

# Giải bài tập trắc nghiệm Java

(file 20 câu trắc nghiệm của Thầy Phan Thượng Càng)

## Một số lưu ý:

- Trong 1 câu có thể có nhiều câu đúng. Chọn câu đúng nhất!
- Cú pháp sai rất ngu, cảm giác cố tình cho sai cú pháp chứ đó không phải là lỗi mà một sinh viên bình thường có thể sai được. Điều đó đi ngược lại với ý nghĩa của hình thức trắc nghiệm( Kiểm tra kiến thức với các mức độ ứng dụng khác nhau)
- Bạn nên bỏ khái niệm Code Clean khi học một số thầy cô trong trường!
- Tài liệu vẫn có nhiều đó, giáo trình khoa bán đã rất oke rồi. Đọc kỹ rồi làm thôi. Phân biệt java OOP và các khái niệm OOP chung cho các ngôn ngữ lập trình.

----Goodluck!----

## Câu 1: D

- A. Sai ý nghĩa sử dụng tham số (Khai báo nhưng không dùng). Tuy nhiên cú pháp đúng => Đúng.
  - B. Code không clean + cmt ngu nên sai cú pháp (Mất } ) => Sai.
  - C. Kiểu trả về void mà có return x => Sai.
  - D. Ép kiểu hẹp phải thực hiện tường minh (Có thể mất dữ liệu!). Đề ép kiểu từ long sang int(Ép ngu vãi ra!) => Đúng
- Câu A và D đều đúng, nhưng D đúng hơn => Chọn D

## Câu 2: B

- A. Đúng.
  - B. Không khai báo kiểu trả về mà trong hàm lại có kiểu trả về => Sai.
  - C. Ký tự char thực chất cũng là số nguyên (Xem bảng mã Unicode) => Đúng.
  - D. Cú pháp đúng, dù hàm viết chỉ để làm ví dụ ngu cho bt trắc nghiệm này => Đúng
- A.

## Câu 3: B

Lưu ý điều kiện đề cho là **cùng một gói!**

- A. Thuộc tính masocb là private nên không truy xuất trực tiếp được => Sai.
- B. Thuộc tính hesoluong là protected nên có thể truy xuất trực tiếp được trong cùng 1 gói => Đúng.
- C. Dạng câu A => Sai.
- D. Do B đúng nên D sai.

Tóm lại phạm vi truy cập:

Public: không giới hạn

Protected: trong cùng lớp, lớp con, cùng gói. (nếu lớp con ngoài package thì vẫn truy xuất dc)

Private: trong cùng lớp

None: trong cùng lớp, cùng gói, không có con cháu (nếu con ngoài lớp)

#### Câu 4: D

- A. Hàm in không phải static nên không truy xuất trực tiếp qua tên class được => Sai
- B. Hàm in không có kiểu trả về (chuỗi không thể cộng void) => Sai
- C. Hàm in không thể truy xuất 1 mình (không thông qua đối tượng) => Sai
- D. Đúng.

#### Câu 5: A

- A. Hàm in nạp chồng => Đúng
- B. Không có kiểu tham số hình thức => Sai
- C. Hai hàm trong cùng 1 lớp không thể cùng tên cùng chữ ký => Sai
- D. Hàm xây dựng (constructor) không có kiểu trả về => Sai

#### Câu 6: C

- A. Do 0.33 mặc định là double nên cần ghi là 0.33f để chuyển về float => Đúng
  - B. Đúng
  - C. Cú pháp rút gọn += đã ép kiểu về phải rồi mới cộng ( mà tham số hình thức cũng đã có kiểu phù hợp => Đúng
  - D. Không sử dụng n nhưng vẫn đúng (float + int = float) => Đúng
- Đúng nhất là C (có sử dụng biến h)

#### Câu 7: C

- A. Sai tên hàm xây dựng => Sai
- B. Định nghĩa tên trùng => Sai
- C. Hàm xây dựng sao chép => Đúng
- D. Sai tên hàm xây dựng => Sai.....

#### Câu 8: B

- B. Không tìm thấy hàm xây dựng phù hợp => Sai
- Các câu còn lại đúng

#### Câu 9: C

Các ý đều đúng. Tuy nhiên có câu C hợp lý nhất (sao chép sâu) => chọn C

### Câu 10: D

Các kiểu khác là truyền giá trị nguyên thủy và chỉ có D là truyền giá trị tham chiếu.

