

(1) プログラム名

ジャンケンをして勝利数をカウントするプログラム

(2) 目的・方針

本プログラムは、Java の G U I を用いて簡易的にジャンケンをし、その勝敗について勝利数を数えて表示するものである。

授業で学んだ JButton やリスナーを使用しながら、Web サイト上で掲載されていたものを再現し、さらに発展させて作る方針で進めました。

(3) 説明に用いるスクリーンショット

```
1  import javax.swing.JFrame;
2  import javax.swing.JButton;
3  import javax.swing.JPanel;
4  import javax.swing.JLabel;
5
6  import java.awt.Container;
7  import java.awt.Font;
8  import java.awt.event.ActionListener;
9  import java.awt.event.ActionEvent;
10 import java.awt.Color;
11 import java.awt.BorderLayout;
12 import java.awt.FlowLayout;
13 import java.awt.GridLayout;
14 import java.util.Random;
15
16 public class Miyazawa1J22F164 extends JFrame implements ActionListener {
17     // JFrameを引き継いでいるから、Miyazawa1J22F164クラスはJFrameの親になっ
    た、という理解
18
19     // メソッドを跨いで使うものは、あらかじめ以下で宣言しておく。
20     Container c;
21     JButton rock, scissors, paper;
22     String[] hands = { "グー", "チョキ", "パー" };
23     JLabel gameLabel;
24     JLabel explanation;
25
26     JLabel playerWin;
27     JLabel playerWinNumDisplay;
28     static int playerWinNum = 0;
29     JLabel enemyWin;
30     JLabel enemyWinNumDisplay;
31     static int enemyWinNum = 0;
32
33     static String serif;
34
    Run | Debug
35     public static void main(String[] args) {
```

```
35 public static void main(String[] args) {
36     Miyazawa1J22F164 mf = new Miyazawa1J22F164();
37     mf.setVisible(b: true);
38 }
39
40 // 基本的な表示の構造を記述しているメソッド
41 public Miyazawa1J22F164() {
42     // JFrame f = new JFrame();
43     setSize(width: 800, height: 600); // フレームのサイズを設定
44     setResizable(resizable: false); // フレームのサイズ変更を不可に
45     setTitle(title: "ジャンケン"); // フレームのタイトルを設定
46     setLocationRelativeTo(c: null); // 画面の中心にフレームを出現させる
47     ggg
48     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // フレームを消せばプ
49     // ヘッダーパネル
50     JPanel header = PanelMaker.makePanel(new Color(r: 180, g: 0,
51     b: 255), width: 800, height: 100);
52     // ヘッダーを作る(背景色、サイズを指定)
53     header.setLayout(new BorderLayout()); // ヘッダーの内部はBorderLayout
54     explanation = TextMaker.makeText(str: "ジャンケンしよう！",
55     size: 25, Color.WHITE); // ラベルを作成(ヘッダーで表示したい、ゲーム
56     // 文字の色を白色に
57     // 文字のサイズ変更
58     explanation.setHorizontalAlignment(JLabel.CENTER); // 説明文を、横方
59     // 向に中心に置く
60     explanation.setVerticalAlignment(JLabel.CENTER); //// 説明文を、縦方
61     // 向に中心に置く
62     header.add(explanation); // ヘッダーに説明文を組み込み
63
64     add(header, BorderLayout.NORTH); // ヘッダーをフレームに組み込む、
65     BorderLayoutで北
```

```

63
64 // センターパネル
65 JPanel center = PanelMaker.makePanel(Color.WHITE);
66 // センターパネルを作る(背景色を指定)
67 center.setLayout(new BorderLayout()); // センターパネルの内側はボーダ
   レイアウト
68 gameLabel = TextMaker.makeText(str: "じゃんけん...", size: 58,
   Color.BLACK); // 真ん中でじゃんけんの誘導と相手の手の説明をするラベル
   を作成
69 // 文字の色は黒で
70 // 文字のサイズ変更
71 gameLabel.setHorizontalAlignment(JLabel.CENTER); // 横方向の中心
72 gameLabel.setVerticalAlignment(JLabel.CENTER); // 縦方向の中心
73 center.add(gameLabel); // センターパネルにgameLabelを組み込み
74 add(center, BorderLayout.CENTER); // センターパネルをフレームに組み込
   み
75
76 // 勝ち星カウントパネル
77 JPanel countWin = PanelMaker.makePanel(Color.BLACK, width: 150,
   height: 600); // 背景色とサイズを指定してパネルを作成
78 countWin.setLayout(new FlowLayout()); // レイアウトはFlow
79 playerWinNumDisplay = TextMaker.makeText(Integer.valueOf
   (playerWinNum).toString(), size: 50, Color.WHITE); // ゲーム参加者の
   勝ち数をtoStringで文字列にして表示するラベル。色とサイズを指定
80 playerWin = TextMaker.makeText(str: "<html><body>あなたが<br/>勝った
   回数</body></html>", size: 20, Color.WHITE); // 「あなたが勝った回
   数」と表示するラベル
81 enemyWinNumDisplay = TextMaker.makeText(Integer.valueOf(enemyWinNum).
   toString(), size: 50, Color.WHITE); // ゲーム対戦相手の勝ち数を
   toStringで文字列にして表示するラベル。色とサイズを指定
82 enemyWin = TextMaker.makeText(str: "<html><body>相手が<br/>勝った回数
   </body></html>", size: 20, Color.WHITE); // 「相手が勝った回数」と表
   示するラベル
83 countWin.setLayout(new GridLayout(rows: 4, cols: 1)); // 縦一列に表示
   する
84 countWin.add(playerWinNumDisplay);

