プログラミング応用レポート

チーム: ムガルパレス名工大店

2025 年 5 月 26 日

1　はじめに

本レポートは、2025 年度名古屋工業大学プログラミング応用の課題レポートである。本レポートでは、我々のチーム「ムガルパレス名工大店」が開発した「WizardSpellNameQuiz」の仕様の解説を行う。本レポートは、以下の構成で進める。

- 1 章: はじめに
- 2 章: プログラムの概要
- 3 章: 解説
- 4 章: おわりに

本プロダクトは、ハリーポッター api[github のリンク] を使用しており、ハリーポッターに登場する魔法の名前を当てるクイズゲームである。

2　プログラムの概要

今回のプログラムは、課題の指示に従い作成されている。課題の指示は以下の通りである。

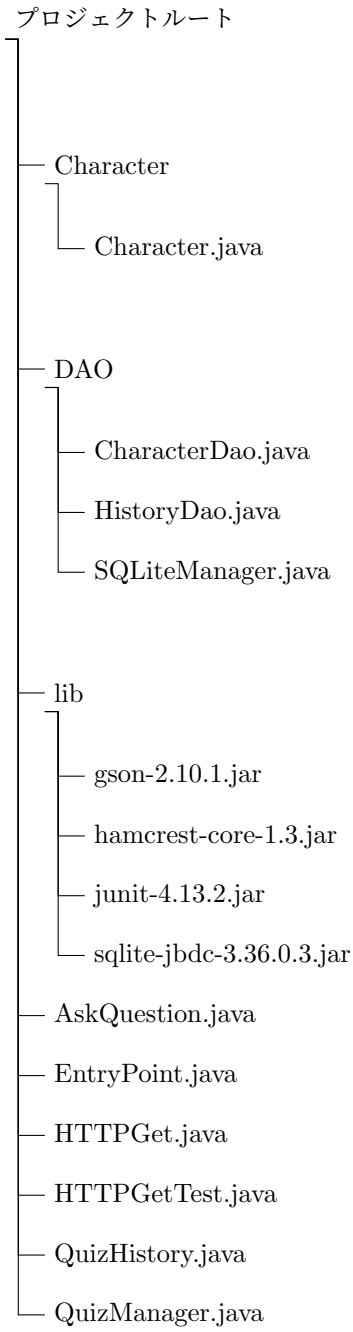
- データベースを使うこと
- Junit を用いたテストコードを作成すること
- GoF のデザインパターンを 1 つ以上使うこと
- Java 17 で新たに採用された言語仕様を調べ、それを用いたプログラムとすること

これらの指示に従い、我々は以下のようなプログラムを作成した。

- データベースを使うこと
  - SQLite を使用して、ハリーポッター api から取得したデータの id、魔法の名前及びその説明をデータベースに保存した。
  - データベースは、SQLite の JDBC ドライバを使用して接続した。

- Junit を用いたテストコードを作成すること
  - Junit を使用して、プログラムの各機能のテストコードを作成した。
  - テストコードは、プログラムの各機能が正しく動作することを確認するために使用した。
- BoF のデザインパターンを 1 つ以上使うこと
  - BoF のデザインパターンとして、algorithm パターンを使用した。
  - algorithm パターン出題形式の変換を行う機能を実装した。
- Java 17 で新たに採用された言語仕様を調べ、それを用いたプログラムとすること
  - Java 17 で新たに採用された言語仕様として、Pattern Matching for switch を使用した。
  - Pattern Matching for switch を使用して、魔法の名前とその説明を取得するクラスを作成した。

今回のプロダクトのアーキテクチャは、以下のようになっている。


プロジェクトルート
Character
Character.java
DAO
CharacterDao.java
HistoryDao.java
SQLiteManager.java
lib
gson-2.10.1.jar
hamcrest-core-1.3.jar
junit-4.13.2.jar
sqlite-jbdc-3.36.0.3.jar
AskQuestion.java
EntryPoint.java
HTTPGet.java
HTTPGetTest.java
QuizHistory.java
QuizManager.java

