

Kiểm tra cuối kỳ

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG - ĐỀ 4

Thời gian: 90 phút, KHÔNG được dùng tài liệu

Câu 1. Trắc nghiệm (3 điểm)

1. Chọn các phát biểu đúng
 - a. Đa hình nghĩa là "Đối tượng thuộc lớp con có thể được đối xử như đối tượng thuộc lớp cha".
 - b. Biến địa phương kiểu boolean nhận giá trị mặc định là false.
 - c. ☒ Các đối tượng được tạo ra đều tồn tại trong bộ nhớ heap cho đến khi chương trình kết thúc.
 - d. Một đối tượng thuộc một lớp cài đặt một interface có thể được coi là một đối tượng thuộc kiểu interface đó.
2. Chọn các phát biểu sai
 - a. Khi một đối tượng thuộc lớp con được khởi tạo, hàm khởi tạo của lớp cha phải được gọi một cách tường minh. ✗
 - b. Nếu một lớp có khai báo các hàm khởi tạo, trình biên dịch sẽ không tạo hàm khởi tạo mặc định cho lớp đó. ✓
 - c. Có thể dùng từ khóa *static* cho hàm khởi tạo. ✗
 - d. Một khối khởi tạo static chạy trước khi hàm khởi tạo của một lớp được chạy. ✓
3. Chọn các phát biểu đúng, từ khóa *super*:
 - a. Có thể được sử dụng để gọi các phương thức *protected* hoặc *public* của lớp cha. ✓
 - b. Có thể được sử dụng để gọi phương thức *m* có mức truy cập *protected* hoặc *public* của lớp cha của lớp cha thông qua lệnh *super.super.m()* ✓
 - c. Chỉ dùng để gọi phương thức khởi tạo của lớp cha. ✗
 - d. Chỉ dùng để gọi phương thức *private* của lớp cha. ✗
4. Chọn các phát biểu đúng
 - a. Khối *catch* có thể có hoặc không có tham số ✗
 - b. Có thể có nhiều khối *catch* tương ứng với một khối *try* ✓
 - c. Thứ tự các khối *catch* không quan trọng ✓
 - d. Có thể tạo và ném ngoại lệ mới trong khối *catch* ✓
5. Chọn các phát biểu đúng: Lớp B thừa kế từ lớp A. Lớp A có định nghĩa phương thức *public final void m()*. Lớp B có định nghĩa phương thức *public void m()*.
 - a. Lỗi khi biên dịch
 - b. Đây là một ví dụ về ghi đè (overriding) ✗
 - c. Đây là một ví dụ về nạp chồng (overloading)
 - d. *m()* trong B bắt buộc phải là *final* ✗
6. Mẫu thiết kế nào được sử dụng để sao chép đối tượng
 - a. Abstract factory
 - b. Factory method

A
↑
B

c. Prototype

d. Adapter

7. Cho khai báo interface dưới đây, phát biểu nào sau đây là đúng về MAX_CONN

```
Interface Config{
    int MAX_CONN = 20;
}
```

a. Là static final

b. Là final

c. Là public static final

d. Không có ràng buộc gì

8. Cho lớp tổng quát Pair<T>, và hai lớp Person và Student, trong đó Student là lớp con của lớp Person. Chọn các phát biểu đúng:

a. Pair<Person> là lớp cha của Pair<Student> ✗

b. Pair<Person> và Pair<Student> không có quan hệ kế thừa ✗

c. Có thể khai báo Pair<? extends Person> p = new Pair<Student>()

d. Tất cả đều sai

Per.
↑
Student

9. Kết quả chạy của chương trình sau là gì? (Sinh viên viết kết quả được in ra khi chạy chương trình)

```
public class Animal {
    public void foo(){
        System.out.println("Animal...");
    }
}
```

```
Public class Dog extends Animal{
    public void foo(){
        System.out.println("Dog...");
    }
}
```

Animal
↑
Dog

```
Public class Test{
    Public static void main(String[] args){
        Animal a = new Dog();
        a.foo(); → dog
        Dog d = (Dog) a;
        d.foo(); → dog
    }
}
```


Smalley

10. Chọn các đáp án đúng

- Các lời gọi phương thức trong Java (ngoại trừ final/private method) đều là liên kết động ✓
- Phương thức thực thể có thể truy xuất các thuộc tính lớp ✓
- Phương thức khởi tạo không thể được khai báo với từ khoá *private* ✗
- Một lớp là trừu tượng bắt buộc phải chứa phương thức trừu tượng ✗

Câu 2. (3 điểm)

- Trong một chương trình Java, chuyện gì xảy ra nếu nhiều hơn một khối *catch* có thể bắt đối tượng ngoại lệ bị ném? Cho ví dụ minh hoạ. (1 điểm)
- Phát biểu: "Đối tượng của tất cả các lớp đều có thể được chuỗi hoá và ghi ra file" là đúng hay sai? Hãy giải thích. (1 điểm)
- Cho một ví dụ và giải thích về việc sử dụng giao diện (interface) làm kiểu của tham số phương thức. (1 điểm)

Câu 3. (2 điểm)

Công ty A muốn xây dựng một hệ thống Payroll. Hàng tháng, công ty cần tính tiền chi trả cho các hoá đơn (**Invoice**) và tiền lương cho nhân viên (**Employee**). Mặc dù các loại chi trả này thuộc các danh mục khác nhau và có cách tính khác nhau, nhưng chúng đều thực hiện nhiệm vụ tính tổng số tiền (*getPaymentAmount*) mà công ty cần chi trả. Trong đó, **Invoice** có thông tin về danh sách hàng hoá (**Goods**), phương thức *getPaymentAmount* tính và trả về tổng số tiền cần phải chi trả cho các hàng hoá trong hoá đơn đó. Một đối tượng **Goods** bao gồm thông tin về tên sản phẩm, số lượng, và giá. **Employee** bao gồm thông tin tên (*name*) và địa chỉ (*address*) của nhân viên, phương thức *getPaymentAmount* tính và trả về số tiền mà công ty cần trả cho nhân viên. Để tính lương cho nhân viên, công ty chia nhân viên thành ba loại: **SalariedEmployee**, **HourlyEmployee**, và **CommissionEmployee**. Cụ thể, **SalariedEmployee** mỗi tuần được trả cố định một khoản *weeklySalary* không quan tâm số giờ họ làm việc. **HourlyEmployee** được trả $h * \text{hourSalary}$, trong đó h và *hourSalary* lần lượt là số giờ làm việc trong tuần và mức lương của một giờ làm việc. **CommissionEmployee** được trả theo tỷ lệ hàng hoá họ bán được, theo công thức $\text{commissionRate} * \text{grossSales}$, trong đó *commissionRate* và *grossSales* lần lượt là tỉ lệ hoa hồng họ được nhận và tổng giá trị họ đã bán được.

- Hãy vẽ biểu đồ thiết kế lớp sử dụng lớp trừu tượng và/hoặc giao diện (interface) một cách thích hợp (1 điểm)
- Viết mã cài đặt các lớp cụ thể (concrete class) ở trên (1 điểm)

Câu 4. (2 điểm)

- Mô tả và cho ví dụ về mẫu thiết kế Singleton (1 điểm)
- Viết một phương thức tổng quát *sort()* với tham số là một mảng gồm các phần tử thuộc kiểu tổng quát, phương thức này sắp xếp các phần tử theo thứ tự tăng dần và trả về mảng đã sắp xếp. (1 điểm)

public static List<T> sort(List