

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационной безопасности
Направление подготовки 09.03.04 Программная инженерия

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ ПОДГОТОВКЕ

Вид практики: Учебная практика

учебная или производственная практика

Тип практики: ознакомительная практика

наименование типа практики в соответствии с ФГОС ВО

Выполнил обучающийся _____ / _____ /
подпись ФИО, курс, номер группы

Дата сдачи отчета: «29» июня 2024 г.

Дата аттестации «29» июня 2024 г.

Оценка _____

Руководитель практической
подготовки от ФГБОУ ВО «АмГУ» _____ / Л.В. Никифорова /
подпись И.О.Фамилия

Заведующий кафедрой _____ / Л.В. Никифорова /
подпись И.О. Фамилия

Благовещенск, 2024

СОДЕРЖАНИЕ

| | |
|--|----|
| Введение | 6 |
| 1 Описание предметной области | 8 |
| 1.1 Особенности создания Windows-приложений на языке C++ в среде Visual Studio 2022 | 8 |
| 1.2 Интеграция C++ с Visual Studio.NET | 9 |
| 2 Разработка приложения и GUI-интеграция | 12 |
| 2.1 Описание программы | 12 |
| 2.2 Тестирование программы | 13 |
| 2.3 Руководство пользователя (оператора) | 15 |
| Заключение | 18 |
| Приложение А Текст программы | 19 |

ВВЕДЕНИЕ

Учебная практика пройдена в Амурском Государственном Университете с «17» июня 2024 года по ««29» июня 2024 года. Юридический адрес: Игнатьевское ш., 21, Благовещенск, Амурская обл.

Руководителем практики является Никифорова Лариса Владимировна.

Целью учебной практики является обеспечение непрерывности и последовательности 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», «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

Во время выполнения 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 C++ в среде Visual Studio 2022

Интегрированная среда разработки (IDE) облегчает создание, отладку и тестирование приложений на Windows. Она поддерживается различными версиями Windows, включая Windows 7, 8, 10 и 11. Одной из самых важных преимуществ этой среды является поддержка различных инструментов для управления зависимостями и сборкой проекта (например, NuGet, MSBuild) [1].

При помощи программного обеспечения под названием Visual Studio Installer устанавливается не только среда разработки Visual Studio, но и зависимости, необходимые для работы с ней. Для работы с языком C++ и выполнения индивидуального задания, необходимо установить «Desktop development with C++», которая включает в себя индивидуальные

компоненты, нужные для работы основных возможностей оболочки, включая редактор кода с учетом синтаксиса, контроль исходного кода и управление рабочими элементами, а также поддержки спецификации CLI (англ. Common Language Infrastructure) [4].

1.2 Интеграция C++ с Visual Studio.NET

Среда Visual Studio.NET содержит удобные средства разработки Windows-приложений, выполняющие вместо программиста рутинную работу создание шаблонов и заготовок обработчиков событий. Однако для интеграции кода чистого C++, необходимо знать особенности её работы:

1) Среда разработки Visual Studio .NET имеет свои собственные инструменты и функциональность, которая может быть использована при разработке проектов на C++. Для эффективной работы с кодом на C++ необходимо знать, как использовать эти инструменты и функции.

2) Visual Studio .NET имеет свою собственную систему сборки и управления проектами, которая немного отличается от других сред разработки. Для успешной интеграции кода на C++ необходимо знать особенности этой системы и уметь правильно настраивать проекты.

3) Несмотря на то, что среда разработки Visual Studio .NET поддерживает C++, для её работы необходимо использовать C++/CLI.

C++/CLI – это расширение стандартного языка программирования C++, которое предоставляет возможность создавать приложения для платформы .NET. Это означает, что можно использовать возможности .NET Framework вместе с языком C++. В C++/CLI можно работать как с нативным кодом C++, так и с кодом, написанным на языке C# или других языках .NET.

