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 в ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА № 1

по дисциплине

**ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ**

# Тема: «Модульные тестирования.»

Работу выполнил

Студент гр.4338

Бусов В.Р.

Принял

Преподаватель Максимов Р.С.

Казань 2025

**Цель работы**

Изучить возможность создания автоматических тестов, для модульного тестирования.

**Задание на лабораторную работу**

1. Разработать программу, содержащую графический интерфейс для работы с тремя модулями
2. В каждом модуле должен находиться один метод сортировки числовой последовательности
3. Разработать unit-тесты для каждого модуля
4. Оформить отчет

**Результат**

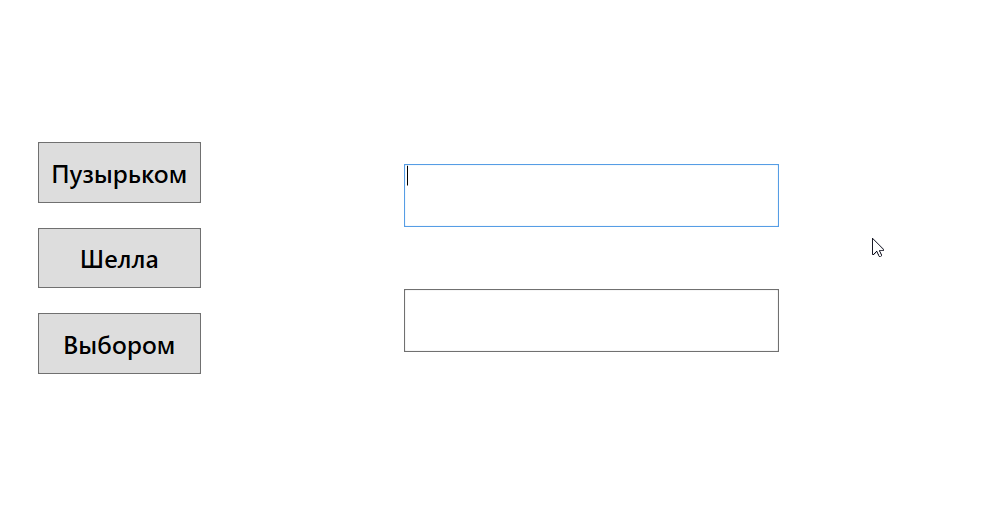


Рисунок 1 – интерфейс разработанной программы

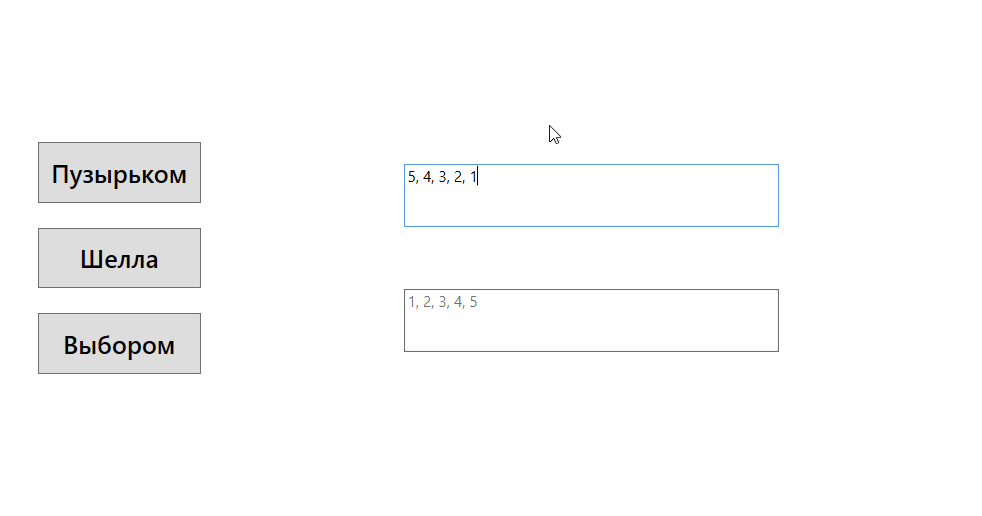


Рисунок 2 – Сортировка методом пузырька



Рисунок 3 – Сортировка методом Шелла

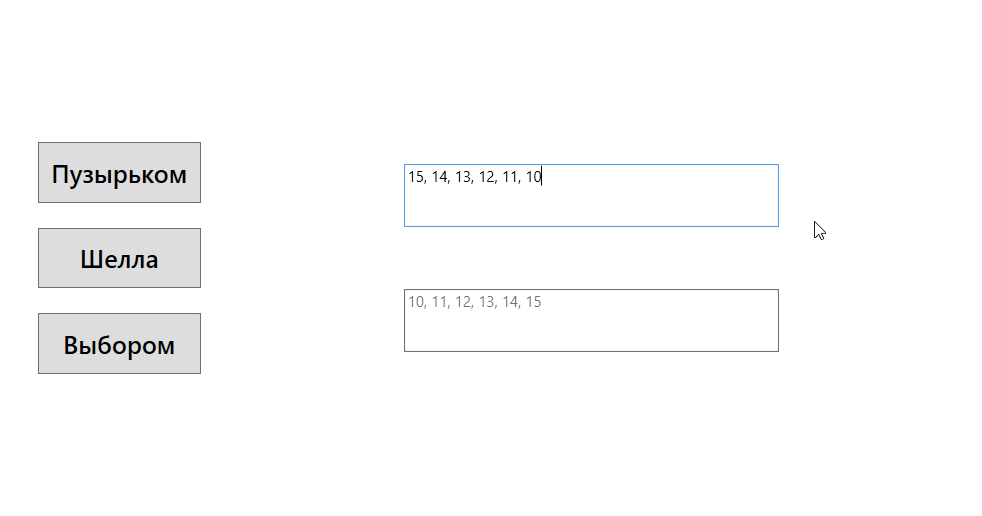


Рисунок 4 – Сортировка методом выбора

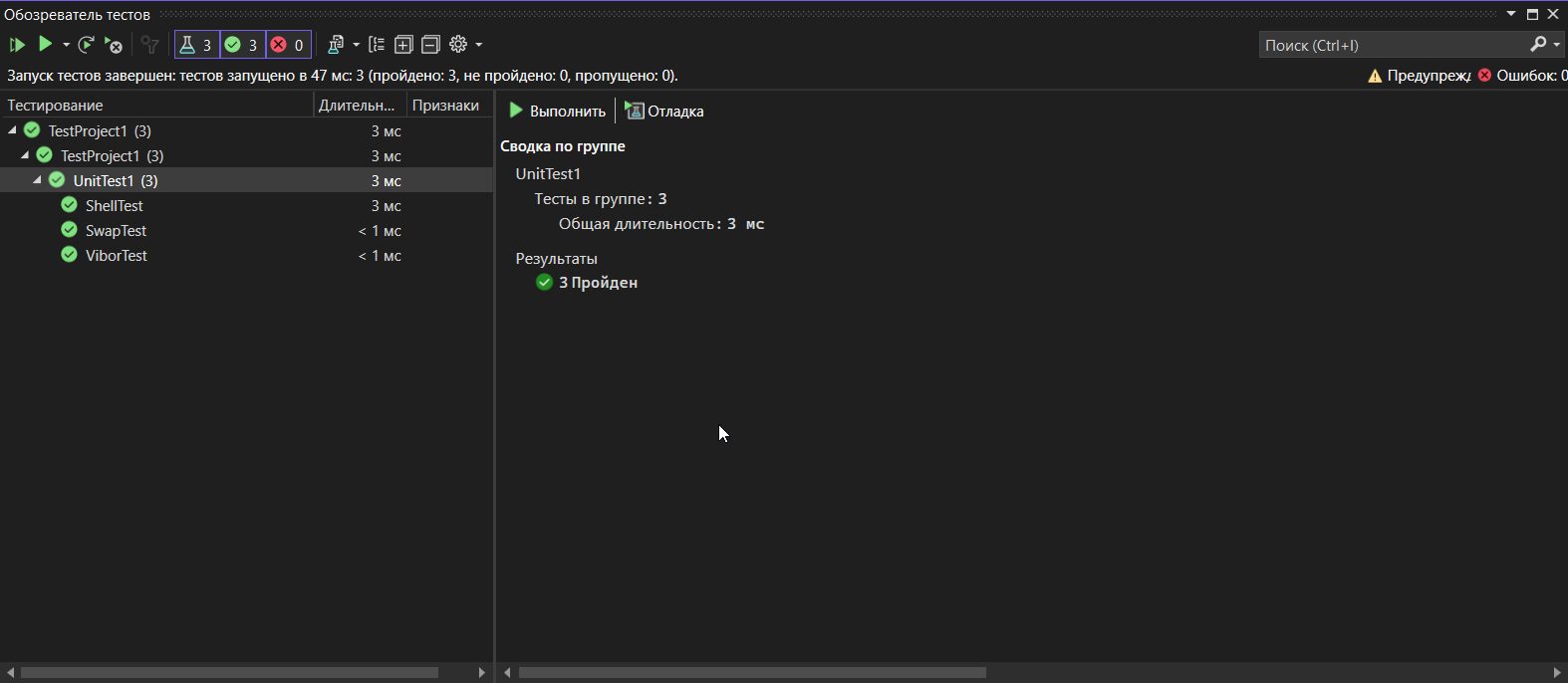


Рисунок 5 – Результат unit-тестирования

