

課題1 次はキーボードから整数を一つ読むコードである。

- ① `Integer.parseInt()`メソッドは文字列が表す整数値を `int` 型に変換する。もし、整数値以外が入力されると `NumberFormatException (Unchecked 例外クラス)` という例外をスローする。整数値以外が入力された場合、この例外をキャッチしてメッセージ「整数値ではありません」を表示しなさい。

※`readLine()`メソッドは `IOException (Checked 例外クラス)` という例外をスローする。この例外が発生した場合はこれまで通り

```
public static void main(String[] args) throws IOException
```

として、呼び出し元メソッドである `main()`メソッドに例外処理を任せる。

- ② 整数が入力されるまで繰り返すコードに変更しなさい。

```
import java.io.*;
class Assignment9_1
{
    public static void main(String[] args) throws IOException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        int num=0;
        String str;

        System.out.println("整数を入力してください。");
        str=br.readLine();
        num=Integer.parseInt(str);
        System.out.println("入力された整数は"+num+"です。");
    }
}
```

課題2 次は除算を管理するクラス `Div` の宣言である。整数 `i` と `j` を受け取り `i/j` を戻り値とする。このとき、`j` がゼロであった場合には除算はできない。例外クラス `Exception` を継承して各自このエラーを表すクラス（クラス名は各自で決めること）を宣言しなさい。さらに、`j` がゼロであった場合にはこの例外をスローするようにクラス `Div` の宣言を変更しなさい。

```
class Div
{
    public static double calc(int i, int j)
    {
        double ans;
        ans=(double)i/j;
        return ans;
    }
}
```

課題3 課題1で作成した例外処理をもつ整数値のキーボード入力により2つの整数を入力して、除算を計算するクラス Div で除算を計算するコードをメインメソッドに記述しなさい。課題2で宣言したゼロでの除算を表す例外が発生しない場合は除算の結果を表示し、例外が発生した場合はこれをキャッチして無限大「 ∞ 」を表示するようにすること。

(実行例1)

```
>java Assignment9_3
整数 i ÷ 整数 j を計算します。
整数値 i を入力してください。
4r
整数値ではありません。
もう一度入力してください。
4
整数値 j を入力してください。
2
4 / 2 = 2.0
-- Press any key to exit (Input "c" to continue) --
```

(実行例2)

```
>java Assignment9_3
整数 i ÷ 整数 j を計算します。
整数値 i を入力してください。
4
整数値 j を入力してください。
0
4 / 0 =  $\infty$ 
-- Press any key to exit (Input "c" to continue) --
```