С применением расширений C++/CLI программист может определять управляемые и неуправляемые классы и функции в пределах одного файла, использовать управляемые и низкоуровневые типы C и C++, как в обычном программном коде на C++, то есть простым подключением заголовочного файла и связыванием с библиотекой. Эти широчайшие возможности можно использовать для создания управляемых типов-оберток, пригодных для

использования в любом языке NET, а также низкоуровневые классы-обертки и функции, пригодные для использования в программном коде на C/C++ [2].

Платформа .NET состоит из двух основных компонентов. Это Common Language Runtime и .NET Framework Class Library. Среда CLR является реализацией спецификации CLI, спецификации общезыковой инфраструктуры компании Microsoft. Его задача – обеспечить выполнение приложений .NET, которые, как правило, закодированы на языке CIL, рассчитаны на автоматическое управление памятью и вообще требуют гораздо больше заботы, чем обычные приложения Windows. Поэтому CLR занимается управлением памятью, компиляцией и выполнением кода, работой с потоками управления, обеспечением безопасности и т.п. [1]

Для того, чтобы связать готовый код на C++ и среду разработки .NET нужно переписать его при необходимости таким образом, чтобы все важные итерации кода происходили вне функции `main()`, т.к. она в дальнейшем будет заменена. Это достигается вынесением всех алгоритмов в отдельные функции. Для удобства создается заголовочный файл с вызовом всех необходимых библиотек и прототипами всех указанных функций.

Несмотря на то, что CLR является подтипом языка C++, большая часть его синтаксиса подчиняется собственным уникальным правилам. Например, чтобы обеспечить выделение динамической памяти при инициализации оператора «new» необходимо добавить «gc». Ключевое слово «_gc» перед объявлением класса говорит компилятору, что класс программы является управляемым и подчиняется правилам среды CLR. В частности, нет необходимости вызывать деструктор для удаления объекта из памяти.

Множество типов данных так же подлежит конвертации. Определяемое пользователем преобразование может быть неявным или явным. Оно должно быть неявным, если преобразование не приводит к потере информации. В противном случае необходимо определить явное преобразование. Так, нельзя записать тип данных `string` из библиотеки `<string>` в среде CLR. Необходимо преобразовать данный тип в `String^`.

Среда выполнения Windows и среда CLR представляют строки в виде объектов, управление выделяемой памятью которых осуществляется автоматически. Это значит, что в случае выхода строковой переменной за пределы области видимости или завершении работы приложения явно отменять память для строки не требуется. Чтобы указать, что управление временем существования строкового объекта должно осуществляться автоматически, следует объявить тип `string` с помощью модификатора дескриптор объекта (^). Знак «^» означает, что объявленная переменная является дескриптором управляемого объекта C++/CLI.

2 РАЗРАБОТКА ПРИЛОЖЕНИЯ И GUI-ИНТЕГРАЦИЯ

2.1 Описание программы

Обозначение и наименование программы: «Пигасов Андрей Александрович, 3105-об, В. 24».

Программное обеспечение, необходимое для функционирования программы: Операционная система Windows (7,8,9,10, 11), архитектура x64 (64-bit).

Языки программирования: C++, C++/CLI.

Функциональное назначение: генерация случайных и заданных пользователем значений матрицы, их сортировка и вывод на экран.

Описание логической структуры: код включает в себя две функции: «generateArray» для генерации массива и «sortArray» для сортировки массива. Каждая функция выполняет определенные операции с массивом целых чисел. Функция «generateArray» генерирует массив по следующему правилу: введенный параметр является третьим элементом массива, следующие семь элементов по правилу: каждый последующий в два раза больше предыдущего, остальные значения задаются с помощью счетчика случайных чисел. Всего генерируется 23 элемента массива. Функция «sortArray» выполняет сортировку массива по следующему правилу: если номер минимального элемента массива меньше 8, отсортировать по возрастанию элементы, расположенные после него, иначе отсортировать элементы с 8 по 18 номер. При необходимости, функция уменьшает минимальный элемент в два раза.

