

問題・じゃんけんゲームをつくる

Ver 1.0

Seiichi Nukayama

2022 年 11 月 28 日

目次

1	設計	1
1.1	コマンド版じゃんけんゲーム	1
1.2	どんなクラスが必要か?	1
1.3	Com クラス	2
1.4	Human クラス	2
1.5	抽象クラス	2
1.6	勝敗判定メソッド	3
1.7	勝ち負け判定アルゴリズム	4
1.8	処理の流れ	4

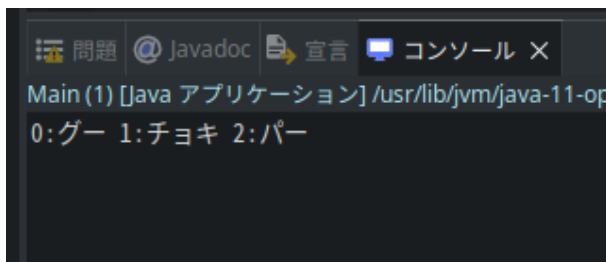
1 設計

以下の設計をもとに、じゃんけんアプリを作成する。

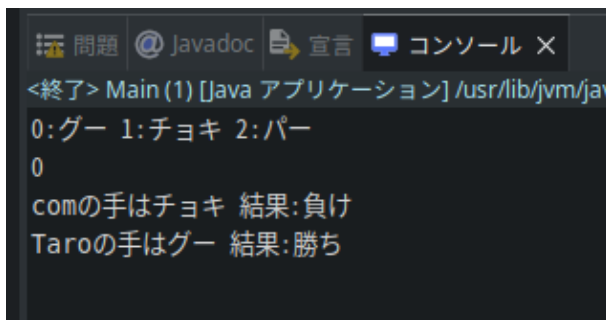
1.1 コマンド版じゃんけんゲーム

今から考えるゲームアプリは、コマンド実行型アプリである。

Main クラスを Eclipse で実行すると、以下のようにコンソールに出力される。



0 か 1 あるいは 2 を入力すると、次のように出力される。



1.2 どんなクラスが必要か？

プレイヤーは 2 人とする。しかし、3 人以上にも拡張できるようにしておく。

プレイヤー 1(人間)	Human クラス
プレイヤー 2(コンピュータ)	Com クラス

それぞれのクラスには、どんな設定 (属性) が必要だろうか？

- 名前 (name)。プレイヤーに名前をつけることができる。
初期値は " コム"、"あなた" とする。
- ジャンケンの手 (hand)。(0: グー、1: チョキ、2: パー)
整数の方が勝敗の判定が楽。
- 勝敗 (result)。各インスタンスが自分の勝敗を覚えておくようにする。
値は "win"、"lose"、"draw" とする。

次は、必要なメソッド (操作) を考えてみる。

- 自分の手と勝敗結果を表示するメソッド。(printResult())

例えば以下のような。

“コムの手はチョキ 結果: 負け”

“あなたの手はグー 結果: 勝ち”

- 次の手を決定するメソッド。(nextHand())

Com クラスの場合は、ランダムに決める。(0...2)

Human クラスの場合は、ユーザーに数字を入力してもらう。

1.3 Com クラス

以下のようなクラス図が考えられる。

Com	
name	: String
hand	: int
result	: String
Com() Com(name:String)	
nextHand()	: void
printResult()	: void