Về bản chất các tham số trong java luôn được truyền bằng giá trị. Tuy nhiên do java có nhiều loại biến nên ý nghĩa của các giá trị truyền có thể khác nhau:

- Kiểu Nguyên Thủy (int, char, double,...): truyền giá trị kiểu nguyên thủy
- Kiểu lớp(Class): Truyền giá trị tham chiếu (Tham chiếu)
- Kiểu mảng(array): Truyền tham chiếu

(\* vậy trong java không thể thay đổi giá trị tham số thực tế bên ngoài. kể cả là truyền giá trị hay tham chiếu đều thì chỉ thay đổi giá trị các tham số hình thức bên trong hàm. Điều này thể hiện tính bao gói trong java)

### Câu 11: C

- A. Truy cập được do ganThongTin nằm trong class CBGD => Đúng
- B. Lớp CBGD là class con của CBCC => truy cập thuộc tính protected được => Đúng
- C. hoten là thuộc tính private của class CBCC nên không truy xuất ở class con được => Sai
- D. namsinh là default (mặc định-chỉ thị này trong java) nên truy xuất được => Đúng

### Câu 12: B

- A. Do super() là gọi hàm xây dựng của cha => Sai
- B. Gọi phương thức nào ở class cha thì phải **super.<Tên method>()**; => đúng
- C. Thiếu thông tin của class cha => Đúng nhưng không đủ
- D. Thiếu thông tin của class con => Đúng nhưng không đủ

### Câu 13: B

- A. Có các thuộc tính không thể truy xuất trực tiếp ở class con => Sai
- B. Cú pháp super(cb); mà cb thuộc class CBCC thực chất là java đã tự động ép kiểu (upcasting) => Đúng
- C. Sai do không thể gọi trực tiếp phương thức ở class cha mà không qua super! => Sai
- D. class CBCC không có constructor thích hợp => Sai

### Câu 14: D

- A. Do namsinh là default truy cập được trong package => Đúng
- B. Đúng
- C. Đúng do class con truy cập các phương thức của class cha được => Đúng
- D. 3 Câu đúng => Đúng hết => Chọn D

### Câu 15: B

- A. namsinh là property không phải method => Sai

- B. hesoluong là protected nên có thể truy cập trong cùng gói => Đúng
- C. Sai trong CBCC không có method ganThongTin(); => Sai
- D. Có câu đúng => Sai

#### Câu 16: B

- A. Sai
- B. Đúng:
  - 1. Có bao gói: các thuộc tính có access modify là private
  - 2. Có đa hình: nạp chồng constructor GiangDay
- C. Sai
- D. Sai

Tóm tắt chút kiến thức:

Java có 2 quan hệ: thừa kế và quan hệ thành phần (hoặc bao gồm)

Mức độ phức tạp: bao gói => đa hình => thừa kế.

Bao gói: giới hạn phạm vi truy cập (có thêm default)

Đa hình: nạp chồng và nạp đè (nạp đè là nạp chồng ở class con)

Thừa kế: tổng hợp tất cả tính chất trên

Java có thêm ảo hóa: interface

#### Câu 17: C

- A. Chép tham chiếu (hai biến cùng tham chiếu 1 vùng nhớ) => Đúng
  - B. Sao chép cạn (phương thức clone() thuộc class Object – dùng sẽ quăng lỗi- chi tiết gg) => Đúng
  - C. Sao chép sâu => Đúng
  - D. GiangDay là constructor => Sai
- Câu A, B, C đều đúng nhưng C là sao chép sâu oke nhất nên nó đúng nhất!

#### Câu 18: B

- A. giaovien là kiểu CBGD => Sai
- B. Đúng
- C. sai như câu A => Sai
- D. CBGD là tên lớp => Sai

#### Câu 19: B

- A. Không có hàm layHoTen() => Sai
- B. Truy cập được thành phần được bảo vệ trong cùng gói => Đúng
- C. Cần có kiểu trả về => Đúng
- D. thuộc tính giaovien kiểu CBGD, mà class này không có thuộc tính CBCC nên không thể sử dụng được phương thức nhap() (có thể giải thích kiểu khác

là hàm nhập() đã nạp đề ở class con (CBGD) nên không thể truy cập phương thức nhập() ở class CBCC=> Sai  
Giữa B và D. Chọn B là đúng hơn!

**Câu 20: B**

- A. Cú pháp sai (dấu ngoặc) => Sai
- B. Hàm lấy gv trả về CBGD nên truy xuất tiếp phương thức tienLuong() được => Đúng
- C. Hàm in bị nạp đề không truy xuất của cha(cũng có thể giải thích như câu 19D) => Sai
- D. Gd.giaovien (giaovien thành phần riêng nên không truy cập được)