Входные и выходные данные: Функция «generateArray» принимает ссылку на вектор array, третий элемент thirdElement, минимальное значение RandMin и максимальное значение RandMax. Функция «sortArray» сортирует массив и, если логическая переменная test приобретает значение «Истинно», уменьшает минимальный элемент в массиве. В таблице 1 перечислены все вышеуказанные элементы, их идентификаторы и типы.

Таблица 1 – Основные переменные программы

| Идентификатор | Тип | Хранимые данные |
|---------------|--------------|--|
| array | vector<int>& | Последовательный контейнер вектор целочисленных значений |
| thirdElement | int | Третий элемент массива |
| RandMin | int | Минимальный диапазон случайных чисел |
| RandMax | int | Максимальный диапазон случайных чисел |
| test | int | Элемент, необходимый для условия |

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 |
|---------------|---------|------------------------|
| Third | Int32 | thirdElement |
| Minr | Int32 | RandMin |
| Maxr | Int32 | RandMax |
| Testr | Boolean | test |

2.2 Тестирование программы

Согласно алгоритму программы, в результате её выполнения мы сможем получить 2 динамических массива вектора с 23 значениями по указанным выше правилам. Предполагается, что при вводных данных, приведённых в таблице 3, результат будет следующим: необработанный массив состоит из 23 элементов, третий элемент имеет значение «2»,

следующие семь элементов – в два раза больше друг друга соответственно, то есть «4», «8», «16», «32», «64», «128» и «256». Случайные числа задаются в диапазоне [-10; 50]. Результат обработанного массива будет являться отсортированный массив, начинающийся после значения, которое является меньше 8. После этого, минимальное значение будет уменьшено в 2 раза.

Таблица 3 – Значения ввода

| Идентификатор | Присвоенное значение |
|---------------|----------------------|
| thirdElement | 2 |
| RandMin | 10 |
| RandMax | 50 |
| test | True |

На рисунке 1 видно, что программа выполнена успешно, согласно прогнозируемым результатам.

