实验报告

东北林业大学

信息与计算机科学技术实验中心

|  |
| --- |
| 1. 实验目的   理解类模板和函数模板的意义及作用；  掌握类模板和函数模板的定义方法及在程序中的使用方法。 |
| 二、实验环境  Codeblocks |
| 三、实验内容及结果  1． 写一个函数模板，求数组中的最大元素，并且使得函数调用时，数组的类型和返回类型可以是整数也可以是双精度类型。  2． 写一个函数模板，使用冒泡排序将数组内容由小到大排列，并且使得函数调用时，数组的类型可以是整数也可以是双精度型。  3． 建立类模板input，在调用构造函数时，完成以下工作：  （1） 提示用户输入；  （2） 让用户输入数据；  （3） 如果数据不在预定范围内，重新提示输入。  input型的对象应当按以下形式定义：  input ob（"promput message”,min\_value,max\_value）  其中，promput message是提示输入信息。可接受的最小值和最大值分别由min\_value与max\_value指定。 |

|  |
| --- |
| 四、实验过程分析与讨论  1． 写一个函数模板，求数组中的最大元素，并且使得函数调用时，数组的类型和返回类型可以是整数也可以是双精度类型。  #include <iostream>  using namespace std;  // 函数模板，用于求数组中的最大元素  template <typename T>  T findMax(T arr[], int size) {  T max = arr[0];  for (int i = 1; i < size; ++i) {  if (arr[i] > max) {  max = arr[i];  }  }  return max;  }  int main() {  int intArr[] = {1, 5, 3, 9, 7};  double doubleArr[] = {1.1, 5.5, 3.3, 9.9, 7.7};  int intMax = findMax(intArr, 5);  double doubleMax = findMax(doubleArr, 5);  cout << "Max in int array: " << intMax << endl;  cout << "Max in double array: " << doubleMax << endl;  return 0;  }  2． 写一个函数模板，使用冒泡排序将数组内容由小到大排列，并且使得函数调用时，数组的类型可以是整数也可以是双精度型。  #include <iostream>  using namespace std;  // 函数模板，用于冒泡排序  template <typename T>  void bubbleSort(T arr[], int size) {  for (int i = 0; i < size - 1; ++i) {  for (int j = 0; j < size - 1 - i; ++j) {  if (arr[j] > arr[j + 1]) {  T temp = arr[j];  arr[j] = arr[j + 1];  arr[j + 1] = temp;  }  }  }  }  int main() {  int intArr[] = {5, 1, 4, 2, 8};  double doubleArr[] = {5.5, 1.1, 4.4, 2.2, 8.8};  bubbleSort(intArr, 5);  bubbleSort(doubleArr, 5);  cout << "Sorted int array: ";  for (int i = 0; i < 5; ++i) {  cout << intArr[i] << " ";  }  cout << endl;  cout << "Sorted double array: ";  for (int i = 0; i < 5; ++i) {  cout << doubleArr[i] << " ";  }  cout << endl;  return 0;  }  3． 建立类模板input，在调用构造函数时，完成以下工作：  （1） 提示用户输入；  （2） 让用户输入数据；  （3） 如果数据不在预定范围内，重新提示输入。  input型的对象应当按以下形式定义：  input ob（"promput message”,min\_value,max\_value）  其中，promput message是提示输入信息。可接受的最小值和最大值分别由min\_value与max\_value指定。  #include <iostream>  #include <string>  using namespace std;  template <typename T>  class Input {  private:  T value;  T minValue;  T maxValue;  string promptMessage;  public:  // 构造函数  Input(string message, T minVal, T maxVal) : promptMessage(message), minValue(minVal), maxValue(maxVal) {  getInput();  }  // 获取输入并进行范围检查  void getInput() {  while (true) {  cout << promptMessage;  cin >> value;  if (value >= minValue && value <= maxValue) {  break;  } else {  cout << "Value out of range. Please try again." << endl;  }  }  }  // 返回输入的值  T getValue() {  return value;  }  };  int main() {  Input<int> intInput("Enter an integer between 1 and 10: ", 1, 10);  cout << "You entered: " << intInput.getValue() << endl;  Input<double> doubleInput("Enter a double between 1.0 and 10.0: ", 1.0, 10.0);  cout << "You entered: " << doubleInput.getValue() << endl;  return 0;  } |
|  |
| 五、指导教师意见  指导教师签字：  年 月 日 |