

# 新人研修報告書

2008年6月30日

森 健一

## 1.目的

Java 教育課題を通じて、基礎構文やコーディング規約など、アプリケーション開発が行えるスキル、Java 言語及び SQL の基礎知識を習得する事を目的とする。

## 2.方法

参考書、Web サイトから Java 言語の基本構文、基礎知識を取得する。習得した内容を応用し、Java 言語を使用したデータベース操作 (SQL) や Web アプリケーションの開発を行うことで、Java 言語に関する知識と理解を深める。

## 3.結果

Java 言語の理解が乏しかったインターフェースや、抽象クラスに関する知識を習得する事が出来た。さらに、Strategy パターンや Factory パターンといったデザインパターンについて理解した。また、Web アプリケーション開発の流れや SQL の基礎知識を習得することが出来た。

## 4.結論

Java 言語の基本構文、Web アプリケーションの開発方法について理解し、アプリケーション開発を行うための基礎知識を習得することが出来た。今後、一人前の技術者になれるようにさらなる知識、スキルの習得を心がける。

## 5.コメント

3ヶ月の Java 研修期間が終了するが、Java に対する学習が終わったわけではなく、今後も必要なスキルであるため、これからも Java 言語の理解を深める。また、7月以降からは SQL と VB やバッチ、Excel マクロなど実際の業務で必要となる知識の取得を行い、今後も更なるプログラミングスキルの習得に向けて精進する。

## 目次

	頁
[1] 目的.....	1
[2] 対象システム.....	1
[3] 方法.....	2
3.1 コーディング規約.....	2
3.2 継承・オーバーライドの習得.....	2
3.3 インターフェースの習得.....	2
3.4 抽象クラスの習得.....	2
3.5 オブジェクト指向基礎の習得.....	2
3.6 Web アプリケーション開発.....	3
[4] 結果.....	4
4.1 コーディング規約.....	4
4.2 継承・オーバーライドの習得.....	4
4.3 インターフェースの習得.....	5
4.4 抽象クラスの習得.....	6
4.5 オブジェクト指向基礎の習得.....	7
4.6 Web アプリケーション開発の習得.....	8
[5] 結論.....	10
[6] 参考文献.....	11
6.1 参考書.....	11
6.2 参考 Web サイト.....	11
[7] 付録.....	12

## [1] 目的

Java 教育課題を通じて、基礎構文やコーディング規約など、アプリケーション開発が行えるスキル、Java 言語及び SQL の基礎知識を習得する事を目的とする。

## [2] 対象システム

Java 教育で使用する PC 及び開発環境を以下に示す。

PC : パソコン工房

OS : Microsoft Windows XP Professional Version2002 Service Pack 2

メモリ : 896MB RAM

Tomcat : Apache Tomcat 6.0.14 Server

JDK : jdk1.5.0\_14

Eclipse : Eclipse SDK 3.3.2

MySQL : MySQL Server 5.0

MySQL Connector : MySQL Connector 5.0.8

Robocode : Version 1.0.6

Struts : Struts-1.3.8

## [3] 方法

### 3.1 コーディング規約

コーディング規約を読むことで学習し、プログラム作成時には常にコーディング規約を意識して教育課題に取り組む。特にマジックナンバや無意味な名前を使わないよう注意する。

### 3.2 継承・オーバーライドの習得

継承・オーバーライドについての知識を習得するために、Robocode を用いて下記のプログラム(ロボット)を作成する。

- ・ Robot クラスを継承して、三角形と四角形の動きを交互にするロボット
- ・ AdvancedRobot クラスを継承して、いろんな方向にスピンしながら移動するロボット
- ・ 画面中央に移動して、360° 敵を詮索し、見つけたら攻撃するロボット

### 3.3 インターフェースの習得

インターフェースについての知識を習得するために、デザインパターンの一つである Strategy パターンを用いて、弾を当てられるたびに三角形と四角形の動きを交互に変えるロボットを作成する。

### 3.4 抽象クラスの習得

抽象クラスについての知識を習得するために、Strategy パターンを適用した三角形と四角形の動きを交互に変えるロボットを、デザインパターンの Factory パターンで生成する。

### 3.5 オブジェクト指向基礎の習得

3.1～3.4 章で習得してきた内容の再確認として、オリジナルロボットを以下の手順で作成する。

1. ロボットの動き(戦略)を文章でまとめる
2. 1を元に、クラス図を作成する。ただし、インターフェースと抽象クラスを用いる
3. 2を元に、コーディングを行う

