

錬成問題

- 単一のソースプログラム内では、のクラスを宣言する（できる）。
 単一のソースプログラム内では、の **public** なクラスを宣言する（できる）。
 ▶ 共通の選択肢：(a)必ず1個 (b)1個以下 (c)任意の個数

- クラス変数・クラスメソッドは、個々のインスタンスに所属せず、そのクラス型の全インスタンスで共有される変数・メソッドである。クラス変数はフィールドと呼ばれ、クラスメソッドはメソッドと呼ばれる。

- クラスメソッドの中では、以下のように変数やメソッドをアクセスできる。
 - そのクラスのクラス変数に 。
 - そのクラスのインスタンス変数に 。
 - そのクラスのクラスメソッドを 。
 - そのクラスのインスタンスメソッドを 。
 ▶ ・の選択肢：(a)アクセスできる (b)アクセスできない
 ▶ ・の選択肢：(a)呼び出させる (b)呼び出せない

- インスタンスメソッドの中では、以下のように変数やメソッドをアクセスできる。
 - そのクラスのクラス変数に 。
 - そのクラスのインスタンス変数に 。
 - そのクラスのクラスメソッドを 。
 - そのクラスのインスタンスメソッドを 。
 ▶ ・の選択肢：(a)アクセスできる (b)アクセスできない
 ▶ ・の選択肢：(a)呼び出させる (b)呼び出せない

- クラスメソッドとインスタンスメソッドをまたがった多重定義は。
 ▶ の選択肢：(a)できる (b)できない

- **Math** クラスのように、内部に状態（ステート）をもたず、クラス変数とクラスメソッドのみを提供するクラスのことを“クラス”と呼ぶ。
 円周率を表すクラス変数は、**Math** クラスの外部からとしてアクセスできる。
 また、**double** 型変数 x の絶対値を求めるメソッド呼出し式はとなり、平方根を求めるメソッド呼出し式はとなる。

- **Integer** クラスでは、**int** 型で表現できる最小値と最大値を表す **public** でなクラス変数が定義されている。最小値を表すクラス変数は、最大値を表すクラス変数はである。

■ 以下に示す `Point3D` は、個々のインスタンスに識別番号を付与する 3次元座標クラスであり、`Point3DTester` は、それを利用するプログラム例である。

最初に生成するインスタンスに付与する識別番号は 1, 101, …, 901 のいずれかの値である。2 番目以降に生成するインスタンスには、一つずつ大きな番号を付与する。

クラス `Point3D` に関して：

- 誤り (1箇所) を指摘せよ。 … (21)
- クラス変数をすべて列挙せよ。 … (22)
- インスタンス変数をすべて列挙せよ。 … (23)

```
//--- 識別番号付き 3次元座標クラス ---//
import java.(24).Random;
public class Point3D {
    private static int counter = 0; // 何番までの識別番号を与えたか
    private int id;                // 識別番号
    private int x = 0, y = 0, z = 0; // 座標
    1 → (25) { Random r = new Random(); counter = (26).nextInt(100)*100; }
    2 → { id = (27); }
    public Point3D()                { }
    public Point3D(int x)           { this.x = x; }
    public Point3D(int x, int y)    { this.x = x; this.y = y; }
    public Point3D(int x, int y, int z) { this.x = x; this.y = y; this.z = z; }
    public (28) int getCounter() { return this.counter; }
    public int getId() { return this.id; }
    public String toString() {
        return "(" + x + "," + y + "," + z + ")";
    }
}
```

```
//--- 識別番号付き 3次元座標クラスのテスト ---//
public class Point3DTester {
    public static void main((29) args) {
        Point3D p1 = new Point3D();
        Point3D p2 = new Point3D(1);
        Point3D p3 = new Point3D(2, 3);
        Point3D p4 = new Point3D(4, 5, 6);
        System.out.println("最後に与えた識別番号：" + (30));
        System.out.println("p1 = " + p1 + " … 識別番号：" + (31));
        System.out.println("p2 = " + p2 + " … 識別番号：" + (32));
        System.out.println("p3 = " + p3 + " … 識別番号：" + (33));
        System.out.println("p4 = " + p4 + " … 識別番号：" + (34));
    }
}
```

最後に与えた識別番号：204
 p1 = (0,0,0) … 識別番号：201
 p2 = (1,0,0) … 識別番号：202
 p3 = (2,3,0) … 識別番号：203
 p4 = (4,5,6) … 識別番号：204

■ クラス `Point3D` 中の ❶ は (35) と呼ばれ、(36) 実行される。❷ は (37) と呼ばれ、(38) 実行される。

- ▶ (36) ・ (38) の選択肢：(a)一度だけ (b)コンストラクタの先頭で
(c)コンストラクタの末尾で