

課題番号7

条件分岐の基本

フィールドの値が0以下ならNG, それ以外の場合, 偶数か奇数かを判定する機能を持つクラス
ConditionChecker.java を作成してみましょう.

このクラスの仕様

- data: 条件判定に利用する値を保持するためのint型フィールド
- コンストラクタ 整数を引数とし、dataに代入する
- showData() dataの値を表示するvoid型メソッド
- checkData() dataの値が0以下の場合『1以上の整数を入力してください』, 偶数の場合『入力された整数は偶数です』, 奇数の場合『入力された整数は奇数です』と表示するvoid型メソッド

クラス図

ConditionChecker
-data: int
+ConditionChecker(inputNum: int) +showData(): void +checkData(): void <u>+main(args: String[]): void</u>

課題7の実行例

正の整数を入力してください: 25

入力された数字は25です

入力された整数は奇数です

正の整数を入力してください: 110

入力された数字は110です

入力された整数は偶数です

正の整数を入力してください: -5

入力された数字は-5です

1以上の整数を入力してください

水色の部分はキーボード入力

プログラムの骨格

```
import java.util.Scanner;
```

```
クラス宣言 {
```

```
    // フィールドの宣言
```

```
    // コンストラクタの定義 { 仕様通りに記述する}
```

```
    // showDataメソッドの定義{dataの値を実行結果にあるように表示}
```

```
    // checkDataメソッドの定義 {
```

```
    // if else-if ラダーを使って, dataが0以下の場合, 偶数の場合, 奇数の場合
```

```
    // それぞれの処理を記述します}
```

```
    public static void main(String[] args) {
```

```
        // キーボード入力のためのスキャナオブジェクト生成
```

```
        int input; // 入力された値を一旦保存しておくための変数
```

```
        // キーボードからの入力をinputに代入
```

```
        // inputの値を引数として、オブジェクトを生成
```

```
        // オブジェクトに各メソッドを順番に実行させる
```

```
    }
```

```
}
```