|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт информационных технологий |
| Кафедра вычислительной техники |
|  |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1** | |
| **по дисциплине** | |
| *«Структуры и алгоритмы обработки данных»* | |
| Выполнил студент группы *ИКБО-02-17* | *Политов А.Ю.* |
| Принял | *Асадова Ю.С* |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_201\_\_ г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *Подпись студента* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г.. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *Подпись руководителя* |

Москва 2019

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

Институт информационных технологий

Кафедра вычислительной техники

Выполнено \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.Ю. Политов/

Зачтено \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ю.С. Асадова/

**Задание на лабораторную работу**

**Дисциплина**: **«**Структуры и алгоритмы обработки данных**»**

**Студент** Политов А.Ю **Шифр** 17И0325 **Группа** ИКБО-02-17

**1. Тема**: «Связные списки»

**2. Срок сдачи студентом законченной работы:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_г.

**3. Исходные данные:** Два списка с измерениями.

**4. Задание:** Найти приближённое значение сопротивления методом наименьших квадратов.

**5. Содержание отчета:**

* титульный лист;
* задание;
* оглавление;
* введение;
* основные разделы отчета;
* заключение;
* список использованных источников;

Руководитель работы Ю.С. Асадова \_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2018г.

подпись

Задание принял к исполнению А.Ю. Политов \_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2018г.

подпись

*Оглавление*

Введение…………………………………………………………………………… 4

1. Теоретическое введение …………………………………………………… 5
2. Постановка задачи………………………………………………………….. 5
3. Определение входных и выходных данных………………………………. 6
4. Краткое описание программы …………………………………………….. 6
5. Разработка программы (код) ……………………………………………… 7
6. Разработка тестов …………………………………………………………. 8

Заключение ……………………………………………………………………….. 9

Список литературы ……………………………………………………………….10

*Введение*

Целью данной лабораторной работы является ознакомление со структурой данных – связный список, приобретение умений реализации типовых операций со связным списком, приобретение навыков использования списков при решении задач, требующих оптимизации ресурсов времени и памяти. Лабораторная работа была выполнена в среде разработки MS Visual Studio на языке С++.

1. ***Теоретическое введение.***

Часто в серьезных программах надо использовать данные, размер и структура которых должны меняться в процессе работы. Динамические массивы здесь не выручают, поскольку заранее нельзя сказать, сколько памяти надо выделить – это выясняется только в процессе работы. Например, надо проанализировать текст и определить, какие слова и в каком количество в нем встречаются, причем эти слова нужно расставить по алфавиту. В таких случаях применяют данные особой структуры, которые представляют собой отдельные элементы, связанные с помощью ссылок – связные списки.

Связный список — базовая динамическая структура данных в информатике, состоящая из узлов, каждый из которых содержит как собственно данные, так и одну или две ссылки на следующий и/или предыдущий узел списка.

***Виды списков:***

Односвязный список. Простейшая реализация списка. В узлах хранятся данные и указатель на следующий элемент в списке.

Двусвязный список. Также хранится указатель на предыдущий элемент списка, благодаря чему становится проще удалять и переставлять элементы.

XOR-связный список. В некоторых случаях использование двусвязного списка в явном виде является нецелесообразным. В целях экономии памяти можно хранить только результат выполнения операции Xor над адресами предыдущего и следующего элементов списка. Таким образом, зная адрес предыдущего элемента, мы можем вычислить адрес следующего элемента.

Циклический список. Первый элемент является следующим для последнего элемента списка.

***Операция над списками***: вставка, удаление и поиск.

1. ***Постановка задачи.***

Задано два списка, которые содержат результаты N-измерений тока и напряжения на неизвестном сопротивлении. Найти приближённое значение сопротивления методом наименьших квадратов.

**Дано:** два списка с измерениями;

**Ограничения на данные:** данные должны быть представлены в численном виде.

**Ограничения на решаемую задачу**: нет

1. ***Определение входных и выходных данных.***

**Входные данные** – числовые значения, вводимые с клавиатуры.

**Результат** – число.

1. ***Краткое описание программы.***

На вход программы подаются два списка со значениями тока и напряжения. Далее программа создаёт дополнительные списки для хранения временных данных. После списки заполняются данными согласно формулам. После все списки объединятся с помощью главной формулы и вычисляется конечное значение.

1. ***Разработка программы (код).***

**#include <iostream>**

**#include <list>**

**#include "Windows.h"**

**using namespace std;**

**double sum(list<double> list) {**

**double sum = 0;**

**for (double item : list) {**

**sum += item;**

**}**

**return sum;**

**}**

**void add(list<double> &voltage, list<double> &current, list<double> &square\_voltage, list<double> &square\_current, list<double> &voltage\_current) {**

