МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

Лабораторная работа №9

**«Создание приложений Windows Forms**

**для формирования и обработки автоматических**

**одномерных массивов»**

**по дисциплине**

**«Алгоритмизация и программирование»**

Выполнил: студент гр. БЭИ2202 Бояркин Е. В.

Вариант №4

Проверил: доц. Воробейчиков Л. А.

Москва, 2023 г.

**Задание:**

Сформировать одномерный массив заданного размера n из случайных целых

чисел в заданном диапазоне. Если в данном массиве есть хотя бы одно нечетное

число, создать новый массив, записывая в него сначала нечетные числа, а затем

все четные числа, в порядке, обратном их расположению в исходном массиве.

**Интерфейс:**

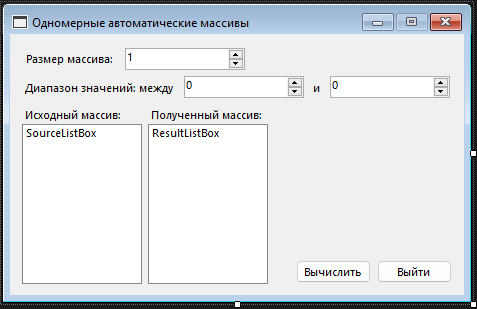


Рисунок 1

**Схемы алгоритмов:**

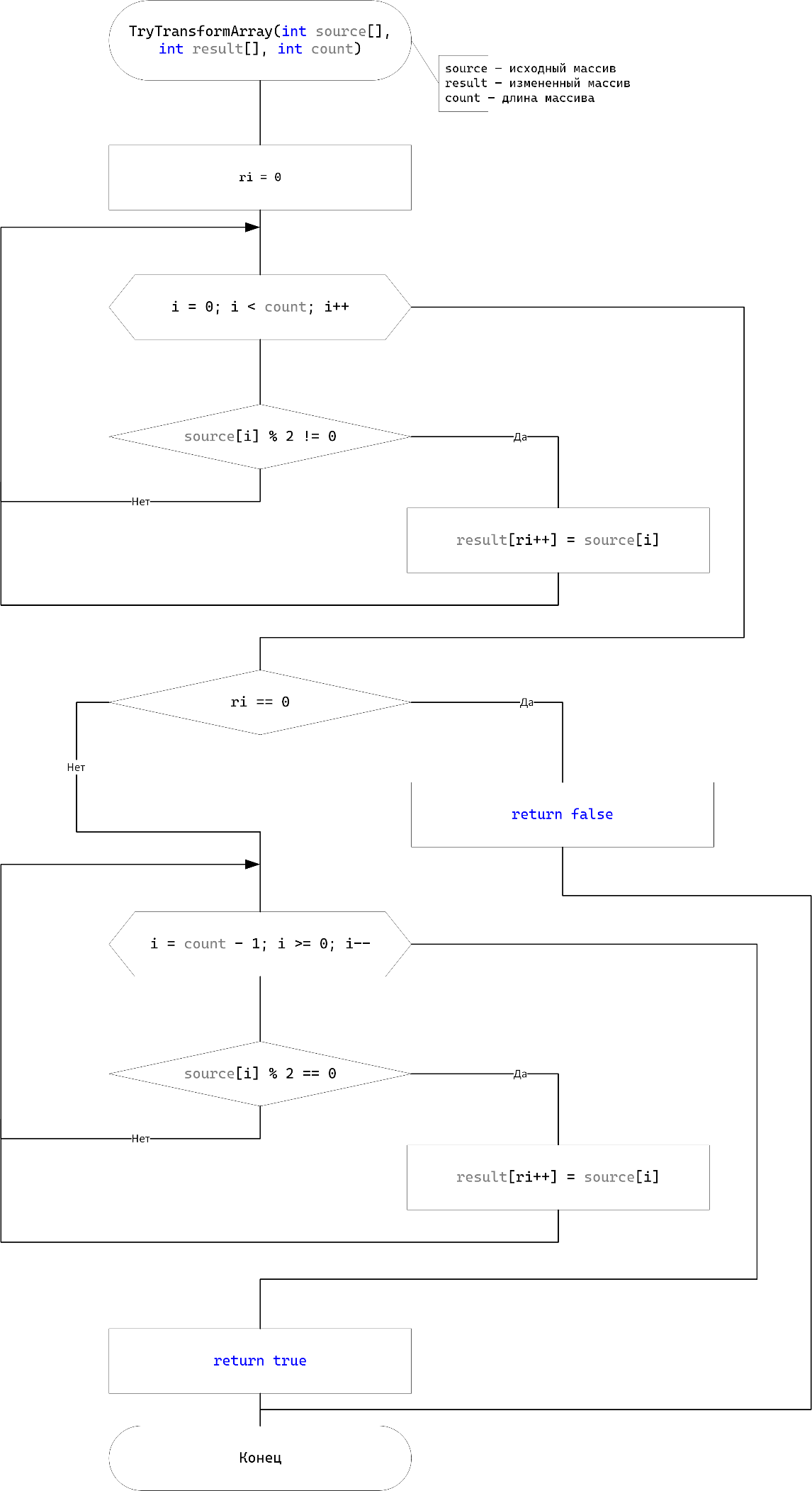


Рисунок 2

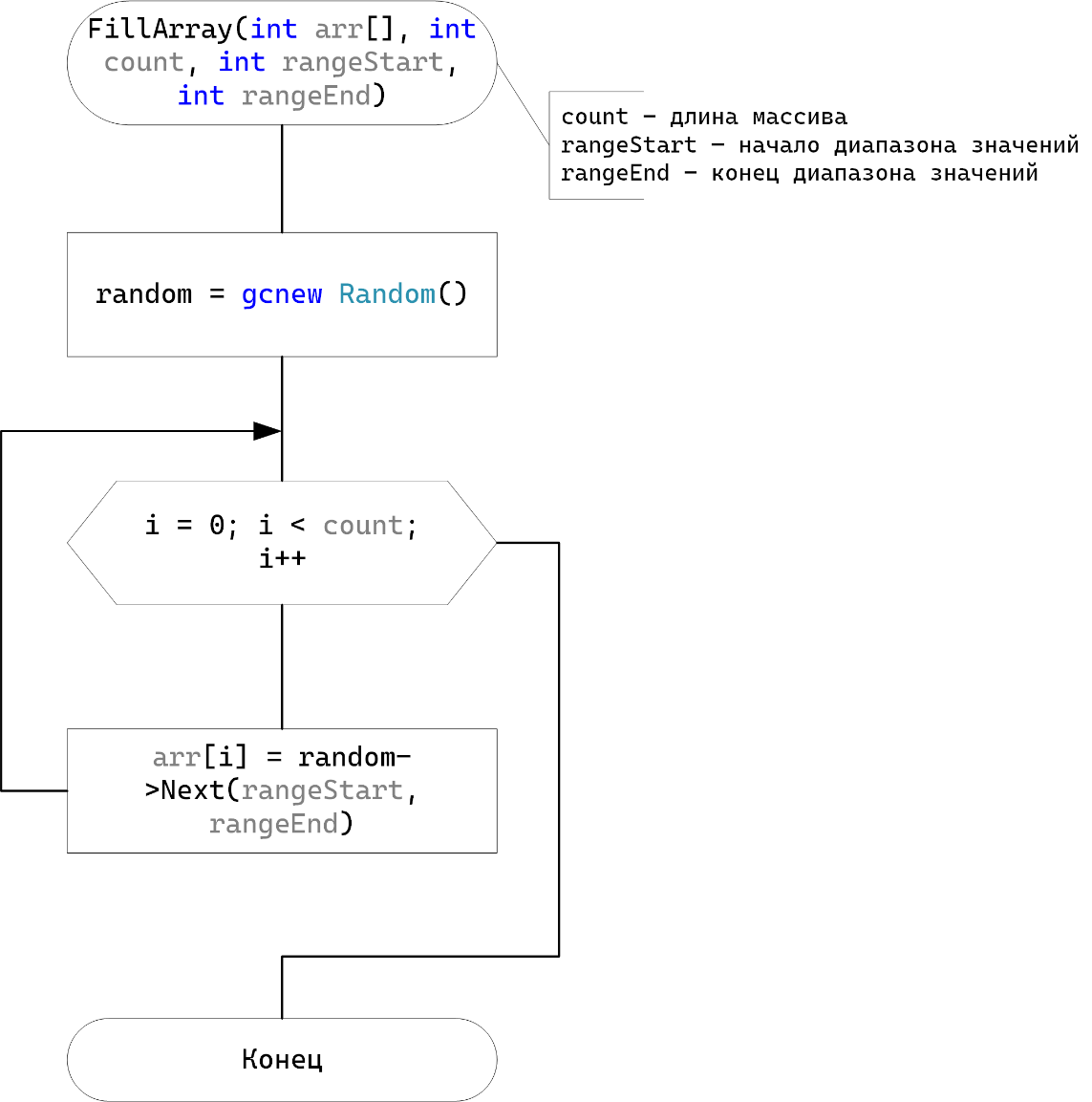


Рисунок 3

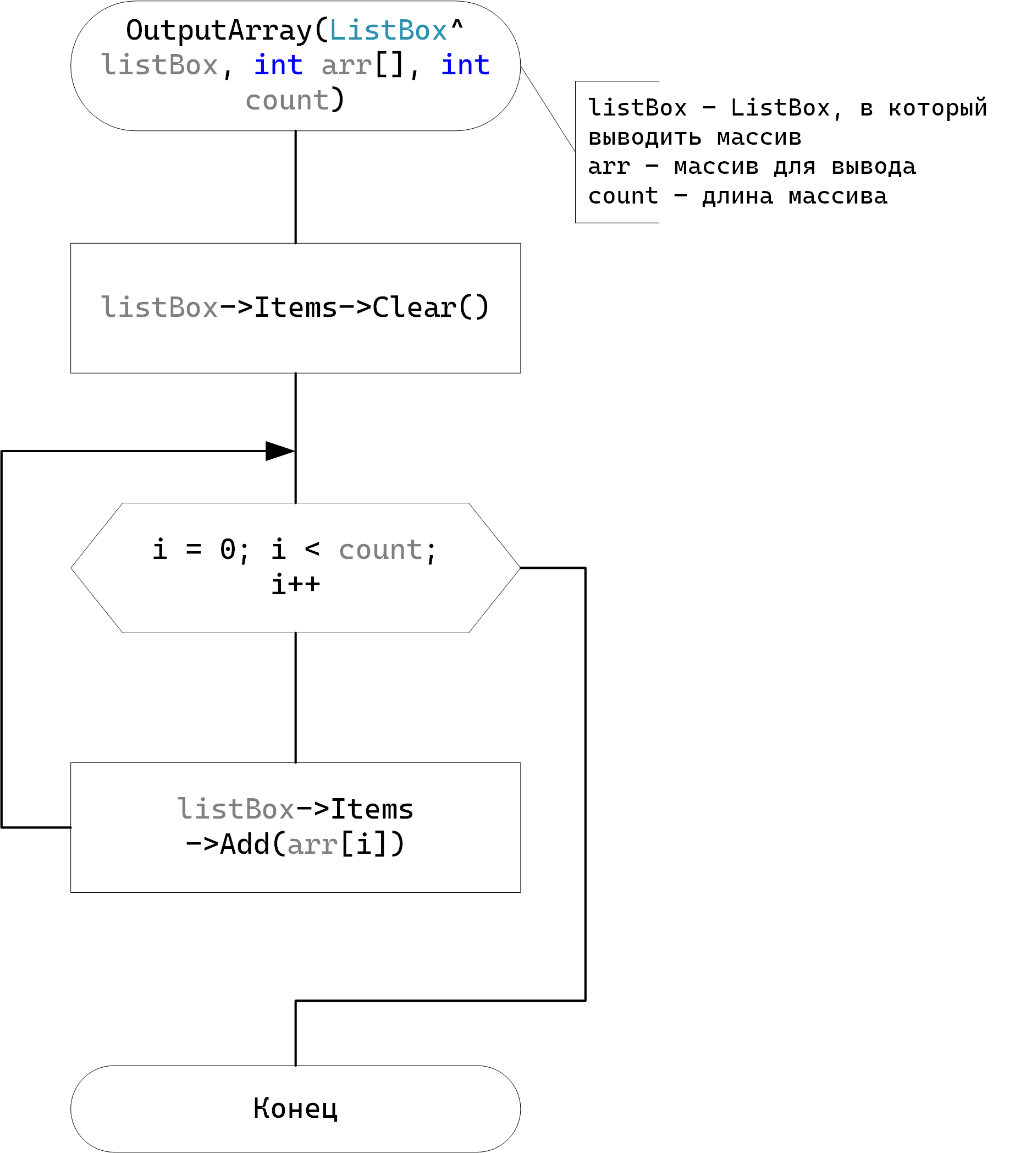


Рисунок 4

**Код:**

Методы для решения задачи:

|  |
| --- |
| #include "pch.h"  #include "Lab.h"  using namespace System;  using namespace System::Windows::Forms;  void FillArray(int arr[], int count, int rangeStart, int rangeEnd)  {  Random^ random = gcnew Random();  for (int i = 0; i < count; i++)  arr[i] = random->Next(rangeStart, rangeEnd);  }  void OutputArray(ListBox^ listBox, int arr[], int count)  {  listBox->Items->Clear();  for (int i = 0; i < count; i++)  listBox->Items->Add(arr[i]);  }  bool TryTransformArray(int source[], int result[], int count)  {  int ri = 0;  for (int i = 0; i < count; i++)  {  if (source[i] % 2 != 0)  result[ri++] = source[i];  }  if (ri == 0)  return false;  for (int i = count - 1; i >= 0; i--)  {  if (source[i] % 2 == 0)  result[ri++] = source[i];  }  return true;  } |

