**1. Sự khác biệt giữa C và C++**

|  |  |  |
| --- | --- | --- |
|  | **C** | **C++** |
|  | Là ngôn ngữ hướng thủ tục | Là ngôn ngữ hướng thủ tục và hướng đối tượng. |
| Hàm nhập xuất | printf, scanf | Cout <<, cin >> |
| Tiền xử lý | #define | Sử dụng hàm inline |
| kiểu nhập xâu | Không có | string |
| Cấp phát động | malloc, calloc | new |
| Giải phóng bộ nhớ | free() | delete |

**2. Tìm hiểu các kiến thức liên quan đến lập trình hướng đối tượng C++**

**a. Khái niệm về lớp**

**-** Lớp là một tập hợp các đối tượng có chung thuộc tính và phương thức. Nó là một thực thể logic.

- Một class sẽ định nghĩa lên một kiểu dữ liệu mới.

- Mối một thể hiện của lớp sẽ cho một đối tượng mới.

**b. Phạmvi truy cập**

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

- Protected: các thuộc tính, các phương thức không thể truy xuất ra bên ngoài class, nó chỉ gọi trong class và class kế thừa (subclass).

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

**c. Hàm khởi tạo, hàm hủy**

- Hàm khởi tạo: là hàm có tên trùng với class, không có kiểu dữ liệu. Là hàm khởi tại giá trị cho đối tượng đầu tiên, nếu không có hàm này khi ta tạo ra một đối tượng vô tình xuất dữ liệu ra thì không có bị lỗi.

- Hàm hủy: là hàm có tên trùng với class, không có kiểu dữ liệu và có dấu “~” ở đầu. Là hàm dùng để giải phóng bộ nhớ nếu class có con trỏ.

**d. Hàm ảo và cơ chế đa hình** (polymorphism**)**

**\*** Hàm ảo

- Hàm ảo là cơ chế của C++ cho phép cài đặt đa cấu hình động, và được khai báo với từ khóa virtual ở đầu tên hàm trong superclass.

- Nếu có nhiều subclass cùng kế thừa superclass, và trong các subclass có tên hàm giống với superclass. Khi ta tạo nhiều đối tượng khác nhau có kiểu dữ liệu của superclass, khi thực thi lời gọi hàm thì chương trình sẽ thực hàm ở trong hàm con. (câu này cũng là câu trả lời cho bài 1 lập trình hướng đối tượng: vì sao các hàm trong lớp K58Menu lại để là virtaul)

\* Cơ chế đa hình.

- Overload (nạp chồng).

Trong một class có 2 or nhiều hàm có tên trùng nhau, nhưng khác kiểu dữ liệu trả về, khác kiểu tham biến.

- Override (ghi đè).

Cái này diễn ra ở trong subclass, khi tên hàm của subclass trùng với tên hàm của superclass.

**e. Kỹ thuật thừa kế.**

\* Đơn thừa kế

- Có nghĩa là, một class dẫn xuất chỉ kế thừa từ một lớp cơ sở.

**-** Cú pháp: class derived-class: access baseA

Trong đó: derived-specifier là tên class dẫn xuất.

Access là public, protected hoặc là private ; nếu để trống phần này, mặc định sẽ là private

baseA là tên lớp cơ sở, trong đó các class đã tồn tại.

ví dụ : class SinhVien : public Nguoi{

}

\* Đa thừa kế.

- Trong C++ cho phép đa thừa kế, tức là một class dẫn xuất có thể kế thừa từ nhiều class cơ sở

- Cú pháp: derived-class: access baseA, access baseB,…

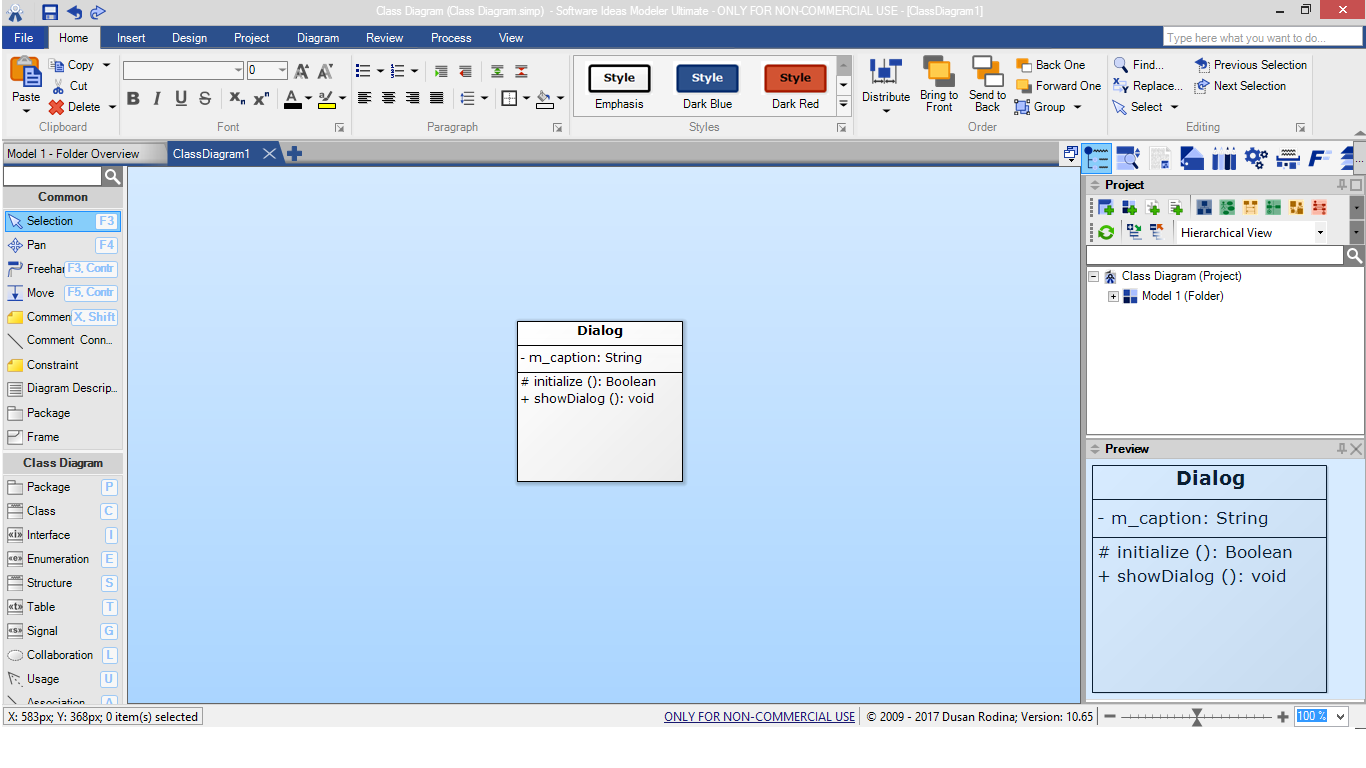
Trong đó: access là public, protected hoặc là private và phải khai báo cụ thể cho mỗi class cơ sở mà class dẫn xuất kế thừa.

Ví dụ:

Class A: public class B, private class C{

}

**3. Biểu diễn UML cho class Dialog**

****