
20 章

◆ 20-1

1. C. 0.0

`System.out.println(x+y);` で出力している `x` と `y` は、`main` メソッドで宣言した `x` と `y` である。`Calc` オブジェクトの `x` と `y` を出力するには、`calc.x` と `calc.y` を使う必要がある。これらは `private` アクセスだが、`main` メソッドは `Calc` クラスのメンバなので、アクセスできる。

◆ 20-2

1. C. 20

変数 `tanaka` は `Person` 型の変数なので、`tanaka.number` は、`Person` クラスの `static` 変数をアクセスする。オーバーライドメソッドとは違って、オブジェクトが入っている変数の型によって、アクセスできる変数が決まることに注意する。

◆ 20-3

1. E

A について、`interface` 宣言に `implements` は書けない。B について、クラス継承の `extends` では、1 つのクラスしか継承できない。C について、`static` は書けない。D について、`interface` 宣言に `implements` は書けない。E について、クラス継承とインタフェースの実装を同時に宣言する正しい書き方。F について、クラス継承の `extends` では、1 つのクラスしか継承できない。

2. A. Hello

`Display` クラスが実装している `Printable` インタフェースと `Output` インタフェースに同じメソッド `print()` が定義されている。しかし、どちらも、実装を持たない抽象メソッドなので、メソッドの衝突は起こらない。`Display` クラスは `print()` メソッドを 1 つ実装するだけでよい。

◆ 20-4

1. B E

A は、要素数を書く場所が間違い。new int[3][]が正しい。B は、正しい書き方。C は、new が抜けている。D は、要素数を指定することはできない。E は、宣言時の初期化の正しい書き方。

2. B E

A について、num[1][1]は 50。B は正しい。C、D、E はどれも同じ。num[0][3]は、存在しない要素なので、実行時例外になる。コンパイラは要素番号の正当性をチェックしないので、要素番号が間違っていると、実行時例外を起こす。

◆ 20-5

1. D

P.508 の解説を参照。