

Work5-1 (P116)

Tacticsインタフェースを利用してじゃんけんの手を決めるようにする

Playerクラス

```
/**
 * じゃんけんでプレイヤーのする動作
 * @author yuna
 */
public class Player{
    //じゃんけんの手を表す定数
    public static final int STONE = 0; //グー
    public static final int SCISSORS = 1; //チョキ
    public static final int PAPER = 2; //パー

    //-----
    //プレイヤークラスの属性
    //-----

    //フィールド
    //プレイヤーの名前
    private String name_;
    //プレイヤーの勝った回数
    private int winCount_ = 0;

    //与えられえた戦略
    private Tactics tactics_;

    //コンストラクタ
    //引数なしのコンストラクタも用意しておく
    Player(){
    /**
     * プレイヤーの名前を引数にとるコンストラクタ
     * @author yuna
     * @param name
     */
    Player(String name){
        //名前を表すフィールドを初期化
        this.name_ = name;
    }
    /**
     * プレイヤーの名前と勝利回数を引数にとるコンストラクタ
     * @param name_
     * @param winCount_
     */
    Player(String name_, int winCount_){
        //名前を表すフィールドを初期化する
        this(name_);
        //勝利回数を表すフィールドを初期化
```

```

        this.winCount_ = winCount_;
    }

    /**
     * 戦略を設定する
     * @param tactics_
     * @author yuna
     * @date 2023/6/19
     */
    public void setTactics(Tactics tactics_) {
        //戦略を引数の設定にする
        this.tactics_ = tactics_;
    }

    //-----
    //プレイヤークラスの操作
    //-----

    /**
     * @return プレイヤーの出す手を返す
     */
    public int readTactics() {
        //プレイヤーに戦略を渡す
        int playerHand = tactics_.readTactics();
        //プレイヤーの出す手を返す
        return playerHand;
    }

    //playerクラスの動作
    //じゃんけんの手を出す
    /**
     * @author yuna
     * @param なし
     * @return じゃんけんの手を値にして返す<br>
     * グー : 0、チョキ:1、パー:2
     */
    public int showHand() {
        //プレイヤーの出す手を戦略をもとに決める
        int playerHand = readTactics();
        //プレイヤーの出す手を返却
        return playerHand;
    }

    /**
     * 審判から勝敗を聞いて勝っていれば勝利回数を数える
     * @author yuna
     * @param result true:勝ち、false:負け
     * @return 無し
     */

```

```

    public void notifyResult(boolean result) {
        //勝った場合
        if(result == true) {
            //勝利回数カウントする
            winCount_ +=1;
        }
    }

    /**
     * 勝利回数を答える
     * @author yuna
     * @return 勝った回数を返却する
     */
    public int getWinCount() {
        //勝利回数を返却
        return winCount_;
    }
}

```

Tacticsクラス

```

/**
 * じゃんけん戦略インタフェース
 * @author yuna
 * @date 2023/6/19
 */
public interface Tactics {
    /**
     * 戦略を読んでじゃんけんの出す手を返却する
     * Playerクラスで定義されたグー・チョキ・パーと対応した
     * 値を返却する
     * @return じゃんけんの出す手を返却する
     * @author yuna
     * @date 2023/6/19
     */
    public int readTactics();
}

```

Work5-2 (P129)

グー・チョキ・パーを順番に出す戦略クラス「CyclicTactics」を作成する

CyclicTacticsクラス

```
/**
 * グー・チョキ・パーを順番に出す戦略クラス
 * @author yuna
 * @date 2023/6/19
 */
public class CyclicTactics implements Tactics{
    //整数を3つに分けるための変数
    private static int classifyNumber = 0;
    /**
     * グー・チョキ・パーを順番に出す戦略を渡す
     * @return プレイヤーの出す手を返却する
     * @author yuna
     * @date 2023/6/19
     */
    public int readTactics() {
        //プレイヤーの出す手を表す変数
        int playerHand = 0;
        //何回戦目かを表す変数
        int NumberOfGame = Judge.countGames;
        //3の倍数の試合回数めの時
        if(NumberOfGame%3 == 0) {
            //変数をインクリメント
            classifyNumber++;
        }
        //3の倍数戦めの場合
        if(NumberOfGame == 3*classifyNumber) {
            //プレイヤーの出す手をパーにする
            playerHand = Player.PAPER;
        }
        //3k+1戦目の場合
        else if(NumberOfGame == 1 + 3*classifyNumber) {
            //プレイヤーの出す手をグーにする
            playerHand = Player.STONE;
        }
        //3k+2戦目の場合
        else{
            //プレイヤーの出す手をチョキにする
            playerHand = Player.SCISSORS;
        }

        //プレイヤーの出す手を返す
        return playerHand;
    }
}
```

ObjectJankenクラス

```
/**
 * オブジェクト指向によるじゃんけんプログラム
 * @author yuna
 * @createDate 2023/6/16
 */
public class ObjectJanken {

    /**
     * @author yuna
     * @createDate 2023/6/16
     * @param args
     */
    public static void main(String[] args) {
        //審判クラスのインスタンスを生成
        Judge saito = new Judge();
        //プレイヤー 1 の名前
        String player1Name = "murata";
        //プレイヤー 2 の名前
        String player2Name = "yamada";

        //プレイヤー 1 のインスタンスを生成
        Player murata = new Player(player1Name);
        //プレイヤー 2 のインスタンスを生成
        Player yamada = new Player(player2Name);
        //村田の戦略のインスタンスを生成
        Tactics murataTactics = new CyclicTactics();
        //村田に戦略クラスを渡す
        murata.setTactics(murataTactics);
        //山田の戦略のインスタンスを生成する
        Tactics yamadaTactics = new RandomTactics();
        //山田に戦略クラスを渡す
        yamada.setTactics(yamadaTactics);
        //murataとyamadaのじゃんけんを開始する
        saito.startJanken(murata,yamada);
    }
}
```

実行画面

（「CyclicTactics」クラスの確認のために、じゃんけんの回数を多くしてます）

【じゃんけん開始】

【1回戦目】
グーvs.チョキ

murataが勝ちました！

【2回戦目】

チョキvs.チョキ

引き分けです！

【3回戦目】

パーvs.グー

murataが勝ちました！

【4回戦目】

グーvs.チョキ

murataが勝ちました！

【5回戦目】

チョキvs.チョキ

引き分けです！

【6回戦目】

パーvs.グー

murataが勝ちました！

【じゃんけん終了】

4対0でmurataの勝ちです！