

## プログラミングⅡ 2022 第1回 演習課題 「プログラミングIの復習」

下記の各問に対して Java のプログラムを作成せよ。次回までに、ソースコードと出力結果を専用の提出用紙のフォーマットに入力して提出せよ。

1. 以下の `Kaishin_A` と `Kaishin_B` のプログラムが同じ処理を行うように、`Kaishin_B` の空欄にメソッド `kaishin` を定義せよ。このように何らかの処理に名前をつけて、使うときに「呼び出す」ことができるのがメソッドである（他のプログラミング言語では、サブルーチン、関数、手続きなどとも呼ぶ）。

```
/* Kaishin_A.java */
public class Kaishin_A {
    public static void main(String[] args) {
        System.out.println("かいしんのいちげき!");
    }
}
```

```
/* Kaishin_B.java */
public class Kaishin_B {
    public static void main(String[] args) {
        kaishin();
    }
}
```

```
}
```

2. 下記の(1)および(2)の指示に従って Java プログラム `Kaishin_C` を作成せよ。

(1) `Kaishin_B` の `main` メソッドの中身だけを変更して、「かいしんのいちげき!」と3回表示されるようにするにはどのように変更すればよいか示せ。ただし、`println` 等は使わないこと。メソッドを使わない場合と比べた場合の利点は何だろうか。

(2) さらに、「かいしんのいちげき!」と表示する回数を「引数」(ひきすう)を使って受け渡すようにする新しい `kaishin` メソッドを定義せよ。引数を利用する利点は何だろうか。このように、名前が同じでも引数が異なるメソッドは共存することができる(オーバーロード)。

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

// (1)の解答

```
        kaishin(5); // 5回表示される
    }
```

```
public static void kaishin() {
    System.out.println("かいしんのいちげき!");
}
```

// (2)の解答

```
}
```

3. 今までキーボードからの入力の処理は、値を読み込むたびに「new java.util.Scanner(System.in).next 何々」という同じ長い決まり文句を書くように習った。しかし、Java に限らず一般的にプログラミング言語には、このように何度も実行するような処理は、なるべく同じことを書かなくてもよくする機能が備わっている。以下に例を示すが、これらの文法的意味については、次回以降に詳しく説明する。

```
// プログラムの冒頭に次の import 文を書くことによって、「java.util.Scanner」と書く代わりに、  
// 単にクラス名の部分の「Scanner」とだけ書けばよいようになる
```

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

```
public class Main {
```

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

```
        // 最初の入力の前に new Scanner(System.in)を Scanner クラスの変数 sc で置き換える  
        // その際、冒頭の import のおかげで java.util.Scanner の代わりに Scanner と書ける  
        Scanner sc = new Scanner(System.in);
```

```
        // 以上で、new java.util.Scanner(System.in)を変数 sc で置き換えたことになるので、  
        // 以降は、毎回それを全部書く代わりに、一度作った sc を何度も使いまわすことができる  
        double x = sc.nextDouble();  
        double y = sc.nextDouble();  
        double z = sc.nextDouble();
```

```
        System.out.println(x + ", " + y + ", " + z);
```

```
    }
```

```
}
```

以上を踏まえ、次の空欄を埋めてキーボードから 10 個の整数をいったん配列に読み込み、合計と平均を計算して出力するプログラムを作成せよ。また、今後はこの方法で Scanner を使用すること。

```
import
```

```
public class Main {
```

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

```
        Scanner sc =
```

```
        int [] data = new int[10];  
        for (int i = 0; i < data.length; i++) {  
            System.out.print("data[" + i + "] = ");
```

```
        }
```

```
        // 配列の中の全要素に対する拡張 for 文
```

```
        int sum = 0;  
        for (int x : data) {
```

```
            double average =
```

```
        }
```

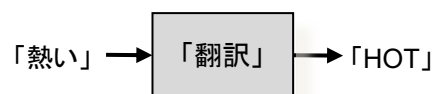
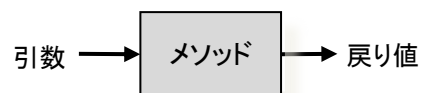
```
    }
```

```
}
```

4. 以下は、キーボードから 2 つの正方形の辺の長さを読み込み、大きい方の面積を表示するプログラムである。数学の関数のような役割を果たすメソッド `square` と `max` を補って完成させよ。このようにメソッドは `return` 文によって値を返す（戻す）ことができる。メソッド定義では戻り値の型を名前の前に記述する。

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

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("正方形 A の辺の長さ: ");
        double a = square(sc.nextDouble());
        System.out.print("正方形 B の辺の長さ: ");
        double b = square(sc.nextDouble());
        double larger = max(a, b);
        System.out.println("大きい方の正方形の面積: " + larger);
    }
}
```



メソッド(関数)の概念

```
public static double square(double x) {
```

```
    return
```

```
}
```

```
public static double max(double x, double y) {
```

```
    if (x >= y) {
        return
    } else {
        return
    }
}
```

```
}
```

5. 以下のプログラムは、キーボードから年齢を読み込み、20 歳以上なら「Adult」、20 歳未満なら「Child」と表示するものである。戻り値が `boolean` 型であるメソッド `isAdult` を定義してプログラムを完成させよ。

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

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Age? ");
        int age = sc.nextInt();

        if (isAdult(age)) {
            System.out.println("Adult");
        } else {
            System.out.println("Child");
        }
    }
}
```

```
}
```