# ЛАБОРАТОРНАЯ РАБОТА № 21

# РАЗРАБОТКА И ОТЛАДКА АЛГОРИТМОВ И ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ ШАБЛОНОВ ФУНКЦИЙ

***Цель работы*:**

Освоить на практике:

* создание шаблонов функций.

# КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

***Шаблоны функций***

В С++ для функции может быть описан шаблон. Это позволит снять достаточно жесткие ограничения, накладываемые механизмом формальных и фактических параметров при вызове функции.

Следует отметить, что шаблон функции – это на самом деле не функция, так как она не приводит к созданию программного кода в памяти. Это просто модель, трафарет для создания множества функций из одной.

# ПРИМЕРЫ

***Шаблон функции***

*Рассмотрим пример функции, вычисляющей сумму нескольких аргументов.*

#include <iostream>

#include <string>

using namespace std;

////////////////////////////////////////////////////

template <class T1,class T2>

T1 sm(T1 a,T2 b) // описание шаблона

{ return (T1)(a+b); // функции c 2 параметрами

}

template <class T1,class T2,class T3>

T1 sm(T1 a,T2 b,T3 c) // описание шаблона функции

{ return (T1)(a+b+c); // функции c 3 параметрами

}

////////////////////////////////////////////////////

int main(){

cout<<"вызов ф-ции суммирования sm(int,int) = "<<sm(4,6)<<endl;

cout<<"вызов ф-ции суммирования sm(int,int,int) = "<<sm(4,6,1)<<endl;

cout<<"вызов ф-ции суммирования sm(int,double) = "<<sm(5,3)<<endl;

cout<<"вызов ф-ции суммирования sm(double,int,short)= " <<

sm(.4,6,(short)1)<<endl;

// cout<<sm("я изучаю","язык С++")<<endl; **error** cannot add two pointers

return 0;

}

В программе описана перегруженная функция sm(), первый экземпляр которой имеет 2, а второй 3 параметра. При этом тип формальных параметров функции определяется при вызове функции типом ее фактических параметров. Используемые типы Т1, Т2, Т3 заданы как параметры для функции с помощью выражения template <class T1,class T2,class T3>. Это выражение предполагает использование типов Т1, Т2 и Т3 в виде ее дополнительных параметров.

Результат работы программы будет иметь вид:

**вызов функции суммирования sm(int,int) = 10**

**вызов функции суммирования sm(int,int,int) = 11**

**вызов функции суммирования sm(int,double) = 8**

**вызов функции суммирования sm(double,int,short)= 7.4**

В случае попытки передачи в функцию sm() двух строк, то есть типов, для которых не определена данная операция, компилятор выдаст ошибку. Чтобы избежать этого, можно ограничить использование шаблона функции sm(), описав явным образом функцию sm() для некоторых конкретных типов данных.

В нашем случае:

char \*sm(char \*a,char \*b) // явное описание функции объединения

{ char \*tmp=a; // двух строк

a=new char[strlen(a)+strlen(b)+1];

strcpy(a,tmp);

strcat(a,b);

return a;

}

Добавление в main() инструкции, например,

cout<<sm("я изучаю"," язык С++")<<endl;

приведет к выводу, кроме указанных выше, сообщения:

я изучаю язык С++

# ЗАДАНИЕ

Реализуйте задания первого и второго уровня своего варианта с использованием шаблона функции. Продемонстрируйте вызов шаблона для различных типов данных.

***Первый уровень сложности***

Составить программу, решающую указанную ниже задачу.

В одномерном массиве, состоящем из *n* (не более 10) вводимых с клавиатуры значений, вычислить заданное значение.

1. Произведение элементов массива, расположенных между максимальным и минимальным элементами.
2. Сумму элементов массива, расположенных между первым и последним нулевыми элементами.
3. Сумму элементов массива, расположенных до последнего положительного элемента.
4. Сумму элементов массива, расположенных между первым и последним положительными элементами.
5. Произведение элементов массива, расположенных между первым и вторым нулевыми элементами.
6. Сумму элементов массива, расположенных между первым и вторым отрицательными элементами.
7. Сумму элементов массива, расположенных до минимального элемента.
8. Сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.
9. Сумму элементов массива, расположенных после последнего элемента, равного нулю.
10. Сумму модулей элементов массива, расположенных после минимального по модулю элемента.
11. Сумму элементов массива, расположенных после минимального элемента.
12. Сумму элементов массива, расположенных после первого положительного элемента.
13. Сумму модулей элементов массива, расположенных после первого отрицательного элемента.
14. Сумму модулей элементов массива, расположенных после первого элемента, рав­ного нулю.
15. Сумму положительных элементов массива, расположенных до максимального элемента.

***Второй уровень сложности***

Написать программу по обработке двухмерного массива. Размеры массива *n*, *m* и значения элементов массива вводятся с клавиатуры.

1. Определить количество строк, не содержащих ни одного нулевого элемента.

2. Определить количество столбцов, не содержащих ни одного нулевого элемента.

3. Определить количество столбцов, содержащих хотя бы один нулевой элемент.

4. Определить произведение элементов в тех строках, которые не содержат отрицательных элементов.

5. Определить сумму элементов в тех столбцах, которые не содержат отрицательных элементов.

6. Определить сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

7. Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

8. Найти сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент.

9. Найти сумму модулей элементов, расположенных ниже главной диагонали.

10. Найти сумму модулей элементов, расположенных выше главной диагонали.

11. Найти количество строк, среднее арифметическое элементов которых меньше введенной с клавиатуры величины.

12. Найти номер первой из строк, содержащих хотя бы один положительный элемент.

13. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент.

14. Найти номер первого из столбцов, не содержащих ни одного отрицательного элемента.

15. Найти номер первой из строк, не содержащих ни одного положительного элемента.

# КОНТРОЛЬНЫЕ ВОПРОСЫ.

1. Дайте определение понятию шаблона.
2. Поясните, каким образом связаны между собой шаблоны и наследование.
3. Приведите синтаксис описания шаблона функции.