新人研修報告書

2008 年 6 月 30 日
佐々木 直大

1.目的

Java 言語および Web アプリケーションについて理解をすることで、Java アプリケーション（Web アプリケーション含む）開発が行えるスキルを習得する。また、プロジェクトの管理をすることで、現場レベルのプロジェクト管理能力を習得することを目的とする。

2.方法

参考書や参考 Web サイトを活用して Java 言語（オブジェクト指向、デザインパターン）、Struts、MySQL についての基礎知識を習得すると共に課題のアプリケーションを作成することで、Java 言語（オブジェクト指向、デザインパターン）、Struts、MySQL についての使用方法やコーディング方法などを理解する。また、プロジェクト全体の進捗を管理することで、プロジェクト管理の技術を習得する。

3.結果

参考書や参考 Web サイトを熟読して Java 言語（オブジェクト指向、デザインパターン）、Struts、MySQL についての基礎知識を習得した。アプリケーションを開発することにより、習得した知識の使用方法を習得した。また、プロジェクトの管理については、毎朝、進捗会を実施したことにより、進捗遅れが発生した時にその原因と対策をメンバーと共有することができ、その結果、期限内に学校給食会システムを納めることができた。

4.結論

Java 言語（オブジェクト指向、デザインパターン）、Struts（JSP／Servlet）、MySQL についての基礎的な技術を習得することができた。ただし、プロジェクトの管理についてはまだまだ課題が多く、今後さらに努力していく必要があると感じた。

5.コメント

研修の 3 ヶ月間、自分が未経験の言語について学んできました。新しい言語にチャレンジする機会が無かったため、新鮮な気持ちで研修期間を過ごせました。今後も現状に満足せず、新しいことにチャレンジしていきたいと思います。

目次

頁

[1] 目的

Java 言語を用いたアプリケーション開発が行える技術と知識を習得する。

[2] 対象システム

Java 教育で使用する環境を以下に示す。

- PC：HP dx2000 ST(PE677AV)
- メモリ：1GB
- JDK：jdk1.5.0
- IDE：Eclipse SDK 3.3.1.1
- MySQL：MySQL Server 5.0
- Tomcat：Tomcat 6.0
- Struts：Struts 1.3.8

[3] 方法

3.1 Robocode の作成

3.1.1 Java 言語の基礎知識およびデザインパターンの習得

Robocode の教育課題を通じて、以下の Java 言語の基本的な知識およびデザインパターンについて知識を習得する。

クラスの継承

「課題１：三角形と四角形の動きを交互にするロボット」のプログラムを作成。

オーバーライド

「課題２：いろんな方向にスピンしながら移動するロボット」のプログラムを作成。

インターフェースと Strategy パターン

「課題３　：弾を当てられるたびに三角形と四角形の動きを交互にするロボット」のプログラムを作成。

TemplateMethod パターン

FactoryMethod パターンの関連パターンとして、参考書「Java 言語で学ぶデザインパターン入門」を使って知識を習得する。

抽象クラスと FactoryMethod パターン

「課題４　：課題３のロボットを Factory パターンを使用して作成」のプログラムを作成。

Singleton パターン

「課題４　：課題３のロボットを Factory パターンを使用して作成」の追加課題「ログ出力機能を追加する」プログラムを作成。

3.2 学校給食会システムの作成

3.2.1 Java 言語による Web アプリケーション開発

学校給食会システムの作成を通じて、以下の Web アプリケーション開発の基本的な知識を習得する。

Struts についての基礎知識習得

参考 Web サイト（Struts－TECHSCORE－）や Struts に付属しているサンプルプログラム（struts-blank-1.3.8 等）を解析して、Struts の構成や処理の流れを理解する。

セッション管理について

セッションとは、ユーザーがあるサイトを訪れてからそこを離れるまでの一連の通信のことである。

Web アプリケーションでは、セッション管理はセキュリティの観点において重要である。

今回開発する学校給食会システムでは、第三者がアプリケーションを使用できないようにログイン画面でユーザー名とパスワードを入力させ、認証されたユーザーのみシステムにアクセスできるようにする必要がある。そこで、ログイン認証されたセッション ID を保持し、ページへのアクセスがあった場合に認証されたセッション ID か判断して認証されたセッション ID であればページを表示させる。

上記処理を行うため、ログイン認証の一部とセッション管理部分の作成を通して、Web アプリケーションでのセッション管理について理解する。

スタイルシートについて

Web ページの見た目を整える方法としては、html タグを使用せずにスタイルシートを用いるのが一般的である。

学校給食会システムの画面整形を通じて、スタイルシートの使用方法を復習する。

Excel への帳票出力

ファイル操作用コンポーネント群「POI」を使用しての Excel への帳票出力方法について理解する。

使用する主な機能は以下の通り。

- **Excel ファイル読み込み**
- **ワークブック読み込み**
- **シート読み込み**
- **シートコピー**
- **行読み込み**
- **セル読み込み**
- **セル書き込み**

