

## 課題番号 12

- 「基準値を事前に用意しておき、その後二つの整数を与えるとその和が基準値以上であるかをチェックする」という機能のクラスSumCheckerを作成してみましょう.
- 仕様は以下の通りとします.
  - フィールドは基準値target(整数)
  - コンストラクタは整数を引数とし、フィールドに代入する
  - 二つの整数を引数とし、その和と基準値を比較するメソッド compSumToTarget を持つ  
compSumToTargetは、引数二つの和が基準値以上ならtrue, そうでなければfalseを返す

# 課題番号 12

- mainメソッドは以下のような流れとします
  1. キーボードから基準値を入力
  2. 基準値を使ってオブジェクト生成
  3. キーボードから繰り返し回数を入力
  4. 以下5～6を、入力された回数繰り返す
    5. キーボードから整数を二つ入力
    6. 入力された二つの整数の和を基準値と比較し、二つの整数の和が基準値以上ならばメッセージを表示する

ヒント

true か false を返すメソッドはそのまま条件式として使えます

## 課題12の実行例(1), (2)

```
基準値を入力してください: 45
回数を入力してください: 4
整数を二つ入力してください: 20 20
整数を二つ入力してください: 30 50
二つの和は基準値以上です
整数を二つ入力してください: 10 34
整数を二つ入力してください: 10 35
二つの和は基準値以上です
```

```
基準値を入力してください: 2000
回数を入力してください: -3
```

からの入力  
青字の部分はキーボード

## 課題12の実行例(3), (4)

基準値を入力してください: 2000

回数を入力してください: 2

整数を二つ入力してください: 1000 500

整数を二つ入力してください: 800 600

基準値を入力してください: 100

回数を入力してください: 3

整数を二つ入力してください: 50 70

二つの和は基準値以上です

整数を二つ入力してください: 120 -200

整数を二つ入力してください: -500 800

二つの和は基準値以上です

青字の部分は  
からの入力  
キーボード

# クラス図

SumChecker
-target: int
+SumChecker(target: int) +compSumToTarget(value1: int, value2: int): boolean <u>+main(args: String[]): void</u>