МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное   
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра технической кибернетики

ЛАБОРАТОРНАЯ РАБОТА № 1

**Знакомство с IDE MS Visual Studio   
и программа “Hello World!”**

по курсу   
Основы программирования

Группа 6101

Студент А. И. Афанасенко

(*подпись*)

Преподаватель,

к.ф.-м.н. Е.С. Козлова

(*подпись*)

Самара 2023

Задание

Задание на лабораторную работу приведено в таблице 1.

**Таблица 1 –** Задание на ЛР № 1

|  |  |
| --- | --- |
| Основное задание на ЛР | 1. Реализовать программу “Hello World!” (программа, выводящая на экран слова “Hello World!” и ожидающая ввода с клавиатуры для завершения работы) в виде консольного приложения в среде разработки MS Visual Studio. Внести различные ошибки в программу и изучить реакцию компилятора (среды разработки).  2. Реализовать одно консольное приложение с функциональностью согласно вариантам. |

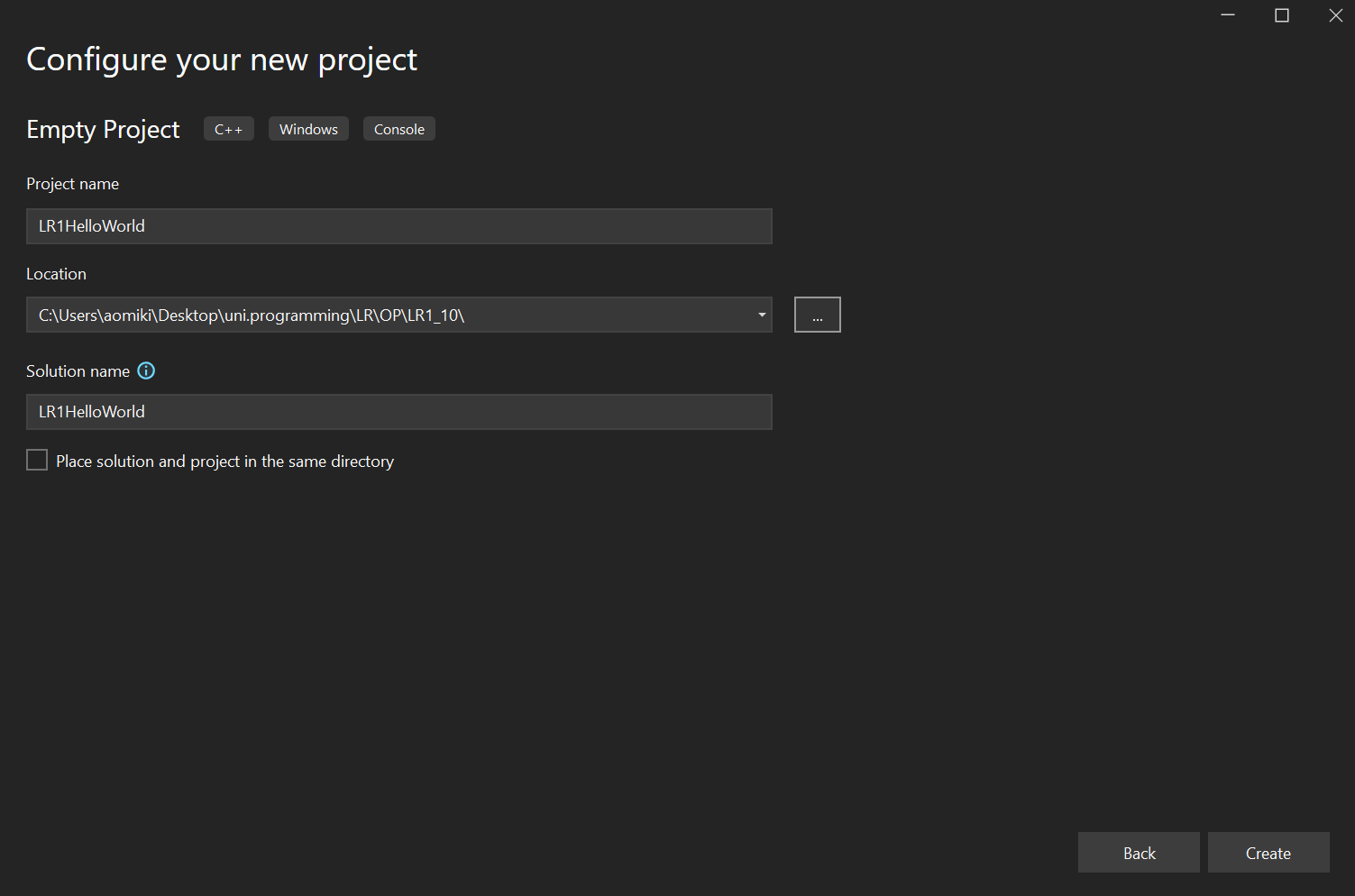
Вариант для второго пункта задания приведен в таблице 2.