### 3.6 Web アプリケーション開発

Web アプリケーション開発を通じて Java の基礎知識を深めると共に、Java 言語のプログラミングスキル向上の為に、学校給食会システムの作成をする。

尚作成する画面は、

- 原料マスタ保守
- 原料区分マスタ保守
- 原料マスタリスト
- 搬入書発行
- 原料情報マスタ保守

の5画面である。これらの画面作成が終了後、単体テスト仕様書作成と単体テストを行う。

## [4] 結果

### 4.1 コーディング規約

コーディング規約を読むことで、分かりやすい名前の付け方や規則を習得した。特に注意したマジックナンバーは、使わないようにすることが出来た。

また、無意味な名前を付けないことに関しては、実際にコーディングを行う際に、分かりやすく短い名前を付けようと意識したが、短くすることで無意味な名前になることがあった。

これからもプログラムを作成する際には、コーディング規約を意識し、客観的な視点でソースチェックを行う様に心がけていく必要がある。

### 4.2 継承・オーバーライドの習得

継承・オーバーライド習得で以下のプログラムを作成した。

- Robot クラスを継承して、三角形と四角形の動きを交互にするロボット
- AdvancedRobot クラスを継承して、いろんな方向にスピンしながら移動するロボット
- 画面中央に移動して、360° 敵を詮索し、見つけたら攻撃するロボット

その結果、以下を理解、習得した。

継承とは、「既に定義されているクラス(スーパークラス)をもとに、拡張や変更を加えた新しいクラス(サブクラス)を定義すること」であることを理解した。

上記に示すロボットは、Robot クラスや AdvancedRobot クラスを継承する。継承の方法は、図 4.2-1 に示すようにコーディングする。

```
public class サブクラス名 extends スーパークラス{  
    処理内容  
}
```

図 4.2-1 . 継承の記述方法

オーバーライドとは、「スーパークラスで定義されているメソッドをサブクラスで定義しなおし、動作を上書きすること」であることを理解した。上記で示したロボットは、Robot クラスには各ロボットの基本動作を設定するメソッドとして定義されている run メソッドがあり、上記のクラスで run メソッドを定義することにより、独自の動作を行えることを確認した。

また、今回のプログラム作成時に Java 言語の基本的な処理となる、if 文・for 文・while 文の使用方法を再確認することが出来た。

### 4.3 インターフェースの習得

インターフェースとは、「定数とメソッドの宣言だけを集めたもので、宣言だけで実際のメソッド本体などは実装しないということ。また、インターフェースを実装したクラスは、インターフェースで定義したメソッドと変数を必ず実装しなければならない」ことを理解した。

三角形と四角形の動きを行うロボットを作成した。図 4.3-1 にクラス図を示す。

図 4.3-1 に示すように、Concrete 1<sup>\*1</sup> クラスと Concrete 2<sup>\*2</sup> クラスを用意する。この2つのクラスは、インターフェース<sup>\*3</sup> を実装する。これにより、アルゴリズムの切り替えが容易にできる。Main クラスでは、この2つのクラス<sup>\*1,\*2</sup>のどちらかを保持する。すなわち、保持するクラスを変えることにより、動作（アルゴリズム）を変えることが出来る。

