

Java 開発試験

目次

1. 試験概要 1

1 目的	1
2 テーマ	1
3 試験形式	2

2. 試験詳細 3

1 Ver1.0 機能要件	3
2 Ver2.0 機能要件	7
3 Ver3.0 機能要件	8
4 Ver4.0 機能要件	9
5 注意事項	9

1. 試験概要

1 目的

Java 研修で学んだ講義内容の理解度、習熟度を測ることを目的とする。

2 テーマ

じゃんけんシステムの開発を課題とする。プレイヤーは「グー」「チョキ」「パー」のいずれかの選択肢をすることができる。コンソール上でコンピューターと対戦できるシステムを作成すること。ルールは現実世界のじゃんけんと同様。

(※じゃんけんの様子は下図に示す通り)

現実世界のジャンケンの様子

これから村田さんと山田さんの2人で実際にジャンケンをしてもらいます。
斉藤さんには審判役をお願いします。それではどうぞ！

斉藤：「それでは村田さん対山田さんのジャンケン3回勝負をはじめます。」

斉藤：「まず1回戦目。ジャンケン・ポン！」

村田：「パー！」

山田：「チョキ！」

斉藤：「はい、パーとチョキで山田さんの勝ちです」

斉藤：「続いて2回戦目。ジャンケン・ポン！」

村田：「グー！」

山田：「チョキ！」

斉藤：「はい、グーとチョキで村田さんの勝ちです」

斉藤：「それでは3回戦目。ジャンケン・ポン！」

村田：「グー！」

山田：「パー！」

斉藤：「はい、グーとパーで山田さんの勝ちです」

斉藤：「これで3回勝負が終わりました。2人の勝った回数を聞いてみましょう」

斉藤：「村田さん何回勝ちましたか？」

村田：「1回です」

斉藤：「山田さんは何回勝ちましたか？」

山田：「2回です」

斉藤：「はい、2対1で山田さんの勝ちです！」

開発は Ver1.0、Ver2.0、Ver3.0、Ver4.0 の4段階に分けて行う（必ず前のバージョンを開発してから次のバージョンを開発すること）。各バージョンの仕様は「2. 演習詳細」で示す。

3 試験形式

- ① 本システムの開発は個人で行うものとする（個人演習）。
※ 周囲の受講生とのコミュニケーションは許可します。

- ② 作成するプロジェクトの名前は以下の命名規則に従うこと。

janken_氏名(ローマ字)

(例：東京太郎の場合)

janken_tokyotaro

- ③ 作成するパッケージの名前は以下の命名規則に従うこと。

Ver1.0 standard

Ver2.0 enhancement

Ver3.0 excellence

Ver4.0 extra

- ④ 作成するファイルは以下の命名規則に従うこと。
※「2. 演習詳細」で示されていない場合はコーディング規約を意識した名前にすること。
- ⑤ これまでに学習したコーディング規約に従うものとする。
※詳しくは「コーディング規約_東京 IT スクール」を参考にすること。
- ⑥ インターネット上で調べることを許可する
※ただし、質問サイト上での試験内容の質問は外部流出に繋がるため避けること。
※開発試験の疑問点はサポーターに確認すること。

2. 試験詳細

1 Ver1.0 機能要件

ユーザーがじゃんけんの手を選択して対戦する機能を実装する。じゃんけんの手を選択は、ユーザーにじゃんけんの手を（0～2）を入力してもらい、勝負を3回繰り返す。コンピューターの手はランダムで決める。3回勝負した結果が、ユーザー側が勝利した場合は、「○対○でプレイヤーの勝ちです！」と出力しシステムを終了させる。コンピューター側が勝利した場合は、「○対○でコンピューターの勝ちです！」と出力しシステムを終了させる。引き分けだった場合は、「○対○で引き分けです！」と出力しシステムを終了させる。○の部分には勝ち数と負け数を表示すること。

