|  |  |
| --- | --- |
| **CODEGYM** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc** |

**BÀI HỌC NGÀY 21.07.2021**

**Học viên:** Trần Tiến Dũng

**Lớp:** C0621G1

|  |  |
| --- | --- |
| **LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG** | |
| 1. **OOP (Object oriented Programming)**  * Mục tiêu * Định nghĩa   - *Các loại OOP*   * *Cú pháp* * *Truy xuất thuộc tính và phương thức của đối tượng* * *Không gian tên phụ (Sub-namespace)* * *Từ khóa “this”*   **1.1. Lập trình thủ tục**   1. ***Hàm Demo***  * *Mục tiêu* | + Trình bày được mô hình Lập trình hướng đối tượng  + Trình bày được khái niệm lớp  + Trình bày được khái niệm đối tượng  + Trình bày được khái niệm phương thức của đối tượng  + Trình bày được 4 đặc trưng của Lập trình hướng đối tượng  + Là lập trình hướng tới các đối tượng: Trừu tượng, mô hình hóa các đối tượng thực vào lập trình và chỉ giữ lại các tính chất quan trọng  + Hướng tiếp cận đa dạng, gần với thực tế,  + Tìm và phân tích mối quan hệ giữa các đối tượng trong bài toán  + Mã lệnh và dữ liệu liên kết trong thể thống nhất  + Ưu thế:   1. Khả năng tái sử dụng cao 2. Ổn định và dễ bảo trì 3. Chi phí giảm dần.   + Đặc tính cơ bản:   1. Tính bao gói (encapsulation): Khả năng truy xuất vào các thành phần của một đối tượng trong khi vẫn đảm bảo che giấu các đặc tính riêng tư bên trong đối tượng 2. Tính kế thừa: Cho phép đối tượng có thể chia sẻ hay mở rộng các thuộc tính hoặc phương thức mà không phải tiến hành định nghĩa lại 3. Tính trừu tượng: Loại bỏ những thuộc tính và hành vi không quan trọng của đối tượng , chỉ giữ lại các thuộc tính và hành vi có liên quan đến vấn đề đang giải quyết 4. Tính đa hình: Thể hiện khi với cùng một phương thức nhưng có thể có cách ứng xử khác nhau ở những lớp cùng giao diện   **+** Lập trình hướng sự kiện  + Lập trình hướng thủ tục: chạy theo một trình tự được đặt ra  Function name\_of\_function (parameter1, parameter2, …){  // code to be execute}  + Truy xuất thuộc tính: “objectName.propertyName;”  + Truy xuất phương thức; “objectName.methodName”   * Dấu chấm (".") để truy xuất các thuộc tính và phương thức của đối tượng. Tên của đối tượng (person) giống như một namespace (không gian tên) - phải được viết trước dấu chấm để có thể truy xuất vào bất cứ thành phần nào được bao gói (encapsulated) bên trong đối tượng. Sau dấu chấm là tên của thành phần bạn muốn truy xuất   VD: name : {  first: 'Bob',  last: 'Smith'},  + Là các giá trị truy xuất    VD: person.name.first  person.name.last  hoặc person.name[0]  person.name[1]   * Cặp (“[]”) đùng để truy xuất các thuộc tính của đối tượng   person.age  person.name.first  có thể dùng  person['age']  person['name']['first']  + Dùng để tham chiếu đến đối tượng hiện hành  + Đặc trưng bởi các thủ tục, bao gồm các tập lệnh tuyến tính nối tiếp nhau  + Chia nhỏ bài toán thành các module chức năng  + Dữ liệu và mã lệnh phân tán  + Hạn chế:   1. Tính ổn định giảm khi hệ thống phát triển 2. Khó bảo trì và tái sử dụng 3. Chi phí phát triển cao   + Khai báo và sử dụng được hàm không tham số  + Khai báo và sử dụng được hàm có tham số  + Khai báo và sử dụng được hàm có return |
|  | |