24\_総合演習④

**課題1**

BINGOゲームで使用するカードを作成して表示するプログラムを作成しなさい。

◆仕様

①カードをマス目状で表示する。

②カードは3×3の9マスとし、各マスに1~9の数値を重複させず、かつランダムに配置する。

③カードの数値データは要素数3×3の二次元配列で扱う。

ファイル名：Kad24\_1.java

＜実行結果＞

BINGOゲーム！

あなたのカードはこちらです。

-------------

| 7 | 9 | 6 |

-------------

| 2 | 1 | 5 |

-------------

| 8 | 4 | 3 |

-------------

＜実行結果＞

BINGOゲーム！

あなたのカードはこちらです。

-------------

| 4 | 1 | 6 |

-------------

| 9 | 5 | 7 |

-------------

| 2 | 3 | 8 |

-------------

|  |  |  |  |
| --- | --- | --- | --- |
|  | 戻り値 | 名前 | 引数 |
| メソッド | int[][] | createCard | なし |

◆createCardメソッドの処理

1~9の数を3×3の二次元配列に重複させず、かつランダムに代入した配列を戻り値として返す。

①以下の配列・変数を宣言します。

int card[][] = new int[3][3]; //数値代入配列

int ransu = 0; //乱数代入変数

boolean[] ransuFlag = new boolean[9]; //乱数フラグ(true:使用済 false:未使用)

②1~9までの乱数を重複させずに二次元配列に代入する。

　二次元配列の行をなぞるループと列をなぞるループの二重ループを使用する。

　乱数を生成し、それが使用されている間生成を繰り返し、未使用の乱数が生成されるまで続ける。

do{

ransu = ra.nextInt(9);

}while(ransuFlag[ransu]);

　未使用の乱数が生成できたら乱数フラグを使用済みに変更し、二次元配列内に順に代入する。

③乱数が代入された二次元配列を戻り値として返す。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 戻り値 | 名前 | 引数 |
| メソッド | void | showCard | int[][] card |

◆showCardメソッドの処理

引数で受け取った二次元配列の値を実行結果のようにマス目状に表示する。

◆mainメソッドの処理

①createCardメソッドを呼び出し、int型二次元配列bingoCardに代入する。

②showCardメソッドを呼び出し、ビンゴカードを表示する。

**課題2**

課題1に機能を追加し、BINGOゲームを行うプログラムを作成しなさい。

◆仕様の追加

①ビンゴから何番のボールが出たかを入力する。

②入力した番号がカードの中に存在する場合、ヒットしたとみなし配列の数値を0に変更する。

③入力する度にカードを表示する。ただし、ヒットした数値はXと表示する。

　※課題1でのshowCardメソッドを修正すること。

④ビンゴチェックを行う。横・縦・斜めのいずれかがビンゴした場合、ゲームを終了する。

⑤入力値は整数のみ受け付ける。整数以外が入力された場合は不正とし、もう一度入力する。

ファイル名：Kad24\_2.java

＜実行結果＞

BINGOゲーム！

あなたのカードはこちらです。

-------------

| 1 | 8 | 3 |

-------------

| 4 | 2 | 9 |

-------------

| 6 | 5 | 7 |

-------------

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？：1

-------------

| X | 8 | 3 |

-------------

| 4 | 2 | 9 |

-------------

| 6 | 5 | 7 |

-------------

※右側へ続く→

※続き

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？：2

-------------

| X | 8 | 3 |

-------------

| 4 | X | 9 |

-------------

| 6 | 5 | 7 |

-------------

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？：7

-------------

| X | 8 | 3 |

-------------

| 4 | X | 9 |

-------------

| 6 | 5 | X |

-------------

BINGO!!

※続き

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？＞6

-------------

| X | X | 8 |

-------------

| X | 4 | X |

-------------

| 3 | X | 2 |

-------------

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？＞2

-------------

| X | X | 8 |

-------------

| X | 4 | X |

-------------

| 3 | X | X |

-------------

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？＞0

-------------

| X | X | 8 |

-------------

| X | 4 | X |

-------------

| 3 | X | X |

-------------

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？＞1.5

入力値が不正です。

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？＞abc

入力値が不正です。

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？＞4

-------------

| X | X | 8 |

-------------

| X | X | X |

-------------

| 3 | X | X |

-------------

BINGO!!

＜実行結果＞

BINGOゲーム！

あなたのカードはこちらです。

-------------

| 5 | 7 | 8 |

-------------

| 1 | 4 | 9 |

-------------

| 3 | 6 | 2 |

-------------

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？＞5

-------------

| X | 7 | 8 |

-------------

| 1 | 4 | 9 |

-------------

| 3 | 6 | 2 |

-------------

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？＞7

-------------

| X | X | 8 |

-------------

| 1 | 4 | 9 |

-------------

| 3 | 6 | 2 |

-------------

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？＞1

-------------

| X | X | 8 |

-------------

| X | 4 | 9 |

-------------

| 3 | 6 | 2 |

-------------

ｶﾞﾗｶﾞﾗｶﾞﾗ...何番の玉が出ましたか？＞9

-------------

| X | X | 8 |

-------------

| X | 4 | X |

-------------

| 3 | 6 | 2 |

-------------

※右側へ続く→

|  |  |  |  |
| --- | --- | --- | --- |
|  | 戻り値 | 名前 | 引数 |
| メソッド | void | checkHit | int[][] card, int ball |

◆checkHitメソッドの処理

ビンゴで出た玉がヒットしている（カード内に存在する）かどうかを調べて、ヒットしていれば配列の値を0に変更する。引数はカード配列と出た玉（入力値）を受け取る。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 戻り値 | 名前 | 引数 |
| メソッド | boolean | checkBingo | int[][] card |

◆checkBingoメソッドの処理

カード内でビンゴ（横・縦・斜めの直線がヒットしている）かどうかを調べて、1列でもビンゴして

いればtrue,してなければfalseを戻り値として返す。

◆mainメソッドの処理

課題1の続きから記述する。

①ビンゴするまで以降の処理を無限ループする。

②何番の玉が出たか入力させ、int型変数bingoBallに代入する。

　この際、文字列が入力された場合の例外処理も記述する。

③checkHitメソッドを呼び出し出た数が当たったかどうか調べる。

④showCardメソッドを呼び出しカードを表示する。

⑤checkBingoメソッドを呼び出し、ビンゴしていた場合はループを抜ける。