3.2.2 Java 言語による DB 操作および SQL

Web サイト（MySQL 4.1 リファレンスマニュアル）を参考にして、Java 言語による DB 操作と SQL についての基本的な知識を習得する。

MySQL についての基本的な知識

Web サイト（MySQL 4.1 リファレンスマニュアル）を参考にして、以下の基本的な知識を習得する。

JDBC

データベースの使用（use 文）

カラムに関する情報の取得（describe 文）

SQL 文についての知識の習得

Web サイト（MySQL 4.1 リファレンスマニュアル）を参考にして、以下の基本的な SQL 文の知識を習得する。

1 つ以上のテーブルからレコードを選択して取り出す（select 文）

既存のテーブルに新しいレコードを挿入する（insert 文）

既存のテーブルレコードのカラムを新しい値で更新する（update 文）

指定されている条件にマッチするレコードを削除する（delete 文）

3.2.3 課題プログラムの作成

3.2.1 項と 3.2.2 項で学んだことを使用して以下のプログラムを作成する。

給食会マスタ保守

レコード登録・追加・削除・検索機能（１件のみ）

給食会マスタリスト

指定されたキーに該当するレコードを Excel へ出力

製品区分マスタ保守

レコード登録・追加機能（１件のみ）

レコード削除機能（複数件）

全件表示機能

納品売上入力

レコード登録・追加機能（複数件）

レコード削除機能（複数件）

レコード検索機能（複数件）

納品売上入力チェックリスト

指定されたキーに該当するレコードを Excel へ出力

請求書発行

指定されたキーに該当するレコードを Excel へ出力

地区別原料使用一覧表

指定されたキーに該当するレコードを Excel へ出力

3.3 構成管理

Robocode および学校給食会システムの構成管理を通じて、構成管理ツール Subversion についての使用方法を習得する。

3.4 全体管理

学校給食会システム開発時に、以下の項目を実施してプロジェクトの管理方法について習得する。

SQL 文についての知識の習得

工程表を使用した全体の進捗状況の把握

他のメンバのサポート

[4] 結果

4.1 Robocode の作成

4.1.1 Java 言語の基礎知識およびデザインパターンの習得

Java 言語の基本的な知識およびデザインパターンについて習得した内容を以下に示す。

クラスの継承

クラスの継承とは、既存のクラス（スーパークラス）のメンバを新しく拡張したクラス（サブクラス）が受け継ぐことである。

「課題１：三角形と四角形の動きを交互にするロボット」を作成し、継承について確認した。
詳細を図 4-1 を使用して以下に説明する。
三角形と四角形の動きをするロボットのメインクラス(②)を、Robot クラス(①)を継承して作成。
メインクラス(②)で、Robot クラスの機能(③)であるロボットの前進を行うメソッド(ahead)やロボットの向きを変えるメソッド(turnLeft)を使用して三角形の動き、四角形の動きを実現することができることを確認することで、継承についての知識を習得した。

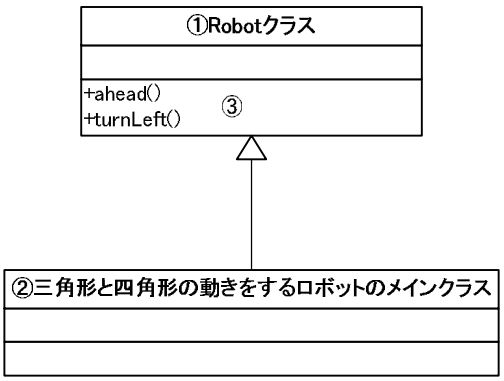

図 4-1 三角形と四角形の動きをするロボットのクラス図

オーバーライド

オーバーライドとは、スーパークラスにおいて定義されているインスタンスメソッドを、サブクラス内で再定義すること。

「課題２：いろんな方向にスピンしながら移動するロボット」を作成し、オーバーライドについて確認した。
詳細を図 4-2 を使用して以下に説明する。

いろんな方向にスピンしながら移動するロボットのメインクラス(②)を、AdvancedRobot クラス(①)を継承して作成。
メインクラスで Robot クラスの機能である壁に当たったというイベント(onHitWall)の内容を、180 度方向転換して前進移動するという処理として再定義(③)し、ロボットが壁に当たったときの動作を確認することで、スーパークラスのメソッドをサブクラスで再定義できるオーバーロードの機能について知識を習得した。

