5. クラスとメソッド

次のプログラムコードに、各設問の条件にあうメソッドを追加しなさい。その後、そのメソッドが正しく動作することを検証するためのプログラムコードを main メソッドの中に追加しなさい。

```
public class Practice {
        // ここに各設問のメソッドを追加する

        public static void main(String[] args) {
             // ここに、追加したメソッドの動作検証を行うプログラムコードを追加する
        }
}
```

例題
クラス名： Practice
メソッド名： getTriangleArea
引数列：double height, double base
戻り値の型： double
処理の内容： 底辺の長さが base、高さが height で表される三角形の面積を返す

解答例

```
public class Practice {
        // (例題)
        static double getTriangleArea(double height, double base) {
                return height * base / 2.0;
        }

        public static void main(String[] args) {
                // (例題) のメソッドの動作検証
                double triangleArea = getTriangleArea(8.2, 4.0);
                System.out.println("三角形の面積は" + triangleArea);
        }
}
```

■ 引数なし、戻り値なし

問題 1

クラス名： Practice1

メソッド名： printHello

引数列：なし

戻り値：なし

処理の内容： "Hello" という文字を出力する

ヒント：メソッドの定義は次のようになる

```
static void printHello() {
  // 処理の内容
}
```

問題 2

クラス名： Practice2

メソッド名： printPI

引数列：なし

戻り値：なし

処理の内容： 円周率（3.14159....）の値を出力する

ヒント：

・円周率の値は Math クラスのクラス変数 PI に格納されている。

・System.out.println(Math.PI); で、この値を出力できる。

問題 3

クラス名： Practice3

メソッド名： printRandomMessage

引数列：なし

戻り値：なし

処理の内容： "こんばんは", "こんにちは", "おはよう" の 3 つのあいさつからランダムに 1 つを表示する。

ヒント：

・次のように記述すると、変数 n には、0, 1, 2 のいずれかの値がランダムに代入される。

```
int n = (int)(3 * Math.random());
```

■ 引数が 1 つ、戻り値なし

問題 4

クラス名： Practice4

メソッド名： printMessage

引数列： String message

戻り値：なし

処理の内容： 引数で渡されたメッセージを出力する

ヒント：

・メソッドの宣言は次のようになる

```
static void printMessage(String message) {
   // 処理の内容
}
```

