МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет

им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации

Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА № 2

по дисциплине

Поддержка и тестирование программных модулей

Тема: «Интеграционное тестирование»

Работу выполнил

Студент гр.4335

Желваков А.С.

Принял

Преподаватель Максимов Р.С.

Казань 2025

**1. Цель работы:** Изучение назначения и задач интеграционного тестирования. Программная реализация тестов, производящих тестирование.

**2. Задание на лабораторную работу:**

**Задание 1.** Откройте выбранную IDE и создайте проект на основе существующих программных кодов, или реализуйте алгоритм.

**Задание 2.**  Подключите к проекту библиотеку unit-тестов.

**Задание 3.** Сгруппируйте модули в кластеры. В данном случае под кластерами подразумеваются классы или наборы классов, имеющие совместную замкнутую функциональность (не зависящий друг от друга).

**Задание 4.** Напишите план интеграционного тестирования.

**Задание 5.**  Исходя из количества описанных кластеров и заявленных для них тестовых примеров (последний пункт плана), создайте соответствующее количество юнит-тестов.

**Задание 6.** Отладьте и запустите все юнит-тесты. При этом не требуется исправление ошибок в исходном коде, если таковые были обнаружены.

**Задание 7.** Оцените результаты выполнения юнит-тестирования и сделайте соответствующие выводы.

**3. Результат выполнения работы:**

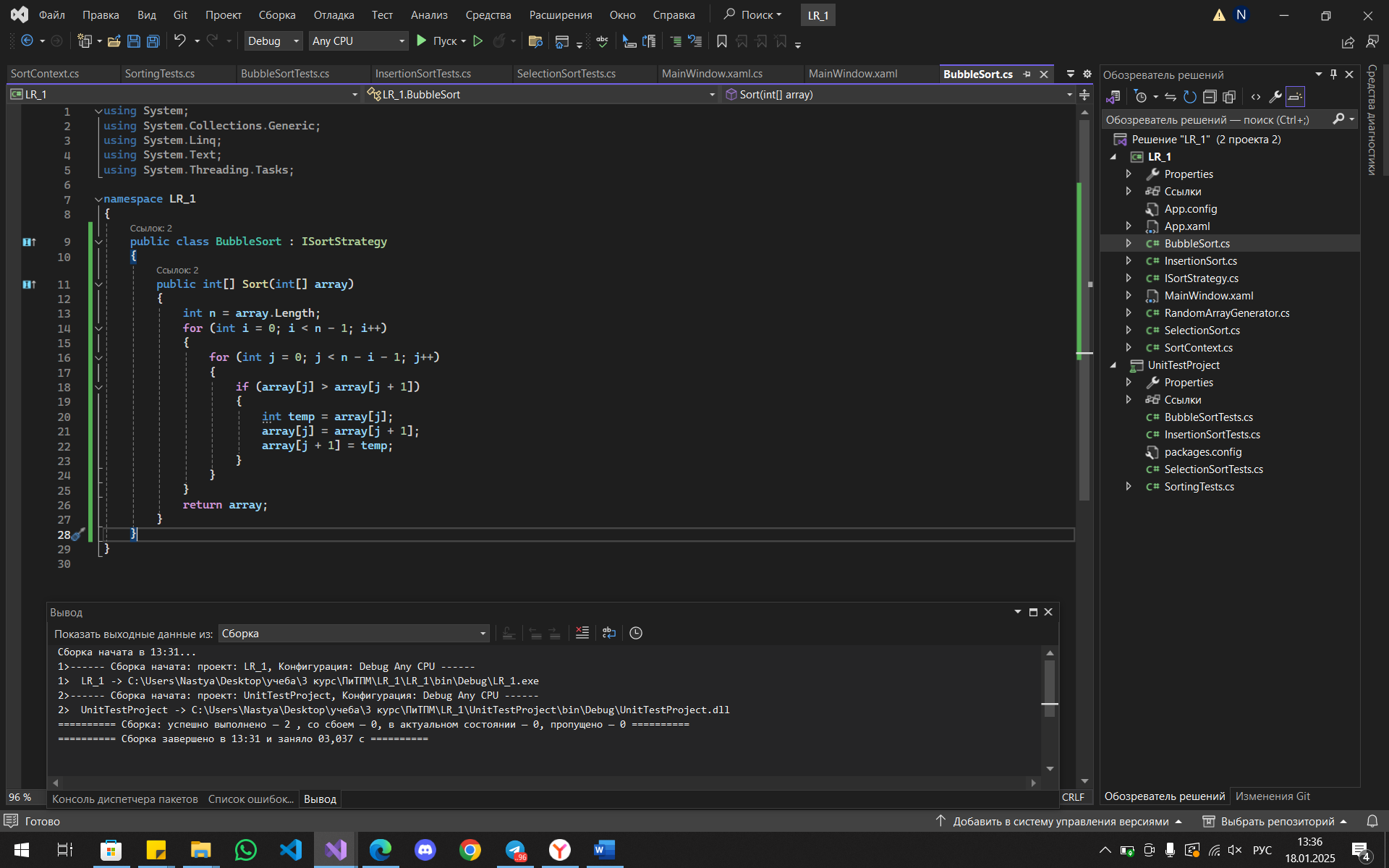
****

Рисунок 1 – Класс BubbleSort

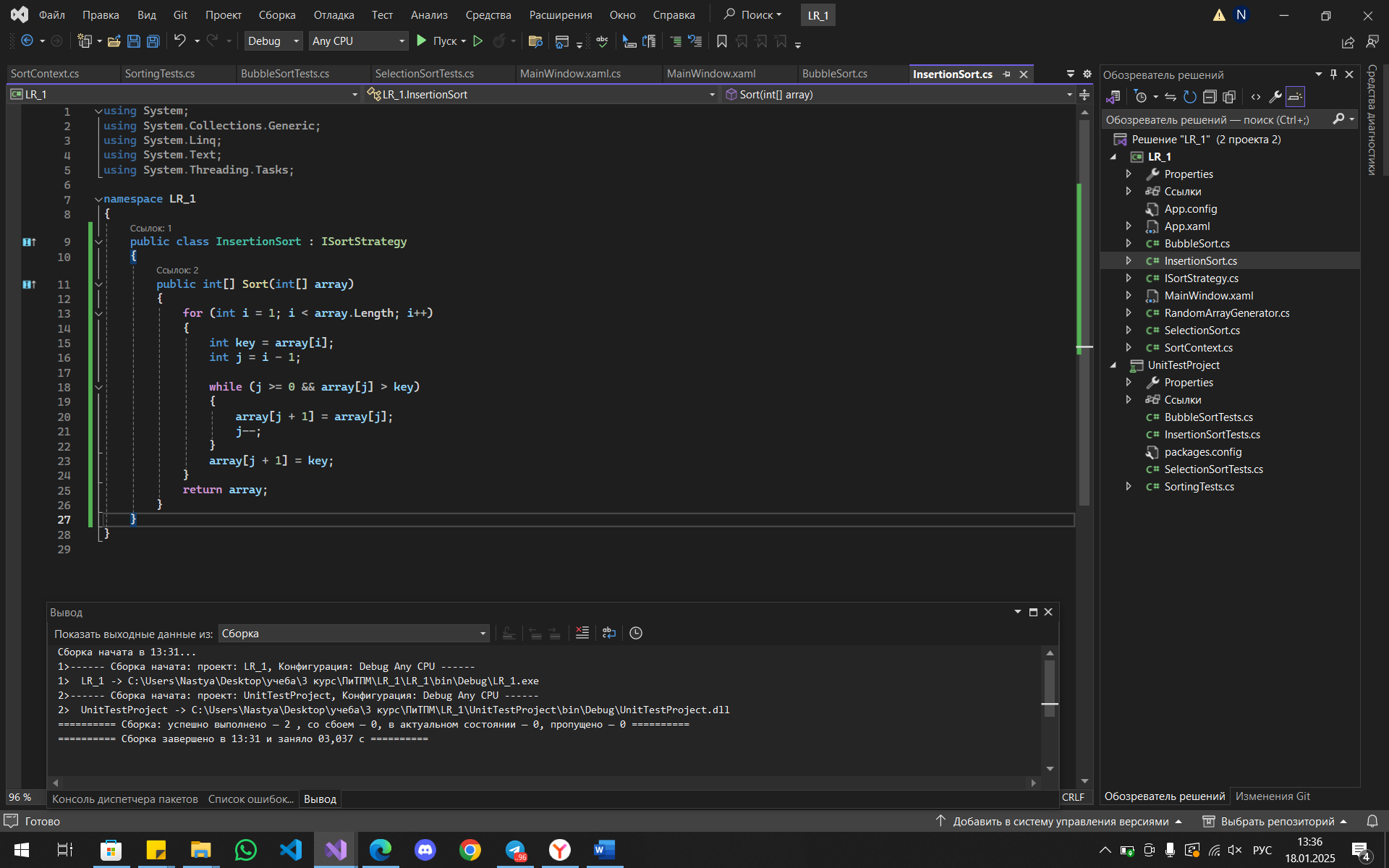


Рисунок 2 – Класс InsertionSort

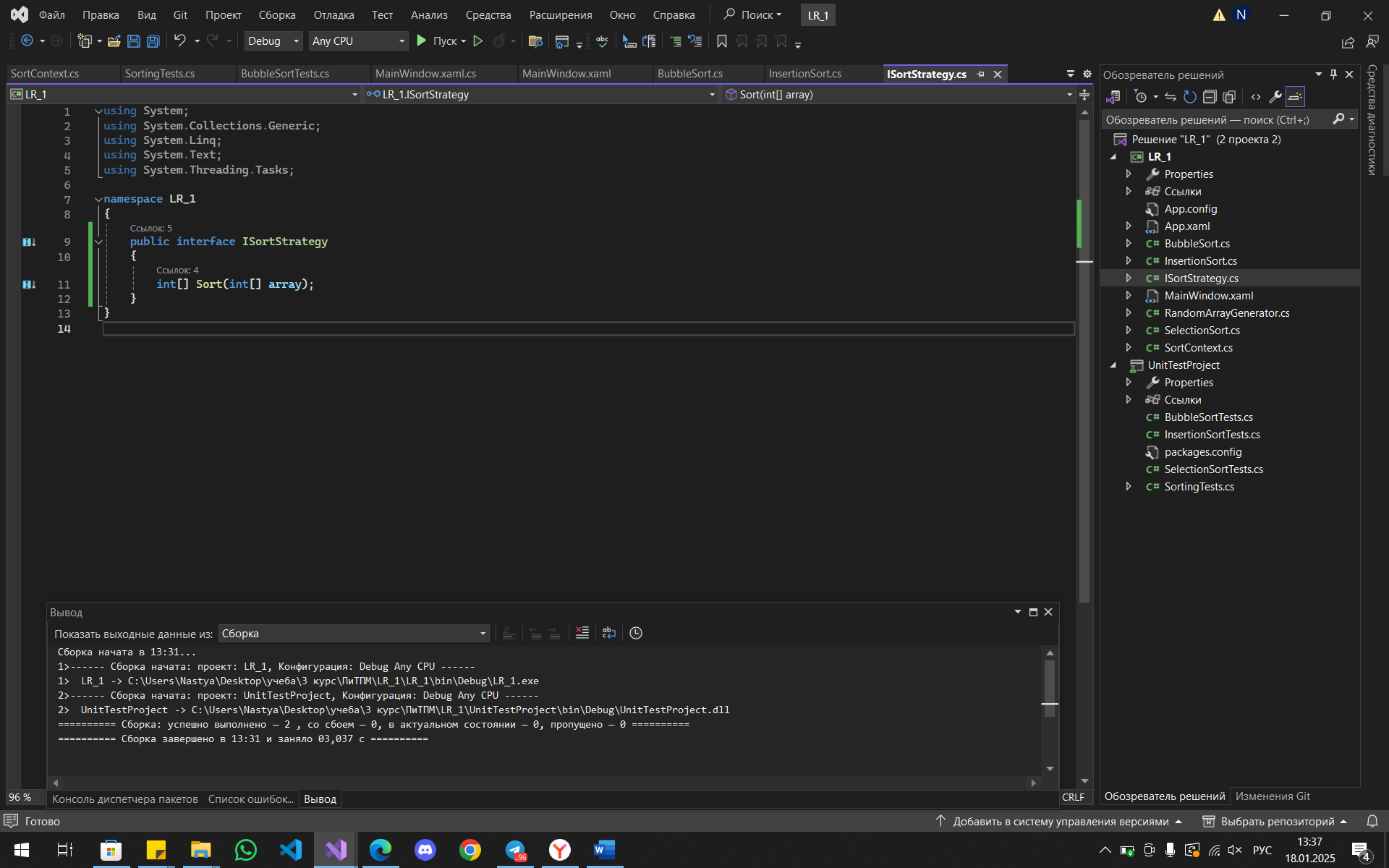


Рисунок 3 – Интерфейс ISortStrategy