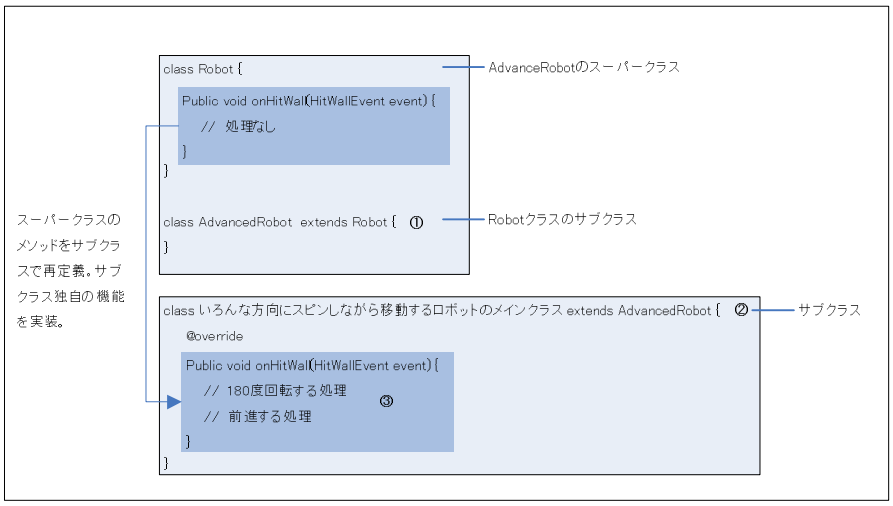

図 4-2 オーバーライドについて

インターフェースと Strategy パターン

インタフェースとは、ある特定の機能の概要を記述したもの。インタフェースを実装したクラスは、同一のインタフェースを実装したクラスとの間においてある特定の機能の概要を共通に持つ。

ある特定の機能の概要をクラスに実装する方法としてスーパークラスを継承すると言う方法もあるが、継承したサブクラスと継承されたスーパークラスの間にクラス関係を持つことになる。スーパークラスとサブクラスの間にクラス関係が見受けられない場合、このような継承を行うことはクラス設計上好ましくなく、この場合、インタフェースを使用するのが好ましい。

また、インターフェースを利用するメリットとして多態性が挙げられる。多態性とは、同じ命令を与えた時にオブジェクトごとに独自の動作を行わせる仕組みのことである。
インターフェースの場合、インターフェースで共通概念化された機能を実装クラスで具体化させることで、インターフェースを使う側は具体化された内容を知る必要がなく、抽象化された機能を使えばオブジェクトごとに具体化された内容が実行されるようになる。

Strategy パターンとは、戦略をクラス化することにより、戦略の切り替えを使用するクラスとは無関係に簡単に行えるようにするもの。戦略とはプログラミングのアルゴリズムに相当する。

「課題３　：弾を当てられるたびに三角形と四角形の動きを交互にするロボット」では Strategy パターンを適用する過程でインターフェースを使用している。
詳細を図 4-3 を使用して以下に説明する。

三角形の動きをするクラス(③)と四角形の動きをするクラス(④)の動くというメソッド(moveStrategy)を、インターフェースとして共通概念化して１つの戦略として定義(②)し、それぞれのクラスにインプリメント(⑤)する。
戦略の切替処理は、三角形と四角形の動きをするロボットのメインクラス(①)で現在の戦略インスタンスを保持するインターフェース型の変数(curStrategyObj)を用意して、被弾したときに curStrategyObj が③のインスタンスであれば④のインスタンスを、curStrategyObj が④

のインスタンスあれば③のインスタンスを curStrategyObj に格納する。戦略の実行は、現在の戦略が三角形の動きか四角形の動きかをメインクラスが知る必要はなく、curStrategyObj の moveStrategy を実行するだけでいい(前述の多態性)。

また、戦略を変更したい場合や新たな戦略を追加したい場合に、上記のような作りにしておくと、メインクラスや他の戦略への影響を最小限に抑えることができるという Strategy パターンの利点についても理解した。

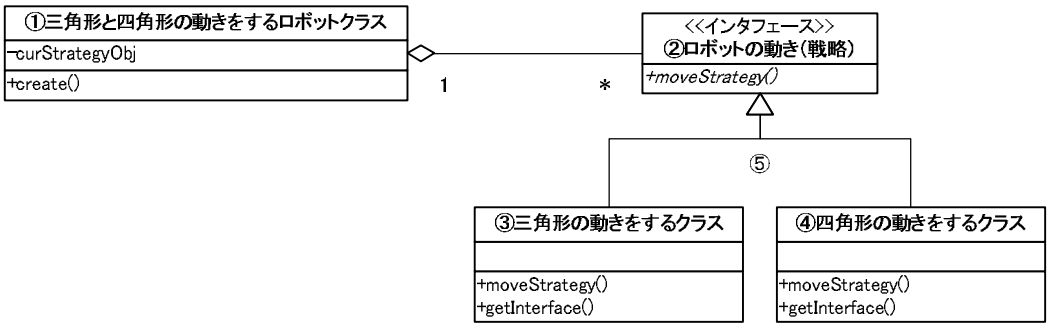

図 4-3 Strategy パターンを使用したクラス図

TemplateMethod パターン

スーパークラスで処理のフレームワーク（枠組み）を定義し、サブクラスで実際の処理内容を記述する。

課題で実際に使用はしていないが、FactoryMethod パターン（後述）や Strategy パターンと関連しているため参考書で上記の知識を習得した。

抽象クラスと FactoryMethod パターン

抽象クラスとは、それ自身がインスタンスを作ることはなく、他のオブジェクトに継承されるためだけに存在するクラスのこと。共通の機能を表現し、個々が持つ独自の機能はそれぞれのサブクラスで実装したい場合に使用する。