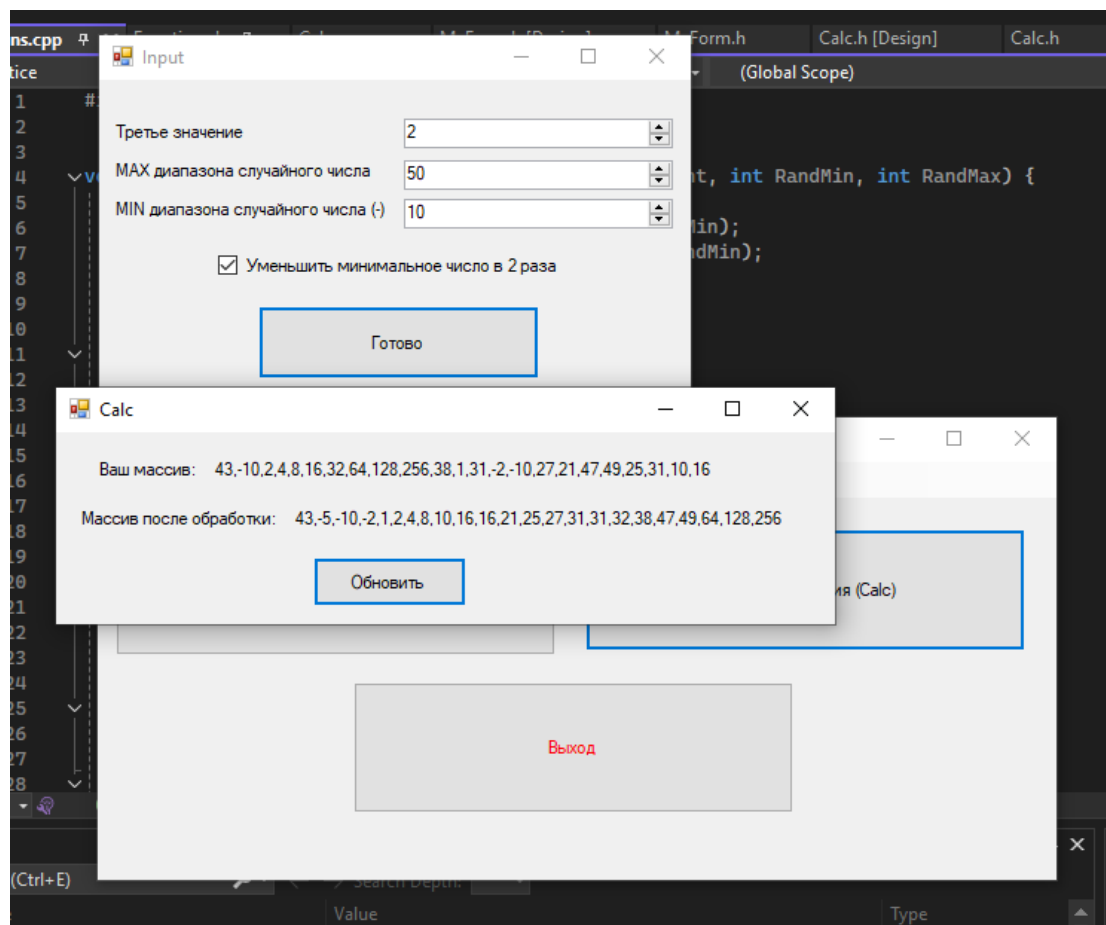


Рисунок 1 – Скриншот выполнения программы

Полученный массив имеет 23 элемента, все случайные значения внутри указанного диапазона, обработанный (отсортированный) массив начинается с минимального числа «-10», уменьшенного в два раза.

2.3 Руководство пользователя (оператора)

Условия применения: программное обеспечение может эксплуатироваться и выполнять заданные функции при соблюдении требований предъявляемых к техническому, системному и прикладному программному обеспечению.

Запуск программы: для входа в пользовательский интерфейс достаточно открыть скомпилированный файл программы с расширением «.exe». После открытия программы откроется меню (рис. 2), содержащее 3 кнопки – «Задать параметры входы (Input)», «Вывести значения (Calc)» и «Выход». В верхнем левом углу есть контекстная кнопка «Задание», которое открывает диалоговое окно с описанием варианта индивидуального задания.

