

# ネームバトラー【マスター編】

ネームバトラー【マスター編】では、【上級編】で作ったネームバトラーを、さらにゲームとして進化させていく内容になっています。

## ■学習内容

### 1. GameManagerクラスにゲームのシステムを移管する

- ・ゲームを管理するマネージャークラス(GameManager)クラスを作成する
- ・ゲームの処理はすべてマネージャークラスで行うようにする
- ※プログラム実行時に実行される `main()` メソッドでは、`static` がついているため、そこから呼び出すメソッドなどは、無駄に `static` をつけないと呼び出せなくなります。そうすると、正しい設計とはことなる実装になってしまうため、`main()` メソッドでは、GameManager のインスタンスを生成し、ゲームのメイン処理を呼び出すだけにします。

```
public static void main(String[] args) {  
    GameManager gm = new GameManager();  
    gm.Start();  
}
```

### 2. パーティーを管理するための Party クラスを作成する

- ・資料後半にある Party クラスのひな型を利用して作成する

### 3. 3対3のパーティー対戦にする

- ・3対3のパーティーバトルとなるように修正する
- ※プレイヤー情報やパーティー構成は、毎回実行時に入力するのが面倒な場合は、プログラミングで固定にしておいても良い
- ・6人の攻撃順はパーティー関係なく、素早さ順で攻撃を行う
- ・攻撃対象は相手パーティーの生存メンバーからランダムに選択で良い
- ・回復魔法は自分に対してのみ使用で良い

### 4. パーティーの状況で行動を変える処理を実装する

- ・一番ダメージを与えられそうな相手を攻撃するとか、攻撃対象を何かしらのルールに基づいて行うようにする
- 例：HPが低いものを狙う、防御が引くものを狙う、魔法使いを狙う...など
- ・自パーティーメンバーのHPが減っている場合にメンバーに回復魔法を使うようにする
- ※ここで作成した処理が「5. 作戦を5つ作る」の1つになる

### 5. 作戦を5つ作る

- ・ドラクエのような「さくせん(AI)」を5つ作る
- 例：魔法を優先して使う、魔法を節約する、バランスよく、いのちを大事に(回復、補助魔法優先)、1人ずつ集中攻撃、など
- ・作戦は作戦毎にクラスを作成する(作戦の基底クラスと実際の作戦毎のロジックを組み込んだサブクラスを作成する)

### 6. 好きな機能を1つ追加する

- ・ゲームとして面白くなるような機能を1つ追加する
- 例：ジョブ固有の特技を追加する、ガード機能を追加する、職業別の相性を設定する、アイテム(回復アイテム、攻撃アイテムなど)を追加する、など

## ■Partyクラス

パーティークラスは以下のひな型を利用して作成してください。

以下の仕様は変更しないでください。

- ・パーティーメンバーは **members** として **ArrayList<Player>** で保持する
  - ・プレイヤーの追加は **AppendPlayer()** メソッドを使用する
  - ・プレイヤーの削除は **RemovePlayer()** メソッドを使用する
- これら以外は自由に変更（追加、修正）して構いません。

```
import java.util.ArrayList;

public class Party {
    // =====
    // フィールド変数
    // =====
    private ArrayList<Player> members;

    // =====
    // コンストラクタ
    // =====
    Party() {
        members = new ArrayList<Player>();
    }

    // =====
    // Getter / Setter
    // =====
    /**
     * パーティーメンバーを ArrayList で取得する
     */
    ArrayList<Player> GetMembers() {
        return members;
    }

    // =====
    // protected メソッド
    // =====

    // =====
    // private メソッド
    // =====

    // =====
    // public メソッド
    // =====
    /**
     * パーティーにプレイヤーを追加する
     * @param player : 追加するプレイヤー
     */
    public void AppendPlayer(Player player) {
    }

    /**
     * パーティーからプレイヤーを離脱させる
     * @param player : 離脱させるプレイヤー
     */
    public void RemovePlayer(Player player) {
    }
}
```