**Таблица 2 –** Вариант второго пункта основного задания на ЛР № 1

|  |  |
| --- | --- |
| Вариант 10 | Подсчет элементов последовательности чисел. Входные данные: последовательность целых чисел. Выходные данные: количество отрицательных и положительных элементов. Особенности: ввод последовательности заканчивается при вводе 0. |

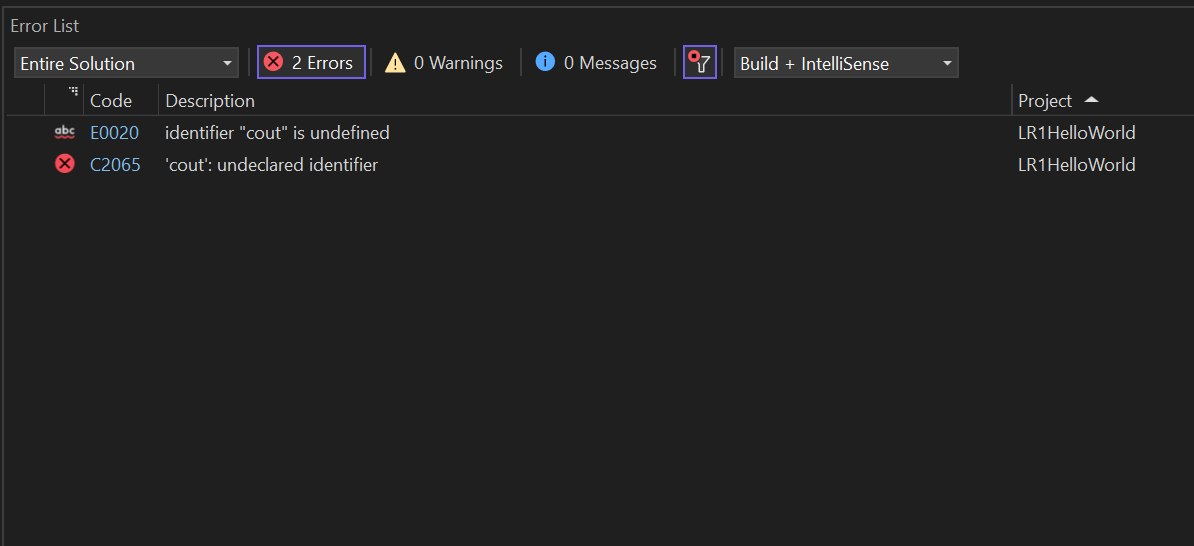
Ход выполнения работы

Для создания программы Hello World!, вначале, я открыл стартовое окно Visual Studio и создал новый проект консольного типа, как показано на рисунке 1.