・printMessage("Hello"); とすると、"Hello" という文字が出力される

問題 5

クラス名： Practice5

メソッド名： printCircleArea

引数列： double radius

戻り値：なし

処理の内容： 引数で渡された値を半径とする円の面積を出力する

ヒント：

・円の面積は 半径×半径×円周率

・printCircleArea(2.0); とすると、半径が 2.0 の円の面積が出力される

問題 6

クラス名： Practice6

メソッド名： printRandomMessage

引数列： String name

戻り値：なし

処理の内容：

・"こんばんは xx さん", "こんにちは xx さん", "おはよう xx さん" の 3 つのあいさつからランダムに 1 つを表示する。

・あいさつ文の xx のところには、引数で渡された name の文字が入る

ヒント：

・同じ名前で引数の異なるメソッドが存在することをオーバーロードという。

■ 引数が複数、戻り値なし

問題 7

クラス名： Practice7

メソッド名： printMessage

引数列： String message, int count

戻り値：なし

処理の内容： 文字列 message を、count の回数だけ繰り返し出力する

ヒント：

・printMessage("Hello", 5); とすると、"Hello" という文字が 5 回出力される

問題 8

クラス名： Practice8

メソッド名： printRectangleArea

引数列： double height, double width

戻り値：なし

処理の内容： 高さが height, 横幅が width の長方形の面積を出力する

問題 9

クラス名： Practice9

メソッド名： printMaxValue

引数列： double a, double b, double c

戻り値：なし

処理の内容： 引数で渡された a, b, c の値のうち、もっとも大きな値を出力する

■ 引数なし、戻り値あり

問題 10

クラス名： Practice10

メソッド名： getMessage

引数列：なし

戻り値の型： String

処理の内容： "よろしくおねがいします" という文字列を返す

問題 11

クラス名： Practice11

メソッド名： getWeatherForecast

引数列：なし

戻り値の型： String

処理の内容：

天気予報メッセージをランダムに生成して、そのメッセージを返す。

天気予報メッセージは、次の中からランダムに組み合わせて作り出すものとする。

{今日・明日・明後日}の天気は{晴れ・曇り・雨・雪}でしょう。

例：明日の天気は雨でしょう。

問題 12

クラス名： Practice12

メソッド名： getSquareRootOf2

引数列：なし

戻り値の型： double

処理の内容： 2 の平方根（√2）を返す

ヒント：Math クラスの sqrt メソッドで平方根を求めることができる。

・double d = Math.sqrt(3.0); で、3 の平方根が変数 d に代入される

■ 引数列：が 1 つ、戻り値あり

問題 13

クラス名： Practice13

メソッド名： getRandomMessage

引数列：String name

戻り値の型： String

処理の内容：

・"こんばんは xx さん", "こんにちは xx さん", "おはよう xx さん" の 3 つのあいさつからランダムに選んだ 1 つを戻り値とする。

・あいさつ文の xx のところには、引数で渡された name の文字が入る

問題 14

クラス名： Practice14

メソッド名： getAbsoluteValue

引数列： double value

戻り値の型： double

処理の内容： 引数で渡された value の値の絶対値を返す。

ヒント：

・5.2 の絶対値は 5.2

・-3.6 の絶対値は 3.6

問題 15

クラス名： Practice15

メソッド名： isEvenNumber

引数列： int value

戻り値の型： boolean

処理の内容： 引数で渡された値が偶数の場合は true、そうでない場合は false を返す。

ヒント：偶数は 2 で割った余りがゼロ

■ 引数列が複数、戻り値あり

問題 16

クラス名： Practice16

メソッド名：getMinValue

引数列：double a, double b

戻り値の型：double

処理の内容： 引数で受け取る 2 の値のうち、小さい方の値を返す

問題 17

クラス名： Practice17

メソッド名：isSameAbsoluteValue

引数列：int i, int j

戻り値の型：boolean

処理の内容： 引数で受け取る 2 の値の絶対値が等しければ true, そうでなければ false を返す

問題 18

クラス名： Practice18

メソッド名： getMessage

引数列：String name, boolean isKid

戻り値の型：String

処理の内容： isKid の値が true なら、"こんにちは。xx ちゃん。" isKid の値が false なら "こんにちは。xx さん。" という文字列を返す。ただし xx には name の値が入る。

■ 引数が配列、戻り値あり

問題 19

クラス名： Practice19

メソッド名： getMinValue

引数列：int[] array

戻り値の型：int

処理の内容： 引数で受け取る配列の要素のうち、最も小さい値を返す

問題 20

クラス名： Practice20

メソッド名： getAverage

引数列：double[] array

戻り値の型：double

処理の内容： 引数で受け取る配列の要素の平均値を返す

問題 21

クラス名： Practice21

メソッド名： getLongestString

引数列：String[] array

戻り値の型：String

処理の内容：

・引数で受け取る配列の要素のうち、最も文字数の大きい文字列を返す

・文字数が同じものが複数存在する場合は、配列の後ろの方の要素を優先する

ヒント：

次のように記述すると、変数 l に文字列 str の長さが代入される。

```
int l = str.length();
```

■ 引数列がクラスの参照（Point クラスを用いる例）

以降の設問に取り組む前に、次のような Point クラスをプログラムコードに追加しなさい。

```
class Point {
    double x;
    double y;

    Point(double x, double y) {
        this.x = x;
        this.y = y;
    }
}
```

問題 22

クラス名： Practice22

メソッド名： getDistanceFromOrigin

引数列：Point p

戻り値の型：double

処理の内容： 引数で受け取る Point オブジェクトの、原点からの距離を返す

ヒント：点(x, y)の原点からの距離は Math.sqrt(x*x+y*y) で求まる

問題 23

クラス名： Practice23

メソッド名： getDistanceBetweenTwoPoints

引数列：Point p0, Point p1

戻り値の型：double

処理の内容： 引数で受け取る 2 つの Point オブジェクト間の距離を返す

ヒント：点(x0, y0)と点(x1, y1)の距離は Math.sqrt((x0 - x1)*(x0 - x1)+(y0 - y1)*(y0 - y1)) で求まる

問題 24

クラス名： Practice24

メソッド名： getBarycenter

引数列：Point[] points

戻り値の型：Point

処理の内容： 引数で受け取る配列に含まれる Point オブジェクトの重心座標を表す Point オブジェクト

ヒント：

・複数の点の重心の x,y 座標は、それぞれすべての点の x 座標の平均と y 座標の平均で表される
・メソッドの中で、新しい Point クラスのインスタンスを生成する

■ 引数列がクラスの参照（Person クラスを用いる例）

以降の設問に取り組む前に、次のような Person クラスをプログラムコードに追加しなさい。

```
class Person {
    private String name;
    private int age;

    Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    String getName() {
        return name;
    }

    int getAge() {
        return age;
    }
}
```

問題 25

クラス名： Practice25

メソッド名： printMessage

引数列：Person person

戻り値：なし

処理の内容：

"こんにちは xx さん" というメッセージを出力する。xx には引数列：で渡された Person オブジェクトの名前(name)を入れる。

ヒント：Person クラスのインスタンス変数 name は private 修飾子が付いているので、直接参照できない。getName メソッドを用いて取得する。

問題 26

クラス名： Practice26

メソッド名： isAdult

引数列：Person person

戻り値の型：boolean

処理の内容：

引数で渡された Person オブジェクトの年齢(age)の値が 20 以上なら true, そうでないなら false を返す。

問題 27

クラス名： Practice27

メソッド名： getYoungestPerson

引数列：Person[] persons

戻り値の型：Person

処理の内容：

配列に含まれる Person オブジェクトの中で、最も年齢の小さなオブジェクトの参照を返す。同じ年齢の Person オブジェクトがある場合には、配列の後ろの方を優先する。

■ Person クラスへのメソッドの追加

Person クラスに、次のメソッドを追加しなさい。

問題 28

クラス名： Practice28

メソッド名： setName

引数列：String name

戻り値：なし

処理の内容： インスタンス変数 name の値を引数の文字列に設定する

問題 29

クラス名： Practice29

メソッド名： setAge

引数列：int age

戻り値：なし

処理の内容：

インスタンス変数 age の値を引数の値に設定する。ただし引数の値がマイナスの場合は何もしない

問題 30

クラス名： Practice30

メソッド名：isSameAge

引数列：Person person

戻り値の型：boolean

処理の内容：

引数で渡された Person オブジェクトの年齢と、自分自身の年齢が同じであれば true、そうでなければ false を返す。