```

```

85     countWin.add(playerWin);
86     countWin.add(enemyWinNumDisplay);
87     countWin.add(enemyWin);
88     add(countWin, BorderLayout.EAST);
89
90     // フッターパネル
91     JPanel footer = PanelMaker.makePanel(new Color(r: 180, g: 0,
92     b: 255), width: 800, height: 100);
93     // フッターを作る()
94     footer.setLayout(new GridLayout());
95
96     // フッターに表示する、じゃんけんの選択肢のボタンをループ文で作る
97     for (String hand : hands) {
98         JButton button = new JButton(hand); // JButtonをインスタンス化
99         button.setFont(new Font(name: "MS ゴシック", Font.BOLD,
100         size: 24)); // ボタンの文字のフォントとサイズを決定
101         button.addActionListener(this);
102         footer.add(button);
103     }
104
105     add(footer, BorderLayout.SOUTH); // フッターをフレームに組み込む、
106     BorderLayoutで南
107
108     // ジャンケンする相手の手を決めるメソッド
109     static int getEnemyHand() {
110         Random random = new Random(); // ランダムクラスをインスタンス化
111         int hand = random.nextInt(bound: 3); // 敵のじゃんけんの手を数字で取得
112         return hand; // 敵の手を返す
113     }
114
115     static void getResult(int playerHand, int enemyHand) {
116         serif = "";
117         int result = (playerHand - enemyHand + 3) % 3; // 数値からあいこor勝

```

```

117 // if文の分岐で、ヘッダーで表示させる文章を変える。switch文でも良かった。
118 if (result == 0) {
119     serif = "あいこ!";
120 } else if (result == 1) {
121     serif = "負け...";
122     enemyWinNum++; // 敵の勝ち数をプラス1する
123 } else if (result == 2) {
124     serif = "勝ち!!!";
125     playerWinNum++; // 自分の勝ち数をプラス1する
126 }
127 }
128
129 @Override
130 // ボタンが押された時の反応
131 public void actionPerformed(ActionEvent e) {
132     int enemyHandNum = getEnemyHand(); // 対戦相手の手を決めて数値で格納
133     String enemyHand = hands[enemyHandNum]; // 数値を文字に変換
134     gameLabel.setText(enemyHand); // 文字をセンターパネルのラベル入れる
135
136     // プレイヤーはボタンを押すだけで、ボタンには文字列しか情報が付属して
    いない。そのため、文字列との一貫性から、プレイヤーが選んだ選択肢を探
    し、数値に置き換える。(for文とif文で)
137     int playerHandNum = 0;
138     for (int i = 0; i < hands.length; i++) {
139         if (hands[i] == e.getActionCommand()) {
140             playerHandNum = i;
141             break;
142         }
143     }
144 }
145 getResult(playerHandNum, enemyHandNum); // getResultメソッドを使って
    相手と自分の手の数値から勝敗を判断する
146 playerWinNumDisplay.setText(Integer.valueOf(playerWinNum).toString
    ()); // 自分の勝ち数の表示を更新する
147 enemyWinNumDisplay.setText(Integer.valueOf(enemyWinNum).toString
    ()); // 敵の勝ち数の表示を更新する

```

```
146      相手と自分の手の数値から勝敗を判断する
      playerWinNumDisplay.setText(Integer.valueOf(playerWinNum).toString
      ()); // 自分の勝ち数の表示を更新する
147      enemyWinNumDisplay.setText(Integer.valueOf(enemyWinNum).toString
      ()); // 敵の勝ち数の表示を更新する
148
149      explanation.setText(serif); // ヘッダーで表示する勝ち負けの状況説明を
      更新する。
150  }
151 }
```