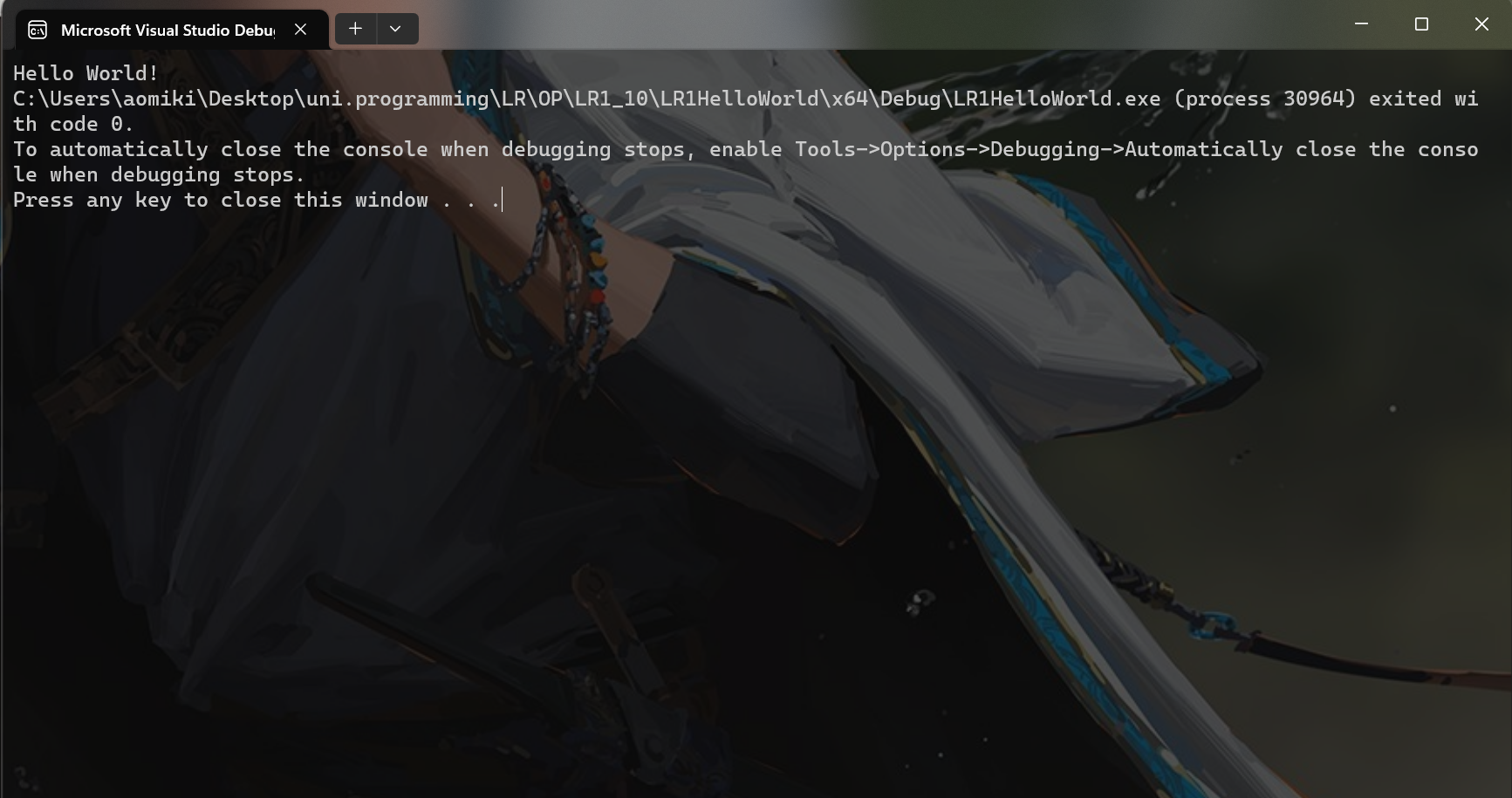
Рисунок 1 — Окно создания проекта

Далее я добавил новый файл Program.cpp. В нем написал функцию int main(), являющейся точкой входа в программу. Далее я подключил библиотеку iostream, написав вверху файла #include <iostream>. Эта библиотека дала доступ к объекту cout, который можно использовать для дальнейшего вывода пользователю текста. После всех проделанных подготовительных шагов, наконец, я написал cout << “Hello World!” в теле функции main.

