プログラミングA 第10回・宿題　回答

1　宿題 1

参考資料：ScoreManager.java

```
import java.io.BufferedReader; import java.io.FileWriter;
import java.io.IOException; import java.io.InputStreamReader;
import java.io.PrintWriter; import java.util.Collections;
import java.util.List; import java.util.ArrayList;
import java.util.Scanner; import java.util.InputMismatchException;
public class ScoreManager {
  private static List<Integer> list = new ArrayList<Integer>();
  public static void add(int n) {
    list.add(n);
  }
  public static void remove(int n) {
    try { list.remove(new Integer(n)); } catch(IndexOutOfBoundsException ioobe) {}
  }
  public static void sort() { Collections.sort(list); }
  public static void list() {
    for(int n: list) System.out.print(n + ","); System.out.println();
  }
  public static void mean() {
    int sum = 0; for(int n: list) sum += n;
    System.out.println(sum / list.size());
  }
  public static void sum() {
    int sum = 0; for(int n: list) sum += n;
    System.out.println(sum);
  }
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    String command = null;
    while(true) {
      System.out.print("> ");
      try {
        command = scanner.next();
        if(command.equals("add")) add(scanner.nextInt());
        else if(command.equals("remove")) remove(scanner.nextInt());
        else if(command.equals("sort")) sort();
        else if(command.equals("list")) list();
        else if(command.equals("mean")) mean();
        else if(command.equals("sum")) sum();
        else if(command.equals("end")) break;
      } catch(InputMismatchException ime) {
      }
```

```
    }
  }
}
```

2 宿題 2

(1)
- 配列とコレクションはいずれも同型の要素の集合を扱える点が共通している。
- 配列では、要素数は最初の確保時点で固定され、それ以上の要素の追加は出来ない。
- 一方、コレクションには可変長な配列としての List があり、List については最初の確保時から要素を追加することが可能である。他にも、要素の追加の仕方や持ち方に応じて Set や HashMap などの様々なデータ構造が含まれる。

(2)
- List と Set は、同型の要素の集合を、要素数について可変長な形で扱える点が共通している。要素の追加や削除、確認等の基本的なメソッドも共通している。
- ただし、List では要素の重複を許し順序が保たれるのに対して、Set では要素の重複は許されず順序も保証されない (その代わりに、要素の検索は多くの場合において ArrayList よりも高速なことが期待できる)。

3 宿題 3

3.1 宿題 3-1

省略。

3.2 宿題 3-2

100000*100000=1410065408。32 ビット int 型で表せる整数の範囲を超えたため。100000×100000-232×2=1410065408(左記は Java の計算式ではないことに注意)

3.3 宿題 3-3

```
float y = 2;
```

3.4 宿題 3-4

```
7.5+3.14=10.64;
```

3.5 宿題 3-5

```
float number = Float.parseFloat(line);
```

3.6 宿題 3-6

値を 3 種類以上用意する。詳細省略。

3.7 宿題 3-7

値を 3 種類以上用意する。詳細省略。

3.8 宿題 3-8

```
int i = 0;
while (i < 10) {
   System.out.print(i + ":" );
   int j = 0;
   while(j < i) {
      System.out.print("*");
      j++;
   }
   System.out.println("");
   i++;
}
```

3.9 宿題 3-9

```
BufferedReader reader = new BufferedReader(
     new InputStreamReader(System.in));
try {
   for(String line; (line = reader.readLine()) != null; ) {
      System.out.println(line);
   }
} catch (Exception e) {
   System.out.println(e);
}
```

3.10 宿題 3-10

```
String s = line.toUpperCase();
```

3.11 宿題 3-11

非負整数乗を正しく計算するが、負数整数乗を正しく計算しない (必ず 1)。また、0 の 0 乗は 1 と定義していることになる。以上を入力例と共に説明。

3.12 宿題 3-12

```
int sum = 0;
int min = values[0];
int max = values[0];
for (int i = 0; i < values.length; i++) {
   sum += values[i];
   if(max < values[i]) {
      max = values[i];
   } else if(min > values[i]) {
      min = values[i];
   }
}
System.out.println("Average: " + (double) sum / values.length);
System.out.println("Min: " + min);
System.out.println("Max: " + max);
java.util.Arrays.sort(values);
System.out.println("Min: " + values[0]);
System.out.println("Max: " + values[values.length - 1]);
```

3.13 宿題 3-13

```
int sum = 0;
int size = 0;
int min = vs[0][0];
int max = vs[0][0];
for (int i = 0; i < vs.length; i++) {
   for (int j = 0; j < vs[i].length; j++){
      sum += vs[i][j];
      size++;
      if(max < vs[i][j]) {
         max = vs[i][j];
      } else if(min > vs[i][j]) {
         min = vs[i][j];
      }
   }
}
System.out.println("Average: " + (double) sum / size);
System.out.println("Min: " + min);
System.out.println("Max: " + max);
```