Bài 13: Class

# Khái niệm

Trong C++, từ khóa "class" được sử dụng để định nghĩa một lớp, là một cấu trúc dữ liệu tự định nghĩa có thể chứa dữ liệu và các hàm thành viên liên quan.

Cú pháp:

| **class** **ClassName** { **private**:  // Các thành phần riêng tư (private) chỉ có thể truy cập bên trong lớp  // Dữ liệu thành viên, hàm thành viên, ...  **protected**:  // Các thành phần bảo vệ (protected) tương tự như private, nhưng có thể truy cập từ lớp kế thừa  **public**:  // Các thành phần công khai (public) được truy cập từ bên ngoài lớp  // Dữ liệu thành viên, hàm thành viên, ...  // Hàm thành viên và các phương thức khác có thể được định nghĩa tại đây  // ...  }; |
| --- |

Ví dụ:

| **class** **HinhChuNhat** {  **public**:  **double** chieuDai;  **double** chieuRong; }; **int** **main**() {  HinhChuNhat hinh1;  hinh1.chieuDai = 10.0;  hinh1.chieuRong = 5.0;   std::cout << "Chieu dai: " << hinh1.chieuDai << '\n';    **return** 0; } |
| --- |

Trong class cho phép khai báo các function như member.

Ví dụ:

| **class** **HinhChuNhat** {  **public**:  **double** chieuDai;  **double** chieuRong;  // Hàm tính diện tích  **double** **tinhDienTich**() {  **return** chieuDai \* chieuRong;  }  **void** **display**(); };  **void** HinhChuNhat::display() {  std::cout << " Hello " << '\n'; } **int** **main**() {  HinhChuNhat hinh1;  hinh1.chieuDai = 10.0;  hinh1.chieuRong = 5.0;   hinh1.display();   std::cout << "Dien tich: " << hinh1.tinhDienTich() << '\n';    **return** 0; } |
| --- |

# Constructor và Destructor

Constructor trong C++ là một function sẽ được tự động gọi khi khởi tạo object. Constructor sẽ có tên trùng với tên của class.

Ví dụ:

| **class** **HinhChuNhat** {  **public**:  **double** chieuDai;  **double** chieuRong;   HinhChuNhat(){  chieuDai = 10;  chieuRong = 9;  }   // Hàm tính diện tích  **double** **tinhDienTich**() {  **return** chieuDai \* chieuRong;  }   }; **int** **main**() {  HinhChuNhat hinh1;    std::cout << "Chieu dai: " << hinh1.chieuDai << '\n';    **return** 0; } |
| --- |

Destructor trong C++ là một function sẽ được tự động gọi khi object được giải phóng. Destructor sẽ có tên trùng với tên của class và thêm ký tự ~ ở phía trước tên.

Ví dụ:

| **class** **HinhChuNhat** {  **public**:  **double** chieuDai;  **double** chieuRong;   HinhChuNhat(){  chieuDai = 10;  chieuRong = 9;  }  ~HinhChuNhat(){  std::cout << "Destructor " << '\n';  }    // Hàm tính diện tích  **double** **tinhDienTich**() {  **return** chieuDai \* chieuRong;  }   };  **int** **main**() {  HinhChuNhat hinh1;  std::cout << "Dien tich: " << hinh1.tinhDienTich() << '\n';    **return** 0; } |
| --- |

# Biến static trong class

Khi một biến trong class được khai báo với từ khóa static, thì tất cả các object sẽ dùng chung địa chỉ của biến này.

Ví dụ:

| **class** **HinhChuNhat** {  **public**:  **double** chieuDai;  **double** chieuRong;    **static** **int** var; };   **int** HinhChuNhat::var;  **int** **main**() {  HinhChuNhat hinh1;  HinhChuNhat hinh2;  HinhChuNhat hinh3;   cout << "address of chieu dai: " << &hinh1.chieuDai << '\n';   cout << "address of chieu dai: " << &hinh2.chieuDai << '\n';   cout << "address of chieu dai: " << &hinh3.chieuDai << '\n';    cout << "address of var: " << &hinh1.var << '\n';   cout << "address of var: " << &hinh2.var << '\n';   cout << "address of var: " << &hinh3.var << '\n';     **return** 0; } |
| --- |