FactoryMethod パターンとは、TemplateMethod パターンをインスタンス生成の場面に適用したもの。インスタンスの作り方をスーパークラスで定義し、具体的な処理内容はサブクラスで行う。

「課題４　：課題３のロボットを Factory パターンを使用して作成」では FactoryMethod パターンを適用する過程で抽象クラスを使用している。
詳細を図 4-4 を使用して以下に説明する。

スーパークラスで①と②を抽象クラスで定義する。そして、①の create メソッドではインスタンスの作り方を指定（②のインスタンスを返す抽象メソッドとして定義）する。②の戦略を切り替えるメソッド（changeStrategy）と戦略を実行するメソッド（executeStrategy）については処理内容を定義しない抽象メソッドとして定義する。

サブクラスで③の create メソッドの処理内容として、④のインスタンスを生成するように指定する。④の戦略を切り替えるメソッド（changeStrategy）は、三角形と四角形の戦略を交互に切り替える処理を実装し、戦略を実行するメソッド（executeStrategy）は、指定された戦略を実行する処理を実装する。

このように、インスタンスの作り方と具体的な処理を分けることで、互いの処理の変更による影響を抑えられる。この課題作成を通じて FactoryMethod パターンと抽象クラスについての知識を習得した。

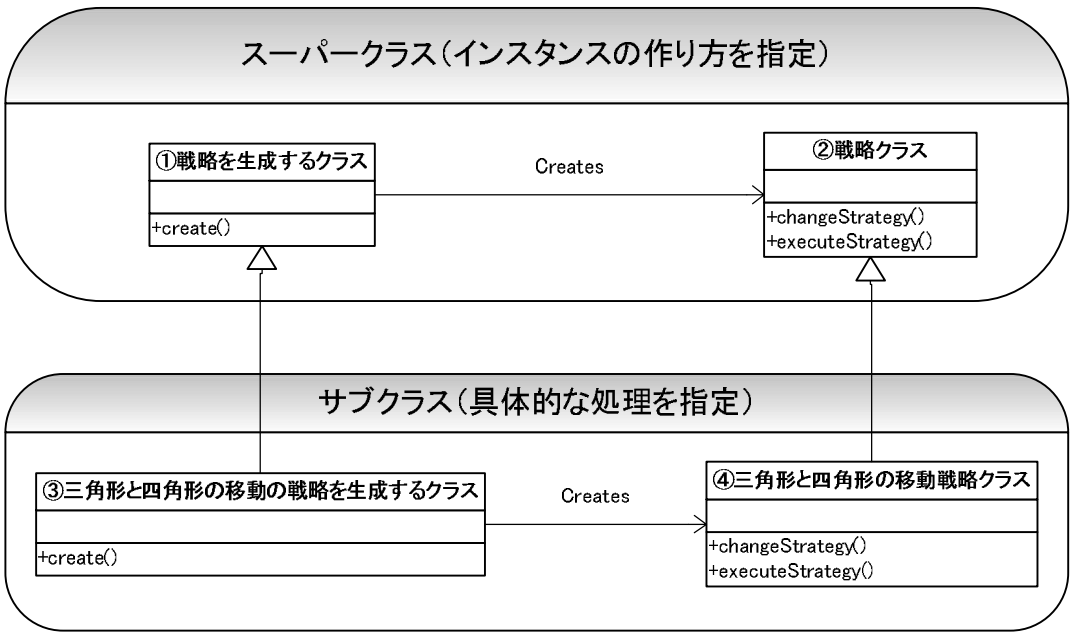

図 4-4 FactoryMethod パターンを使用したクラス図

Singleton パターン

指定したクラスのインスタンスが絶対に１個しか存在しないことを保証したい場合に用いる。

「課題４　：課題３のロボットを Factory **パターンを使用して作成」の追加課題「ログ出力機能を追加する」ではロボットの動作確認などのログを出力するクラス（**LobocodeLogger**）を作成する際、このパターンを使用した。このパターンを用いた理由として、インスタンスごとに個別の情報を保持させる必要がないので生成されるインスタンスは１つで十分だからである。**

ログ出力クラスのコンストラクタを privete **にして外部からインスタンス生成をされないようにし、外部から唯一のインスタンスを得るメソッドとして** getInstance **を用意する。**getIncetance **はインスタンスが生成されていなければ、１度だけインスタンスの生成を行いそのインスタンスを返す。既にインスタンスが生成されている場合、そのインスタンスを返す。**
このようにすることで、生成されるインスタンスが１個しか存在しないことが保障される。

このログを出力するクラスの作成を通じて Singleton **パターンの実装方法について知識を習得した。**

4.2 学校給食会システムの作成

4.2.1 Java 言語による Web アプリケーション開発

Java 言語による Web アプリケーション開発について習得した内容を以下に示す。

Struts についての基礎知識習得

Struts は MVC システムデザインをベースに設計されたフレームワークである。MVC デザインパターンでは、アプリケーションを以下の 3 つのレイヤ（層）に分割している。

