

Lập trình hướng đối tượng HKI 2020 – 2021

Kiểm tra cuối kỳ

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

Thời gian: 90 phút, *KHÔNG* được dùng tài liệu.

**Câu 1.** Chọn các phương án trả lời đúng cho các câu hỏi sau (3 điểm)

1. Những câu nào đúng với phương thức khởi tạo

- ☒ a. Có cùng tên với lớp ✓
- b. Lớp con được kế thừa phương thức khởi tạo của lớp cha ?
- ☒ c. Không có giá trị trả về ✓
- d. Phải có mức độ truy cập là *public*

2. Chọn các câu đúng về lớp trừu tượng

- ☒ a. Lớp trừu tượng bắt buộc được khai báo với từ khóa *abstract* ✓
- b. Lớp trừu tượng phải có phương thức trừu tượng
- c. Không cần thiết khai báo lớp trừu tượng với từ khóa *abstract* nếu có chứa phương thức trừu tượng
- ☒ d. Lớp cụ thể không thể chứa phương thức trừu tượng ✓

3. Từ khóa *super*:

- ☒ a. Có thể được sử dụng để gọi các phương thức là *protected* hoặc *public* của lớp cha ✓
- b. Có thể được sử dụng để gọi các phương thức là *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. Gọi phương thức *private* trong lớp cha.

4. Giả sử có các câu lệnh: *A obj=new A(); m(obj);* ngay sau khi thực hiện xong phương thức *m()*

- a. Giá trị của các thuộc tính của *obj* không thể bị thay đổi
- b. *obj* có thể trỏ đến một đối tượng khác
- ☒ c. Giá trị của các thuộc tính của *obj* có thể bị thay đổi ✓
- ☒ d. Tham chiếu *obj* không thể trỏ đến một đối tượng khác ✓

5. Chọn các phát biểu đúng về phương thức tĩnh (static)

- a. Phải khai báo là *public*
- b. Không được khai báo là *private*
- ☒ c. Phạm vi truy cập được khai báo giống như những phương thức không phải là tĩnh (non-static). ✓
- ~~d. Phạm vi truy cập mặc định là *public* ?~~ package-private



Lập trình hướng đối tượng HKI 2020 – 2021

6. Lớp cụ thể *B* kế thừa từ lớp cụ thể *A*. Các câu lệnh *A a = new B(); a.m()* biên dịch thành công trong các trường hợp nào?
- ☒ a. *B* có định nghĩa phương thức *m()* là *public* ✓
  - ☒ b. *A* có định nghĩa phương thức *m()* là *public* ✓
  - ☐ c. Cả *A* và *B* có định nghĩa phương thức *m()* là *public* ✓
  - ☐ d. Chỉ cần *A* hoặc *B* có phương thức *m()* là *public*
7. Có thể tạo lớp dẫn xuất từ các loại lớp nào sau đây?
- ☐ a. abstract
  - ☐ b. public
  - ☒ c. private ✓
  - ☐ d. final
8. Khi một đối tượng thuộc lớp con được tạo ra, phương thức khởi tạo của lớp nào sẽ được thực hiện xong trước.
- ☒ a. Phương thức khởi tạo của lớp cha ✓
  - ☐ b. Phương thức khởi tạo của lớp con
  - ☐ c. Không xác định được
  - ☐ d. Tùy vào đoạn mã tạo đối tượng lớp con đấy
9. Câu nào mô tả đúng nhất về phương thức trừu tượng
- ☐ a. Là phương thức được khai báo và định nghĩa trong một lớp trừu tượng
  - ☒ b. Là phương thức chỉ có phần khai báo ✓
  - ☒ c. Không thể được khai báo trong lớp cụ thể ✓
  - ☐ d. Là phương thức có phần định nghĩa rỗng (chỉ gồm 2 ngoặc, không có câu lệnh nào)
10. Để chạy được chương trình Java từ một lớp *A* (chọn câu đúng nhất)
- ☐ a. Trong lớp *A* phải có định nghĩa phương thức *main(String[] args)*
  - ☒ b. Lớp *A* không có định nghĩa phương thức *main(String[] args)* thì *A* phải được kế thừa phương thức này ✓
  - ☐ c. *A* chỉ cần một có một phương thức được khai báo là *public static*
  - ☐ d. Không câu nào ở trên đúng

## Câu 2 (3 điểm)

- a. Cho một ví dụ để thể hiện chức năng của khối *finally* trong xử lý ngoại lệ (1 điểm)
- b. Giao diện (interface) có thể sử dụng như một kiểu dữ liệu. Cho một ví dụ và giải thích về việc sử dụng giao diện làm tham số của một phương thức (1 điểm)

nh hướng đối tượng HKI 2020 – 2021

- c. Phương thức *boolean equals(Object anotherObject)* dùng để so sánh hai đối tượng có “bằng nhau” hay không. Hãy cho một ví dụ thể hiện cách định nghĩa và cách sử dụng phương thức đó. (1 điểm)

**Câu 3 (2 điểm)**

Giả sử chúng ta đang viết ứng dụng quản lý nhân sự cho công ty phần mềm ABC. Mỗi nhân viên (*Employee*) chỉ có thể là lập trình viên (*Developer*), kiểm thử viên (*Tester*), hoặc quản lý (*Manager*). Người quản lý (*Manager*) giám sát một số nhân viên (bao gồm cả những quản lý khác). Thông tin về nhân viên ở đây bao gồm họ tên (*name*) và các công việc hiện tại (*listOfTask*). Lớp *Employee* cung cấp phương thức *displayInfo()* để in ra tên, danh sách công việc, và tên của người quản lý của nhân viên. Lớp *Manager* có phương thức *addEmployee(Employee e)* để thêm một nhân viên vào danh sách những nhân viên do người đó quản lý.

- Hãy thiết kế lớp cho ứng dụng trên (sinh viên có thể vẽ biểu đồ lớp hoặc viết mã Java). (1 điểm)
- Cài đặt phương thức *displayInfo()* (0,5 điểm)
- Cài đặt phương thức *addEmployee(Employee e)* (0,5 điểm)

**Câu 4 (2 điểm)**

- Cài đặt một lớp tổng quát biểu diễn cặp 2 phần tử thuộc hai lớp khác nhau (với các phương thức *get/set*) (0,5 điểm)
- Hãy viết một phương thức tĩnh *findMax(...)* dạng tổng quát (generic method) của lớp tiện ích *MyUtility* để tìm phần tử “lớn nhất” trong một mảng chứa các phần tử có kiểu chưa xác định trước. (0,5 điểm)
- Loại trừ mẫu thiết kế (design pattern) *Singleton*, mô tả 2 trong số các mẫu thiết kế khác mà sinh viên biết. (1 điểm)

-----