Lý thuyết quan trọng về lập trình hướng đối tượng c++

1. Lớp và đối tượng

Lớp: là một khuôn mẫu để tạo ra đối tượng, Nó định nghĩa các thuộc tính và hành vi chung cho các đối tượng của nó

2. Thuộc tính là các dữ liệu mà một đối tượng có thể lưu trữ. Những thuộc tính này định nghĩa các đặc điểm hoặc trạng thái của đối tượng. Trong một lớp, bạn có thể định nghĩa các thuộc tính để mô tả các đặc điểm của đối tượng được tạo từ lớp đó.

3. private: public: protected: là các từ khóa sử dụng để xác định mức độ truy cập của các thành viên trong một lớp.

II. hàm không đối số, hàm đối số, hàm hủy

1. hàm đối số

Trong lập trình là một hàm mà bạn có thể truyền giá trị vào khi bạn gọi nó. Đối số là các giá trị hoặc biểu thức mà bạn truyền vào hàm khi gọi nó. Đối số là các giá trị hoặc biểu thức mà bạn truyền vào hàm khi gọi nó. Hàm đối số được sử dụng để chấp nhận dữ liệu từ bên ngoài và xử lý dữ liệu này bên trong hàm.

Cấu trúc :

Void function(dataType parameter1, dataType parameter 2,...){

// code của hàm }

Void là kiểu dữ liệu trả về

function: tên hàm

dataType parameter1,.... là danh sách các đối số mà hàm chấp nhận mỗi đối số có 1 kiểu dữ liệu riêng

void print(int a, int b){

int sum = a+b;

cout<<” sum: “<<sum<<endl;

công dụng:

tổ chức mã nguồn: hàm đối số giúp bạn tổ chức mã nguồn bằng cách chia nhỏ các tác vụ thành các hàm nhỏ, linh hoạt để xử lý các giá trị đầu vào khác nhau

Tái sử dụng mã: bạn có thể tái sử dụng hàm đối số cho các giá trị đầu vào khác nhau, mà không cần phải viết mã hàm mới

Chia nhỏ vấn đề: hàm đối số cho phép bạn chia nhỏ vấn đề lớn thành các vấn đề nhỏ hơn, giúp quản lý mã nguồn dễ dàng hơn và làm cho mã trở nên dễ đọc và hiểu hơn

2. Hàm không đối số

Hàm không đối số là một loại hàm trong lập trình không chấp nhận bất kỳ đối số nào khi được gọi. Điều này có nghĩa là hàm không nhận bất kỳ giá trị hoặc thông tin nào từ bên ngoài khi được gọi hàm. Hàm không đối số được định nghĩa một lần và có thể được gọ nhiều lần mà không cần truyền bất kỳ giá trị nào.

a, Thực hiện một hành động cụ thể

void print(){

cout<<” chao mừng đến với chúng tôi”<<endl;

b, tổng quát và linh hoạt

Hàm không đối số thường được sử dụng khi hành động không cần dữ liệu đầu vòa cụ thể chúng chỉ đơn giản thực hiện một công việc cụ thể

b, giảm sự lặp lạc trong mã nguồn

Khi một đoạn mã được sử dụng nhiều lần trong chương trình, bạn có thể đặt nó vào một hàm không đối số dể tránh việc lặp lại mã đó.

C, Giảm sự lặp lạc trong mã nguồn

Khi một đoạn mã được sử dụng nhiều lần trong chương trình, bạn có thể đặt nó vào một hàm không đối số dể traqnhs việc lặp lại mã đó

D, tạo giao diện người dùng

3. Hàm hủy

Là một hàm đặc biệt được sử dụng để giải phóng bất kỳ tài nguyên nào mà một đối tượng có thể đã cấp phát trong quá trình sống. Mỗi lớp trong c++ có thể có mọt hàm hủy và hàm này được tự động gọi khi một đối tượng của lớp đó bị hủy hoặc bị ra khỏi phạm vi của nó

Public:

~phanso(){}

A, giải phóng bộ nhớ động

Giải phóng bộ nhớ động mà một đối tượng có thể đaqx cấp phát trong quá trình sôgns. Điều này giúp tránh rò rỉ bộ nhớ

III. cách nhập và in dữ liệu

Cách nhập:

cin>>nam>>endl;

Cách in

Cout<<” Ho va ten là “<<nam<<endl;

Phanso(): mauso(0),tuso(1);

: trong trường hợp này là danh sách khởi tạo

Cout<<mauso<<”/”<<tuso<<endl;

Dấu / dùng để hiển thị dấu phân chia giữa 2 giá trị tử số và mẫu số