**LỘ TRÌNH ĐÀO TẠO CHƯƠNG 1**

1. Các thời kì trong lập trình:
   1. Thời kì 1: Lập trình tuyến tính.
      * Là thời kì các chương trình được viết ra giấy theo ngôn ngữ máy (hệ nhị phân).
      * Khó lưu trữ, khó hiểu.
   2. Thời kì 2: Lập trình hướng thủ túc – hàm.

* Là lập trình lấy hành động làm trung tâm.
* Phân chương trình thành các công việc nhỏ hơn để giải quyết được gọi là Hàm.

VD: Hành động Nấu: cơm, canh,.... || Hành động Chiên: cơm, rau, thịt, cá…

VD trong lập trinh:

+ Viết hàm Nhập tên học sinh, điểm, năm sinh,…

+ Viêt hàm Xuất tên học sinh, điểm, năm sinh,…

+ Viết hàm Tính điểm TB.

+ Viết hàm main chạy kiểm tra kết quả.

void main() {

HocSinh hs;

Nhap(hs);

Xuat(hs);

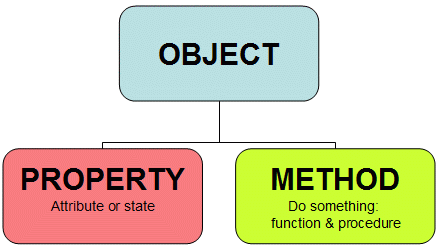
Cout << “Điểm TB: ” << DiemTB(hs);

}

* 1. Thời kì 3: Lập trình hướng đối tượng. (Object – oriented Programing)
* Là phương pháp dựa trên kiến trúc đối tượng (object) và lớp (class).

c.1. Đối tượng (Object)

* + Là thực thể mà chúng ta có thể cảm nhận được. Trong đối tượng có 2 thành phần quan trọng:
    - Thuộc tính (attribute): mô tả tính chất đối tượng.
    - Phương thức (method): các hành động liên quan đến đối tượng.



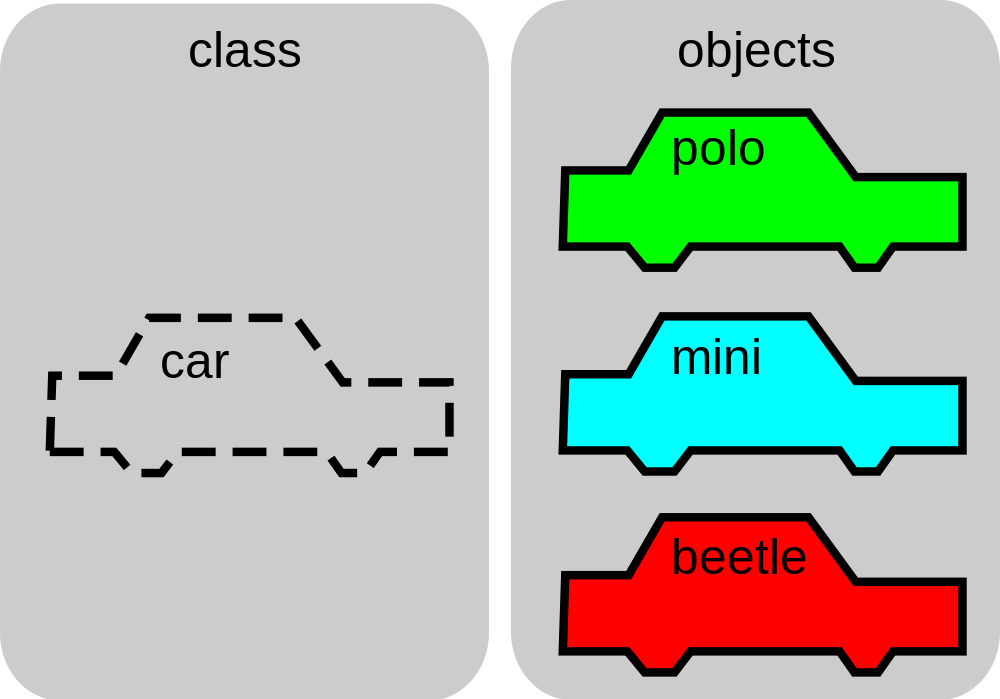
VD: XE, CHÓ, MÈO, NHÀ,…. là một đối tượng

VD: Trong đối tượng CHÓ có: Attribute: màu lông, màu mắt, 4 cái cẳng,….

Method: sủa, ngủ, ăn, “đánh dấu lãnh thổ”

c.2. Lớp (class)

* + Là mô hình hóa các nhóm đối tượng cùng loại.



VD: Trong lớp XE: có loại xe Langbogini, Ferrari, Toyota,….



NOTE: + Đối tượng được nhìn thấy đầu tiên sau đó mô hình hóa thành Lớp (Đối tượng có trước).

