

オープン授業

第7回 関数

# 目次

- 配列の復習
- 関数とは
- 関数の呼び出し
- 戻り値
- ローカル変数

# 前回の復習（配列）

- 配列の宣言

例 ) `int x[] = {0, 1, 2, 3};`と宣言すると、

`x[0] = 0`

`x[1] = 1`

`x[2] = 2`

`x[3] = 3`

と格納される。

- 配列の要素は0番目から始まる。

# 前回の復習（配列）

- 配列の宣言

例 ) `int x[] = {0, 1, 2, 3};`と宣言すると、

`x[0] = 0`

`x[1] = 1`

`x[2] = 2`

`x[3] = 3`

と格納される。

- 配列の要素は0番目から始まる。

# 復習問題

- 4つの要素を持つ配列に、10, 11, 12, 13の順に値を格納し、次スライドのような実行結果になるようなプログラムを作成する。

# 実行結果

4問の問題を出します

0番目の配列の値は?12

不正解です。

1番目の配列の値は?11

正解です。

2番目の配列の値は?34

不正解です。

3番目の配列の値は?13

正解です。

以上で問題を終了します。

# 解答例

```
1 import java.util.Scanner;
2
3 public class Example {
4
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         int x[] = {10, 11, 12, 13};
8         System.out.println("4問の問題を出します");
9         for(int i = 0; i < 4; i++) {
10             System.out.printf("%d番目の配列の値は?", i);
11             int n = scanner.nextInt();
12             if(x[i] == n) {
13                 System.out.println("正解です。");
14             }
15             else {
16                 System.out.println("不正解です。");
17             }
18         }
19         System.out.println("以上で問題を終了します。");
20         scanner.close();
21     }
22
23 }
```

# 関数とは

- 一定の処理を一つのグループとしてまとめること。
- 関数の宣言

関数の型 static 関数名(型 引数1, 型 引数2...)

{

    中身

    return 式;

}

# 関数の呼び出し

- 関数名(引数1, 引数2,...);  
(宣言した関数と、関数を呼び出すときの引数の数と型を同じにする。)
- ```
public static void main(String[] args) {  
    関数名(引数1, 引数2,...);  
}
```

上記の文はメインメソッドと呼ばれる。  
その文の中身で関数を呼び出すことでその関数を実行できる。

# 関数を用いて書き換えた復習問題

```
1 import java.util.Scanner;
2
3
4 public class Example2 {
5     public static void question() {
6         Scanner scanner = new Scanner(System.in);
7         int x[] = {10, 11, 12, 13};
8         System.out.println("4問の問題を出します");
9         for(int i = 0; i < 4; i++) {
10            System.out.printf("%d番目の配列の値は?", i);
11            int n = scanner.nextInt();
12            if(x[i] == n) {
13                System.out.println("正解です。");
14            }
15            else {
16                System.out.println("不正解です。");
17            }
18        }
19        System.out.println("以上で問題を終了します。");
20        scanner.close();
21    }
22
23    public static void main(String[] args) {
24        question();
25    }
26
27 }
```

- `public static void question() {`  
  
`}`

上記の関数でquestionという関数の宣言。

- `public static void main(String[] args) {`  
`question();`  
`}`

上記のメインメソッドでquestion関数を呼び出している。

# 戻り値について

- return 式;
- 関数は呼び出し元の関数に値を返すことができる。その値を戻り値という。

例) `int x = 10;`

`return x;`

上記のように入力すると、10の値を返すことができる。(戻り値の型は宣言した変数と同じ型になる。)

# 例題1

- メインメソッドで任意の $x, y$ を入力する。そして、二つの数字の掛け算を行う関数を用いて、 $x, y$ を掛け算したあと、その結果を表示するプログラムを作成する。

# 実行結果

xの値を入力してください:45

yの値を入力してください13

x\*yの値は585

# 解答例

```
1 import java.util.Scanner;
2
3
4 public class Example01 {
5
6     public static int multiply(int x, int y) {
7         int i;
8         i = x * y;
9         return i;
10    }
11
12    public static void main(String[] args) {
13        int x, y, result;
14        Scanner scanner = new Scanner(System.in);
15        System.out.print("xの値を入力してください:");
16        x = scanner.nextInt();
17        System.out.print("yの値を入力してください:");
18        y = scanner.nextInt();
19        result = multiply(x, y);
20        System.out.printf("x*yの値は%dです", result);
21    }
22 }
```