M（Model）：ロジックの中核やデータベースアクセスなどを行う。

V（View）：入力フォームや処理結果の出力などを行う。

C（Control）：画面遷移や View・Model のマッピングを制御する。

Struts を使用した Web アプリケーションの流れは図 4-5 のようになる。

Struts ではアクションサーブレットが中心的役割を果たしている。他に重要なものとして、JSP ファイル、struts-config.xml、Action クラス、ActionFormBeans、JavaBeans がある。

以下に上で挙げた項目について理解した内容を図 4-5 を用いて示す。

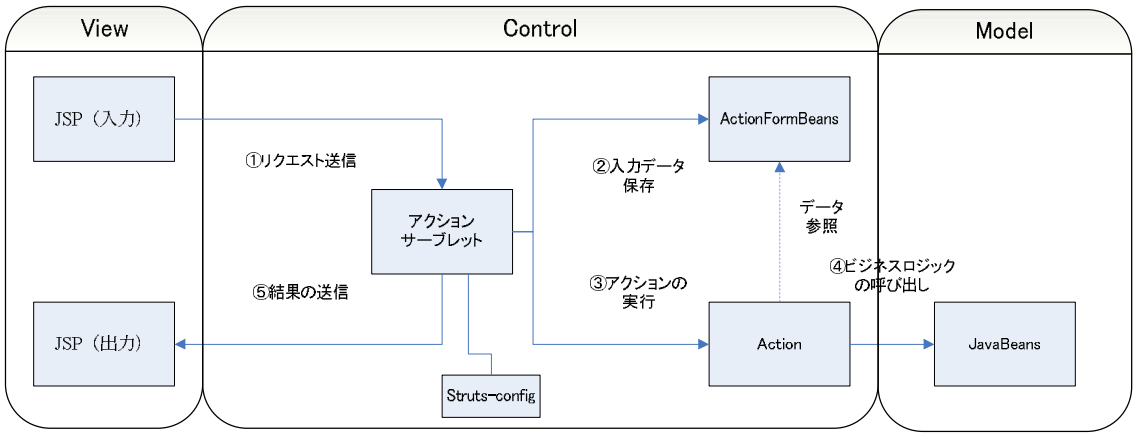

図 4-5 Struts アプリケーションの流れ

・アクションサーブレット

struts-config.xml（後述）を参照し、画面遷移（⑤）や、画面間のデータ受け渡し（①、②）、ロジックを制御する（③）役割を持つ。

・ActionFormBeans

ユーザーからの入力データを格納し、ビジネスロジックに引き渡す（②）役割を持つ。

データを保持するためのプライベート変数とプライベート変数への読み書きを行う setter／getter メソッドを持つ。
また、入力データの妥当性の検証や HTML カスタムタグに対してデフォルト値をセットする役割も持つ。

・JSP

Struts における View の役割を持つ（①、⑤）。JSP は View としての位置づけのため、できるだけロジックは JavaBeans に記述して表示のみの機能になるよう心がける。

・Action クラス

ActionFormBeans に格納したリクエストパラメータを基に、実際のビジネスロジックを実行する役割を持つ（④）。

・JavaBeans

ビジネスロジックのある機能を１つにまとめたクラスを指す。
ビジネスロジックは保守性を重視して、直接 Action クラスに記述せず、JavaBeans として別個に記述することが推奨されている。

・struts-config.xml

Struts アプリケーションの画面遷移や JSP、ActionFormBeans、Action クラスを関連付ける一連の制御情報を記述するもの。
<action>タグで JSP と Action クラスの関連付けを行い、<form-bean>タグで<action>タグと ActionFormBean の関連付けを行う。また、<forward>タグでは結果の出力先を指定する。

参考 Web サイト（Struts－TECHSCORE－）や Struts に付属しているサンプルプログラム（struts-blank-1.3.8 等）を解析することで、上記の Struts の構成や処理の流れを理解した。

・セッション管理について

図 4-6 に学校給食会システムで作成したセッション管理の遷移図を示す。以下、概略を説明する。(尚、説明の丸数字は図 4-6 の丸数字を示す)

① ログイン画面でユーザーID とパスワードを入力しログインボタンを押下すると、入力された ID、パスワードが登録されているものか認証を行なう。

② ID、パスワードの認証が登録されているものであった場合は、ログイン Action クラスが実行される。

③ ログイン Action クラスでは、認証されたセッション ID をセッションに保存する。

④ ログイン Action 終了後、メニュー画面へ遷移する。

⑤ メニュー画面 JSP では、メニュー画面にアクセスしているユーザーがログイン認証されたものであるか、セッション ID を確認することで判断する。ログイン認証されたユーザーであればメニュー画面を表示し、不正なアクセスであればログインエラーページを表示する。