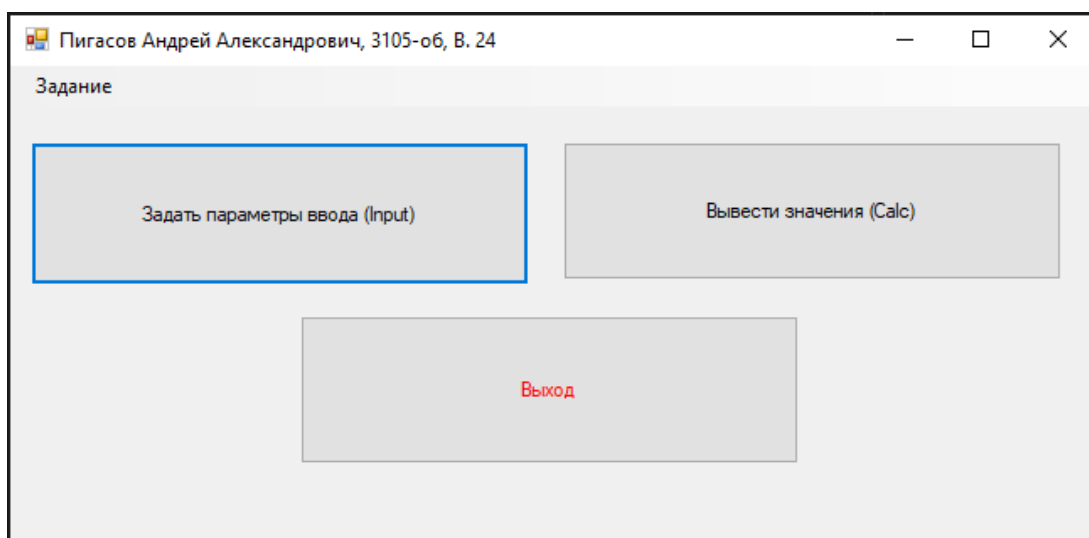


Рисунок 2 – Интерфейс программы в меню

Выполнение программы: для задания значений необходимо нажать на кнопку «Задать параметры входы (Input)». В открытом окне (рис. 3) пользователю предлагается ввести 3 значения: значение третьего элемента, минимальное и максимальное значение диапазона случайных чисел. Ниже есть флажок, который выполняет уменьшение минимального числа массива в 2 раза при желании пользователя. После ввода желанных значений,

оператору необходимо нажать на кнопку «Готово», чтобы результаты были переданы в форму для расчёта и вывода массива.

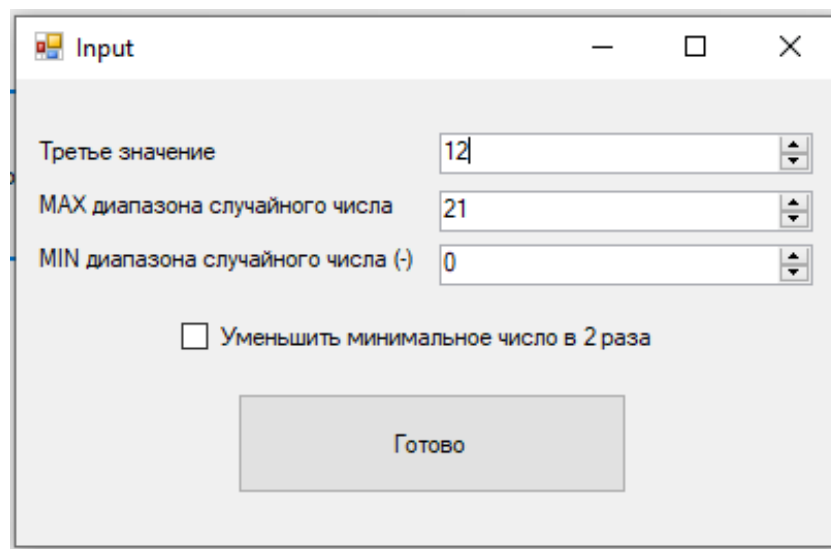


Рисунок 3 – Интерфейс диалогового окна «Input».

Теперь, при желании, диалоговое окно с вводом можно закрыть. Чтобы увидеть созданный и отсортированный массивы, пользователю необходимо вернуться в меню программы, и оттуда нажать кнопку «Вывести значения (Calc)». В открывшемся окне данные будут не обновлены, для появления корректных данных необходимо нажать на кнопку «Обновить» (рис. 4). В поле с меткой «Ваш массив» появится полученный на основе введенных данных массив. В поле «Массив после обработки» будет выведен отсортированный массив.

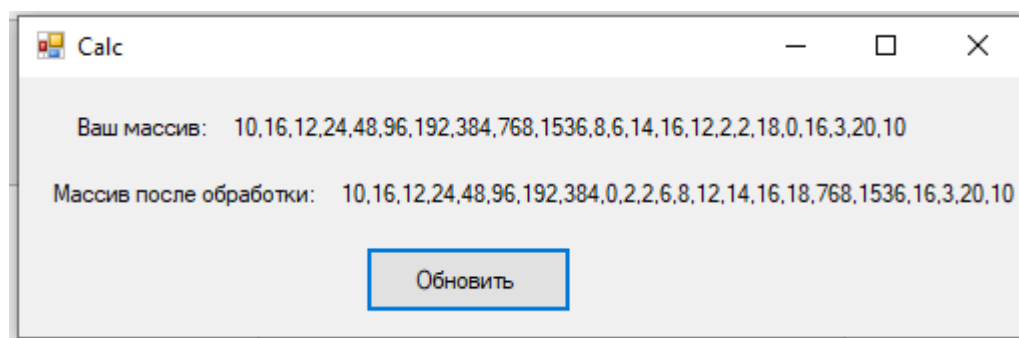


Рисунок 4 – Интерфейс диалогового окна «Calc»

Для выхода из приложения достаточно нажать кнопку «Выход» в меню.