また、今後の解説に使用するためプログラムの実行結果を以下に示す。

```
Hello, Mugal Palace in Nitech!
This is a Harry Potter quiz game.

Please choose an option:
[1] Take a quiz
[2] Change mode
[3] View history
[4] Clear history
[5] Exit
Enter a number: 1
説明: Conjures and projects Lord Voldemort's Dark Mark
1: Morsmordre
2: Avada Kedavra
3: Mucus Ad Nauseam
4: Obliviate
正しい名前の番号を選んでください（1-4）: 数字で1-4を入力してください: 4
不正解。正解は 1: Morsmordre です。

Please choose an option:
[1] Take a quiz
[2] Change mode
[3] View history
[4] Clear history
[5] Exit
Enter a number: 2

--- Change Quiz Mode ---
Current mode: 4択モード
Select new mode:
[0] 4択問題モード
[1] 2択問題モード
Your choice: 1
モードを 2択問題モード に変更しました。

Please choose an option:
[1] Take a quiz
[2] Change mode
[3] View history
[4] Clear history
[5] Exit
Enter a number: 1
説明: Reveals the presence of another person
名前: Homenum Revelio
この説明と名前の組み合わせは正しいですか？（○: y / ×
    : n）: y または n で入力してください: n
```

```
不正解。正解は ○ です。

Please choose an option:
[1] Take a quiz
[2] Change mode
[3] View history
[4] Clear history
[5] Exit
Enter a number: 3
履歴を表示します。

--- Quiz History ---
[2025-05-25 13:18:01] 問題: Reveals the presence of another person | 表示名: Homenum
    Revelio |  正解: Homenum Revelio | 回答: n | 正誤: × | モード: ×
    ○
[2025-05-25 13:17:32] 問題: Conjures and projects Lord Voldemort's Dark Mark | 表示名:
    Obliviate | 正解: Morsmordre | 回答: 4 | 正誤: ×
     | モード: 4択
[2025-05-25 11:36:42] 問題: Links objects together for better communication | 表示名:
    Geminio | 正解: Protean Charm | 回答: 2 | 正誤: ×
     | モード: 4択

Please choose an option:
[1] Take a quiz
[2] Change mode
[3] View history
[4] Clear history
[5] Exit
Enter a number: 5
```

3 解説

ここからは、実行結果に基づいて解説を行う。

まず、以下の部分の解説を行う。

```
  Please choose an option:
[1] Take a quiz
[2] Change mode
[3] View history
[4] Clear history
[5] Exit
Enter a number: 1
説明: Conjures and projects Lord Voldemort's Dark Mark
1: Morsmordre
2: Avada Kedavra
3: Mucus Ad Nauseam
```

```
4: Obliviate
正しい名前の番号を選んでください（1-4）：数字で1-4を入力してください: 4
不正解。正解は 1: Morsmordre です。
```

この部分は、クイズの出題と回答の部分である。ユーザは、1 を入力することでクイズを開始することができる。クイズでは、ハリーポッターに登場する魔法の説明が表示され、その説明に対応する魔法の名前を選択肢から選ぶ形式である。選択肢は、1 から 4 までの番号で表示され、ユーザはその中から正しい番号を選ぶ必要がある。不正解の場合は、正解の番号とその魔法の名前が表示される。

実際にこの部分のロジックを解説を行う。まず、HTTP.Java にある「fetchAndSaveSpells」という関数を用いてハリーポッター api から情報を取得して、事前に構築した Character.java のクラスに保存する。その後、保存した情報を jbdc を用いてデータベースに保存する。ここまでの処理は、実際にプロダクトを動かす前に行う必要がある。最初にプロダクトを起動する際には EntryPoint.java の「main」関数を実行する。その後、QuizManager.java の「startQuiz」関数を実行することでクイズが開始される。そして、関数の実行により AskQuestion.java の「start」関数が呼び出される。それにより、クイズの出題が行われる。

実際にクイズの出題のロジックは、2 択の場合はデータベースからランダムにデータを取得して、その説明と名前のペアを保存しておく、その後、ユーザに説明と名前を表示し、それらが一致するかどうかを出題する。これらの情報は、事前に作成した QuizHistory.java の History クラスに保存後、History テーブルに保存される。 4 択の場合は、データベースからランダムに 4 つのデータを取得して、そのうち 1 つを正解として、残りの 3 つを不正解として選択肢を作成する。

次に、以下の部分の解説を行う。

```
  Please choose an option:
[1] Take a quiz
[2] Change mode
[3] View history
[4] Clear history
[5] Exit
Enter a number: 2
--- Change Quiz Mode ---
Current mode: 4択モード
Select new mode:
[0] 4択問題モード
[1] 2択問題モード
Your choice: 1
モードを 2択問題モード に変更しました。
```