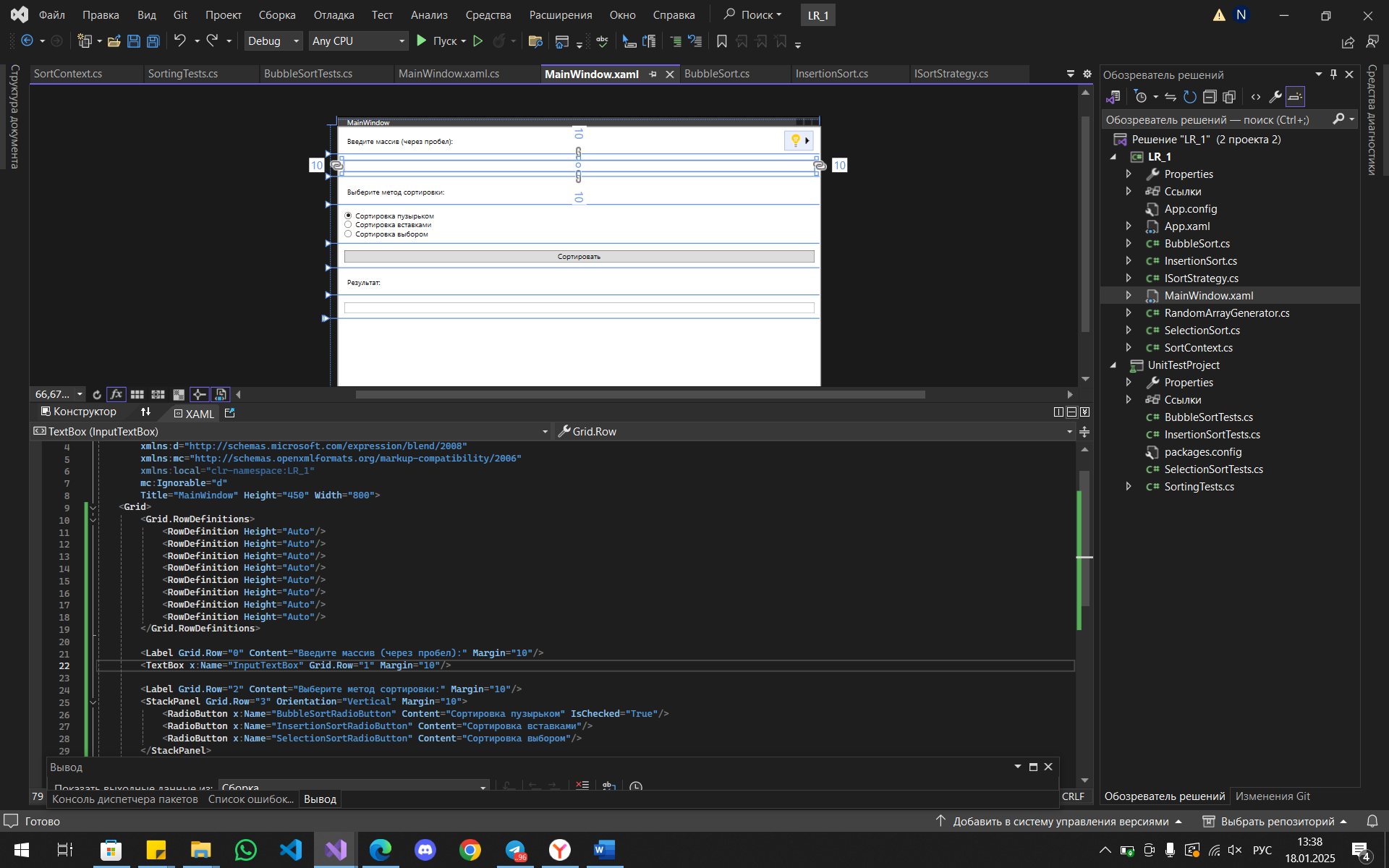


Рисунок 4 – Окно при запуске

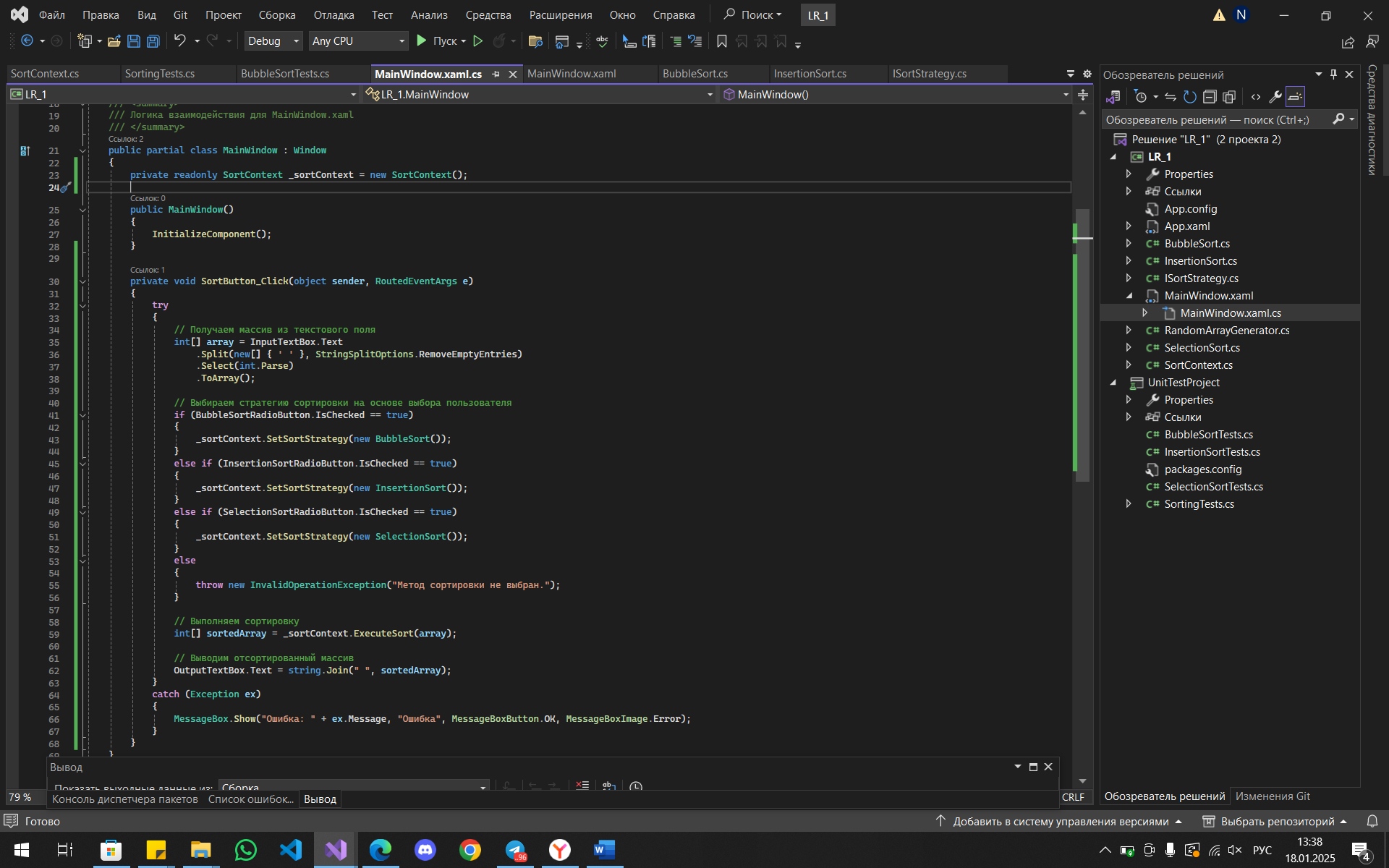


Рисунок 5 – Обработчики событий

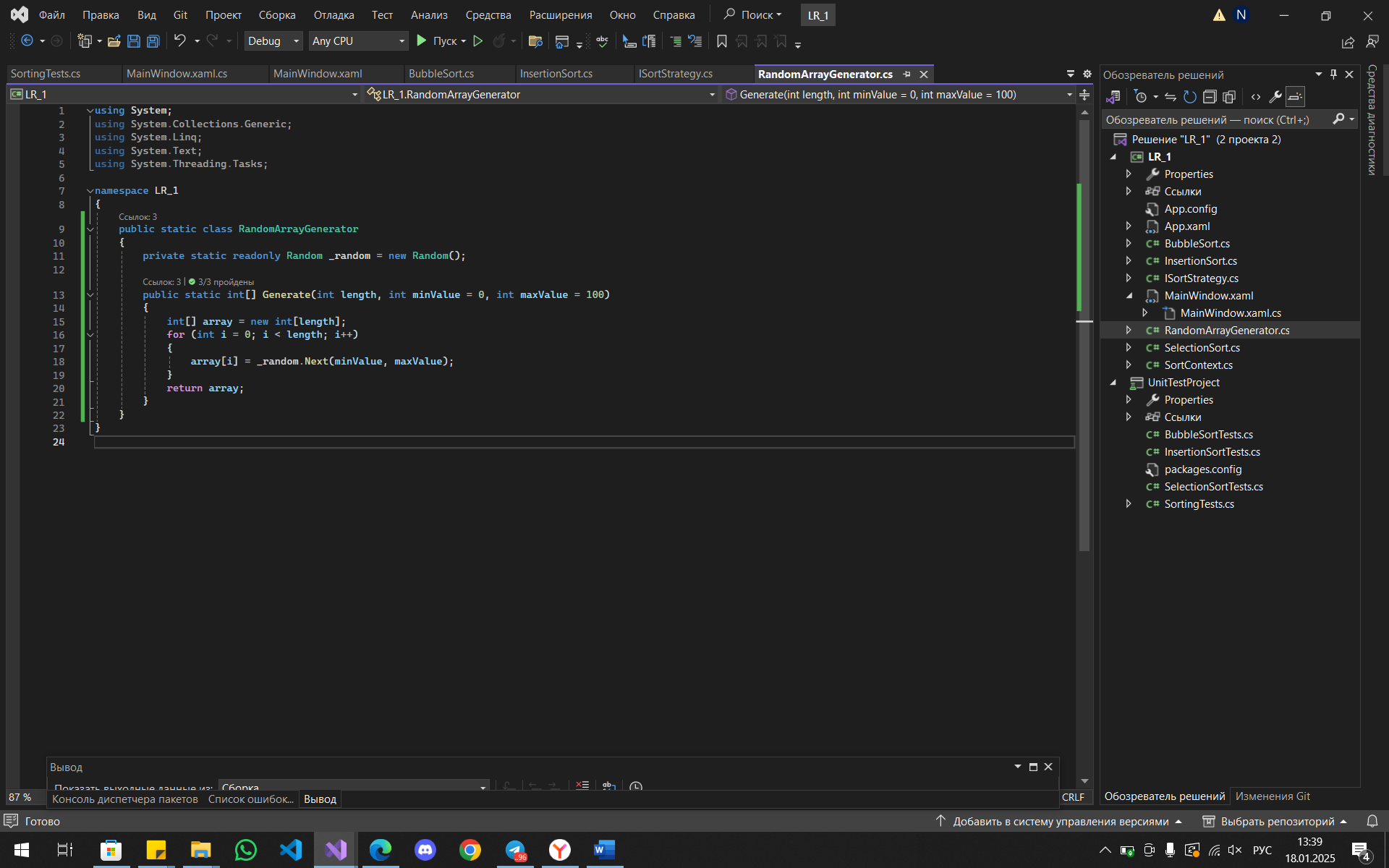


Рисунок 6 – Класс RandomArrayGenerator

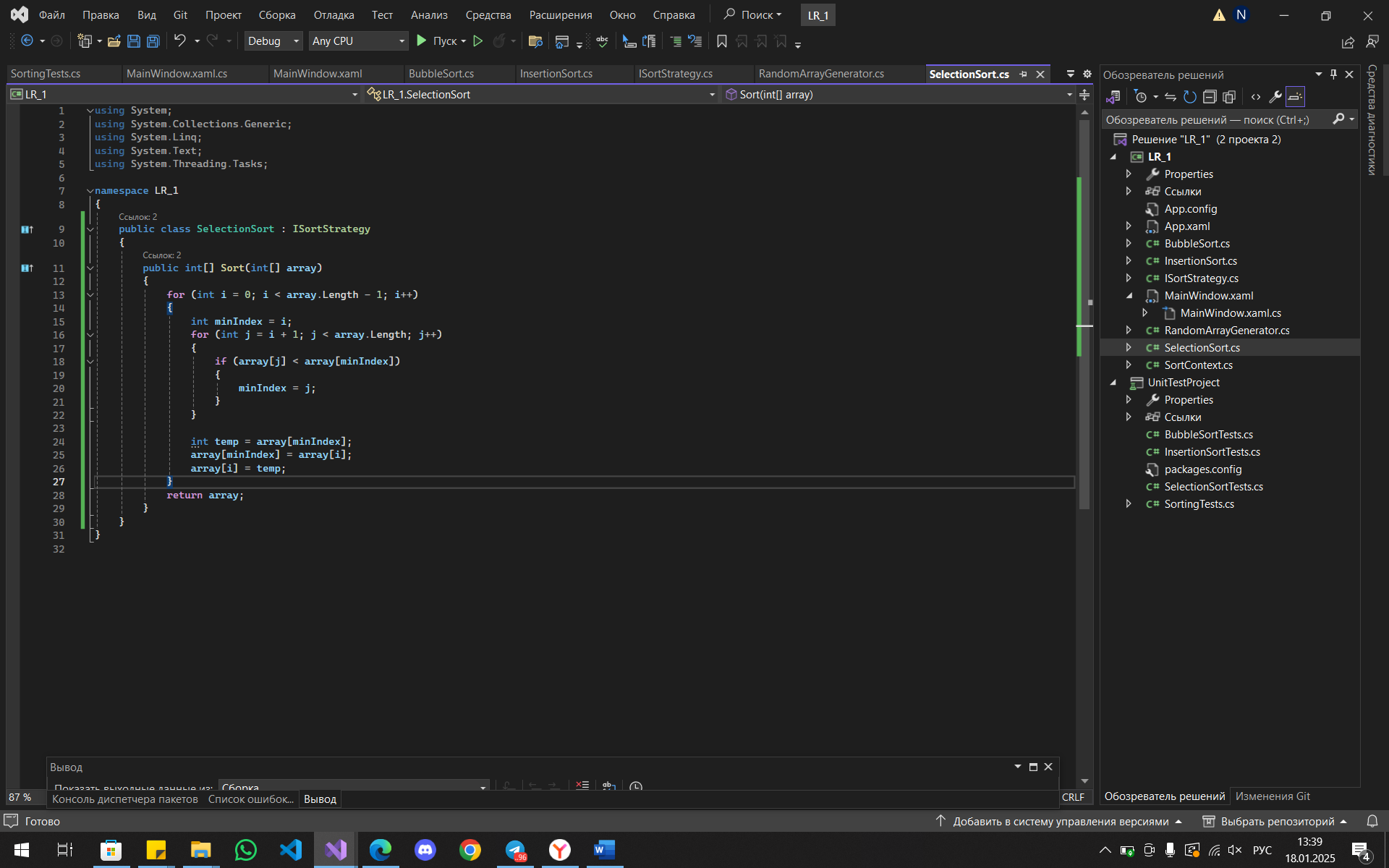


Рисунок 7 – Класс SelectionSort

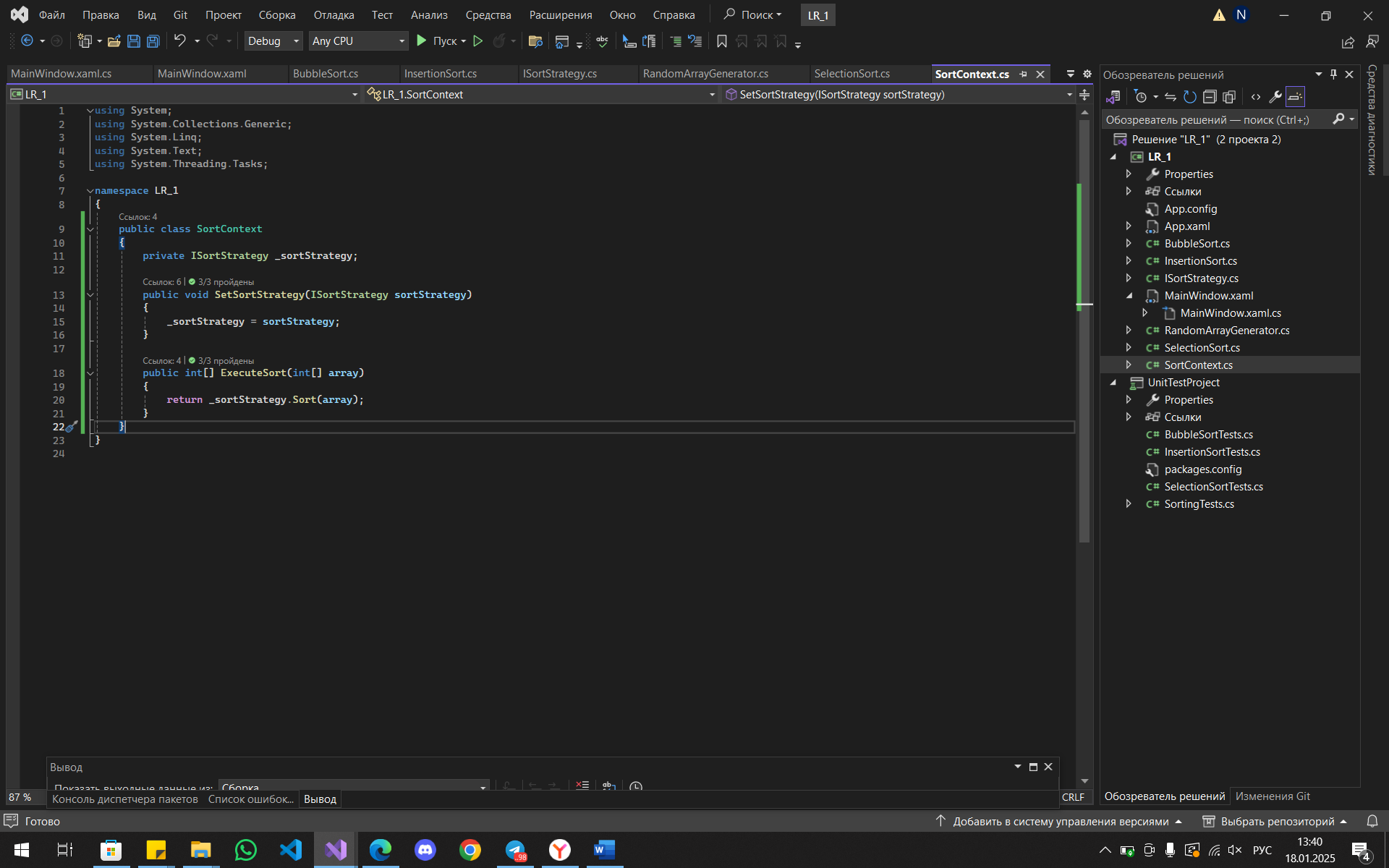


Рисунок 8 – Класс SortContext

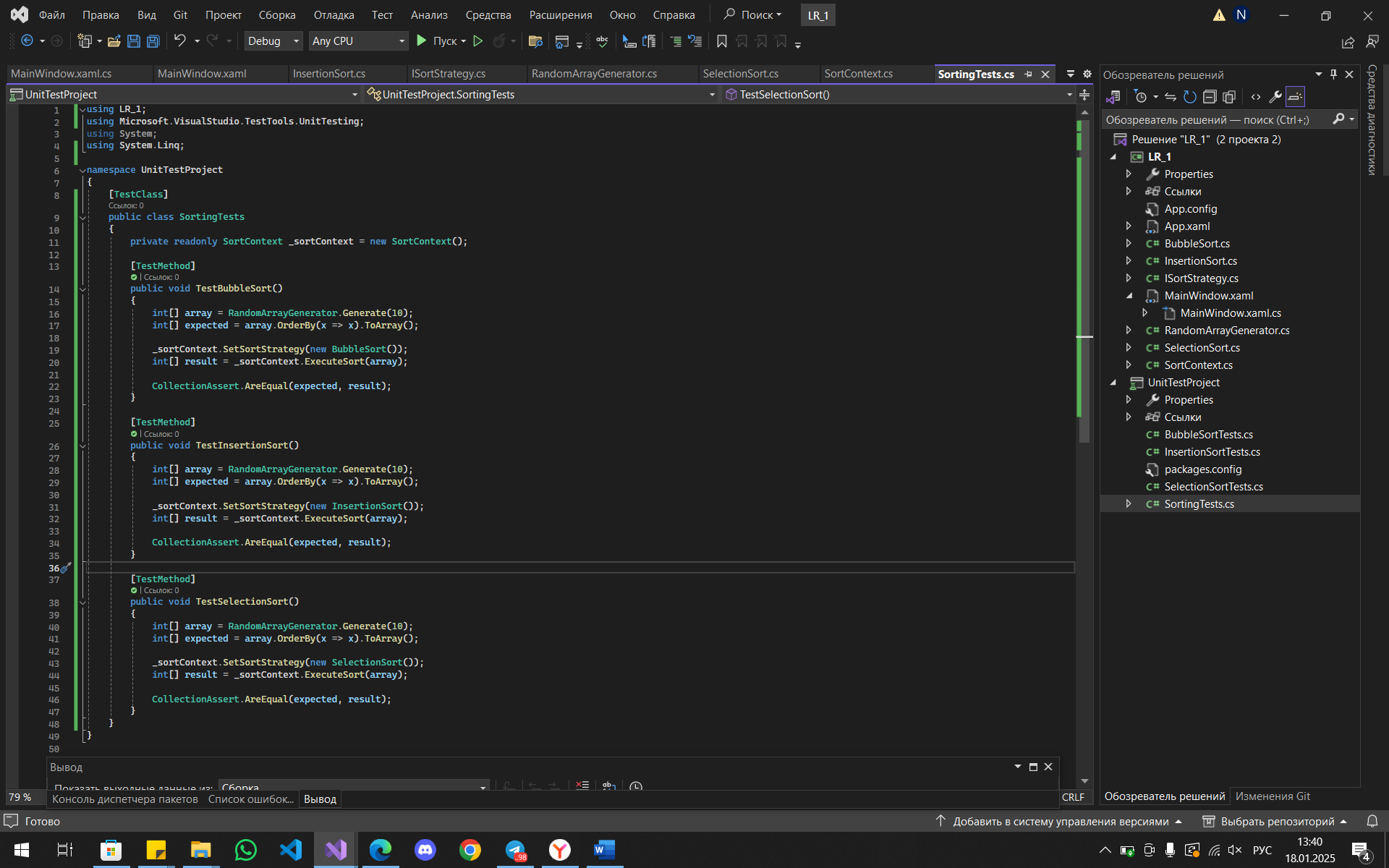