Заголовочный файл:

|  |
| --- |
| #pragma once  using namespace System::Windows::Forms;  void FillArray(int arr[], int count, int rangeStart, int rangeEnd);  void OutputArray(ListBox^ listBox, int arr[], int count);  bool TryTransformArray(int source[], int result[], int count); |

Файл кода программной части формы:

|  |
| --- |
| #include "pch.h"  #include "MainForm.h"  #include "Lab.h"  using namespace WfApp;  using namespace System;  using namespace System::Windows::Forms;  void MainForm::EvaluateButtonClick(Object^ sender, EventArgs^ e)  {  const int MaxArraySize = 10000;  int n = safe\_cast<int>(ArraySizeNumericUD->Value),  rangeStart = safe\_cast<int>(RangeStartNumericUD->Value),  rangeEnd = safe\_cast<int>(RangeEndNumericUD->Value);  if (rangeStart > rangeEnd)  {  int temp = rangeStart;  rangeStart = rangeEnd;  rangeEnd = temp;  }  int sourceArr[MaxArraySize],  resultArr[MaxArraySize];  FillArray(sourceArr, n, rangeStart, rangeEnd);  OutputArray(SourceListBox, sourceArr, n);  if (TryTransformArray(sourceArr, resultArr, n))  OutputArray(ResultListBox, resultArr, n);  else  {  MessageBox::Show(L"В исходном массиве нет нечётных чисел.", L"Информация", MessageBoxButtons::OK, MessageBoxIcon::Information);  ResultListBox->Items->Clear();  }  }  void MainForm::ExitButtonClick(Object^ sender, EventArgs^ e)  {  Application::Exit();  } |

**Результат выполнения:**

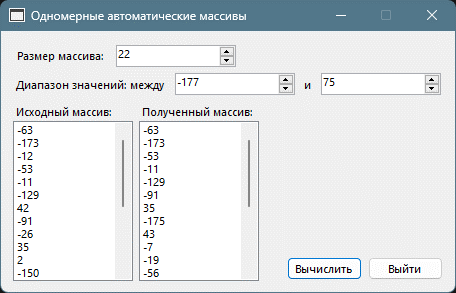


Рисунок 5

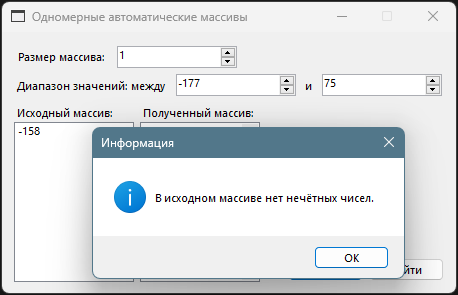


Рисунок 6

**Доказательство правильности результатов:**

В первом случае все записанные в новый массив элементы соответствуют заданию. Во втором случае в исходном массиве нечётных чисел нет, поэтому выведено соответствующее сообщение.

Тесты:

|  |
| --- |
| #include "pch.h"  #include "Lab.h"  #include <assert.h>  void TestSingle(int source[], int result[], int count, bool success)  {  int actualResult[MaxArraySize];  bool actualSuccess = TryTransformArray(source, actualResult, count);  assert(success == actualSuccess && L"Проверка на наличие нечетных чисел не пройдена");  if (!success)  return;  for (int i = 0; i < count; i++)  assert(result[i] == actualResult[i] && L"Полученный массив не равен требуемому");  }  void TestAll()  {  const int TestsCount = 3;  int counts[TestsCount] = { 3, 1, 7 };  int sources[TestsCount][MaxArraySize] = { { 1, 2, 3 }, { 0 }, { -51, 4236, 2, 4, 6, 12323, 3} };  int results[TestsCount][MaxArraySize] = { { 1, 3, 2 }, { }, { -51, 12323, 3, 6, 4, 2, 4236 } };  int successes[TestsCount] = { true, false, true };  for (int i = 0; i < TestsCount; i++)  TestSingle(sources[i], results[i], counts[i], successes[i]);  } |