この部分は、クイズのモードを変更する部分である。この部分では課題の要件である GoF のデザインパターンを使用している。実際に使用しているデザインパターンは、4 択の出題形式と 2 択の出題形式という複数のアルゴリズムを切り替える、Strategy パターンを使用した。

具体的には、EntryPoint.java のにより QuizManager.java の「changeMode」関数が呼び出される。その後、QuizManager.java の「changeMode」関数により AskQuestion.java に使用される変数を変更することで

出題形式を変更することができる。

次に、以下の部分の解説を行う。

```
  Please choose an option:
[1] Take a quiz
[2] Change mode
[3] View history
[4] Clear history
[5] Exit
Enter a number: 1
説明: Reveals the presence of another person
名前: Homenum Revelio
この説明と名前の組み合わせは正しいですか？（〇: y / ×
    : n）: y または n で入力してください: n
不正解。正解は 〇 です。
```

この部分は、2 択の出題形式でのクイズの出題と回答の部分である。先ほどの解説で説明したため、ここでは詳細な解説は省略する。

次に、以下の部分の解説を行う。

```
  Please choose an option:
[1] Take a quiz
[2] Change mode
[3] View history
[4] Clear history
[5] Exit
Enter a number: 3
履歴を表示します。
--- Quiz History ---
[2025-05-25 13:18:01] 問題: Reveals the presence of another person | 表示名: Homenum
    Revelio |  正解: Homenum Revelio | 回答: n | 正誤: × | モード: ×
    〇
[2025-05-25 13:17:32] 問題: Conjures and projects Lord Voldemort's Dark Mark | 表示名:
    Obliviate | 正解: Morsmordre | 回答: 4 | 正誤: ×
     | モード: 4択
[2025-05-25 11:36:42] 問題: Links objects together for better communication | 表示名:
    Geminio | 正解: Protean Charm | 回答: 2 | 正誤: ×
     | モード: 4択
```

この部分は、クイズの履歴を表示する部分である。

QuizManager.java の「showHistory」関数が呼び出されることで、QuizHistory.java の「getHistory」関数が呼び出され、出題の際に History クラスに保存されていた、出題の説明、表示名、正解、回答、正誤、モードを取得する。その後、取得した履歴を表示する。

次に、以下の部分の解説を行う。

```
  Please choose an option:
[1] Take a quiz
[2] Change mode
[3] View history
[4] Clear history
[5] Exit
Enter a number: 5
```

この部分は、プログラムを終了する部分である。

EntryPoint.java の Switch 文により、ユーザが 5 を入力した場合はプログラムを終了する。

以上が、実行結果に基づく解説である。また、実行結果では示していないが、EntryPoint.java の起動後に 4 を入力すると、History テーブルのデータを全て削除することができる。

今回のプロダクトは、ハリーポッター api を使用しており、ハリーポッターに登場する魔法の名前を当てるクイズゲームである。

プログラムの実行結果を見てわかるように、クイズの出題形式を 4 択と 2 択で切り替えることができる。

また、クイズの履歴を表示することができる。

今回のプロダクトは、Java 17 で新たに採用された Pattern Matching for switch を使用しており、それを使用している部分のコードを以下に示す。

```
1   Object obj = choice;
2       switch (obj) {
3         case Integer i:
4         switch (i) {
5           case 1:
6             QuizManager.startQuiz(scanner);
7             break;
8           case 2:
9             QuizManager.changeMode(scanner);
10            break;
11          case 3:
12            System.out.println("履歴を表示します。");
13            QuizHistory.showHistory();
14            break;
15          case 4:
16            QuizHistory.clearHistory();
17            return;// または break + while 条件に false
18          case 5:
19            System.out.println("Exiting the game. Goodbye!");
20            scanner.close();
21            return; // または break + while 条件に false
22          default:
23            System.out.println("無効な入力です。1〜4を選んでください。");
24        }
```

このコードでは、Pattern Matching for switch を使用して、Object 型の変数 obj を Integer 型にキャストしている。これにより、switch 文の case で Integer 型の変数 i を使用することができる。

4　おわりに

今回のプロダクトは、事前にメンバー内で仕事を細分化し、仕事担当領域を割り当てて開発を行った為、スムーズに開発を進めることができた。また、GitHub を使用してコードの管理を行い、メンバー間でのコミュニケーションを円滑にすることができた。きちんと準備をしたチーム開発は、ストレスが少なく制作が進んでいくので、今後もこのような開発スタイルを続けていきたい。