+ Khi muốn tạo Đối tượng thì phải có Lớp (Lớp có trước).

c.3. Tầm vực trong hướng đối tượng:

* Private: các thuộc tính hoặc các phương thức không thể truy xuất ra bên ngoài class, chỉ được gọi trong phạm vi class.

VD: Tiền của mình là để chính mình dùng, người khác không thể tự lấy tiền của mình dùng được.

* Public: các thuộc tính hoặc các phương thức có thể truy xuất ra bên ngoài class.

VD: Khi đi khỏi nhà nhưng không đóng và khóa cửa lại, ăn trộm, hàng xóm, bất cứ ai muốn cũng có thể vào nhà thích cái gì lấy cái ấy.

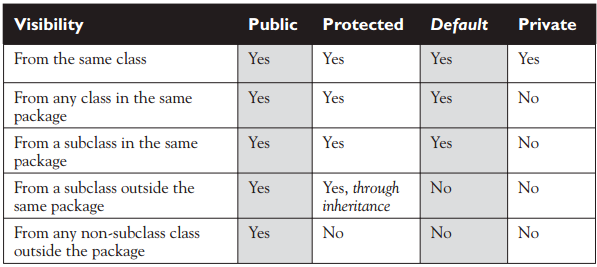
* Protected: các thuộc tính các phương thức không thể truy suất ra bên ngoài class, nó chỉ được gọi trong lớp và lớp thừa kế.

VD: Tiền của ba mẹ con cái có thể dùng được nhưng ông hàng xóm không thể lấy tiền đó để sử dụng được.

VD: Gia tài của cha mẹ được người con thừa kế.

* Quay lại ví dụ đầu: Hành động Nấu: cơm, canh,.... || Hành động Chiên: cơm, rau, thịt, cá…
* Giờ thành: Đối tượng cơm: có các hành động nấu, chiên,..

Đối tượng Cá: có các hành động hấp, chiên, rán, kho,....



c.4. Các bước trong hướng đối tượng:

* Bước 1: Xác định các lớp và đối tượng có trong đề bài (GIẤY).
  + Thuộc tính (Attribute)
  + Phương thức (Method)
* Bước 2: Khai báo các lớp và đối tượng (MÁY).
* Bước 3: Khai báo các phương thức.
* Bước 4: Cài đặt các xử lý của phương thức đó.
* Bước 5: Tạo đối tượng, gọi các phương thức trong hàm main.

c.5. Các kỹ thuật cơ bản:

* Viết hàm main(): + Khi tạo project, click vào ổ khởi tạo hàm main.

+ Bấm main -> giữ ctrl và bấm space -> enter.

* Viết chú thích: + Bấm // hoặc /\*\*/

+ Tổ hợp phìm cmt nhanh ctrl + / (cmt từng dòng).

+ Tổ hợp ctrl + shift + / (cmt khối), để tắt cmt khối bấm ctrl + shift + \.

c.6. Kiểu dữ liệu và dạng cơ bản:



* Gán giá trị VD: boolean check = true;
* Biểu thức toán học cơ bản: +, -, \*, /, %, ++, --, =, ==, &&, ||. >=, <=, !=
* Giả sử biến A giữ giá trị 10, biến B giữ giá trị 20, thì:

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Miêu tả** | **Ví dụ** |
| + | Phép cộng | A + B sẽ cho kết quả 30 |
| - | Phép trừ: trừ toán hạng trái cho toán hạng phải | A - B sẽ cho kết quả -10 |
| \* | Phép nhân | A \* B sẽ cho kết quả 200 |
| / | Phép chia | B / A sẽ cho kết quả 2 |
| % | Phép chia lấy phần dư | B % A sẽ cho kết quả 0 |
| ++ | Phép lượng gia: lượng gia giá trị toán hạng thêm 1 | B++ sẽ cho kết quả 21 |
| -- | Phép lượng giảm: lượng giảm giá trị toán hạng đi 1 | B-- sẽ cho kết quả 19 |

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Miêu tả** | **Ví dụ** |
| == | Kiểm tra nếu giá trị của hai toán hạng có cân bằng hay không, nếu có thì điều kiện là true. | (A == B) là không true. |
| != | Kiểm tra nếu giá trị hai toán hạng là cân bằng hay không, nếu không cân bằng, thì điều kiện là true | (A != B) là true. |
| > | Kiểm tra nếu toán hạng trái có lớn hơn toán hạng phải hay không, nếu có thì điều kiện là true | (A > B) là không true. |
| < | Kiểm tra nếu toán hạng phải có lớn hơn toán hạng trái hay không, nếu có thì điều kiện là true | (A < B) là true. |
| >= | Kiểm tra nếu toán hạng trái có lớn hơn hoặc bằng toán hạng phải hay không, nếu có thì điều kiện là true | (A >= B) là không true. |
| <= | Kiểm tra nếu toán hạng phải có lớn hơn hoặc bằng toán hạng trái hay không, nếu có thì điều kiện là true | (A <= B) là true. |