

隨堂練習卷

建議作答 50 分鐘 · 總分 118

§ 1 一、選擇題 (每題 6 分, 共 30 分)

1. **(B)** — static 屬類別、整個類別共享一份；實例變數屬物件、每個物件各一份。2. **(C)** — 同名、參數列不同叫多載 (overloading)。3. **(A)** — 欄位是另一個物件是組合 (has-a)；繼承才是 is-a。4. **(C)** — static 方法用「類名.方法()」呼叫，不能用 this、不能直接存取實例變數。5. **(C)** — ordinal 從 0 起：MON=0、TUE=1、WED=2。

§ 2 二、概念追蹤 (共 16 分)

6. | 動作 | 此時 count | |--| |-- | 第 1 個 new | 1 | | 第 2 個 new | 2 | | 第 3 個 new | 3 | (count 是 static，三個物件共享同一份，每 new 一次 +1) 最後輸出：**3** 7. 輸出：

```
a.id=1 b.id=2
count=2
```

「count」是 static、共享一份；「id = ++count」讓 a 拿到 1、b 拿到 2，且 count 累加到 2。「getCount()」讀的是這份共享的 count，所以是 2。「a」與「b」的 **id 不同** (各自的實例變數)，但讀同一份 count 會得到相同的目前總數。

§ 3 三、改錯 (每題 6 分, 共 18 分)

8. static 方法不能存取實例變數「name」(編譯錯「non-static variable name cannot be referenced from a static context」)。改成實例方法：`public String get() { return name; }` (拿掉「static」)。9. 「this(...)」必須是建構子的**第一個敘述**，且不能與其他「this.» 賦值並存於前。改：`public Time(int h){ this(h, 0); }` (單純委派三參數版即可，不要先「this.hour=h」再「this(...)」)。10. 「final」變數只能賦值一次，「PI = 3.14159;」二次賦值會編譯錯。改：直接在宣告就給最終值「`public static final double PI = 3.14159;`」，刪掉後面的重新賦值。

§ 4 四、程式設計 (共 54 分)

11. (18 分)

```
public class Student {
    private String name;
    private static int total = 0;
    public Student(String name) { this.name = name; total++; }
    public String getName() { return name; }
    public static int getTotal() { return total; }
    public static void main(String[] args) {
        Student s1 = new Student("小明");
        Student s2 = new Student("小華");
        System.out.println(s1.getName());
        System.out.println(s2.getName());
        System.out.println("總人數 = " + Student.getTotal()); // 2
    }
}
```

(評分：static total 宣告與 total++ 共 8 分、getName/getTotal 6 分、main 建物件與輸出 4 分。)

12. (18 分)

```
public class Car {
    private String model;
    private Engine engine;
    public Car(String model, Engine engine) { this.model = model; this.engine = engine; }
    public Engine getEngine() { return engine; }
    public String getModel() { return model; }
    public static void main(String[] args) {
        Car car = new Car("Sedan", new Engine(120));
        System.out.println(car.getModel() + " 馬力 = " + car.getEngine().getHorsepower()); // Sedan 馬力 = 120
    }
}
class Engine {
    private int horsepower;
    public Engine(int hp) { this.horsepower = hp; }
    public int getHorsepower() { return horsepower; }
}
```

(評分：Car 含 Engine 欄位的組合 8 分、層層點號 `getEngine().getHorsepower()` 6 分、建物件與輸出 4 分。)

13. (18 分)

```
import java.math.BigDecimal;
import java.math.RoundingMode;

public class Receipt {
    public static void main(String[] args) {
        BigDecimal price = new BigDecimal("19.99"); // 用字串建構才精確
        BigDecimal qty = new BigDecimal("3");
        BigDecimal subtotal = price.multiply(qty); // 用方法，不能寫 price * qty
        System.out.println("小計 = " + subtotal); // 59.97

        BigDecimal total = subtotal.multiply(new BigDecimal("1.05"))
            .setScale(2, RoundingMode.HALF_UP);
        System.out.println("含稅 = " + total); // 62.97
    }
}
```

(評分：字串建構 + `multiply` 算小計 8 分、`setScale(2, RoundingMode.HALF_UP)` 四捨五入 6 分、輸出 4 分。關鍵：不可用 `double`、不可對 `BigDecimal` 用 `*+`。)