尚、このアルゴリズムを変える仕組みは Strategy パターンと呼ばれる。

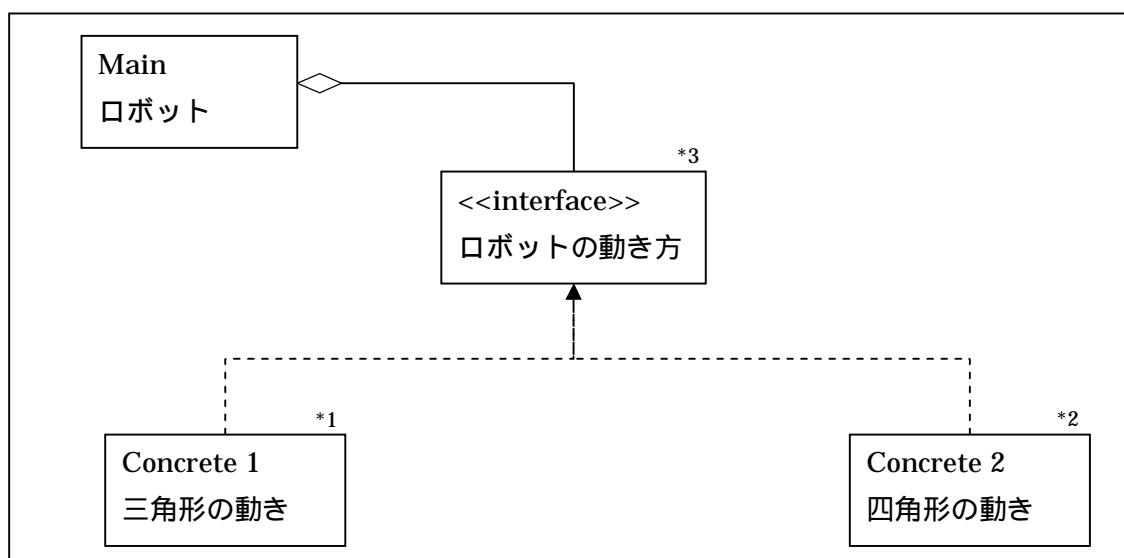


図 4.3-1 . Strategy パターン

#### 4.4 抽象クラスの習得

抽象クラスとは、「抽象メソッドが、一つでも含まれているクラスである」ことを理解した。また、「サブクラスでは、抽象メソッドを必ず実装する必要がある」ことも理解した。

4.3 章同様に、三角形と四角形の動きを行うロボットを作成する。尚、ロボットの動作を制御するクラスを新たに作成した。図 4.4-1 にクラス図を示す。

図 4.4-1 に示したように、戦略クラスの Concrete クラス<sup>\*1</sup> と、それを生成する Factory クラス<sup>\*2</sup> を用意する。尚、この 2 つのクラスはそれぞれ、AbstractFactory クラスと AbstractConcrete クラスを継承させる。

Main クラスで Factory<sup>\*2</sup> を利用して、ロボットの動作を制御するクラス<sup>\*1</sup> のインスタンスを生成する。