※補足

Ver1.0 ではクラスとメソッドは一つのみで構いません。

例外処理は不要です、追加機能で実装してください。

表示形式は下記イメージを参考に、正としてください。

クラス名は自由とします。命名規則に従い、わかりやすい名前を付けてください。

不明な点はサポーターに相談すること。

画面（コンソール）のイメージは以下の通り。次ページに続きます。

① プレイヤーが勝利した場合の画面（>の右にユーザーの手を入力する）

【ジャンケン開始】

【1回戦目】

プレイヤーの手を決めてください

（グー：0 チョキ：1 パー：2）

> 0

グー vs. チョキ

プレイヤーが勝ちました！

【2回戦目】

プレイヤーの手を決めてください

（グー：0 チョキ：1 パー：2）

> 1

チョキ vs. パー

プレイヤーが勝ちました！

【3回戦目】

プレイヤーの手を決めてください

（グー：0 チョキ：1 パー：2）

> 0

グー vs. パー

コンピューターが勝ちました！

【ジャンケン終了】

2対1でプレイヤーの勝ちです！

② コンピューターが勝利した場合の画面（>の右にユーザーの手を入力する）

【ジャンケン開始】

【1 回戦目】
プレイヤーの手を決めてください
(グー : 0 チョキ:1 パー:2)
> 2
パーー vs. チョキ
コンピューターが勝ちました！

【2 回戦目】
プレイヤーの手を決めてください
(グー : 0 チョキ:1 パー:2)
> 0
グー vs. パー
コンピューターが勝ちました！

【3 回戦目】
プレイヤーの手を決めてください
(グー : 0 チョキ:1 パー:2)
> 0
グー vs. パー
コンピューターが勝ちました！

【ジャンケン終了】
0 対 3 でコンピューターーの勝ちです！

③ 引き分けだった場合の画面（>の右にユーザーの手を入力する）

【ジャンケン開始】

【1 回戦目】

プレイヤーの手を決めてください

（グー：0 チョキ:1 パー:2）

> 0

グー vs. チョキ

プレイヤーが勝ちました！

【2 回戦目】

プレイヤーの手を決めてください

（グー：0 チョキ:1 パー:2）

> 2

パー vs. パー

引き分けです！

【3 回戦目】

プレイヤーの手を決めてください

（グー：0 チョキ:1 パー:2）

> 0

グー vs. パー

コンピューターが勝ちました！

【ジャンケン終了】

1 対 1 で引き分けです！

2 Ver2.0 機能要件

Ver1.0 で作成したクラスの main メソッドから throws IOException を外し、main メソッド内で例外処理を行うように変更する。また、ユーザーが入力したじゃんけんの手が数値で入力されているが、誤っている場合（4 や 100 や -5）は再度入力させるように変更する。
（※パッケージを分け、クラスを再度作成してください）

2-1 例外発生時に出力するメッセージ

システムエラーが発生しました。

システムを終了します。

2-2 じゃんけんの手を再度入力させる時に出力するメッセージ

入力した値が誤っています、再度入力してください

3 Ver3.0 機能要件

Ver2.0 で作成したシステムを、オブジェクト指向に乗っ取ったシステムに改修すること。
下記で指定されたように、クラスとメソッドを分割したシステムにすること。表示内容自体は Ver2.0 から変更はありません。

このシステムの次の 3 つのクラスを連携させて作成する。

SystemMain クラス (SystemMain.java) …… システムのメインとなるクラス
JankenJudge クラス (JankenJudge.java) …… じゃんけんの判定クラス
Player クラス (Player.java) …… プレイヤークラス、キーボード入力処理などを実装
また、コンピューターのじゃんけんの手も Player クラスに実装すること

SystemMain クラスの仕様を以下に示す。

SystemMain クラス

フィールド	なし	利用しない。
メソッド	Main	システムを開始した後、Player クラスを呼び出し、じゃんけんの手を入力させる。その後、コンピューターの手をランダムで決めた後に、JankenJudge クラスを呼び出し判定を行う。 3 回繰り返し、勝敗を表示し、システムを終了させる。 例外発生時や予期せぬエラーがあった場合はその旨のメッセージを表示してシステムを終了させる。
戻り値	Void	利用しない。
引数	String[] args	利用しない。

※補足

JankenJudge クラスと Player クラス内のメソッド記述は自由とする。

個数と命名は問わない、ただしコーディング規約を参考にし、わかりやすい名前を付けること。

4 Ver4.0 機能要件

Ver3.0 に下記機能を追加すること。クラスやメソッドはこれまでの学習内容を参考に独自で追加して構わない。ただし、不要なクラスや処理を記述しないこと。

- ① 対戦回数を自由に選択できるようにする。
コンソール上に表示する内容は自由に決めて構わない。ユーザーの使いやすさを考えること。
- ② コンピューターの手をランダムではなく戦略性を実装すること。
例：同じ手を連続でださない、ユーザーの手を真似る
パターンに関しては自由に決めて構わない。何でも良いのでひとつ実装をすること。
- ③ 対戦人数を自由に選択できるようにする。
コンピューターの人数を増やし、判定を実施する。
現実世界の多人数のじゃんけんと同様のルールで機能実施すること。
コンソール上に表示する内容は自由に決めて構わない。ユーザーの使いやすさを考えること。
- ④ 上記 3 点の機能を実施できた場合は、オリジナルの機能を実装を試みること。
不要なバグを発生させないように注意。

5 注意事項

- ① 採点基準は下記とする。
 - ・ Ver4.0 含め、すべての機能が実装されている。
 - ・ クラスやメソッドが正しく分けられている
 - ・ コメントと javadoc が正しく付与されている
 - ・ 例外処理がきちんと実装されている
 - ・ 命名規則などがコーディング規約通りに実装されている
- ② 納品はプロジェクトを zip 形式で圧縮し、提出すること。
- ③ 開発が途中の状態でも必ず提出。また、提出時間はサポーターが提示した時間を厳守すること。
- ④ システムの仕様に関して不明な点があればサポーターに適宜質問して良い。技術的に不明な点があれば、まずは自分で調べ、それでも分からない場合はサポーターに質問しても良い。システムのコアの部分は答えることができない可能性があるので注意。