Пользователь может не задавать начальные значения в диалоговом окне «Input» и напрямую перейти к выводу значений. В таком случае программа

будет использовать значения по умолчанию. Ниже приведена таблица значений по умолчанию (таблица 4).

Таблица 4 – Значения по умолчанию

| Идентификатор | Значение по умолчанию |
|---------------|-----------------------|
| thirdElement | 0 |
| RandMin | 0 |
| RandMax | 20 |
| test | False |

В диалоговом окне «Input» значения, вводимые пользователем, ограничены. Значение поля «Третье значение» задается в диапазоне [0;100], значение «MAX диапазона случайного числа» в [20;100] и «MIN диапазона случайного числа (-)» в [0;100]. Флажок «Уменьшить минимальное число в 2 раза» по умолчанию пустой.

ЗАКЛЮЧЕНИЕ

В процессе разработки приложения на C++/CLI в Microsoft Visual Studio было замечено, что практика имела значительное влияние на закрепление и углубление теоретических знаний. Участие в проекте позволило применить полученные ранее знания и навыки на практике, что повысило уровень понимания материала.

Кроме того, разработка приложения способствовала приобретению новых практических навыков и умений в области программирования на C++/CLI, а также в создании графического пользовательского интерфейса и интеграции консольного кода в него.

В целом, можно сделать вывод, что практика позволила успешно применять свои знания на практике и развивать свои профессиональные навыки. Во время прохождения практики все задания были успешно выполнены, и разработка рабочего приложения в качестве индивидуального задания была завершена.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Александров, Э. Э Программирование на языке С в Microsoft Visual Studio 2010: учебное пособие / Э. Э Александров, В. В. Афонин. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 569 с.
2. Голдштейн С. Оптимизация приложений на платформе .NET / С. Голдштейн, Д. Зурбалева, И. Флатов – М.: ДМК Пресс. 2022. – 524 с.
3. Макаров, А. В. Common Intermediate Language и системное программирование в Microsoft.NET: учебное пособие / А. В. Макаров, С. Ю. Скоробогатов, А. М. Чеповский. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 397 с.
4. Тельнова Ю. Ф. Информационные системы и технологии: науч. издание/ Ю. Ф. Тельнова – М.: ЮНИТИ-ДАНА, 2022. – 303 с.

ПРИЛОЖЕНИЕ А

ТЕКСТ ПРОГРАММЫ

```
//Functions.h

#pragma once

#include <vector>
#include <sstream>
#include <algorithm>
#include <cstdlib>
#include <ctime>
#include <Windows.h>
#include <vcclr.h>
#include <msclr/marshal_cppstd.h>

using namespace std;
using namespace System;

void generateArray(vector<int>& array, int thirdElement, int RandMin, int
RandMax);

void sortArray(vector<int>& array, int test);
string arrayToString(const vector<int>& array);

String^ ConvertStrtoCLR(string& os, String^ s);
string& Convert_StrToCLR(String^ s, string& os);

//Functions.cpp
#include "Functions.h"
```

```

void generateArray(vector<int>& array, int thirdElement, int RandMin, int
RandMax) {
    srand(time(NULL) % 1000);
    array.push_back(rand() % (RandMax+RandMin) - RandMin);
    array.push_back(rand() % (RandMax + RandMin) - RandMin);
    array.push_back(thirdElement);

    int nextElement = thirdElement;
    for (int i = 0; i < 7; ++i) {
        nextElement *= 2;
        array.push_back(nextElement);
    }

    for (int i = 0; i < 13; ++i) {
        array.push_back(rand() % (RandMax + RandMin) - RandMin);
    }
}

void sortArray(vector<int>& array, int test) {
    auto minElement = min_element(array.begin(), array.end());
    int minIndex = distance(array.begin(), minElement);

    if (minIndex < 8) {
        sort(array.begin() + minIndex, array.end());
    }
    else {
        sort(array.begin() + 8, array.begin() + 19);
    }
    if (test == true) { *minElement /= 2; }
}