Рисунок 9 – Тест

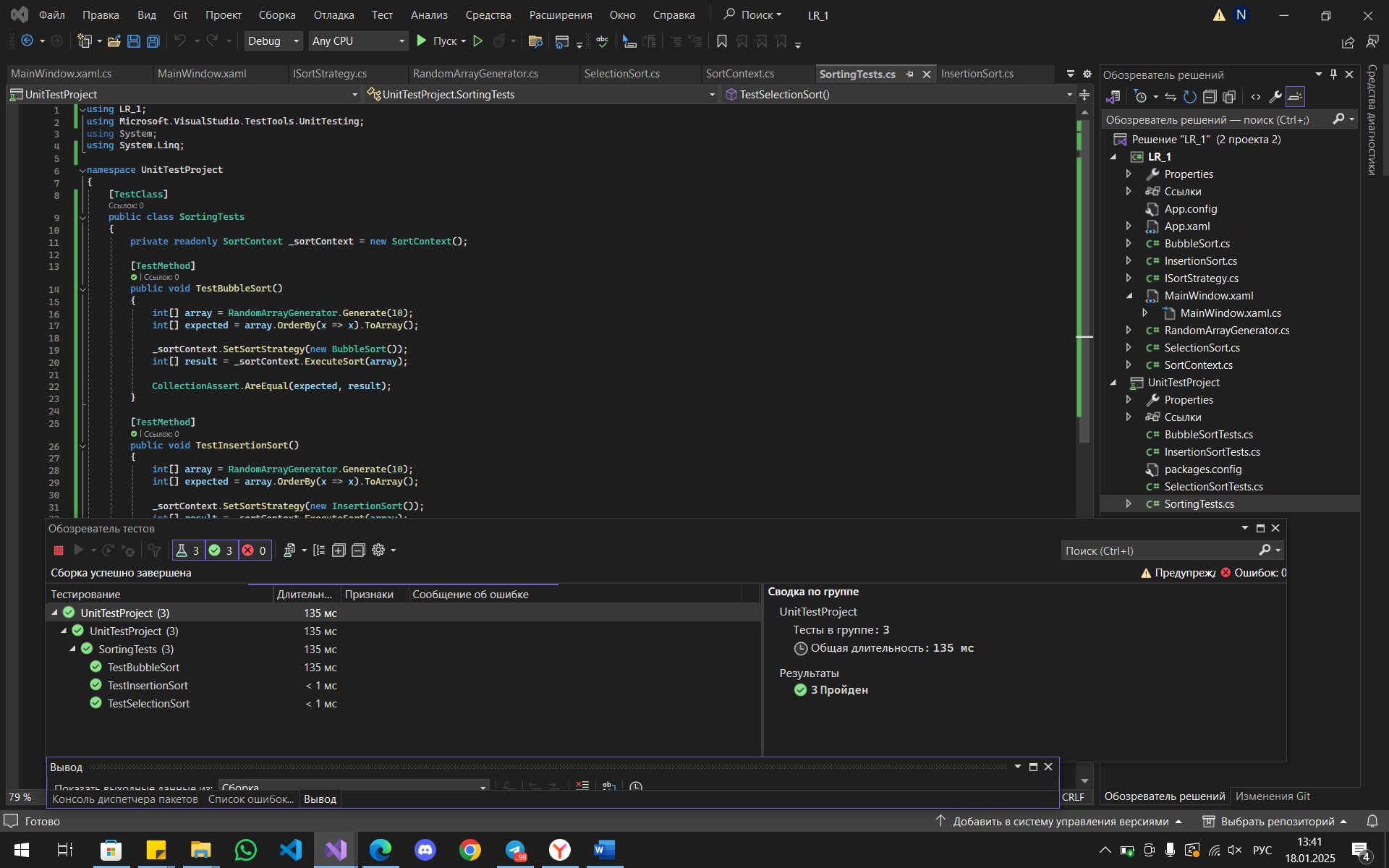


Рисунок 10 – Результат теста

**Листинг**

*SortingTests.cs*

using LR\_1;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Linq;

namespace UnitTestProject

{

[TestClass]

public class SortingTests

{

private readonly SortContext \_sortContext = new SortContext();

[TestMethod]

public void TestBubbleSort()

{

int[] array = RandomArrayGenerator.Generate(10);

int[] expected = array.OrderBy(x => x).ToArray();

\_sortContext.SetSortStrategy(new BubbleSort());

int[] result = \_sortContext.ExecuteSort(array);

CollectionAssert.AreEqual(expected, result);

}

[TestMethod]

public void TestInsertionSort()

{

int[] array = RandomArrayGenerator.Generate(10);

int[] expected = array.OrderBy(x => x).ToArray();

\_sortContext.SetSortStrategy(new InsertionSort());

int[] result = \_sortContext.ExecuteSort(array);

CollectionAssert.AreEqual(expected, result);

}

[TestMethod]

public void TestSelectionSort()

{

int[] array = RandomArrayGenerator.Generate(10);

int[] expected = array.OrderBy(x => x).ToArray();

\_sortContext.SetSortStrategy(new SelectionSort());

int[] result = \_sortContext.ExecuteSort(array);

CollectionAssert.AreEqual(expected, result);

}

}

}