尚、このインスタンスを生成する仕組みは Factory パターンと呼ばれる。

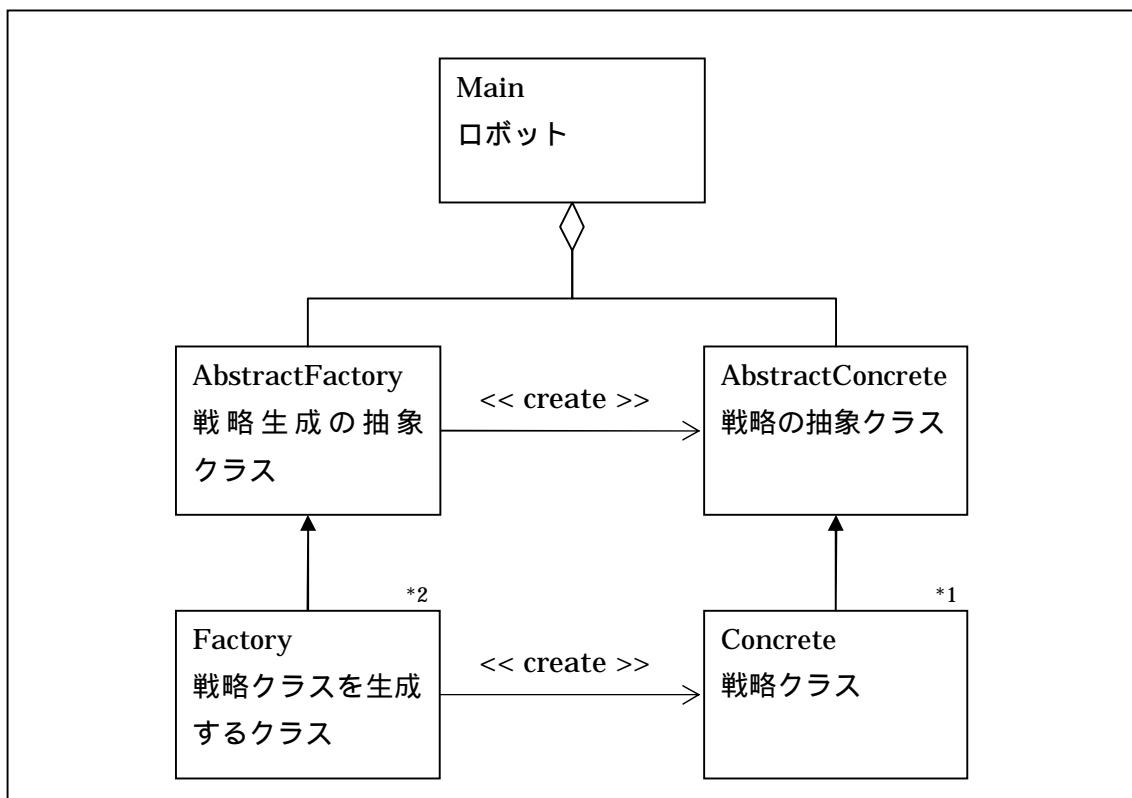


図 4.4-1 . Factory パターン



#### 4.5 オブジェクト指向基礎の習得

オリジナルロボットを作成することで、オブジェクト指向の基礎（継承、オーバーライド、インターフェース、抽象クラス）に関して、習得できていることを確認した。オリジナルロボットは以下の動き（戦略）を持つ。

- ・ 攻撃を受けると移動パターンを変更
- ・ 自機のエネルギー状況により攻撃パターンを変更

このロボットを、以下の手順で作成した。尚、( ) は各作業の成果である。

1. オリジナルロボットの動作戦術を検討（文章化）
2. 1で文章化した動作戦術を実現するためのクラス図を検討（クラス図(図 4.5-1)）  
条件：インターフェースと抽象クラスを用いること
3. コーディング（クラス）

その結果、以下を理解、習得した。

1. 自分の考えを文章にすることによって、乱雑なイメージを整理する方法を習得した。
2. クラス図の描き方を習得した。また、クラス図を作成することで、コーディングを行う際の流れをイメージすることが容易になった。
3. クラス図により、インターフェースや抽象クラスの流れを確認しつつ作業が出来る事を理解した。

