Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

Факультет среднего профессионального образования

**Отчет о**

**лабораторной работе № 3**

по теме: Типы данных, определяемые пользователями

по дисциплине: Разработка программных модулей

Специальность:

09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| Проверил:  Антонов М.Б. \_\_\_\_\_\_\_\_\_\_  Дата: «2» ноября 2019 г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2333  \_\_\_\_\_\_ Максимов Д.О. |

Санкт-Петербург 2019

1. ПОСТАНОВКА ЗАДАЧИ

Для усвоения полученных теоретических знаний было предложено задание на работу с файлами:

Дан файл имеющий следующий вид (Таблица 1):

Таблица 1 – Исходный файл

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № операции | Операция | Тип данных (int/float/string) | Первый оператор | Второй оператор | Ожидаемый результат |
| 1 | + | int | 26 | 2 | int |
| 2 | - | float | 3.56 | 31.93 | float |
| 3 | \* | int | 23 | 4 | int |
| 4 | / | float | 3.81 | 4.94 | float |
| 5 | pow | int | 9 | 2 | int |

Требуется считать все операции из файла и вывести их результат в отдельный файл (Таблица 2):

Таблица 2 – Новый файл

|  |  |  |
| --- | --- | --- |
| № операции | Результат | Тип |
| 1 | 28 | int |
| 2 | -28.37 | float |

Так как вывод в консоль не предусмотрен, в качестве результатов выполнения предоставлять: выходной файл.

1. Код программы

#include <iostream>

#include <fstream>

#include <sstream>

#include <math.h>

using namespace std;

int main() //начало функции main

{

ifstream dataFile;

ofstream resFile;

dataFile.open("data.txt");

if (!dataFile.is\_open())

{

return 23;

}

resFile.open("res.txt", ios::out | ios::trunc);

if (!resFile.is\_open())

{

return 45;

}

string number;

string operation;

string type\_operands;

string operand1;

string operand2;

string type\_res;

string prev\_number = "qk";

{

dataFile >> number;

if (number == prev\_number)

{

break;

}

dataFile >> operation;

dataFile >> operand1;

dataFile >> operand2;

dataFile >> type\_res;

resFile << number << ' ';

int i;

if (operation == "+") i = 1

if (operation == "-") i = 2;

if (operation == "\*") i = 3;

if (operation == "/") i = 4;

if (operation == "pow") i = 5;

switch (i)

{

case 1:

{

if (type\_operands == "int")

{

stringstream streamOp1(operand1);

int x = 0;

streamOp1 >> x;

stringstream streamOp2(operand2);

int y = 0;

streamOp2 >> y;

resFile << x+y << ' ';

}

else

{

if (type\_operands == "float")

{

stringstream streamOp1(operand1);

float x = 0;

streamOp1 >> x;

stringstream streamOp2(operand2);

float y = 0;

streamOp2 >> y;

if (type\_res == "int")

{

float res\_float = x+y;

int res\_int = static\_cast<int>(res\_float+0.5);

resFile << res\_int << ' ';

}

else

{

resFile << x+y << ' ';

}

}

else

{

resFile << operand1+operand2 << ' '

}

}

break;

}

case 2:

{

if (type\_operands == "int")

{

stringstream streamOp1(operand1);

int x = 0;

streamOp1 >> x;

stringstream streamOp2(operand2);

int y = 0;

streamOp2 >> y;

resFile << x-y << ' ';

}

else

{

stringstream streamOp1(operand1);

float x = 0;

streamOp1 >> x;

stringstream streamOp2(operand2);

float y = 0;

streamOp2 >> y;

if (type\_res == "int")

{

float res\_float = x-y;

int res\_int = static\_cast<int>(res\_float+0.5);

resFile << res\_int << ' ';

}

else

{

resFile << x-y << ' ';

}

}

break;

}

case 3:

{

if (type\_operands == "int")

{

stringstream streamOp1(operand1);

int x = 0;

streamOp1 >> x;

stringstream streamOp2(operand2);

int y = 0;

streamOp2 >> y;

resFile << x\*y << ' ';

}

else

{

stringstream streamOp1(operand1);

float x = 0;

streamOp1 >> x;

stringstream streamOp2(operand2);

float y = 0;

streamOp2 >> y

if (type\_res == "int")

{

float res\_float = x\*y;

int res\_int = static\_cast<int>(res\_float+0.5);

resFile << res\_int << ' ';

}

else

{

resFile << x\*y << ' ';

}

}

break;

}

case 4:

{

if (!(operand2 == "0"))

{

if (type\_operands == "int")

{

stringstream streamOp1(operand1);

int x = 0;

streamOp1 >> x;

stringstream streamOp2(operand2

int y = 0;

streamOp2 >> y;

resFile << x/y << ' ';

}

else

{

stringstream streamOp1(operand1);

float x = 0;

streamOp1 >> x;

stringstream streamOp2(operand2);

float y = 0;

streamOp2 >> y;

if (type\_res == "int")

{

float res\_float = x/y;

int res\_int = static\_cast<int>(res\_float+0.5);

resFile << res\_int << ' ';

}

else

{

resFile << x/y << ' ';

}

}

}

else

{

resFile << "NaN" << ' ';

}

break;

}

case 5:

{

if (type\_operands == "int")

{

stringstream streamOp1(operand1);

int x = 0;

streamOp1 >> x;

stringstream streamOp2(operand2);

int y = 0;

streamOp2 >> y;

resFile << pow(x, y) << ' ';

}

else

{

stringstream streamOp1(operand1);

float x = 0;

streamOp1 >> x;

stringstream streamOp2(operand2);

float y = 0;

streamOp2 >> y;

if (type\_res == "int")

{

float res\_float = pow(x, y);

int res\_int = static\_cast<int>(res\_float+0.5);

resFile << res\_int << ' ';

}

else

{

resFile << pow(x, y) << ' ';

}

}

break;

}

default:

{

break;

}

}

resFile << type\_res << endl;

prev\_number = number;

}

resFile.close();

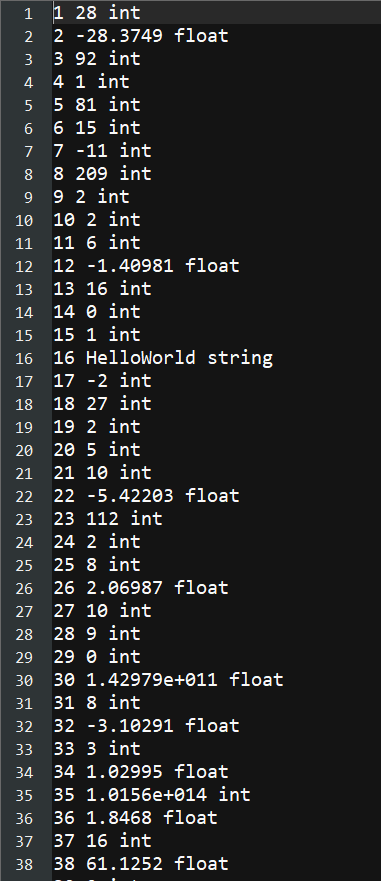
dataFile.close();

return 0;

}

1. протокол программы

Ниже представлен результирующий файл.



1. Выводы

В ходе выполнения практической работы были получены навыки работы с файлами, а именно: проверка открытия файла, чтение из файла, запись в файл, закрытие файла. Соответственно с поставленными задачами все требования были выдержаны, а именно: был считан файл, содержащий в себе номера операций, операторы, действия над ними и их типы и типы результатов операций, были записаны в новый файл номера операций, результаты и их типы.