Оглавление

[ВВЕДЕНИЕ 2](#_Toc169899067)

[1 ОСНОВНАЯ ЧАСТЬ 4](#_Toc169899068)

[**1.1 Особенности создания Windows-приложений на языке С++ в** 4](#_Toc169899069)

[**среде Visual Studio 2022** 4](#_Toc169899070)

[**1.2 Интеграция C++ с Visual Studio.NET** 5](#_Toc169899071)

[2 РАЗРАБОТКА ПРОГРАММЫ И ИНТЕРФЕЙСА 8](#_Toc169899072)

[**2.1 Описание программы** 8](#_Toc169899073)

[**2.2 Тестирование программы** 10](#_Toc169899074)

# ВВЕДЕНИЕ

Учебная практика была пройдена в Амурском Государственном Университете с «17» июня 2024 года по ««29» июня 2024 года. Юридический адрес Амурского Государственного Университета: Игнатьевское шоссе 21, город Благовещенск, Амурская область.

Руководителем практики являлась Никифорова Лариса Владимировна.

Целью учебной практики заключалась в овладении студентами профессиональной деятельности согласно требованиям к уровню подготовки бакалавров по направлению подготовки 09.03.04 Программная инженерия.

Задачами практики являются:

Углубление знаний по дисциплинам, полученным за время обучения на первом курсе, таких как: «Программирование», «Информатика», «Компьютерные и информационные технологии в профессиональной деятельности», «Цифровая грамотность», «Линейная алгебра и теория матриц».

Развитие практических навыков разработки прикладного программного обеспечения и применения современных инструментальных средств для их создания;

Развитие практических навыков инсталляции и настройки программного обеспечения общего назначения и специализированных программ;

Формирование навыков подготовки и систематизации необходимых материалов и научно-технической информации для выполнения задания; - создание условий для практического применения знаний в области общепрофессиональных, специализированных компьютерных и математических дисциплин.

Формирование и совершенствование базовых профессиональных навыков и умений в области применения современных информационных технологий.

Формирование информационной компетентности с целью успешной работы в профессиональной сфере деятельности.

Приобретение навыков создания отчетов, в том числе и научно-технических, обеспечение успеха дальнейшей профессиональной карьеры.

# 1 ОСНОВНАЯ ЧАСТЬ

При выполнении учебной практики было проведено ознакомление со следующими разделами её программы:

Со структурой и основными принципами организации работы во время учебной практики Амурского Государственного Университета.

С видами деятельности, связанными с направлением подготовки 09.03.04 Программная инженерия.

С интерфейсом программного обеспечения, необходимого для

разработки алгоритма решения, кодирования программы и ее отладки.

С подходами и методологиями для успешного выполнения

индивидуального задания.

В процессе учебной практики была получена информация, необходимая для успешного выполнения индивидуального задания. Проведена инсталляция программного обеспечения Microsoft Visual Studio, а также изучена теоретическая часть, необходимая для событийно-управляемого программирования в среде разработки Visual Studio.NET.

## **1.1 Особенности создания Windows-приложений на языке С++ в**

## **среде Visual Studio 2022**

Среда разработки Visual Studio 2022 позволяет создавать, отлаживать и проводить тестирование приложений на Windows. Данная среда разработки поддерживается такими версиями Windows как Windows 11 и Windows 10. Одной из самых важных преимуществ этой среды является поддержка различных инструментов для управления зависимостями и сборкой проекта. Visual Studio 2022 предоставляет мощные инструменты для разработки Windows-приложений на C++. Для начала работы необходимо было сделать следующее:

Установить Visual Studio 2022 с компонентами для разработки на C++. Для выполнения задания было необходимо было установить компонент "Desktop development with C++". Данный компонент позволяет контролировать исходный код, управлять рабочими элементами, учитывать синтаксис при редактировании кода.

Настроить среду разработки, выбрав необходимые компоненты и SDK для Windows. Для возможности работы с Windows Form необходимо было установить расширение C++/CLI.

## **1.2 Интеграция C++ с Visual Studio.NET**

Среда Visual Studio.NET содержит удобные средства разработки Windows-приложений, позволяющие избавить программиста от рутинной работы. Однако при интеграции среды .NET в язык программирования C++, возникают следующие особенности:

Среда разработки Visual Studio .NET поддерживает C++, но для работы необходимо использовать расширение C++/CLI. C++/CLI – это расширение позволяющее использовать возможности платформы .NET Framework для языка программирования C++.

Необходимо знать, как использовать специальные инструменты и функции, так как среда разработки Visual Studio .NET имеет собственную функциональностью и инструменты.