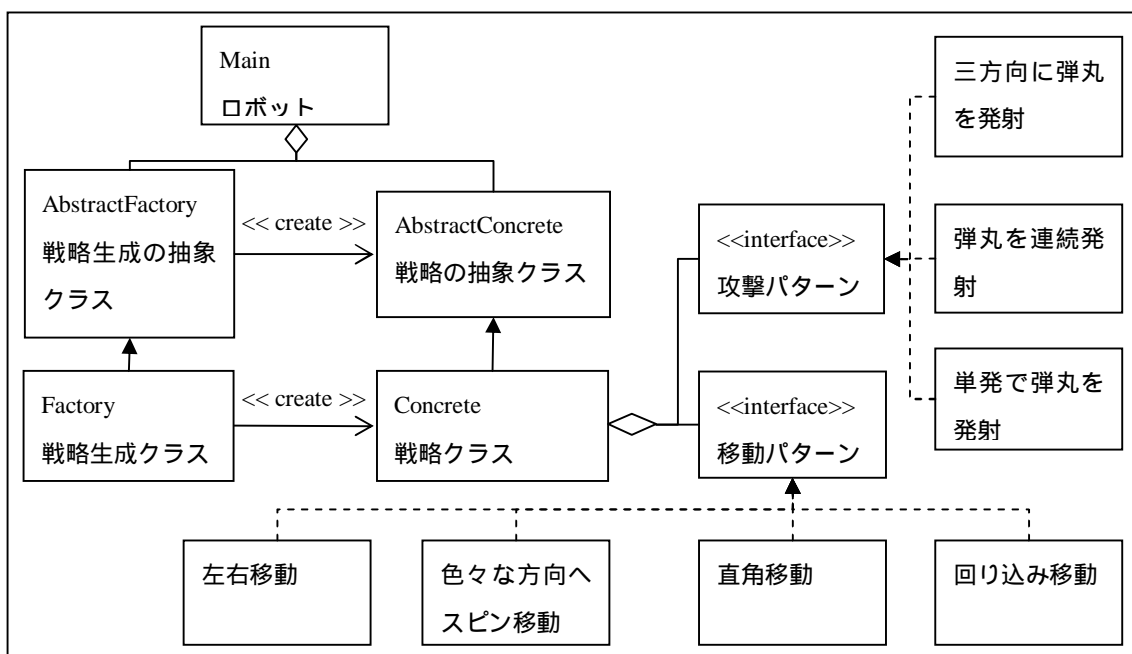


図 4.5-1 . オリジナルロボットのクラス図

## 4.6 Web アプリケーション開発の習得

Web アプリケーション開発で、以下のプログラムを作成した。尚、付録に各画面で作成したクラスを記す。

- 原料マスタ保守  
原料マスタ保守のテーブル(m\_material)にあるデータを、追加・更新・削除する。
- 原料区分マスタ保守  
原料区分マスタ保守のテーブル(m\_material\_kubun)にあるデータを、追加・更新・削除する。
- 原料マスタリスト  
原料マスタリストのデータを、Excel シートで表示する。
- 搬入書発行  
搬入書発行のテーブル(t\_nouhin\_denpyo)にあるデータで、搬入書発行先を指定して、Excel シートで表示する。
- 原料情報マスタ保守  
原料情報マスタ保守のテーブル(m\_material\_info)にあるデータを、追加・更新・削除する。
- 単体テスト仕様書  
作成したプログラムを使用している時に起こる可能性のある動作パターンを洗い出す。
- 単体テスト  
単体テスト仕様書の内容通りにプログラムを実行させ、正しく動作していればエビデンスを作成し、実行記録を残す。

その結果、以下を理解、習得した。

- データベースを操作するためにデータベース言語である SQL 使用した。また、検索・追加・更新・削除については、以下のようにして記述することを理解した。
  - データベースのデータを確認や検索を行う際には、「SERECT 列名 FROM テーブル名;」と記述する。
  - 追加を行う際は、「INSERT INTO テーブル名 (列名 1 , 列名 2 , ... 列名 n) VALUES (値 1 ,値 2, ...値 n);」と記述する。
  - 更新を行う際は、「UPDATE テーブル名 SET 列名 = 値 WHERE 条件式」と記述する。
  - 削除を行う際は、「DELETE FROM テーブル名 WHERE 条件式」と記述する。

- Session とは、「ある特定のクライアントがサーバと接続され、データを送受信している状態のこと」であることを理解した。また、Session にデータを格納し、名前を付けることが出来る。そして、Session に格納されているデータを取得したい場合、格納時に付けた名前を使うことでデータを取得することが出来ることを理解した。
- Excel シートの任意の位置にデータを設定する際に、POI ライブラリを使用した。Excel シートにデータを設定するまでの流れは、図 4.6-1 に示す。