TextMaker.java ファイル

```
Miyazawa1J22F164.java  TextMaker.java ×  PanelMaker.java
期末課題 > TextMaker.java > TextMaker > makeText(String, int, Color)
1  import java.awt.Color;
2  import java.awt.Font;
3
4  import javax.swing.JLabel;
5
6  public class TextMaker {
7
8      // テキストを作るメソッド
9      static JLabel makeText(String str, int size, Color color) {
10         JLabel label = new JLabel(str); // インスタンス化
11         label.setForeground(color); // テキストの文字を設定
12         label.setFont(new Font(name: "MS ゴシック", Font.BOLD, size)); // テキストのフォントを設定
13         label.setHorizontalAlignment(JLabel.CENTER); // テキストの位置を、水平方向の中心にする
14         label.setVerticalAlignment(JLabel.CENTER); // テキストの位置を、垂直方向の中心にする
15         return label; // 作ったテキストを返す
16     }
17
18 }
```

PanelMaker.java ファイル


```
J Miyazawa1J22F164.java  J TextMaker.java  J PanelMaker.java ×
期末課題 > J PanelMaker.java > PanelMaker > makePanel(Color)
4  import javax.swing.JPanel;
5
6  public class PanelMaker {
7      // サイズを指定してパネルを作るメソッド
8      static JPanel makePanel(Color color, int width, int height) {
9          JPanel panel = new JPanel(); // インスタンス化
10         panel.setBackground(color); // パネルの背景色を設定
11         panel.setPreferredSize(new Dimension(width, height)); // パネルのサイズを設定
12         return panel; // パネルを返す
13     }
14
15     // サイズ指定なしでパネルを作るメソッド
16     static JPanel makePanel(Color color) {
17         JPanel panel = new JPanel(); // インスタンス化
18         panel.setBackground(color); // パネルの色を変更する
19         return panel; // 作ったパネルを返す
20     }
21 }
```

(4) 概要

(6)でももう一度示すが、全体の構成として、次の web サイトに掲載されていたものを利用した。

https://original-game.com/java-janken-game-program/#m_heading-4

基本的には、コメントアウトでその都度説明している通りだが、簡単に大きな流れを説明する。

今回の期末課題は、3つのファイルから構成されている。

TextMaker ファイルでは、引数から文字の色や大きさを設定しながらラベルを作成する makeText メソッドを定義している。

PanelMaker ファイルでは、背景色やサイズを設定しながらパネルを作成する makePanel メソッドを定義している。このメソッドは二つあり、引数の数によって、サイズの指定の有無に対応している。

そして Miyazawa1J22F164.java のファイルが main メソッドを含む、中心となるファイルだ。

Miyazawa1J22F164 クラスは、JFrame を継承している。(16 行目

また、他のクラスと共通して使う変数は、main メソッドの記述に入る前にあらかじめ宣言している。

(20-33 行目

main メソッドでは、Miyazawa1J22F164 メソッドをインスタンス化し、可視化するためのメソッドにしてある。(35-38 行目

その後は順に、フレームの各種設定(42-46 行目)、ヘッダーの作成と各種設定(48-61 行目)、中心のパネル(センターパネル)の作成と各種設定(63-73 行目)、じゃんけんの勝ち星を表示するパネルの作成と各種設定(75-87 行目)、フッターの作成と各種設定(89-102 行目)となっている。

その下には、勝負する相手の手をランダムで決めるメソッドを記述している(106-111 行目)。

その下には、じゃんけんの勝敗を判定するメソッドを記述している(113-126 行目)。

その下には、フッターにあるボタンが押された時の反応を記述している。(128-149 行目)。

フッターのボタンが押されることで、リスナーをたどって、自分の手と相手の手が数値化され、計算で勝負が決まり、表示する文字や勝利数のカウントなどが分岐によって処理されている。

勝利数をそのまま表示する部分では、扱い上は int 型だが、表示する前に toString によって String 型に変換されている。その際、Integer.valueOf(enemyWinNum).toString()と、valueOf を用いた。

(5) 必須項目を満たしている部分の説明

(1) 独自のクラスを作成し、利用している

main メソッドが含まれているファイル以外に TextMaker.java ファイルと PanelMaker.java ファイルを用意し、それぞれの中に独自のクラスを作成した。また、Miyazawa1J22F164.java ファイルの中で、それぞれのクラスに含まれるメソッドを呼び出し、利用した
(49,53,64,67,76,78,79,80,81,90 行目)。

(2) GUI(Swing)を用いている

main メソッドを含む Miyazawa1J22F164 クラスは、Swing の JFrame を継承している(16 行

目).

PanelMaker.java ファイルの PanelMaker クラスでは、JPanel をインスタンス化し、作ったパネルにさまざまな設定を付け足すメソッドを記述している。

その他、各部分で Swing を利用している。

(6) 参考としたプログラミング

全体のジャンケンをする仕組みや GUI の構成などは、以下のサイトを参考にした。

サイト名：オリジナルゲーム.com 「【絶対できる！】Java でのじゃんけんゲームの作り方」

https://original-game.com/java-janken-game-program/#m_heading-4