⑥ メニュー画面にある各処理画面へのリンクがクリックされると、該当する処理画面の JSP へ遷移する。

⑦ 該当する処理画面の JSP では、処理画面にアクセスしているユーザーがログイン認証されたものであるか、セッション ID を確認することで判断する。ログイン認証されたユーザーであれば処理画面を表示し、不正なアクセスであればログインエラーページを表示する。

セッション管理のプログラム作成を通じて、第三者がアプリケーションを使用できないように、ログイン画面でユーザー名とパスワードを入力させ、認証されたユーザーのみシステムにアクセスできるようにする方法について知識を習得した。

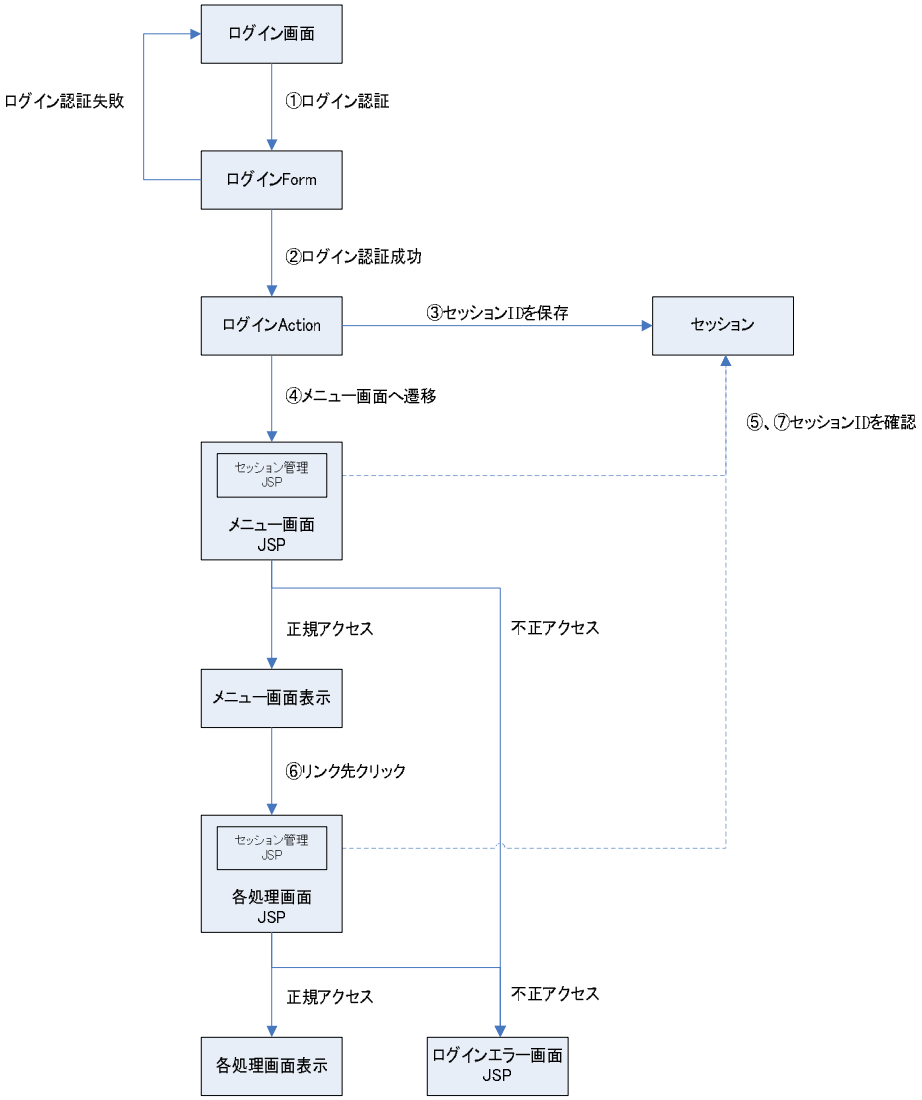

図 4-6　セッション管理の遷移図

・スタイルシートについて

スタイルシートとは、ホームページのデザインとレイアウトを HTML **を分けて書くための方法。**
スタイルシートを用いることで、適用されている全てのレイアウトを簡単に統一でき、レイアウトの変更もスタイルシートを変更するだけで適用されている全てのページに反映できる。

学校給食会システムの画面整形を通じて、上記の利点や記述方法について再確認した。

・Excel への帳票出力

図 4-7 に POI を使用した Excel への出力方法のサンプルを示す。以下、概略を説明する。（尚、説明の丸数字は図 4-7 の丸数字を示す）

・Excel ファイル読み込み(①)

POI に Excel ファイルを読み込ませるには、まず、読み込ませる Excel ファイルをバイト入力ストリーム（FileInputStream）でバイナリデータとして読み込んだあと、POIFSFileSystem クラスを使用して、POI でバイナリデータを読める形にする。

・ワークブック読み込み(②)

既存のワークブックを読み込むには、HSSFWorkBook クラスのコンストラクタの引数に前述の POIFSFileSystem クラスインスタンスを指定して、HSSFWorkBook クラスのインスタンスを生成する。

・シート読み込み(③)

