 行削除 インクルード表示 参照

Webロボでの開発作業の多くは、上記画面での入出力項目の制御文を記述することです。

なるべく共通関数を作ること、重複作業を回避できます。

関数として登録した部品はデータベース化され、どのプログラムでどの関数が、どの関数がどのプログラムで使用されているか瞬時にわかります。

②関数表

関数表			作成日付	2015/10/12	ページ	1
関数名	行番号	担当口	関数定義			
@会員レコード後退	1	01 1	//@会員レコード後退			
	2	01 1	if (!get\$varh("kaiincd").equals("")){			
	3	01 1	\$db.sql="select * from kaiin where kaiincd<"+get\$varh("kaiincd")			
	4	01 1	+ " order by kaiincd desc limit 1";			
	5	01 1	\$db.host="/Applet.do";			
	6	01 1	dbreadflg=DbAccess.sqlRead(\$db);			
	7	01 1	if (dbreadflg==3){\$msg1="例外が発生しました！"+\$db.sql;return;}			
	8	01 1	if (dbreadflg==2 \$d.\$db.r_count==0){\$msg1="該当するレコードがありません！";			
	9	01 1	return;}			
	10	01 1	\$den_row=\$d.\$db.r_count;			
	11	01 1	denpyo_read(\$den_row);			
	12	01 1	}			
	13	01 1	else {\$msg1="会員コードが無指定です！"; return;}			
	14	01 1	// end			

登録した関数を印刷します。

③関数別プログラム一覧表

関数別プログラム別一覧表

2015/10/12

ページ 1

関数名	番号	プログラム名	タイトル名	挿入場所
@登録日	12	nyu00160.jsp	会員登録 2	詳細項目部

④プログラム別関数別一覧表

プログラム別関数一覧表

2015/10/12

ページ 1

番号	プログラム名	タイトル名	挿入場所	関数名
3	nyu00030.jsp	会員検索	詳細項目部	@test1
10	nyu00120.apl	タブレット会員参照	詳細項目部	@test1
8	nyu00130.apl	タブレット会員検索	表頭部 where	#kaiin-list-sql
12	nyu00160.jsp	会員登録 2	表頭部 atosyor i	@nyu00100後処理
12	nyu00160.jsp	会員登録 2	表頭部 maesyor i	@nyu00100前処理
12	nyu00160.jsp	会員登録 2	詳細項目部	@登録日

⑤プログラム使用関数明細

使用関数明細表 プログラム nyu00140.jsp

タイトル アプレット会員一覧

作成日付 2015/10/12 ページ 1

関数名	行番号	関数定義
#kaiin-list-sql	1	//#kaiin-list-sql
	2	prepaesql = "select * from kaiin where kaiinod>=?";
	3	typelist.add("int");
	4	datalist.add(String.valueOf(dynaForm.get("kaisi")));
	5	prepaesql = prepaesql + " and kaiinod <= ?";
	6	typelist.add("int");
	7	datalist.add(String.valueOf(dynaForm.get("syuryo")));
	8	prepaesql = prepaesql + " order by kaiinod";
	9	//end
@当日印刷	1	//@当日印刷
	2	Calendar cal = Calendar.getInstance();
	3	int month = cal.get(Calendar.MONTH) + 1;
	4	String today = cal.get(Calendar.YEAR) + "/" + month + "/"
	5	+ cal.get(Calendar.DATE);
	6	outrec.setSakuseibi (today);
	7	// end

5 : その他

①Form. xmlの自動生成

```
1 //*****↓
2 //
3 // 会員登録 (JSP) ↓
4 // Nyu00020Form.xml ↓
5 // create 2015/10/07 ↓
6 // version 2.0 ↓
7 // 担当者 ↓
8 //*****↓
9
10 ↓
11 <form-bean name="Nyu00020Form" ↓
12     type="org.apache.struts.validator.DynaValidatorForm" ↓
13     <form-property name="tourokubi" type="java.lang.String" /> ↓
14     <form-property name="kaiincd" type="java.lang.String" /> ↓
15     <form-property name="kaiinmei" type="java.lang.String" /> ↓
16     <form-property name="kaiinkubun" type="java.lang.String" initial="1" /> ↓
17     <form-property name="out_kaiinkubun" type="java.lang.String" /> ↓
18     <form-property name="jyusyo" type="java.lang.String" /> ↓
19     <form-property name="denwa" type="java.lang.String" /> ↓
20     <form-property name="syumi" type="java.lang.String" /> ↓
21     <form-property name="kakunin" type="java.lang.String" /> ↓
22 </form-bean> ↓
23
24 ↓
25 <form-bean name="Nyu00021Form" ↓
26     type="org.apache.struts.validator.DynaValidatorForm" ↓
27     <form-property name="tourokubi" type="java.lang.String" /> ↓
28     <form-property name="kaiincd" type="java.lang.String" /> ↓
29     <form-property name="kaiinmei" type="java.lang.String" /> ↓
30     <form-property name="kaiinkubun" type="java.lang.String" initial="1" /> ↓
31     <form-property name="out_kaiinkubun" type="java.lang.String" /> ↓
32     <form-property name="jyusyo" type="java.lang.String" /> ↓
33 </form-bean> ↓
```

