Câu 22:

a) Lập trình hướng đối tượng là phương pháp lập trình lấy đối tượng làm nền tảng để xây dựng chương trình. Hay lập trình hướng đối tượng đó là phương pháp lập trình dựa trên kiến trúc lớp (class) và đối tượng (object).

b) Đối tượng là những sự vật, sự việc mà nó có những tính chất, đặc tính, hành động giống nhau và ta gom góp lại thành đối tượng giống trong thực tế cuộc sống. Khi lập trình OOP, chúng ta sẽ định nghĩa các lớp (class) để gom (mô hình) các đối tượng thực tế.

VD: người, vật hoặc 1 bảng dữ liệu, . . .

Một đối tượng bao gồm 2 thông tin: thuộc tính và phương thức.

Thuộc tính chính là những thông tin, đặc điểm của đối tượng. Ví dụ: một người sẽ có họ tên, ngày sinh, màu da, kiểu tóc, . . .

Phương thức là những thao tác, hành động mà đối tượng đó có thể thực hiện. Ví dụ: một người sẽ có thể thực hiện hành động nói, đi, ăn, uống, . . .

c) Các đối tượng có các đặc tính tương tự nhau được gom lại thành 1 lớp đối tượng. Bên trong lớp cũng có 2 thành phần chính đó là thuộc tính và phương thức. Ngoài ra, lớp còn được dùng để định nghĩa ra kiểu dữ liệu mới.

d) Các đặc trưng cơ bản của lập trình hướng đối tượng:

- Tính đóng gói: Các dữ liệu và phương thức có liên quan với nhau được đóng gói thành các lớp để tiện cho việc quản lý và sử dụng. Ngoài ra, đóng gói còn để che giấu một số thông tin và chi tiết cài đặt nội bộ để bên ngoài không thể nhìn thấy.

VD: các thuộc tính họ tên, tuổi, nghề nghiệp được đóng gói trong 1 class người để thuận tiện quản lí, và khi để private thì bên ngoài không thế thấy được thông tin bên trong lớp người.

- Tính trừu tượng: Khi viết chương trình theo phong cách hướng đối tượng, việc thiết kế các đối tượng ta cần rút tỉa ra những đặc trưng chung của chúng rồi trừu tượng thành các interface và thiết kế xem chúng sẽ tương tác với nhau như thế nào.

VD: Bạn chạy xe tay ga thì có hành động là tăng ga để tăng tốc, thì chức năng tăng ga là đại diện cho trừu tượng (abstraction). Người dùng chỉ cần biết là tăng ga thì xe tăng tốc, không cần biết bên trong nó làm thế nào.

- Tính kế thừa: Lớp cha có thể chia sẽ dữ liệu và phương thức cho các lớp con, các lớp con khỏi phải định nghĩa lại, giúp chương trình ngắn gọn.

VD: học sinh, giáo viên đều có chung các thuộc tính họ tên, năm sinh, quê quán,… thì khi đó ta sẽ tạo 1 lớp cha person chứa các thuộc tính đó và các lớp con học sinh, giáo viên kế thừa lại và sử dụng các thuộc tính của lớp cha.

- Tính đa hình: Là hiện tượng các đối tượng thuộc các lớp khác nhau có thể hiểu cùng một thông điệp theo các cách khác nhau.

VD: Ta có 2 con vật: chó, mèo. Cả 2 con vật này đều là lớp động vật. Nhưng khi ta bảo cả 2 động vật kêu thì con chó sẽ kêu gâu gâu, con mèo sẽ kêu meo meo. Vậy trong ví dụ chó, mèo xem như là các đối tượng. 2 con vật có thể hiểu cùng kêu nhưng theo các cách khác nhau.

Câu 23:

Overload (nạp chồng) là hai phương thức cùng tên nhưng số lượng tham số trong các phương thức khác nhau hoặc các loại dữ liệu của tham số khác nhau và nằm trong một class.