**double value;**

**cout << "Введите ток и напряжение: ";**

**while (!(cin >> value)) {**

**cout << "error\n";**

**cin.clear();**

**fflush(stdin);**

**}**

**voltage.push\_front(value);**

**while (!(cin >> value)) {**

**cout << "error\n";**

**cin.clear();**

**fflush(stdin);**

**}**

**current.push\_front(value);**

**cout << "значения успешно введены" << endl;**

**list<double>::iterator it\_vol = voltage.begin();**

**list<double>::iterator it\_cur = current.begin();**

**square\_voltage.push\_front(\*it\_vol \* \*it\_vol);**

**square\_current.push\_front(\*it\_cur \* \*it\_cur);**

**voltage\_current.push\_front(\*it\_cur \* \*it\_vol);**

**}**

**void show(list<double> &voltage, list<double> &current) {**

**cout << "+------+------+------+------+------+" << endl;**

**cout << "| V | I | V^2 | I^2 | V\*I |" << endl;**

**cout << "+------+------+------+------+------+" << endl;**

**for (size\_t i = 0; i < voltage.size(); i++) {**

**list<double>::iterator it\_vol = voltage.begin();**

**list<double>::iterator it\_cur = current.begin();**

**advance(it\_vol, i);**

**advance(it\_cur, i);**

**cout << "|";**

**cout.precision(5);**

**cout.fill(' ');**

**cout.width(6);**

**cout << \*it\_vol << "|";**

**cout.precision(5);**

**cout.fill(' ');**

**cout.width(6);**

**cout << \*it\_cur << "|";**

**cout.precision(5);**

**cout.fill(' ');**

**cout.width(6);**

**cout << \*it\_vol \* \*it\_vol << "|";**

**cout.precision(5);**

**cout.fill(' ');**

**cout.width(6);**

**cout << \*it\_cur \* \*it\_cur << "|";**

**cout.precision(5);**

**cout.fill(' ');**

**cout.width(6);**

**cout << \*it\_cur \* \*it\_vol << "|" << endl;**

**}**

**cout << "+------+------+------+------+------+" << endl;**

**}**

**int main() {**

**SetConsoleCP(1251);**

**SetConsoleOutputCP(1251);**

**list<double> voltage;**

**list<double> current;**

**list<double> square\_voltage;**

**list<double> square\_current;**

**list<double> voltage\_current;**

**// пользовательские значения**

**cout << "Введите количество измерений: ";**

**int count;**

**while (!(cin >> count)) {**

**cout << "error\n";**

**cin.clear();**

**fflush(stdin);**

**}**

**for (int i = 0; i < count; i++) {**

**add(voltage, current, square\_voltage, square\_current, voltage\_current);**

**}**

**show(voltage, current);**

**cout << endl << "Некоторые значения сумм" << endl;**

**double voltage\_sum = sum(voltage);**

**cout << "V: " << voltage\_sum << endl;**

**double current\_sum = sum(current);**

**cout << "I: " << current\_sum << endl;**

**double square\_voltage\_sum = sum(square\_voltage);**

**cout << "V^2: " << square\_voltage\_sum << endl;**

**double square\_current\_sum = sum(square\_current);**

**cout << "I^2: " << square\_current\_sum << endl;**

**double voltage\_current\_sum = sum(voltage\_current);**

**cout << "V\*I: " << voltage\_current\_sum << endl;**

**cout << "+------+------+------+------+------+" << endl;**

**// Супер-пупер-формула подсчета сопротивления**

**double R = (((int)voltage.size() \* voltage\_current\_sum) - current\_sum \* voltage\_sum)**

**/ ((int)voltage.size() \* square\_voltage\_sum - voltage\_sum \* voltage\_sum);**

**cout << endl << "Приближённое значение сопротивления: " << R << "(Ом)"<< endl;**

**system("pause");**

**}**

1. ***Разработка тестов***

В таблице 2 представленный сведения о тестировании программы.

Таблица 2 – Тестирование программы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  теста | Входные данные | Эталон результата | Результат программы | Результат анализа  теста |
|  |  |  |  |
| 1 | 123 34  34 42  0 0  123 123  123 12 | 0.37041 | 0.37041 | + |

На рисунке 1 показана работа программы для данных, указанных в таблице 2.

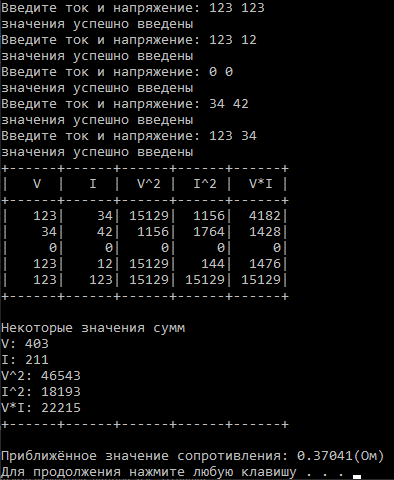


Рисунок 1 - Результат выполнения теста 1

***Заключение***

В ходе выполнения работы были получены практические навыки при работе структурой данных – связный список, приобретены умения реализации типовых операций со связным списком, навыки использования списков при решении задач, требующих оптимизации ресурсов времени и памяти на языке С++.

***Список литературы***

1. «qaru.site» — информационный сервис для разработчиков [Электронный ресурс]. URL: <http://qaru.site> (дата обращения: 24.02.2019)
2. «cppstudio» — информационный сервис для разработчиков [Электронный ресурс]. URL: <http://cppstudio.com> (дата обращения: 25.02.2019)
3. [Кей Хорстманн, Гари Корнелл «С++. Библиотека профессионала. Том 1».10-е издание (2016](https://t.me/progbook/168)) (дата обращения: 25.02.2019)