

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)** — 泛型的兩大好處：編譯期型別安全（放錯型別當場編譯錯）、取出免手動轉型。
2. **(C)** — 泛型方法的 `` 宣告在修飾子（`static`）之後、回傳型別之前。
3. **(C)** — 型別實參必須是參考型別；基本型別 `int` 不行（要用 `Integer`）。`int[]` 是陣列、屬參考型別可用。
4. **(B)** — 要在方法內呼叫 `compareTo`，須加上界 `>`（泛型一律用 `extends`）。
5. **(B)** — `` 是上界萬用字元，適合讀、不能 `add`（PECS：Producer-Extends）。(A) `List` 不能 add（除 null）；(C) 泛型不協變；(D) `super` 適合寫。

§ 2 二、概念追蹤 (共 16 分)

6. 輸出：

```
Integer: 42
String: hi
Double: 3.14
```

泛型方法 `show(T x)` 套用到任何型別；`getClass().getSimpleName()` 取執行期實際型別名（42 自動裝箱成 Integer、3.14 成 Double）。7. **(b)** `list.add(0);` 會編譯錯。參數型別是 `List`（上界萬用字元），編譯器只知道元素是「某個 Number 的子型別」但不知道確切是哪一個，為了型別安全**不允許 add**（除了 `null`）。(a) 把元素當 `Number` 讀是安全的、(c) 回傳 double 沒問題。口訣 PECS：extends 只能讀。

§ 3 三、改錯 (每題 6 分，共 18 分)

8. 泛型方法的 `` 要放在**回傳型別之前**，不是接在方法名後面。改：`public static void printArray(T[] a) { ... }`。9. 基本型別不能當型別實參。改用包裝類別：`Box b = new Box<>(42);`（自動裝箱會把 `42` 轉成 `Integer`）。10. 沒有上界時 `T` 只當 `Object`，呼叫不到 `compareTo`（編譯錯）。改加上界：`public static > T max(T a, T b) { return a.compareTo(b) >= 0 ? a : b; }`。

§ 4 四、程式設計 (共 54 分)

11. (18 分)

```
public class PrintArray {
    public static <T> void printArray(T[] a) {
        for (T e : a) System.out.printf("%s ", e);
        System.out.println();
    }
    public static void main(String[] args) {
        Integer[] ints = {1, 2, 3};
        String[] words = {"x", "y", "z"};
        printArray(ints);    // 1 2 3
        printArray(words);  // x y z
    }
}
```

(評分：`` 宣告位置正確 6 分、增強 for 印出 8 分、main 兩種型別各呼叫 4 分。)

12. (18 分)

```
public class Pair<K, V> {
    private K key;
    private V value;
    public Pair(K key, V value) { this.key = key; this.value = value; }
    public K getKey() { return key; }
    public V getValue() { return value; }
    public static void main(String[] args) {
        Pair<String, Integer> p = new Pair<>("score", 95);
        System.out.println(p.getKey() + " = " + p.getValue());    // score = 95
    }
}
```

(評分：`` 兩型別參數與欄位 8 分、兩個 getter 回對應型別 6 分、main 建物件與輸出 4 分。)

13. (18 分)

```
public class MaxThree {
    public static <T extends Comparable<T>> T max(T a, T b, T c) {
        T m = a;
        if (b.compareTo(m) > 0) m = b;
        if (c.compareTo(m) > 0) m = c;
        return m;
    }
    public static void main(String[] args) {
        System.out.println(max(3, 9, 5));    // 9
        System.out.println(max("apple", "pear", "fig"));    // pear
    }
}
```

(評分：上界 ``>` 8 分、用 ``compareTo` 比三者取最大 6 分、main 兩種型別呼叫與輸出 4 分。關鍵：必須有上界才能呼叫 compareTo、物件比大小不可用 ``>`。)