14\_静的メソッド

**課題1**

入力した文字を入力した回数分表示するプログラムを作成しなさい。

表示処理は静的メソッドを作成して使用すること。

また、整数以外が入力された場合の例外処理も実装すること。

ファイル名：Kad14\_1.java

＜実行結果＞

表示文字＞専門学校

表示回数＞abc

入力値が不正です。

＜実行結果＞

表示文字＞コンピュータ

表示回数＞4

コンピュータ

コンピュータ

コンピュータ

コンピュータ

＜実行結果＞

表示文字＞ECC

表示回数＞3

ECC

ECC

ECC

|  |  |  |  |
| --- | --- | --- | --- |
|  | 戻り値 | 名前 | 引数 |
| メソッド | void | show | String data, int count |

showメソッドの処理

data文字列をcount回、画面に表示する。

public static void show(String data, int count){

～省略～;

}

mainメソッドの処理

①表示する文字列を入力し、String型変数inStrへ代入する。

System.out.print("表示文字＞");

String inStr = sc.next();

②表示する回数を入力し、int型変数inNumへ代入する。

　この際、整数以外が入力された場合の例外処理を try-catch 文で行い、以降の処理は行わない。

　例外が発生しなかった場合、入力文字列と入力回数を引数にshowメソッドを呼び出す。

try{

int inNum = Integer.parseInt(sc.next());

～省略～;

}catch(NumberFormatException nfe){

System.out.println("入力値が不正です。");

}

**課題2**

ランダムに生成された1~100までの整数を当てるプログラムを作成しなさい。

結果は「小さい」「大きい」「正解」の3パターンとし、正解するまで入力し続けること。

また、整数以外が入力された場合の例外処理も実装すること。

ファイル名：Kad14\_2.java

\*\*\* 数当てゲーム \*\*\*

何番でしょう？(1~100)＞abc

入力値が不正です。

何番でしょう？(1~100)＞50

もっと小さいよ！

何番でしょう？(1~100)＞25

もっと大きいよ！

何番でしょう？(1~100)＞def

入力値が不正です。

何番でしょう？(1~100)＞37

もっと小さいよ！

何番でしょう？(1~100)＞33

正解！

\*\*\* 数当てゲーム \*\*\*

何番でしょう？(1~100)＞50

もっと大きいよ！

何番でしょう？(1~100)＞75

もっと小さいよ！

何番でしょう？(1~100)＞63

もっと大きいよ！

何番でしょう？(1~100)＞69

もっと大きいよ！

何番でしょう？(1~100)＞72

もっと小さいよ！

何番でしょう？(1~100)＞71

正解！

\*\*\* 数当てゲーム \*\*\*

何番でしょう？(1~100)＞50

もっと小さいよ！

何番でしょう？(1~100)＞25

もっと大きいよ！

何番でしょう？(1~100)＞37

もっと小さいよ！

何番でしょう？(1~100)＞30

もっと大きいよ！

何番でしょう？(1~100)＞33

もっと大きいよ！

何番でしょう？(1~100)＞35

正解！

|  |  |  |  |
| --- | --- | --- | --- |
|  | 戻り値 | 名前 | 引数 |
| メソッド | int | check | int n, int ans |

◆checkメソッドの処理

引数n（入力値）が引数ans（答え）と一致しているか判定する。

入力値が答えより大きければ1、小さければ2、一致していれば3を戻り値として返す。

◆mainメソッドの処理

①1~100までの乱数を生成してint型変数ransuに代入する。

final int MAX = 100; //乱数最大値

Random rnd = new Random();

int ransu = rnd.nextInt(～省略～) + 1; //1~MAXまでの乱数生成

②無限ループの中で以下③④の処理を行う

③何番かを入力し、その入力値と乱数を引数にcheckメソッドを呼び出す。

　checkメソッドの戻り値をint型変数resultに代入する。

この際、整数以外が入力された場合の例外処理をtry-catch文で行い、再入力させる。

System.out.print("何番でしょう？(1~" + MAX + ")＞");

int inNum = Integer.parseInt(sc.next());

int result = ～省略～

④戻り値（1or2or3）によって分岐し、結果文字列を表示する。正解の場合はループを抜ける。

**課題3**

西暦と月を入力して、その月の最終日（うるう年も考慮する）を表示するプログラムを作成しなさい。

また、整数以外が入力された場合や存在しない月が入力された場合の例外処理も実装すること。

ファイル名：Kad14\_3.java

＜実行結果＞

ある月の日数を求めます。

西暦＞2025

月＞abc

入力値が不正です。

＜実行結果＞

ある月の日数を求めます。

西暦＞2021

月＞2

2021年2月の最終日は28日です。

＜実行結果＞

ある月の日数を求めます。

西暦＞2020

月＞2

2020年2月の最終日は29日です。

＜実行結果＞

ある月の日数を求めます。

西暦＞1999

月＞13

その月は存在しません。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 戻り値 | 名前 | 引数 |
| メソッド | boolean | isLeapYear | int year |

◆isLeapYearメソッドの処理

引数yearがうるう年の場合はtrue、そうでない場合はfalseを戻り値として返す。

※うるう年とは4で割り切れて、かつ100で割り切れない年、または400で割り切れる年のこと。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 戻り値 | 名前 | 引数 |
| メソッド | int | getDayOfMonth | int year, int month |

◆getDayOfMonthメソッドの処理

引数yearと引数monthを元に月の最終日を求めて、戻り値として返す。

①1~12月の最終日定数配列を宣言する。

final int[] LASTDAYS = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

②2月かつ、うるう年の場合は入力月の最終日に+1した数値を戻り値として返す。

　それ以外の場合は入力月の最終日を戻り値として返す。

◆mainメソッドの処理

①西暦と月の入力値をそれぞれint型変数yearとmonthに代入する。

この際、整数以外が入力された場合の例外処理をtry-catch文で行い、以降の処理は行わない。

②入力した西暦と月を引数としてgetDayOfMonthメソッドを呼び出し、戻り値を表示する。

　この際、存在しない月が入力された場合の例外処理も行う。