**Ответы на контрольные вопросы**

1. **Модульное тестирование**, или **юнит-тестирование** (англ. unit testing) — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы. Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок. Цель модульного тестирования — изолировать отдельные части программы и показать, что по отдельности эти части работоспособны
2. **Assert** в C# — это метод, используемый в юнит-тестах для проверки того, что условие (например, равенство значений) выполняется. Если условие не выполняется, тест считается неудачным, и выбрасывается исключение. Это позволяет убедиться, что код работает корректно.

**Assert.AreEqual** — это конкретный метод из библиотеки NUnit или MSTest, который используется для проверки, что два значения равны. Например, можно проверить, что результат выполнения функции совпадает с ожидаемым значением.

1. -**Assert.AreNotEqual**: проверяет, что два значения не равны. Если они равны, тест проваливается.

-**Assert.IsTrue**: проверяет, что условие истинно. Если условие ложно, тест проваливается.

-**Assert.IsFalse**: проверяет, что условие ложно. Если условие истинно, тест проваливается.

-**Assert.IsNull**: проверяет, что объект равен null. Если объект не равен null, тест проваливается.

-**Assert.IsNotNull**: проверяет, что объект не равен null. Если объект равен null, тест проваливается.

4. Возникла сложность при сравнении двух контейнеров List<T>.