После нажатия на кнопку запуска программы, Visual Studio показала ошибку сборки, как видно на рисунке 2. Ошибка, простыми словами, говорит что компилятор не понимает что значит слово cout. Суть этой проблемы состоит в том, что cout определен в пространстве имен std, и это нужно явно указать в коде. Для этого, над функцией main, я пишу using namespace std, что импортирует данное пространство имен в мою программу.

Рисунок 2 — ошибка, неопределенный идентификатор

После этого я нажал кнопку запустить, и в консоли появился вывод программы — Hello World! (Рисунок 3).

Рисунок 3 — вывод программы Hello World

В рамках второй программы я написал алгоритм, подсчитывающий в последовательности число отрицательных и положительных чисел. Программу, в целом, можно разделить на три шага:

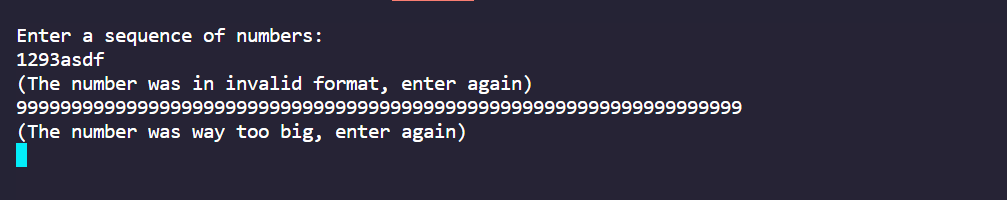
1. Чтение ввода и сохранение его в массив целых чисел

2. Подсчет положительных и отрицательных чисел в массиве

3. Вывод результата подсчета пользователю

Также, эти три шага оборачивает цикл, предлагающий после шага 3 начать выполнение заново с шага 1.

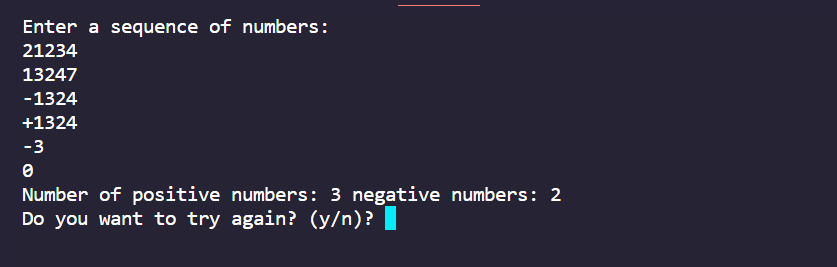
В первом шаге я реализовал функцию чтения строки целого числа readIntLine. Она использует getline чтобы получить строку, затем использует stoi чтобы конвертировать строку в целое число. Эти действия обернуты в конструкцию try catch, которая ловит ошибки конвертации, и показывает пользователю что он неправильно ввел число, как можно видеть на рисунке 4. Странной особенностью функции stoi, является то, что она без ошибок конвертирует строку, которая содержит числа и следующие за ними буквы (например 371falf). Для исключения такого случая, я использовал регулярное выражение ^[+-]?[0-9]+$, которое проверяет, является ли строка целым числом.

Рисунок 4 — ошибки при неправильном формате числа

Функция readIntLine вызывается до тех пор, пока она не вернет 0.

На втором шаге цикл итерирует через массив введенных чисел и инкрементирует две переменных, для положительных и отрицательных чисел.

Третий шаг просто выводит результат, как показано на рисунке 5.