②Action. xmlの自動生成

```
1 //*****↓
2 //
3 // アプレット会員登録 ↓
4 // Nyu00100Map.xml ↓
5 // create 2015/10/07 ↓
6 // version 2.0 ↓
7 // 担当者 ↓
8 //*****↓
9
10 ↓
11 <action ↓
12     path="/Nyu00100" ↓
13     type="action.Nyu00100Action" ↓
14     validate="false" ↓
15     <forward name="success" path="/WEB-INF/classes/jsp/Nyu00100.jsp" /> ↓
16     <forward name="top" path="/WEB-INF/classes/jsp/Top.jsp" /> ↓
17     <forward name="tokenvalid" path="/WEB-INF/classes/jsp/Tokenvalid.jsp" /> ↓
18     <forward name="urlcheck" path="/WEB-INF/classes/jsp/Urlcheck.jsp" /> ↓
19     <forward name="logincheck" path="/WEB-INF/classes/jsp/Logincheck.jsp" /> ↓
20     <forward name="norole" path="/WEB-INF/classes/jsp/Norole.jsp" /> ↓
21     <forward name="nokengenrec" path="/WEB-INF/classes/jsp/Nokengenrec.jsp" /> ↓
22     <forward name="reigai" path="/WEB-INF/classes/jsp/Exception.jsp" /> ↓
23 </action> ↓
24 [EOF]
```

③Valid. xmlの自動作成

```
1 //*****↓
2 //
3 // 会員登録 (JSP) ↓
4 // Nyu00020Valid.xml ↓
5 // create 2015/10/07 ↓
6 // version 2.0 ↓
7 // 担当者 ↓
8 //*****↓
9
10 ↓
11 <form name="Nyu00020Form"> ↓
12 ↓
13     <field ↓
14         property="kaiincd" ↓
15         depends="required,maxlength,integer"> ↓
16         <msg name="required" key="errors.required" /> ↓
17         <arg0 name="required" key="string.kaiincd" /> ↓
18         <msg name="maxlength" key="errors.maxlength" /> ↓
19         <arg1 name="maxlength" key="string.kaiincd" /> ↓
20         <var> ↓
21             <var-name>maxlength</var-name> ↓
22             <var-value>5</var-value> ↓
23         </var> ↓
24         <msg name="integer" key="errors.integer" /> ↓
25         <arg2 name="integer" key="string.kaiincd" /> ↓
26     </field> ↓
27     <field ↓
```

④メッセージの自動作成

```

ファイル(F) 編集(E) 検索(S) 表示(V) ウィンドウ(W) ツール(T) ヘルプ(H)
01
2 //
3 // MessagesJP.CSV(ソース) ↓
4 // Messages.properties (日本語変換) ↓
5 // create 2015/10/07 ↓
6 // version 2.0 ↓
7 // 担当者 ↓
8 //
9 //***** ↓
10 ↓
11 errors.header=<font color="red"> ↓
12 errors.footer=</font> ↓
13 errors.suffix=<br> ↓
14 errors.required={0}を入力して下さい。 ↓
15 errors.range={0}以上、{1}以下です。 ↓
16 errors.minlength={0}桁以上です。 ↓
17 errors.maxlength={0}桁以下です。 ↓
18 errors.invalid=文字列が該当しません。 ↓
19 errors.byte=数値がbyteに対応できません。 ↓
20 errors.short=数値がshortに対応できません。 ↓
21 errors.integer=数値がintegerに対応できません。 ↓
22 errors.long=数値がlongに対応できません。 ↓
23 errors.float=数値がfloatに対応できません。 ↓
24 errors.double=数値がdoubleに対応できません。 ↓
25 errors.date=日付形式ではありません。 ↓
26 errors.jihun=時分形式ではありません。 ↓
27 errors.creditcard=クレジットカード形式ではありません。 ↓
28 errors.email=メールアドレス形式ではありません。 ↓
29 errors.url=ホームページアドレス形式ではありません。 ↓
30 errors.eregular=不適切な文字が含まれます。 ↓
31 errors.lessut=数値が範囲より小さい。 ↓
32 errors.oversut=数値が範囲より大きい。 ↓
33 errors.ataierror=数値の値がおかしい。 ↓

