

# 課題 1

|              |                                |
|--------------|--------------------------------|
| クラス名         | DoubleNum                      |
| <u>メンバ関数</u> |                                |
| コンストラクタ      | 引数でメンバ変数「value_」を初期化           |
| set          | 引数を2倍してメンバ変数「value_」に代入        |
| show         | メンバ変数「value_」を「値: (数値)」という形で出力 |
| <u>メンバ変数</u> |                                |
| value_       | 整数型                            |

# 課題 2

|              |   |
|--------------|---|
| クラス名         | Num                                       |
| <u>メンバ関数</u> |   |
| コンストラクタ      | メンバ変数「num1_」と「num2_」を0で初期化                |
| set          | 引数1をメンバ変数「num1_」に、<br>引数2をメンバ変数「num2_」に代入 |
| getMin       | メンバ変数「num1_」と「num2_」のうち、小さいほうを返す          |
| getMax       | メンバ変数「num1_」と「num2_」のうち、大きいほうを返す          |
| <u>メンバ変数</u> |   |
| num1_        | 整数型                                       |
| num2_        | 整数型                                       |

# 課題 3

|              |   |
|--------------|---|
| クラス名         | RandomNum   |
| <u>メンバ関数</u> |   |
| コンストラクタ      | 引数1でメンバ変数「min_」、<br>引数2でメンバ変数「count_」を初期化<br>メンバ変数「value_」は-1で初期化 |
| set          | メンバ変数「value_」に最小値min_で、count_個の乱数を代入                              |
| show         | メンバ変数「value_」を「値: (数値)」という形で出力                                    |
| <u>メンバ変数</u> |   |
| value_       | 整数型   |
| min_         | 整数型/生成する乱数の最小値  |
| count_       | 整数型/生成する乱数の個数   |

# 問題 4

| クラス名         | InputNum  |
|--------------|---|
| <u>メンバ関数</u> |   |
| コンストラクタ      | 引数1でメンバ変数「min_」、引数2でメンバ変数「max_」を初期化<br>メンバ変数「value_」は-1で初期化                                       |
| set          | 入力された数値をメンバ変数「value_」に代入<br>ただし、メンバ変数「min_」より小さい数値、<br>またはメンバ変数「max_」より大きい数値が<br>入力された場合は、入力をやり直す |
| get          | メンバ変数「value_」を返す  |
| <u>メンバ変数</u> |   |
| value_       | 整数型/入力された数値   |
| min_         | 整数型/入力される最小値  |
| max_         | 整数型/入力される最大値  |

# 問題 5

|              |  |
|--------------|--|
| クラス名         | ArrayNum   |
| <u>メンバ関数</u> |  |
| コンストラクタ      | メンバ変数「values_」の要素全てを0で初期化<br>メンバ変数「size_」を5で初期化                  |
| set          | メンバ変数「values_」の引数1番目の要素に引数2の値を代入<br>ただし、引数1番目の値が配列の範囲外の場合は、何もしない |
| get          | メンバ変数「values_」の引数番目の要素を返す<br>ただし、引数番目の値が配列の範囲外の場合は、-1を返す         |
| getSize      | メンバ変数「size_」を返す  |
| show         | メンバ変数「values_」の値をすべて出力<br>「N: (N番目の要素)」とという形式で出力                 |
| <u>メンバ変数</u> |  |
| values_      | 整数型の配列変数（サイズ5個）  |
| size_        | 配列変数のサイズ   |