Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №4

по курсу «Объектно-ориентированное программирование»

на тему «Изучение шаблонов в языке C++»

Выполнили студенты группы 19ВВ3:

Ланцов Андрей

Трубаненко Александр

Приняли:

Евсеева Ю.И.

**2021**

**Цель работы:** Изучить шаблоны в языке C++

**Задания:**

4. Класс − множество set. Дополнительно перегрузить следующие

операции:

+ − добавить элемент в множество (типа set+item);

+ − объединение множеств;

\* − пересечение множеств;

**Ход работы:**

Листинг:

#include <iostream>

#include <vector>

#include <memory>

template <class T>

class Set {

std::vector<T> set;

int level;

public:

Set() {

level = 0;

}

Set(Set<T>& left) {

\*this = left;

}

Set(T\* \_set, int arr\_size) {

this->level = 0;

for (int i = 0; i < arr\_size; ++i) {

if (!has(\_set[i])) {

\*this += \_set[i];

this->level++;

}

}

}

~Set() {

set.clear();

}

public:

bool has(T& item) {

int current\_level = set.size();

for (int j = 0; j < current\_level; ++j) {//check value is already in set

if (set[j] == item) {

return 1;

}

}

return 0;

}

void operator+=(T& item) {

if (!has(item)) {

set.push\_back(item);

}

}

Set<T>\* operator+(Set<T>& left) {

Set<T>\* new\_set = new Set<T>;

\*new\_set = Set(\*this);

new\_set->level = this->level;//copy right one into new set

int left\_level = left.level;

for (int i = 0; i < left\_level; ++i) {//unite with left one

if (!this->has(left.set[i])) {

\*new\_set += left.set[i];

new\_set->level++;

}

}

return new\_set;

}

Set<T>\* operator\*(Set<T>& left) {

Set<T>\* new\_set = new Set<T>;

int left\_level = left.level;

for (int i = 0; i < left\_level; ++i) {

if (this->has(left.set[i])) {

\*new\_set += left.set[i];

new\_set->level++;

}

}

return new\_set;

}

void print() {

for (size\_t i = 0; i < this->level; i++){

std::cout << set[i]<< ',';

}

std::cout << '\n';

}

};

int main() {

int arr1[] = { 1,2,3 };

int arr2[] = { 1,3,3 };

int arr3[] = { 2,2,4,5 };

std::string strs1[] = { "foo","bar","baz" };

std::string strs2[] = { "foo","foo","baz" };

std::string strs3[] = { "foobar","bar","baz","foo","foo"};

Set<int> set\_int1(arr1, 3);

Set<int> set\_int2(arr2, 3);

Set<int> set\_int3(arr3, 4);

Set<int> int\_sets[] = { set\_int1,set\_int2,set\_int3 };

int int\_sets\_count = sizeof(int\_sets) / sizeof(int\_sets[0]);

Set<std::string> set\_str1(strs1, 3);

Set<std::string> set\_str2(strs2, 3);

Set<std::string> set\_str3(strs3, 5);

Set<std::string> str\_sets[] = { set\_str1,set\_str2,set\_str3 };

int str\_sets\_count = sizeof(str\_sets) / sizeof(str\_sets[0]);

Set<int>\* new\_int\_set1 = set\_int2 + set\_int3;

Set<int>\* new\_int\_set2 = set\_int1 \* set\_int3;

Set<int>\* new\_int\_set3 = set\_int2 \* set\_int3;

Set<std::string>\* new\_str\_set1 = set\_str1 + set\_str3;

Set<std::string>\* new\_str\_set2 = set\_str2 \* set\_str3;

std::cout << "Int sets:\n";

for (size\_t i = 0; i < int\_sets\_count; i++) {

std::cout << "Set " << i+1 << ":\n";

int\_sets[i].print();

}

std::cout << "\nset2 + set3:\n";

new\_int\_set1->print();

std::cout << "set1 \* set3:\n";

new\_int\_set2->print();

std::cout << "set2 \* set3:\n";

new\_int\_set3->print();

std::cout << "String sets:\n";

for (size\_t i = 0; i < str\_sets\_count; i++){

std::cout << "Set " << i+1 << ":\n";

str\_sets[i].print();

}

std::cout << "\nset1 + set3:\n";

new\_str\_set1->print();

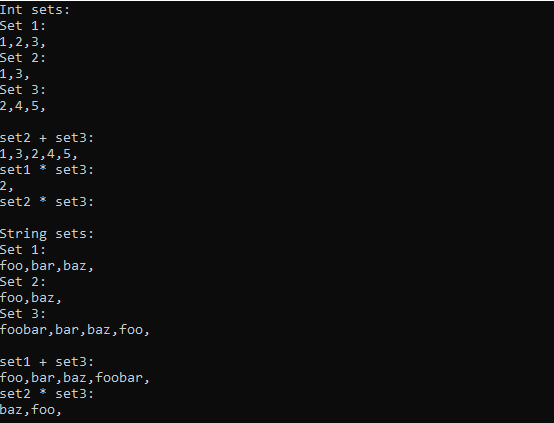
std::cout << "set2 \* set3:\n";

new\_str\_set2->print();

return 0;

}

**Результат работы программы**

****

**Вывод**

В результате выполнения работы были изучена работа шаблонов в языке C++.