```

⑤プログラマー一覧表

プログラマー一覧表 作成日付 2015/10/12 ページ 1

番号	プログラム名 インクルード名	タイトル名	パターン 風聞区分	企コ 担コ 担当者	テーブル名	次プログラム	戻り/MENU
1	nyu00010.apl	会員マスタ保守	77*let72	01 1 nyumon	kaiin		
2	nyu00020.jsp	会員登録(JSP)	j s p通	01 1 nyumon	kaiin	close	close
3	nyu00030.jsp	会員検索	j s p検	01 1 nyumon		nyu00040.jsp リンク	close
4	nyu00040.jsp	会員一覧表示	j s p一	01 1 nyumon	kaiin		close
5	nyu00050.jsp	会員保守(JSP)	j s p通	01 1 nyumon	kaiin	close	close
6	nyu00100.apl	アプレット会員登録	アプレッ	01 1 nyumon	kaiin		
8	nyu00110.apl	アプレット会員保守	アプレッ	01 1 nyumon	kaiin		
10	nyu00120.apl	アプレット会員参照	アプレッ	01 1 nyumon	kaiin		
8	nyu00130.apl	アプレット会員検索	アプレッ	01 1 nyumon		nyu00140.jsp	
11	nyu00140.jsp	アプレット会員一覧	j s p一	01 1 nyumon	kaiin		close
7	nyu00150.jsp	会員一覧印刷	P d f一	01 1 nyumon	kaiin	close	
12	nyu00160.jsp	会員登録2	j s p通	01 1 nyumon	kaiin	close	close

⑥担当別プログラマー一覧表

担当別プログラマー一覧表 作成日付 2015/10/12 ページ 1

担コ 企コ 担当者	番号	プログラム名 インクルード名	タイトル名	パターン 風聞区分	テーブル名	次プログラム	戻り/MENU
1 01 nyumon	1	nyu00010.apl	会員マスタ保守	44 77*let72	kaiin		
1 01 nyumon	2	nyu00020.jsp	会員登録(JSP)	1 j s p通	kaiin	close	close
1 01 nyumon	3	nyu00030.jsp	会員検索	7 j s p検		nyu00040.jsp リンク	close
1 01 nyumon	4	#kaiin-list-sql nyu00040.jsp	会員一覧表示	4 j s p一	kaiin		close
1 01 nyumon	5	nyu00050.jsp	会員保守(JSP)	2 j s p通	kaiin	close	close
1 01 nyumon	6	nyu00100.apl	アプレット会員登録	35 アプレッ	kaiin		
1 01 nyumon	8	nyu00110.apl	アプレット会員保守	36 アプレッ	kaiin		
1 01 nyumon	10	nyu00120.apl	アプレット会員参照	24 アプレッ	kaiin		
1 01 nyumon	8	nyu00130.apl	アプレット会員検索	39 アプレッ		nyu00140.jsp	
1 01 nyumon	11	#kaiin-list-sql nyu00140.jsp	アプレット会員一覧	4 j s p一	kaiin		close
1 01 nyumon	7	nyu00150.jsp	会員一覧印刷	32 P d f一	kaiin	close	
1 01 nyumon	12	nyu00160.jsp	会員登録2	1 j s p通	kaiin	close	close

⑦他プロジェクトからの取り込み

他のプロジェクトのテーブル定義、関数を取り込み、新たなプロジェクトで再利用できます。

以下がその操作メニューです。

The screenshot shows the 'Web Robo' application window. At the top, the title bar says 'Webロボ' and the URL is 'http://www.webrobo.jp'. Below the title bar, there are two input fields: 'プロジェクトフォルダー' (Project Folder) with the value 'c:\tomcat5.0\webapps\myumon' and 'Webロボバージョン' (Web Robo Version) with the value 'c:\webrobo'. A menu bar is located below these fields, containing the following items: '環境設定' (Environment Settings), 'テーブル定義' (Table Definition), 'プログラム定義' (Program Definition), 'プログラム生成実行' (Program Generation Execution), '翻訳' (Translation), '仕様書作成' (Specification Book Creation), '修正プログラム管理' (Correction Program Management), '退避外部入出力' (Backup External Input/Output), and '終了' (Exit). The '退避外部入出力' menu is currently open, displaying a grid of buttons. The buttons are arranged in two columns. The left column contains: 'テーブルエクスポート' (Table Export), 'テーブルWK取込' (Table WK Import), '個別テーブル取込' (Individual Table Import), and 'クリエイト定義取込' (Create Definition Import). The right column contains: '関数エクスポート' (Function Export), 'プロジェクト関数WK取込' (Project Function WK Import), '個別関数取込' (Individual Function Import), and '表題、詳細エクスポート' (Table Title, Detailed Export). To the right of this grid, there are two additional buttons: '定義データクリア' (Clear Definition Data) and '退避処理' (Backup Process).