```

```

}

string arrayToString(const vector<int>& array) {
    std::stringstream ss;
    for (size_t i = 0; i < array.size(); ++i)
    {
        if (i != 0)
            ss << ",";
        ss << array[i];
    }
    std::string s = ss.str();
    return s;
}

string& Convert_StrToCLR(String^ s, string& os){
    using namespace Runtime::InteropServices;
    const char* chars =
        (const char*)(Marshal::StringToHGlobalAnsi(s)).ToPointer();
    os = chars;
    Marshal::FreeHGlobal(IntPtr((void*)chars));
    return os;
}

String^ ConvertStrtoCLR(string& os, String^ s)
{
    s = gcnew System::String(os.c_str());
    return s;
}

//MyForm.cpp
#ifndef UNICODE

```

```

#define UNICODE
#endif

#ifndef _UNICODE
#define _UNICODE
#endif

#include <iostream>
#include <locale.h>
#include <Windows.h>
#include <tchar.h>
#include "MyForm.h"
#include "Calc.h"
#include "Input.h"

using namespace System;
using namespace System::Windows::Forms;
[STAThreadAttribute]

int main(array<String^>^ arg) {
    setlocale(LC_ALL, "RU");
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);

    Practice::MyForm form;
    Application::Run(% form);

    return 0;
}

```

```

System::Void Practice::MyForm::button1_Click(System::Object^ sender,
System::EventArgs^ e)
{
    Input^ inp = gcnew Input;
    inp->Show();
}

```

```

System::Void Practice::MyForm::button2_Click(System::Object^ sender,
System::EventArgs^ e)
{
    Calc^ calc2 = gcnew Calc;
    calc2->Show();
}

```

```

System::Void
Practice::MyForm::ToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    MessageBox::Show(_T("Введенный параметр является третьим
элементом массива, следующие семь элементов по правилу: каждый
последующий в два раза больше предыдущего, остальные значения задаются
с помощью счетчика случайных чисел. Если номер минимального элемента
массива, содержащего 23 значения, меньше 8, отсортировать по возрастанию
элементы, расположенные после него, иначе отсортировать элементы с 8 по
18 номер. Минимальный элемент уменьшить в два раза."), _T("Информация о
программе."));
}

//Calc.cpp
#include "Functions.h"
#include "Calc.h"

```

```

        System::Void Practice::Calc::b_update_Click(System::Object^ sender,
System::EventArgs^ e)
    {
        vector<int> array;

        try {
            generateArray(array, Third, Minr, Maxr);
        }
        catch (String^ exception) { MessageBox::Show(exception,"Внимание!");
Console::WriteLine(exception);
        }

        vector<int> array2 = array;
        std::string text = arrayToString(array);
        l_Array->Text = ConvertStrtoCLR(text,l_Array->Text);;
        try {
            sortArray(array2, Testr);
        }
        catch (String^ exception) {
            MessageBox::Show(exception, "Внимание!");
Console::WriteLine(exception);
        }

        string text2 = arrayToString(array2);
        l2_Array->Text = ConvertStrtoCLR(text2, l2_Array->Text);;
    }

//Input.cpp
#include "Input.h"
#include "Calc.h"
#include "MyForm.h"

System::Void Practice::Input::button1_Click(System::Object^ sender,
System::EventArgs^ e)
    {

```

```

int thirdElement = Convert::ToInt32(num_Third->Value);
int RandMax = Convert::ToInt32(num_RanMax->Value);
int RandMin = Convert::ToInt32(num_RanMin->Value);
Boolean Test = false;
if (checkBox1->Checked)
    Test = true;
else Test = false;

Calc^ calc = gcnew Calc();
calc->Third = thirdElement;
calc->Maxr = RandMax;
calc->Minr = RandMin;
calc->Testr = Test;
return System::Void();
}

```