



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий
Кафедра практической и прикладной информатики

ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ

По дисциплине

Теория вычислительных процессов

Работа выполнена	« » _____ 2024г.	Воеводский А.В.
Зачтено	« » _____ 2024г.	Павлова Е.С.

Москва, 2024г.

Практическая работа №1

Цель работы

Использование структуры данных: «множество». Вставка и удаление элементов. Слияние множеств.

Задание на практическую работу

Используя MS Visual Studio (или любую IDE под C++) реализовать слияние, объединение и пересечение двух множеств.

Ход работы

Реализуем пустое множество. Создание пустого множества представлено в листинге 1.

Листинг 1 – main.cpp

```
#include <iostream>
#include <set>

int main()
{
    std::set<int> numbers;    // пустое множество чисел int
}
```

Листинг 2 – main1.cpp

```
#include <algorithm>
#include <iostream>
#include <string>
#include <set>
#include <vector>
using namespace std;

void unification(const set<int>& a, set<int> b) {
    for (auto number : a) {
        if (!b.contains(number)) {
            b.insert(number);
        }
    }
    cout << "Unification: " << endl;
    for (auto number : b) {
        cout << number << " ";
    }
    cout << endl;
}
```

```

void merge(set<int> a, set<int> b) {
    multiset<int> temp;
    temp.insert(a.begin(), a.end());
    temp.insert(b.begin(), b.end());
    cout << "Merge: " << endl;
    for (auto number : temp) {
        cout << number << " ";
    }
    cout << endl;
}

void intersection(set<int> a, set<int> b) {
    set<int> temp;
    for (auto number : a) {
        if (b.contains(number)) {
            temp.insert(number);
        }
    }
    cout << "Intersection: " << endl;
    for (auto number : temp) {
        cout << number << " ";
    }
    cout << endl;
}

int main() {
    set<int> numbers2{1, 2, 3, 4, 5};
    std::cout << "Insert elements of the set\nInsert ends
with 0" << std::endl;
    set<int> numbers;
    int a;
    cin >> a;
    while (a!=0) {
        numbers.insert(a);
        cin >> a;
    }
    cout<< "Your numbers: ";
    for (auto number : numbers) {
        cout << number << " ";
    }
    cout << endl;
    cout << "Operations with set: 1 2 3 4 5" << endl;
    unification(numbers, numbers2);
    merge(numbers, numbers2);
    intersection(numbers, numbers2);
    return 0;
}

```

Далее протестируем программную реализацию.

На рис. 1 представлено ручное заполнение множества.

```
Insert elements of the set
Insert ends with 0
9 7 1 2 5
0
Your numbers: 1 2 5 7 9
```

Рисунок 1 – Ручное заполнение множества

На рис. 2 представлен результат слияния, объединения и пересечения введенного множества с множеством 1 2 3 4 5.

```
Operations with set: 1 2 3 4 5
Unification:
1 2 3 4 5 7 9
Merge:
1 1 2 2 3 4 5 5 7 9
Intersection:
1 2 5|
```

Рисунок 1 – Результат выполнения операций слияния, объединения и пересечения

Выводы

В данной работе произошло ознакомление со структурой данных множество. Реализованы 3 метода: слияние, объединение и пересечение; методы протестированы.