6：プログラムパターン単価と項目単価による見積の提案

人月工数による見積が一般的ですが、要員の能力によって工数は大きく異なり、より客観的な基準による見積をすべきです。

Webロボは、プログラムパターンと項目数による見積によるべきと考え、プログラムパターンの本数単価と項目数単価を登録すれば、定義データより自動的に見積書を印刷するプログラムを用意しています。

予めユーザーとこれらの単価を決めておけば、仕様変更に伴う見積変更でのユーザーとのトラブルを回避することが可能です。

又、開発後提出見積と実績を比較評価することで、社内基準の確立と要員の能力評価が可能になります。

パターン別見積単価

管理コード	プログラムボタン	基本単価	項目単価	
1	j s p通常登録	50,000	3,000	
2	j s p通常保守	50,000		
3	j s pメニュー	50,000		
4	j s p一覧閲覧	50,000		
5	j s p明細表示			
6	j s pログイン			
7	j s p検索画面			
8	j s p座標登録			
9	j s pマッピング			
10	j s pアップロード			
11	アプレットJTABLE入力			
12	アプレットJTABLE保守			
13	j s p暦入力			
14	j s p予定登録			
15	j s pメール送信			
16	j s p更新処理			
17	j s pインポート			
18	j s pエクスポート			
19	j s pファイル整理			
24	アプレット一覧表示			
27	j s pファイル出力			
28	j s pレコード削除			
29	アプレット登録印刷			

タイプ別プログラム別見積一覧表

作成日付 2015/12/23 ページ 1

番号	プログラム名	タイトル名	パターン	基本料	項目数	項目単価	項目金額	プログラム金額
1	nyu00020. jsp	会員登録 (J S P)	j s p通常登録	50,000	7	3,000	21,000	71,000
2	nyu00160. jsp	会員登録 2	j s p通常登録	50,000	7	3,000	21,000	71,000
	小計				14			142,000
3	nyu00050. jsp	会員保守 (J S P)	j s p通常保守	50,000	7			50,000
	小計				7			50,000
4	nyu00040. jsp	会員一覧表示	j s p一覧閲覧	50,000	7			50,000
5	nyu00140. jsp	アプレット会員一覧	j s p一覧閲覧	50,000	6			50,000
	小計				13			100,000
6	nyu00030. jsp	会員検索	j s p検索画面		2			
	小計				2			
7	nyu00120. ap l	アプレット会員参照	アプレット一覧表示		7			
	小計				7			
8	nyu00150. jsp	会員一覧印刷	P d f 一覧表印刷		7			
	小計				7			
9	nyu00100. ap l	アプレット会員登録	アプレット通常登録		11			
	小計				11			
10	nyu00110. ap l	アプレット会員保守	アプレット通常保守		10			
	小計				10			
11	nyu00130. ap l	アプレット会員検索	アプレット検索		2			
	小計				2			
12	nyu00010. ap l	会員マスタ保守	アプレット登録保守		7			
	小計				7			
	合 計				80			292,000

7 : パッケージ開発用システムの提供

弊社ではすでに、会計システム、販売管理システムをWebロボで開発しており、現在給与システムを開発中です。

これらの基本パッケージシステムを低価格でソース提供し、一からシステム開発するコストの低減をはかることができます。

以下よりPDF資料をご覧くださいことが可能です。

会計システム <http://www.webrobo.jp/siryou/gkkaikei.pdf>

合併漁協用に開発された会計システムですが、一般企業でも十分利用可能です。科目レベルが5階層で、本店、支店及び部門損益が必要なそれなりの規模企業には、大いに利用価値が高いシステムです。

販売管理システム <http://www.webrobo.jp/siryou/gkkoubai.pdf>

当システムも漁協用購買管理システムとして開発しています。

十分、一般企業用システムへの転用は可能です。

最後に

最後までご辛抱いただき、お読みいただきましてありがとうございます。

当説明書をお読みいただき、少しいじってみたいとお思いになられた方への入門編コースを用意しています。Webロボの主なパターンのプログラム11本を、テキストに沿って実際プログラムを作ってもらい、Webロボの機能と開発効率を体験していただき、Webロボでの開発が貴社にとって有効化どうかの判断をしていただきます。