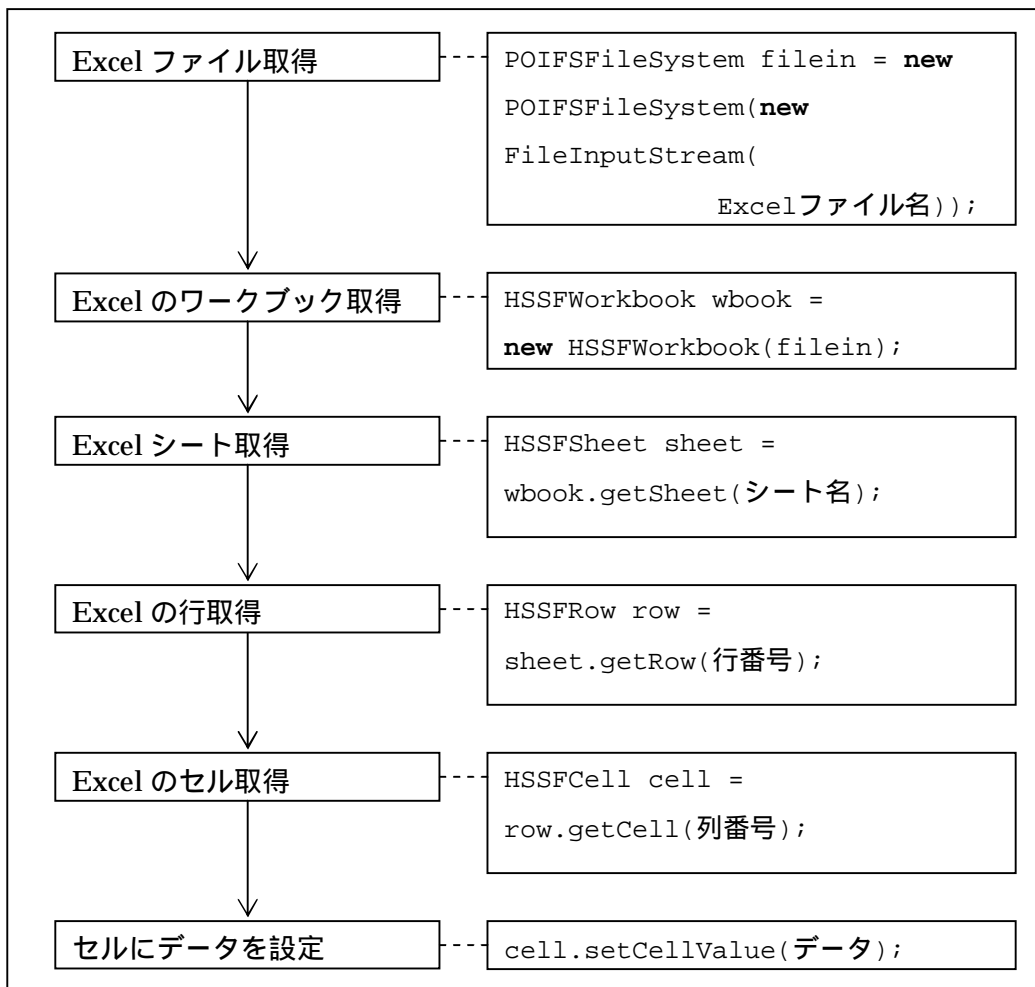


図 4.6-1 . データを設定するまでの流れ

- 単体テスト仕様書を作成する際は、プログラマとしての視点からだけでなく、ユーザーとしての視点から確認することも大切である。なぜなら、そのプログラムを使用するのはユーザーであり、ユーザーが使用する際に不具合を発生させないためである。
- 単体テストではバグが多く、自分のプログラミングスキルの未熟さを再確認することになった。また、単体テストを行うことで開発したアプリケーションの品質を保つことができることを理解した。

## [5] 結論

Java 言語の基本構文や、Web アプリケーションの開発方法について理解し、アプリケーション開発を行うための基礎知識を習得することが出来た。しかし、基礎的なスキルを習得しただけなので、より深い知識やスキルの習得を心がける。

## [6] 参考文献

### 6.1 参考書

Java 言語プログラミングレッスン上  
Java 言語プログラミングレッスン下  
デザインパターン入門  
基礎からのサーブレット/JSP 改訂版  
やさしい Java 第 3 版

### 6.2 参考 Web サイト

JavaA2Z  
[<http://www.kab-studio.biz/Programing/JavaA2Z/>]  
Java で Hello World  
[<http://www.hellohiro.com/>]  
C# デザインパターン UML Rarestyle へようこそ  
[<http://www.rarestyle.net/main/index.aspx>]  
サルでもわかる 逆引きデザインパターン  
[<http://www.nulab.co.jp/designPatterns/designPatterns1/designPatterns1-1.html>]  
Java 道  
[<http://www.javaroad.jp/index.htm>]  
Servlet および JavaServer Pages API ドキュメント  
[<http://www.ingrid.org/jakarta/servletapi/servletapi-4.0/docs-ja/index.html>]  
Java™ Platform, Standard Edition, v 1.3 API 仕様  
[<http://java.sun.com/j2se/1.3/ja/docs/ja/api/index.html>]  
FrontPage - きよえり PukWiki  
[<http://kiyoeri.gotdns.org/~kiyoeri/pukiwiki/>]  
SAK Streets  
[<http://homepage2.nifty.com/sak/>]

## [7] 付録

学校給食会システムで作成した 5 画面の、各画面で作成したクラスについての説明を以下に記す。

### ➤ 原料マスタ保守



図 7.1 . 原料マスタ保守画面

- アクションクラス  
原料マスタ保守画面での登録・削除ボタン押下時に、SQL を使用してデータベースを参照、登録・削除実行をする処理について記述したクラス。
- フォームクラス  
原料マスタ保守画面での登録・削除ボタン押下時に入力されたデータに対し、エラー有無の確認を行う処理について記述したクラス。
- 初期表示アクションクラス  
メインメニューから原料マスタ保守画面に移動したとき、テキストボックスやコンボボックス、レコードを表示するために必要なデータを取得する処理について記述したクラス。
- レコードを保持する Bean クラス  
原料マスタ保守のアクションクラスとフォームクラスで、受け渡しが必要となるデータを保持するためのクラス。
- 原料マスタ保守の画面 JSP  
画面全体のレイアウトについて記述したクラス。

