Function Template, Class Template

Nguyễn Văn Khiết

Nội dung

- Function Template
- Class Template

Function Template

- Function Template (Hàm khuôn mẫu) là những hàm đặc biệt làm việc với các kiểu dữ liệu tổng quát.
- ➔ có thể định nghĩa những hàm mà chức năng của nó có thể áp dụng cho những kiểu dữ liệu khác nhau (mà không cần phải viết lại code).

Function Template

```
template <class T>
T Max (T a, T b) {
  T result;
  result = (a>b)? a : b;
  return (result);
}

int main() {
      int i=5, j=6;
      int k=Max<int>(i,j);
      cout << k << endl;

      double l=10.2, m=5.7;
      double n=Max<double>(l,m);
      cout << n << endl;

      n = Max(l,m);
      cout << n << endl;
}
```

Function Template

- Với ví dụ trên:
  - Thay vì viết các hàm Max(int, int), Max(double, double), chỉ cần viết 1 hàm Max cho kiểu dữ liệu generic T
  - Khi sử dụng gọi hàm Max<…> với kiểu dữ liệu tương ứng. VD: Max<int>(3,4)
  - C++ có thể tự detect được kiểu dữ liệu của các tham số truyền vào. VD: Max(3.2, 5.6);
  - Nếu hai tham số truyền vào hàm Max khác kiểu dữ liệu ➔ C++ không thể detect được ➔ báo lỗi

Function Template

- Có thể định nghĩa hàm template với nhiều kiểu generic cùng lúc
- **template <class T, class U>**

  …

Function Template

```
template <class T, class U>
T Min (T a, U b) {
  return (a<b?a:b);
}

{
     //int j;
     //double m, n;
     …
      n = Min(m, i);
     cout << n << endl;
}
```

➔ C++ sẽ tự detect được Min<double, int>

Function Template

- Function template
  - Cách viết hàm tổng quát cho nhiều kiểu dữ liệu tổng quát khác nhau
  - Có thể có nhiều kiểu dữ liệu tổng quát trong hàm template
  - Kiểu cụ thể sẽ được xác định trong lời gọi hàm

Bài tập luyện tập

- Bài 1: Viết hàm template để sắp xếp mảng có kiểu dữ liệu bất kỳ

Class Template

- Có thể viết các lớp khuôn mẫu, trong đó các thành viên (thuộc tính, phương thức) sử dụng các kiểu dữ liệu tổng quát

Class Template

```
template <class T>
class TwoNumber {
protected:
        T mNumberOne;
        T mNumberTwo;
public:
        TwoNumber(T, T);
        T getmax ();
};

template <class T>
TwoNumber<T>::TwoNumber(T one, T two)
{
        mNumberOne = one;
        mNumberTwo = two;
}

template <class T>
T TwoNumber<T>::getmax ()
{
        T retval;
        retval = mNumberOne>mNumberTwo? mNumberOne : mNumberTwo;
        return retval;
}
```

Class Template

```
#include “twonumber.h”
int main() {
    TwoNumber<int> myobject (10, 5);
    cout << myobject.getmax();
}
```

Class Template

- Phần cài đặt của class template phải nằm chung file với phần định nghĩa class template
- Đối với từng phương thức, phải bắt đầu bằng template <class T>

Bài tập luyện tập

- Bài 2: Định nghĩa lớp mảng các phần tử với kiểu dữ liệu tổng quát (không dùng các lớp thuộc bộ thư viện STL)
  - Cho phép khởi động mảng với kích thước cho trước
  - Lấy giá trị từng phần tử trong mảng
  - Gán giá trị cho từng phần tử của mảng