1.4 Human クラス

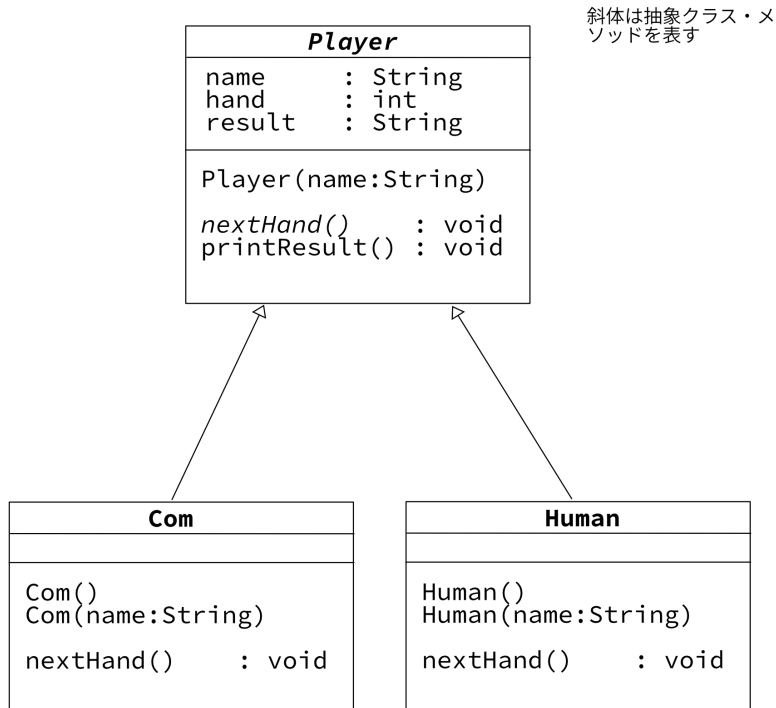
Human クラスについて考えてみると、nextHand() メソッドの内容が違っただろうことを除けば、他は全く同じである。

Human	
name	: String
hand	: int
result	: String
Human() Human(name:String)	
nextHand()	: void
printResult()	: void

1.5 抽象クラス

Com クラスと Human クラスで共通なフィールドとメソッドを抽象クラスとしてまとめることができる。
実装内容が違うメソッドは、抽象メソッドにしておいて、各具象クラスでオーバーライドすることにする。

じゃんけんゲーム・クラス図



1.6 勝敗判定メソッド

勝敗を判定するメソッドを `judge()` とする。

このメソッドをどのクラスが実装するのだが、ここでは `Game` クラスというのを考えて、それに実装させることにしたい。

このメソッドの内容だが、`Com` クラスのインスタンスと `Human` クラスのインスタンスを引数にとり、それぞれの `hand` フィールドをみて、勝敗を判定することにする。

たとえば、`Com` の勝ちだとすると、`Com` の `result` フィールドに "win" という文字列をセットし、`Human` の `result` フィールドに "lose" という文字列をセットすることになる。

このメソッド定義を次のように書くことができるが、

```

1 public void judge(Com com, Human human) {
2     ... (処理内容) ...
3 }
4
5

```

Java の "多態性" を使って、次のように書ける。

```

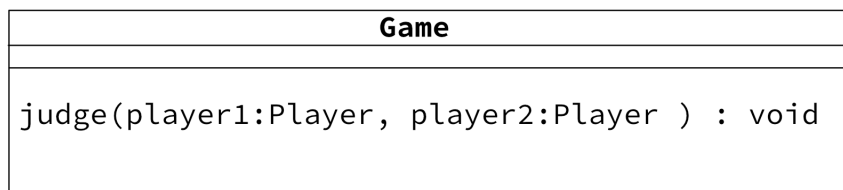
1 public void judge(Player player1, Player player2) {
2     ... (処理内容) ...
3 }

```

```
4 }  
5 }
```

つまり、抽象クラス Player の規格を満たすクラスであれば、どんなインスタンスでも引数として指定できるのである。

クラス図はこのようになる。



1.7 勝ち負け判定アルゴリズム

仮りに int 型の comHand と int 型の humanHand として考えてみる。それぞれには、0, 1, 2 のどれかがセットされている。

引分けの場合 — `comHand == humanHand` である。

comHand の勝ちの場合は、

comHand	humanHand
0	1
1	2
2	0

となる。

このとき、`(comHand + 1) % 3 == 0` という式が成り立つ。

それ以外は、humanHand の勝ちである。

1.8 処理の流れ

以下は Main クラスでの処理の流れである。

1. Com クラスのインスタンスを生成。
2. Human クラスのインスタンスを生成。
3. Com クラスの名前を設定。
4. Human クラスの名前を設定。
5. Com クラスの次の手を決め、それを hand に設定する。
6. Human クラスの次の手は、ユーザーが入力する。(0, 1, 2)
それを hand に設定する。
7. Game クラスのインスタンスを生成。
8. Game クラスの judge() メソッドに、引数として、Com クラスと Human クラスのインスタンスを渡

す。

すると、勝敗結果がそれぞれのインスタンスの result フィールドにセットされる。

9. Com クラスの printResult() メソッドを実行して、結果を表示する。

10. Human クラスの printResult() メソッドを実行して、結果を表示する。