# ローカル変数について

- 関数内で宣言された変数のこと。
- ローカル変数は宣言された関数内でしか使うことができない。

## 例題2

- 正の整数を入力して、入力された数字を表示、そして奇数か偶数か判断して、それを表示する。その後入力した正の整数に新たな整数を足して、また数字を表示し奇数偶数判断をするプログラムを作成する。

(数字を表示する関数、奇数偶数判断をする関数、新たな足し算をする関数の三つの関数を用いて作成する。)

# 実行結果

---

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

入力された整数は奇数です

新たな整数を入力してください:7

5に7を足します

入力された数字は12です

入力された整数は偶数です

# 解答例

```
1 import java.util.Scanner;
2
3 public class Example02 {
4     public static void showData(int data) {
5         System.out.printf("入力された数字は%dです\n", data);
6     }
7
8     public static void checkData(int data) {
9         if (data <= 0 ) {
10            System.out.printf("1以上の整数を入力してください\n");
11        } else if (data % 2 == 0) {
12            System.out.printf("入力された整数は偶数です\n");
13        } else {
14            System.out.printf("入力された整数は奇数です\n");
15        }
16    }
17
18    public static int multiplyData(int data) {
19        Scanner keyBoardScanner = new Scanner(System.in);
20        int data2, multiply;
21        System.out.print("新たな整数を入力してください:");
22        data2 = keyBoardScanner.nextInt();
23        System.out.printf("%dに%dを足します\n", data, data2);
24        multiply = data + data2;
25        return multiply;
26    }
27
28    public static void main(String[] args) {
29        Scanner keyBoardScanner = new Scanner(System.in);
30        int input;
```

```
31     System.out.print("正の整数を入力してください:");
32     input = keyBoardScanner.nextInt();
33     showData(input);
34     checkData(input);
35     input = multiplyData(input);
36     showData(input);
37     checkData(input);
38     keyBoardScanner.close();
39 }
40 }
```

---

## 例題3

- 配列のサイズを入力し、その配列の中に値を格納する。そして、その配列の最大値と最小値を表示するプログラムを作成する。

(最大値を求める関数と最小値を求める関数を作成する。)

※配列の大きさは、変数.lengthで求まる。

- 例) `int x[] = new int[4];`

`int i = x.length;`

この場合のiは、4となる。

# 実行結果

配列サイズを入力してください:5

値を入力してください:4 10 5 67 9

最小値は4です。

最大値は67です。

# 解答例

```
1 import java.util.Scanner;
2
3 public class MaxAndMinOfNumbers {
4
5     public static int max(int numbers[]) {
6         int currentMax = numbers[0];
7         for(int i = 0; i < numbers.length; i++) {
8             if(numbers[i] > currentMax) {
9                 currentMax = numbers[i];
10            }
11        }
12        return currentMax;
13    }
14
15    public static int min(int numbers[]) {
16        int currentMin = numbers[0];
17        for(int i = 0; i < numbers.length; i++) {
18            if(numbers[i] < currentMin) {
19                currentMin = numbers[i];
20            }
21        }
22        return currentMin;
23    }
24
25
26    public static void main(String[] args) {
27        Scanner keyBoardScanner = new Scanner(System.in);
28        System.out.print("配列サイズを入力してください:");
29        int input = keyBoardScanner.nextInt();
30        int[] numbers = new int[input];
```

```
31     System.out.print("値を入力してください:");
32     for(int i = 0; i < input; i++) {
33         numbers[i] = keyBoardScanner.nextInt();
34     }
35     System.out.printf("最小値は%dです.\n", min(numbers));
36     System.out.printf("最大値は%dです.\n", max(numbers));
37     keyBoardScanner.close();
38
39 }
40
41 }
```

# まとめ

- 関数は処理をまとめて書くことができる。
- 宣言された関数は他の関数上で呼び出すことができる。
- 関数間での数値の移動は引数と戻り値を用いる。
-