
16 章

1-1. B. 0/null

`Product product = new Product();` によって、初期化の規定値で初期化された `Product` オブジェクトが生成される。既定値では、数値項目は 0、オブジェクト (`String` 型もオブジェクト) は `null` に初期化される。

1-2. C. コンパイルエラー

`disp()` は、`Item` オブジェクトの中にあるメソッドなので、まず `Item` オブジェクトを作成し、それを通して `disp()` メソッドを使わねばならない。例えば、次のようにする。

```
public class Item {
    private String name;
    public void disp(){
        System.out.println("hello");
    }
    public static void main(String[] args) {
        Item item = new Item();
        item.disp();
    }
}
```

2. D. Exec の 2 行目と 5 行目でコンパイルエラー

`Product` クラスには `public` がついていないので、デフォルトアクセスである。そのため、他のパッケージ内のクラスからは不可視となってアクセスできない。`import` 文を正しく書いていても、アクセスできないのでコンパイルエラーになる。2 行目がコンパイルエラーになると、正しくインポートできないので、`Product` クラスを使う 5 行目もコンパイルエラーになる。

3. A. import 文は package 文とクラス宣言の間に書かねばならない

B. について、完全修飾名を使えばインポート文は不要。C. について、`package` 文とクラス宣言の間に書かねばならない。D. について、書き誤りはコンパイルエラーになる。E. について、別々に書くことはできない。クラスの先頭にまとめる必要がある。

4. C. 2 行目でコンパイルエラー

ソースファイル名が `Exec.java` なので、`public` を付けることができるのは、`Exec` クラスだけである。