Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №7**

За 4 семестр

По дисциплине: “ООТПиСП”

Тема: “Шаблоны функций и классов”

**Вариант №9**

**Выполнил**:

студент 2 курса

группы ПО-7 (перв.)

Крупенков М.Д.

**Проверил:**

Хацкевич М.В.

Брест 2022

**Цель.** Получить практические навыки создания шаблонов и использования их в программах С++.

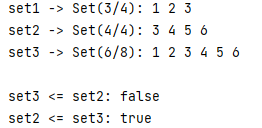
**Основное содержание работы.** Создать шаблон заданного класса и использовать его для данных различных типов.

**Порядок выполнения работы:**

Класс − множество set. Дополнительно перегрузить следующие операции:

* () − конструктор множества (в стиле конструктора для множественного типа в языке Pascal);
* + − объединение множеств;
* <= − сравнение множеств.

Результаты программы



Код программы

**main.cpp**

#include **"Set.h"**#include **"iostream"  
  
int** main() {  
 std::cout << **"set1 -> "**;  
 Set<**int**> set1;  
 set1.append(1);  
 set1.append(2);  
 set1.append(3);  
 set1.print();  
  
 std::cout << **"set2 -> "**;  
 Set<**int**> set2(4, 3, 4, 5, 6);  
 set2.print();  
  
 std::cout << **"set3 -> "**;  
 **auto** set3 = set1 + set2;  
 set3.print();  
  
 std::cout << **"\nset3 <= set2: "** << std::boolalpha << (set3 <= set2);  
 std::cout << **"\nset2 <= set3: "** << std::boolalpha << (set2 <= set3);  
}

**Set.h**

#ifndef **SET\_H**#define **SET\_H**#include **<iostream>**#include **<stdarg.h>  
  
  
template**<**class** T>  
**class** Set {  
 **unsigned int** count;  
 **unsigned int** countReal;  
 T \*array;  
  
**public**:  
 Set() : count(0), countReal(0), array(**nullptr**) {}  
  
 Set(**unsigned int** n, ...) : count(0), countReal(n), array(**new** T[countReal]) {  
 va\_list factor;  
 **va\_start**(factor, n);  
 T arg;  
 **for** (**int** i = 0; i < n; i++) {  
 arg = **va\_arg**(factor, T);  
 **if** (!in(arg)) {  
 array[i] = arg;  
 count++;  
 }  
 }  
 **va\_end**(factor);  
 }  
  
 Set<T>(**const** Set<T> &other) : count(other.count), countReal(other.countReal) {  
 array = **new** T[countReal];  
 **for** (**int** i = 0; i < count; ++i) {  
 array[i] = other.array[i];  
 }  
 }  
  
 **bool** in(**const** T t) **const** {  
 **for** (**int** i = 0; i < count; ++i)  
 **if** (array[i] == t) **return true**;  
 **return false**;  
 }  
  
 **void** append(**const** T t) {  
 **if** (!count) {  
 array = **new** T(t);  
 count = countReal = 1;  
 **return**;  
 }  
 **if** (in(t)) **return**;  
 **if** (count != countReal) {  
 array[count++] = t;  
 **return**;  
 }  
 countReal = count \* 2;  
 **auto** newSet = **new** T[countReal];  
 **for** (**int** i = 0; i < count; ++i) {  
 newSet[i] = array[i];  
 }  
 newSet[count++] = t;  
 **delete**[] array;  
 array = newSet;  
 }  
  
 **void** print() **const** {  
 std::cout << **"Set("** << count << **"/"** << countReal << **"):"**;  
 **for** (**int** i = 0; i < count; ++i) {  
 std::cout << **" "** << array[i];  
 }  
 std::cout << std::endl;  
 }  
  
 Set<T> &**operator**+=(**const** Set<T> &other) {  
 T t;  
 **for** (**int** i = 0; i < other.count; ++i) {  
 t = other.array[i];  
 **if** (!in(t)) append(t);  
 }  
 **return** \***this**;  
 }  
  
 Set<T> &**operator**+(**const** Set<T> &other) **const** {  
 **auto** set = **new** Set<T>(\***this**);  
 \*set += other;  
 **return** \*set;  
 }  
  
 **bool operator**<=(**const** Set<T> &other) **const** {  
 **for** (**int** i = 0; i < count; ++i) {  
 **if** (!other.in(array[i])) **return false**;  
 }  
 **return true**;  
 }  
};  
  
#endif *//SET\_H*

**Вывод:** Было получено преисполнение при работе с шаблонными классами в С++.