既存のシートを読み込むには、HSSFWorkBook クラスの getsheet メソッドを使用し、引数に読み込みたいシートのシート名またはシート番号を指定する。そして、戻り値である HSSFSheet クラスのインスタンスを取得する。

・シートコピー(④)

既存のシートをコピーするには、HSSFWorkBook クラスの clonesheet メソッドを使用し、引数にコピー元となるシートのシート番号を指定する。そして、戻り値である HSSFSheet クラスのインスタンスを取得する。

・行読み込み(⑤)

シート内の特定の行を読み込むには、HSSFSheet クラスの getRow メソッドを使用し、引数に読み込みたい行の行番号を指定する。そして、戻り値である HSSFRow クラスのインスタンスを取得する。

・セル読み込み(⑥)

シート内の特定のセルを読み込むには、HSSFRow クラスの getCell メソッドを使用し、引数に読み込みたいセルの列番号を指定する。そして、戻り値である HSSFCell クラスのインスタンスを取

得する。

・セル書き込み(⑦)

シート内の特定のセルに値を書き込むには、書き込みたいセルを読み込んで HSSFCell クラスのインスタンスを取得し、setCellValue メソッドを使用し、引数に書き込みたい値を指定する。

```
// xls/sample.xlsファイルのパスを取得します。
final String REP_MASTER_FILE
    = "/xls/" + “sample”;
String filePath = this.getServlet()
    .getServletContext().getRealPath(REP_MASTER_FILE);

// ファイルを読み込みます。
POIFSFileSystem filein
    = new POIFSFileSystem(new FileInputStream(filePath));  ・・・・・・①

// ワークブックを読み込みます。
HSSFWorkbook wb = new HSSFWorkbook(filein);  ・・・・・・②

// シート(sheet1)を読み込みます。
HSSFSheet sheet = wb.getSheet(“sheet1”);  ・・・・・・③

// sheet1からシートをコピーする
sheet = wbook.cloneSheet(wbook.getSheetIndex(“sheet1”));  ・・・・・・④

// 1行目を取得します。
HSSFRow row = sheet.getRow(0);  ・・・・・・⑤

// 1列目(A)のセルを取得します。
HSSFCell cell = row.getCell((short) 0);  ・・・・・・⑥

// 1行目のセルAに数値1000を出力します。
cell.setCellValue(1000);  ・・・・・・⑦
```

図 4-7　POI サンプルコード

4.2.2 Java 言語による DB 操作および SQL

Java 言語による DB 操作と SQL について習得した内容を以下に示す。

・MySQL についての基本的な知識

学校給食会システムや Web サイト（MySQL 4.1 リファレンスマニュアル）を熟読して理解した内容を以下に示す。

・JDBC

Java プログラムからデータベースにアクセスするための API。実際に Java のアプリケーションから MySQL のデータベースへ接続するには JDBC ドライバを使用して行う。

学校給食会システムで実際に JDBC ドライバを介してデータソースの確立やデータソースへのクエリーおよび更新文の送信処理を行うことで、JDBC に対する理解を深めた。

・データベースの使用（use 文）

構文：use db_name;

db_name に指定したデータベースを、後続のクエリのデフォルトデータベースとして使用する。

学校給食会のデータベース（schoollunchsystem）内のテーブルを参照する場合は、use schoollunchsystem; で使用するデータベースを指定した後にテーブルを操作しなければならないことを理解した。

・カラムに関する情報の取得（desc 文）

構文：desc table_name;

table_name に指定したテーブルのカラムに関する情報を出力する。

上記命令で指定したテーブルのカラムに関する情報を取得できることを理解した。

・SQL 文についての知識の習得

Web サイト（MySQL 4.1 リファレンスマニュアル）を参考にして、理解した内容を以下に示す。

・1 つ以上のテーブルからレコードを選択して取り出す（select 文）

構文：select フィールド名 [[as] フィールド別名]，...
from テーブル名 [[as] テーブル別名]，...
[where 条件式]
[group by フィールド名，...]
[order by フィールド名[desc]，...];

・フィールド名

抽出するフィールド名をカンマ区切りで指定する。*を使用すると、全てのフィールドを抽出する。また、フィールドには別名を付ける事ができ、別名を付ける場合は、スペースを入れて別名を指定する。

・from 句

対象のテーブル名をカンマ区切りで指定する。

・[where 句]

レコードを抽出するための条件式を記述する。

・[group by 句]

指定したフィールド単位でグルーピングする。

・[order by 句]

指定したフィールドで昇順または降順にソートする。

上記内容を理解することで、select 文についての知識を習得した。

・既存のテーブルに新しいレコードを挿入する（insert 文）

構文：insert [into] テーブル名[(フィールド名，...)]
values(値，...) ;

・values

対応するフィールドの値を設定する。

このとき、指定以外のフィールドには、default の値か null が入る。

上記内容を理解することで、insert 文についての知識を習得した。

・既存のテーブルレコードのカラムを新しい値で更新する
（update 文）
構文：update テーブル名 set フィールド名=値，...
[where 条件式];
・where 句
where の条件に一致したレコードのフィールドを更新する。where で条件を指定しない場合、全てのレコードのフィールドが更新される。