VD: Calculator có hai method với cùng một tên là sum tuy nhiên method thứ hai được định nghĩa với số lượng tham số nhiều hơn method thứ nhất. Chúng ta nói method sum thứ hai overload method sum thứ nhất

Override (ghì đè) là hai phương thức có cùng tên, cùng kiểu dữ liệu trả về, cùng phạm vi truy cập (Access Modifie) nằm ở hai class khác nhau.

VD: có hai class là Animal và Bird trong đó class Bird kế thừa class Animal. Cả hai class đều định nghĩa một method với tên là canFly() và method này có cùng số lượng tham số. Chúng ta nói method canFly() trong lớp cha Animal bị override bởi lớp con Bird.

Câu 24:

Constructor là một phương thức đặc biệt, nó được dùng để khởi tạo và trả về đối tượng của lớp mà nó được định nghĩa. Constructor sẽ có tên trùng với tên của lớp mà nó được định nghĩa và chúng không được định nghĩa một kiểu giá trị trả về.

Nếu trong một đối tượng, chúng ta không định nghĩa một constructor nào cả thì mặc định Java sẽ thêm một constructor mặc định vào đối tượng của chúng ta.

Constructor mặc định sẽ không chứa bất kỳ tham số nào và khi được gọi để khởi tạo đối tượng, nó cũng gọi constructor của lớp cha và khởi tạo giá trị mặc định cho các instance variable.

VD: để tạo ra 1 đối tượng học sinh mới thì ta cần gọi 1 constructor qua toán từ new tùy thuộc tham số truyền vào, có hoặc không.

Câu 25:

|  |  |  |
| --- | --- | --- |
| **STT** | **Lớp trừu tượng** | **Interface** |
| 1 | Lớp trừu tượng có thể có các phương thức abstract và non-abstract | Interface chỉ có thể có phương thức abstract |
| 2 | Lớp trừu tượng không hỗ trợ đa kế thừa | Interface hỗ trợ **đa kế thừa** |
| 3 | Lớp trừu tượng có thể có các biến **final, non-final, static và non-static** | Interface chỉ có các biến **static và final** |
| 4 | Lớp trừu tượng có thể có phương thức static, phương thức main và constructor | Interface không thể có phương thức static, main hoặc constructor. |
| 5 | Từ khóa abstract được sử dụng để khai báo lớp trừu tượng | Từ khóa interface được sử dụng để khai báo Interface |
| 6 | Lớp trừu tượng có thể cung cấp trình triển khai của Interface | Interface không cung cấp trình triển khai cụ thể của lớp abstract |
| 7 | Ví dụ: public abstract class Shape{ public abstract void draw(); } | Ví dụ: public interface Drawable{ void draw(); } |

Câu 26:

a) Exception là một tình trạng bất thường. Trong Java, Exception là một sự kiện mà phá vỡ luồng chuẩn của chương trình. Nó là một đối tượng mà được ném tại Runtime. Một exception (ngoại lệ) trong Java là một vấn đề xảy ra trong quá trình thực hiện của chương trình. Một ngoại lệ có thể xảy ra với nhiều lý do khác nhau, như dưới đây:

- Người dùng nhập dữ liệu không hợp lệ

- Một file cần được mở nhưng không thể tìm thấy.

- Kết nối mạng bị ngắt trong quá trình thực hiện giao tiếp hoặc JVM hết bộ nhớ.

b) Xử lý ngoại lệ để duy trì luồng chuẩn của ứng dụng. Khi gặp lỗi chương trình vẫn thực hiện.

c) Finally : Nó nằm ở cuối cùng của khối lệnh try – catch – finally, nó luôn luôn được thực hiện ngay cả khi chương trình có bắt được lỗi hay không!

d) Throws : được sử dụng để khai báo một ngoại lệ. Nó thể hiện thông tin cho lập trình viên rằng có thể xảy ra một ngoại lệ, vì vậy nó là tốt hơn cho các lập trình viên để cung cấp các mã xử lý ngoại lệ để duy trì luồng bình thường của chương trình.