Рисунок 5 — результат работы программы

Заключение

Цель лабораторной работы – написать две программы, первая - которая выводит Hello World!, и вторая — которая считает положительные и отрицательные числа, достигнута. В ходе выполнения лабораторной работы я изучил базовые операции ввода и вывода в C++, приобрел навыки по использованию массива vector и применению регулярных выражений. Самой интересной частью работы было написание регулярного выражения для валидации ввода пользователя, так как нужно было учесть все типы ввода числа, такие как +12, -12, 10.

Литература

1. Безус, Е.А. Лабораторные работы по курсу «Методы программирования» (язык программирования C++): Методические указания [Текст] / Сост. Е.А. Безус, Е.С. Козлова, С.С. Стафеев Самара, 2019. – 56 с.
2. Страуструп, Б. Язык программирования С++ / Б. Страуструп. – М:Бином, 2011 –1136 с.
3. Шилдт. Г. С++. Полное руководство / Г. Шилдт. – М: "ООО И.Д. Вильямс", 2019 – 800 с.
4. Страуструп, Б. Программирование: принципы и практика использования С++, испр. изд.: Пер. с англ. / Б. Страуструп. – М: "ООО И.Д. Вильямс", 2011 –1244 с.
5. Прата, С. Язык программирования С++. Лекции и упражения , 6-е изд.: Пер. с англ. / С. Прата. – Киев: "Диалектика", 2018 –1248 с.
6. Библиографическое описание документа. Общие требования и правила составления [Электронный ресурс] / сост.: В.С. Крылова, С.М. Григорьевская, Е.Ю. Кичигина // Официальный интернет-сайт научной библиотеки Томского государственного университета. – Электрон. дан. – Томск, [2010]. – <http://www.lib.tsu.ru/win/produkzija/metodichka/metodich.html> (дата обращения: 10.09.2019).

Приложение

Код первого пункта основного задания

#include <iostream>

using namespace std;

int main()

{

cout << "Hello world!";

return 0;

}

Код второго пункта основного задания

#include <iostream>

#include <vector>

#include <string>

#include <regex>

using namespace std;

/// @brief Reads int and performs input validation, asks for input again if needed

/// @return input int

int readIntLine()

{

regex reg\_matchInts = regex(R"(^[+-]?[0-9]+$)");

int result = 0;

bool done = false;

while (!done)

{

try

{

string input;

getline(cin, input);

result = stoi(input);

//regex to block the case when input is smth like "100ffas"

if(!regex\_match(input, reg\_matchInts))

throw invalid\_argument("");

done = true;

}

catch(const invalid\_argument& e)

{

cout << "(The number was in invalid format, enter again)\n";

}

catch(const out\_of\_range& e)

{

cout << "(The number was way too big, enter again)\n";

}

}

return result;

}

/// @brief Count elements of negative and positive integers in a sequence

int main()

{

//loop for repeating algo if user needs it

bool doContinue = false;

do

{

doContinue = false;

cout << "Enter a sequence of numbers:\n";

//reads numbers and saves them to an array

vector<int> numbers;

int currNumber;

currNumber = readIntLine();

while (currNumber != 0)

{

numbers.push\_back(currNumber);

currNumber = readIntLine();

}

//counts positive and negative numbers

int positiveNum = 0, negativeNum = 0;

for (size\_t i = 0; i < numbers.size(); i++)

{

if (numbers[i] > 0)

positiveNum++;

else

negativeNum++;

}

//shows result to user

cout << "Number of positive numbers: " << positiveNum << " negative numbers: " << negativeNum << "\n";

string str\_doContinue;

while (str\_doContinue != "y" && str\_doContinue != "Y" && str\_doContinue != "n" && str\_doContinue != "N")

{

//asks whether to execute algo again or not

cout << "Do you want to try again? (y/n)? ";

getline(cin, str\_doContinue);

if (str\_doContinue == "y" || str\_doContinue == "Y")

doContinue = true;

}

} while (doContinue);

return 0;

}