➤ 原料区分マスタ保守



図 7.2 . 原料区分マスタ保守

- アクションクラス  
原料区分マスタ保守画面での登録・削除ボタン押下時に、SQL を使用してデータベースを参照、登録・削除を実行する処理について記述したクラス。
- フォームクラス  
原料区分マスタ保守画面での登録・削除ボタン押下時に、入力されたデータに対し、エラー有無の確認を行う処理について記述したクラス。
- 初期表示アクションクラス  
メインメニューから原料区分マスタ保守画面に移動したとき、テキストボックスやレコードを表示するために必要なデータを取得する処理について記述したクラス。
- レコードを保持する Bean クラス  
原料区分マスタ保守のアクションクラスとフォームクラスで、受け渡しの必要となるデータを保持するクラス。
- 原料区分マスタ保守の画面 JSP  
画面全体のレイアウトについて記述したクラス。

➤ 原料マスタリスト



図 7.3 . 原料マスタリスト画面

- ・ アクションクラス  
原料マスタリスト画面での Excel 出力ボタン押下時に必要なデータ  
を取得するために SQL を使用しデータベースを参照し、Excel へ出力  
するための処理について記述したクラス。
- ・ フォームクラス  
原料マスタリスト画面での Excel 出力ボタン押下時に入力されたデ  
ータに対し、エラーの有無の確認を行う処理について記述したクラス。
- ・ Excel シート  
原料マスタリスト画面での Excel 出力ボタン押下時に表示する Excel  
シートのレイアウト。
- ・ 原料マスタリストの画面 JSP  
画面全体のレイアウトについて記述したクラス。



➤ 搬入書発行



図 7.4 . 搬入書発行画面

- アクションクラス  
搬入書発行画面での Excel 出力ボタン押下時に必要なデータを取得するために SQL を使用しデータベースを参照し、Excel へ出力するための処理について記述したクラス。
- フォームクラス  
搬入書発行画面での Excel 出力ボタン押下時に入力されたデータに対し、エラーの有無の確認を行う処理について記述したクラス。
- Excel シート  
搬入書発行画面での Excel 出力ボタン押下時に表示する Excel シートのレイアウト。
- 搬入書発行の画面 JSP  
画面全体のレイアウトについて記述したクラス。

➤ 原料情報マスタ保守



図 7.5-1 . 原料情報マスタ保守画面



図 7.5-2 . 原料情報マスタ保守(期首在庫量登録)画面

- ・ アクションクラス

原料情報マスタ保守画面での登録・明細登録・検索・削除ボタン押下時に、SQL を使用してデータベースの参照、登録・削除を実行する処理について記述したクラス。

- ・ フォームクラス

原料情報マスタ保守画面での登録・明細登録・検索・削除ボタン押下時に入力されたデータに対し、エラー有無の確認を行う処理について記述したクラス。

- 初期表示アクションクラス  
メインメニューから原料情報マスタ保守画面に移動したとき、コンボボックスやレコードを表示させるために必要な処理について記述したクラス。
- 明細アクションクラス  
原料情報マスタ保守(期首在庫量登録)画面での登録・削除・確定ボタン押下時に、SQL を使用してデータベースの参照や、レコードの登録、削除を行う処理について記述したクラス。
- 明細フォームクラス  
原料情報マスタ保守(期首在庫量登録)画面での登録・削除・確定ボタン押下時に入力されたデータに対し、エラー有無の確認を行う処理について記述したクラス。
- 明細初期表示アクションクラス  
原料情報マスタ保守画面から原料情報マスタ保守(期首在庫量登録)画面に移動したとき、コンボボックスやレコードを表示させるのに必要な処理について記述したクラス。
- レコードを保持する Bean クラス  
原料情報マスタ保守の 2 画面間と、アクションクラスとフォームクラスで、受け渡しが必要となるデータを保持するクラス。
- 原料情報マスタ保守の画面 JSP  
画面全体のレイアウトについて記述したクラス。
- 原料情報マスタ保守(期首在庫量登録)画面 JSP  
画面全体のレイアウトについて記述したクラス。