У Visual Studio .NET имеется собственная система сборки и управления проектами, которая отличается от среды разработки С++. Именно из-за этого программисту необходимо знать специализированные функции и инструменты.

При применении расширения C++/CLI программист может использовать стандартный функционал С++ с возможностью использовать управляемые типы и классы. Данная возможность позволяет создавать типы-обертки, которые можно будет потом использовать в любом языке NET.

Платформа .NET состоит из следующих компонентов:

Common Language Runtime (CLR) - реализацией спецификации CLI. Так как приложения .NET закодированы на языке CIL, то основная задача CLR заключается в обеспечении выполнения приложений .NET.

Библиотека классов Framework (FCL) - это компонент Microsoft .NET Framework, реализующий систему виртуального выполнения CLI. FCL необходим для реализации базовых стандартных библиотек CLI.

Что бы связать код на С++ и среду NET необходимо чтобы все основные части кода находились вне функции main(), из-за того что во время компиляции функция main будет заменена. Это можно достичь вынесением алгоритмов в отдельные функции и типы данных.

При использовании CLR необходимо помнить, что определенная часть синтаксиса подчиняется своим особым правилам. Например, при создании объекта управляемого класса в динамической памяти необходимо знать, что вместо «new» нужно использовать «gcnew». Приписка «gc» позволяет указать компилятору то, память выделяется под управляемый класс, который не требует ручной очистки и данный объект будет подчинятся правилам среды CLR.

Типы данных, реализованных в С++ при использовании в управляемых классах CLR необходимо конвертировать. Конвертация должна происходить в явном или не явном виде. Использование неявного вида не допустимо при возможности потери информации.

Среда выполнения Windows и среда CLR представляют свои типы данных в виде объектов, управление выделяемой памятью которых осуществляется автоматически. Это значит, что в случае выхода переменной за пределы области видимости или завершении работы приложения явно отменять память для переменной не потребуется.

# 2 РАЗРАБОТКА ПРОГРАММЫ И ИНТЕРФЕЙСА

## **2.1 Описание программы**

Обозначение и наименование программы: «Буханов Денис Евгеньевич, группа 3105 об, 3 вариант».

Программное обеспечение, необходимое для работы программы: Операционная система Windows 10 или Windows 11, с поддержкой архитектуры x86-64

Язык программирования: C++/CLI.

Функциональное назначение: генерация случайных и заданных пользователем значений массива, сортировка массива и вывод его на экран.

Описание логической структуры: код включает в себя метод «get\_inform» класса «InputForm» который генерирует массив и метод «sort» класса «My\_arr» который необходим для сортировки массива.

Метод «get\_inform» генерирует массив по следующему правилу:

Введенный параметр является центральным и последним элементом массива, остальные элементы создаются с помощью счетчика случайных чисел. Всего в массиве содержится 25 элементов.

Метод «sort» выполняет сортировку массива следующем образом: если положительных элементов массива больше чем отрицательных, то отсортированные по убыванию положи-тельные элементы расположить в начале массива. Иначе в начале массива расположить отсортированные отрицательные элементы.

Входные и выходные данные: Метод «get\_inform» класса «InputForm» при своей работе считывает данные с формы которую заполнил пользователь, генерирует вектор размерностью 25 и возвращает данный вектор. Если флажок типа CheckBox будет иметь значение «Истинна», то в вектор будут добавлены числа, созданные с помощью генератора случайных чисел. После полученный вектор используется при создании объекта класса «My\_arr». Метод «sort» класса «My\_arr» позволяет отсортировать вектор по выше изложенному правилу. В таблице 1 перечислены все вышеуказанные элементы, их идентификаторы и типы.

Таблица 1 – Основные переменные программы

|  |  |  |
| --- | --- | --- |
| Идентификатор | Тип | Хранимые данные |
| min\_range | int | Минимальный диапазон  случайных чисел |
| max\_range | int | Максимальный диапазон  случайных чисел |
| items | vector<int> | Последовательный  контейнер вектор  целочисленных значений |
| my\_arrey | My\_arr<int>\* | Массив содержащий последовательный контейнер |

В последствие, для интеграции кода C++ с Средой Visual Studio.NET данные переменные будут переконвертированы в аналогичные типы данных, поддерживаемые языком C++/CLI. В таблице 2 указаны аналоги этих переменных в соответствующем порядке.

Таблица 2 – Переменные после конвертации

|  |  |  |
| --- | --- | --- |
| Идентификатор | Тип | Аналогичная переменная |
| min\_range\_random->Value | Int32 | min\_range |
| max\_rande\_random->Value | Int32 | max\_range |

## **2.2 Тестирование программы**

Согласно алгоритму программы, в результате её выполнения можно будет получить 2 массива с 25 значениями по указанным выше правилам. Предполагается, что при вводных данных, приведённых на рисунке 1, результат будет следующим: необработанный массив состоит из 25 элементов, центральный и последний элемент имеет значение «89», все остальные элементы задаются случайно в диапазоне от [-100; 100].

