

### Câu 1. (2.0 điểm)

- a) Phương thức trừu tượng có thể khai báo tĩnh không? Tại sao?
- b) Cho biết sự khác nhau phạm vi truy cập protected và default trong Java?
- c) Cho biết sự khác nhau thành viên tĩnh (static) và thành viên thường trong lớp đối tượng và trong quan hệ kế thừa?

### Câu 2. (2.0 điểm) Cho biết các kết quả xuất ra của chương trình sau và giải thích?

<pre> 1  class A { 2      protected int value; 3      public A(int x) { 4          this.value = x; 5      } 6      public A() { 7          this(1); 8      } 9      public void inc1() { 10         this.value += 1; 11     } 12     public void inc2() { 13         this.value += 2; 14     } 15     public final void show() { 16         System.out.print(this.value); 17     } 18 } 19 class B extends A { 20     public void inc2() { 21         this.inc1(); 22         this.value += 2; 23     } 24 }</pre>	<pre> 25    class C extends B { 26        public C(int y) { 27            this.value += y; 28        } 29        public void inc1() { 30            this.value += 4; 31        } 32        public void inc2() { 33            super.inc2(); 34            super.inc1(); 35        } 36    } 37 // Các lệnh trong main 38 C a1 = new C(10); 39 a1.inc1(); 40 a1.show(); 41 A a2 = new B(); 42 a2.inc1(); 43 a2.inc2(); 44 a2.show(); 45 A a3 = new C(3); 46 a3.inc1(); 47 a3.inc2(); 48 a3.show();</pre>
--	---

### Câu 3. (2.0 điểm) Cho biết lỗi trong các khai báo sau, giải thích và nêu cách khắc phục?

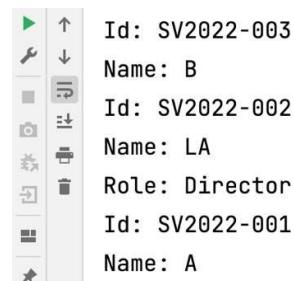
<pre> 1  interface IA { 2      IA(); 3      void show(); 4  } 5 6  class A implements IA { 7      protected String name; 8      public A(String name) { 9          this.name = name; 10     } 11 }</pre>	<pre> 12 class B extends A { 13     void show() { 14         System.out.print(this.name); 15     } 16 }</pre>
--	---

**Câu 4. (2.0 điểm)** Cho interface và các lớp như sau

```
interface IEm {  
    void output();  
    String getSortField();  
}  
class Em {  
    private String id;  
    private String name;  
}  
class Leader {  
    private String role;  
}  
  
class EmployeeManager {  
    private List<IEm> em = new ArrayList<>();  
  
    public void addEmployee(IEmployee e) {  
        this.em.add(e);  
    }  
    public void output() {  
        this.em.forEach(e -> e.output());  
    }  
}
```

Bô sung **các lớp** (*KHÔNG thay đổi các khai báo đã có*) trên để kết quả của đoạn chương trình trong phương thức main() như hình bên dưới.

```
EmployeeManager m = new EmployeeManager();  
m.addEmployee(new Em("A"));  
m.addEmployee(new Leader("LA", "Director"));  
m.addEmployee(new Em("B"));  
  
m.sort();  
m.output();
```



Chú ý: id của lớp Em là chuỗi có định dạng SV<năm-hiện-tại>-<3 chữ số tự tăng>, phương thức sort () trong EmployeeManager sắp xếp nhân viên giảm dần theo id.

**Câu 5. (2.0 điểm)** Một hệ thống quản lý đề cương môn học tại một trường Đại học. Các môn học bao gồm các thông tin mã môn, tên, số tín chỉ, mục tiêu, chuẩn đầu ra và nội dung môn học. Đối với các môn học có triển khai kết hợp LMS, yêu cầu bao gồm thông tin nội dung triển khai và phần trăm giảng dạy trên LMS. Mỗi môn học có thể có tối đa 3 môn học trước (học trước khi học môn trong đề cương) và 3 môn học tiên quyết (bắt buộc học và đã đạt môn học trước khi học môn trong đề cương). Để đánh giá kết quả môn học đạt chuẩn đầu ra, môn học được phép thiết kế tối đa 4 cột điểm đánh giá và tối thiểu 2 cột điểm. Mỗi cột điểm đánh giá bao gồm thông tin phương pháp đánh giá, nội dung đánh giá. Mỗi hình thức đánh giá có thể áp dụng cho các môn học khác nhau và tỷ trọng (tỷ lệ điểm chiếm trên điểm toàn môn học) của phương pháp đánh giá phụ thuộc giảng viên thiết kế từng môn có thể khác nhau. Hệ thống yêu cầu các chức năng cơ bản tạo một đề cương mới, thêm môn tiên quyết, môn học trước vào một đề cương, thêm hình thức đánh giá vào một đề cương.

Thiết kế sơ đồ lớp (class diagram) giải quyết các yêu cầu trên.