

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №7.
"Контейнерные классы"

Выполнил:
Ст. 2 курса гр. АС-53
Демидович А. Г.
Проверила:
Давидюк Ю. И.

Брест, 2020

1. **Цель.** Получить практические навыки работы с контейнерными классами STL.

2. Постановка задачи (Вариант 8)

1. Новая последовательность должна содержать все элементы исходной, модуль которых находится в заданном промежутке $[a; b]$.
2. Произведение элементов последовательности, расположенных между первым и вторым нулевыми элементами.

3. Код программы

Используются STL const-итераторы.

Для 2 задания использовался исходный вектор.

```
#include <iostream>
#include <math.h>
#include <vector>

int main() {
    std::vector<double> values;
    std::vector<double> results;
    double a = 0, b = 5;
    double mult = 1;
    int countNull = 0;
    std::vector<double>::const_iterator iter, firstNull, secondNull;
    for (int i = 0; i < 10; i++)
        if (i % 5 == 0)
            values.push_back(0);
        else
            values.push_back(-10 + rand() % 20);

    for (double element : values) // Добавляем в список элементы входящие в [a, b]
        if (fabs(element) > a && fabs(element) < b)
            results.push_back(element);

    std::cout << "Source vector:" << std::endl;
    for (int i = 0; i < values.size(); i++)
        std::cout << values[i] << " ";
    std::cout << std::endl;

    std::cout << "Task 1:" << std::endl;
    for (int i = 0; i < results.size(); i++)
        std::cout << results[i] << " ";

    firstNull = values.begin();
    secondNull = values.begin();
    for (iter = values.begin(); iter != values.end(); iter++) { // Ищем первое и второе
нулевые значения
        if (*iter == 0) {
            switch (countNull) // Какой нулевой элемент ищем
            {
                case 0:
                    firstNull = iter;
                    countNull++;
                    break;
                case 1:
                    secondNull = iter;
                    countNull++;
            }
        }
    }
}
```

```

        break;
    default:
        break;
    }
    if (countNull >= 2)
        break;
}

for (iter = ++firstNull; iter < secondNull; iter++) {
    mult *= *iter;
}

std::cout << "\nTask 2. Multiply: " << mult << std::endl;
return 0;
}

```

4. Результат выполнения программы

```

Source vector:
0; -9; -3; 4; -10; 0; -1; -6; 8; 8;
Task 1:
0; -3; 4; 0; -1;
Task 2. Multiply: -1080

```

5. Вывод:

Получил практические навыки работы с контейнерными классами C++.