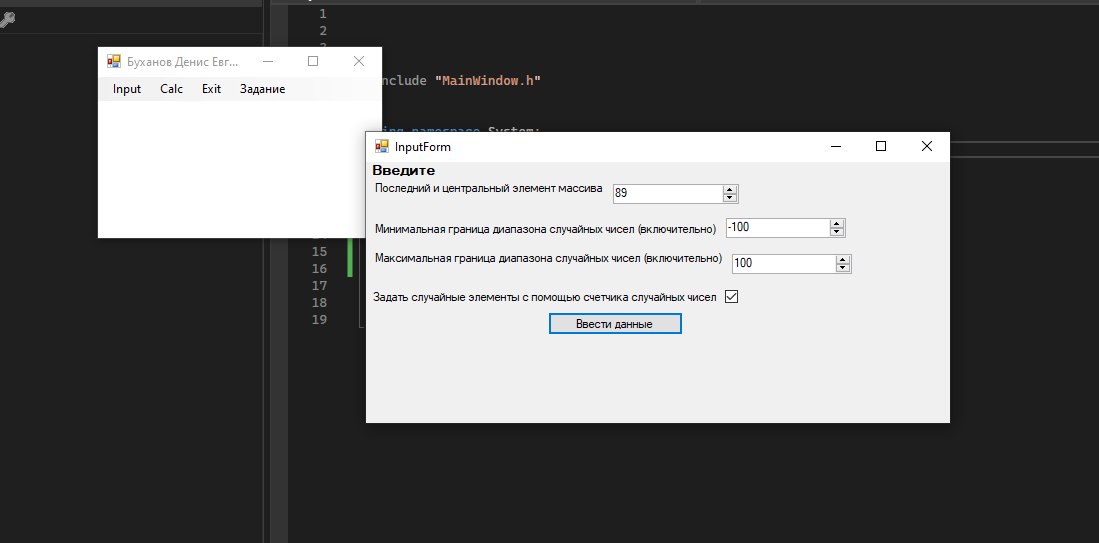


Рисунок 1 – Скриншот ввода данных в программу

Результат обработанного массива будет являться отсортированный массив, начинающийся с отсортированных в порядке убывания отрицательных чисел если отрицательных чисел в массиве больше, чем положительных или с отсортированных в порядке убывания положительных чисел если положительных чисел в массиве больше, чем отрицательных. На рисунке 2 видно, что программа выдала ожидаемый результат.

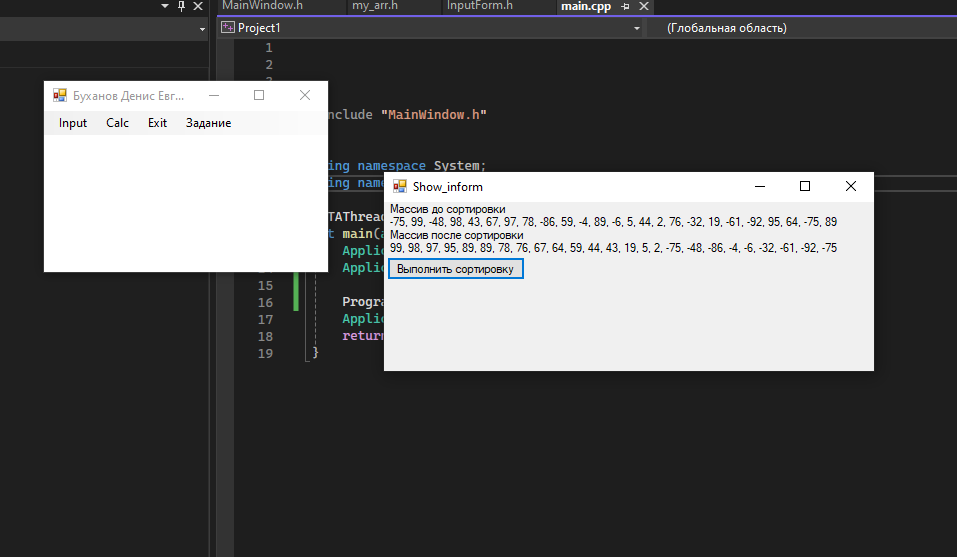


Рисунок 2 – Скриншот выполнения программы

Полученный массив имеет 25 элементов, все случайные значения не выходят за приделы диапазона, отсортированный массив начинается с максимального числа «99».