上記内容を理解することで、update 文についての知識を習得した。

・指定されている条件にマッチするレコードを削除する（delete 文）
構文：delete from テーブル名
[where 条件式];
・where 句
where の条件に一致したレコードを削除する。where で条件を指定しない場合、全てのレコードが削除される。

上記内容を理解することで、delete 文についての知識を習得した。

4.2.3 課題プログラムの作成

4.2.1 項と 4.2.2 項で習得した項目を、実際のシステムに適用することで Struts や MySQL についての応用力を身に付けた。
また、画面構成や画面遷移などのユーザーインターフェースの重要性について再認識できた。

4.3 構成管理

Subversion の使用方法について習得した。

Subversion を使用していて、操作に慣れるのに苦労したのがチェックアウトである。

自分がこれまで使用してきた VisualSourceSafe とチェックアウトに対する考え方（競合についてを参照）が大きく違い、慣れるまで違和感があったが、ポリシーを理解することで操作に対する違和感が軽減した。

＊競合について

学校給食会システムでは、複数の人が同一ファイルを編集するケースがあり、競合が度々発生していた。

私が知っている VisualSourceSafe では、チェックアウト時点でファイルにロックがかかるため、コミットする際に競合することはない。

一方、Subversion では設定でチェックアウト時にファイルにロックはかからない（ファイルを自動的にロックをする設定もある）ため、競合が発生するケースがある。

この操作方法の相違は、Subversion に「競合したファイルは当事者同士で話し合って解決する」というポリシーがあるためと思われる。

競合した場合は、運用で回避する（共通ファイルはロックする）か、競合後にマージするなどして解決する。

4.4 全体管理

・新人全員参加の進捗会の実施

共通の問題・連絡事項について認識合わせができた。
また、メンバーの良いコミュニケーションの場にもなった。

反省点は、各自の進捗報告に対するチェックが甘かった点。
今後は、全体のスケジュールに対して各自の立てたスケジュールで、本当に間に合うか厳しくチェックすることが必要だと感じた。

・工程表を使用した全体の進捗状況の把握

Excel アドオンを使用した進捗管理で、進捗状況が一目で分かって進捗管理に非常に役立った。これからも機会があれば使用してみたい。

・他のメンバのサポート

管理とは若干異なるが、メンバで進捗が遅れている人の技術的なサポートを実施した。
Web 開発は初めてで自分自身のことで手一杯な部分があり、十分にサポートできていなかった。また、教えることの難しさというものも感じた。

これからはもっと全体を見渡す余裕を持たなければならないと実感した。
そのためにも自分の持ち分について早めに終わらせ、進捗に余裕を持たせるよう努力していきたい。
また、人にものを教える際にはただ知っていることを伝えるだけでなく、どうやったら相手に伝わるかを自分なりに工夫していきたい。

[5] 結論

技術的な教育課題（Java 言語、Struts、MySQL）について、知識を習得できた。
しかし、全体の管理の面ではまだまだ課題が多いと実感した。
今後、技術的な面の向上はもちろんのこと、全体の管理能力の向上という面に重点をおいて実務をこなしていきたい。

[6] 参考文献

6.1 参考書

- Java 言語で学ぶデザインパターン入門
- Apach Struts 入門
- Apache Struts ハンドブック
- MySQL 活用ガイド

6.2 参考 Web サイト

- 闘え、Robocode（ロボコード）！
  [http://www.ibm.com/developerworks/jp/java/library/j-robocode/]
- Java 2 プラットフォーム SE 1.4.0
  [http://sdc.sun.co.jp/java/docs/j2se/1.4/ja/docs/ja/api/index.html]
- チューニングのための JavaVM 講座（前編）
  [http://www.atmarkit.co.jp/fjava/rensai3/javavm01/javavm01_1.html]
- チューニングのための JavaVM 講座（後編）
  [http://www.atmarkit.co.jp/fjava/rensai3/javavm02/javavm02_1.html]
- MySQL 4.1 リファレンスマニュアル
  [http://dev.mysql.com/doc/refman/4.1/ja/index.html]
- Struts を使う Web アプリケーション構築術
  [http://www.atmarkit.co.jp/fjava/rensai3/struts01/struts01_1.html]
- Apache Struts Documentation
  [http://www.ingrid.org/jajakarta/struts/struts1.0/ja/target/documentation/api/overview-summary.html]
- Struts－TECHSCORE－
  [http://www.techscore.com/tech/ApacheJakarta/Struts/index.html]
- Struts による Web アプリケーション開発
  [http://www.infoscience.co.jp/technical/struts/index.html]
- とほほのスタイルシート入門
  [http://www.tohoho-web.com/css/index.htm]
- POI API Documentation
  [http://poi.apache.org/apidocs/index.html]
- 原色大辞典
  [http://www.colordic.org/]