|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
|  | Институт информационных технологий (ИТ) |
|  | Кафедра инструментального и прикладного программного обеспечения (ИиППО) |

|  |  |  |  |
| --- | --- | --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7** | | | |
| **по дисциплине** | | | |
| **«Тестирование и верификация программного обеспечения» на тему**  **«Разработка через тестирование»** | | | |
| Выполнил студент группы ИКБО-20-19 | | Московка А.А.. | |
|  | |  | |
| Принял | | Мельников Д.А. | |
| Практические работы выполнены | «\_14\_»\_декабря\_2021 г. | | Московка А.А..  (подпись студента) | |
| «Зачтено» | «\_\_»\_\_2021 г. | | Мельников Д.А.  (подпись руководителя) | |
|  |  | |  | |

Москва 2021

1. **Состав команды:**

1. Ильин Андрей Юрьевич

2. Городнов Сергей Алексеевич

3. Московка Артём Александрович

1. **Поведенческая система**

В качестве поведенческой системы был выбран цифровой автомат.

Реализовать конечный автомат Мили в виде класса. Начальным состоянием автомата является A. Если вызываемый метод не реализован для некоторого состояния, необходимо вызвать исключение.

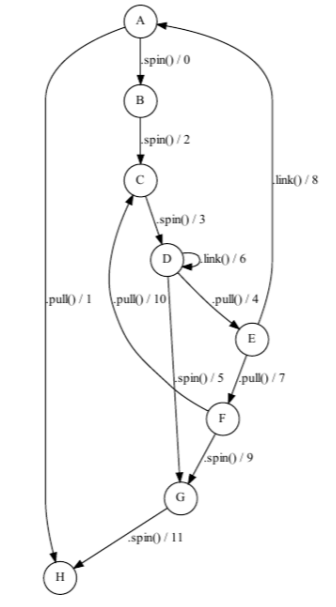


Рис. 1 – Автомат

1. **Созданная модель**

Описанный выше цифровой автомат был реализован на C#.

У автомата есть 3 метода pull (), spin (), link (), которые принимают определенные значения для переходов по узлам автомата.

*Листинг 1 - Интерфейс методов автомата*

interface Condition

    {

        int pull(Automat automat);

        int spin(Automat automat);

        int link(Automat automat);

    }

*Листинг 2 - Код класса автомата*

class Automat

    {

        public Condition condition;

        public int pull()

        {

            try

            {

                return this.condition.pull(this);

            }

            catch (NotImplementedException)

            {

                return -1;

            }

        }

        public int spin()

        {

            try

            {

                return this.condition.spin(this);

            }

            catch (NotImplementedException)

            {

                return -1;

            }

        }

        public int link()

        {

            try

            {

                return this.condition.link(this);

            }

            catch (NotImplementedException)

            {

                return -1;

            }

        }

    }

*Листинг 3 - Код узла А*

class A : Condition

    {

        public int link(Automat automat)

        {

            throw new NotImplementedException();

        }

        public int pull(Automat automat)

        {

            automat.condition = new H();

            return 1;

        }

        public int spin(Automat automat)

        {

            automat.condition = new B();

            return 0;

        }

    }

Остальные узлы были сформированы аналогичным образом в соответствии с изначальным условие построения автомата.

1. **Юнит-тесты**

Были сформировано 5 модульных тестов на проверку истинность построенного автомата.

*Листинг 4 – Запуск всех тестов*

static void Main(string[] args)

        {

            test1();

            Console.WriteLine("Тест 1 пройден\n");

            test2();

            Console.WriteLine("Тест 2 пройден\n");

            test3();

            Console.WriteLine("Тест 3 пройден\n");

            test4();

            Console.WriteLine("Тест 4 пройден\n");

            test5();

            Console.WriteLine("Тест 5 пройден\n");

            Console.WriteLine("Все тесты были пройдены успешно");

            Console.ReadLine();

        }

**Первый тест:** Запуск автомата с условием для перехода по узлам со значениями {0, 2, 3, 5, 11}

*Листинг 5 – Код первого теста*

static void test1()

        {

            Automat automat = new Automat();

            automat.condition = new A();

            List<int> conditions = new List<int>();

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            Debug.Assert(conditions.SequenceEqual(new List<int> { 0, 2, 3, 5, 11 }));

        }

**Второй тест:** Запуск автомата с условием для перехода по узлам со значением {1}

*Листинг 6 – Код второго теста*

static void test2()

        {

            Automat automat = new Automat();

            automat.condition = new A();

            List<int> conditions = new List<int>();

            conditions.Add(automat.pull());

            Debug.Assert(conditions.SequenceEqual(new List<int> { 1 }));

        }

Рис. 8 Листинг кода второго теста

**Третий тест:** Запуск автомата с условием для перехода по узлам со значениями {0, 2, 3, 6, 4, 8, 1}

*Листинг 7 – Код третьего теста*

static void test3()

        {

            Automat automat = new Automat();

            automat.condition = new A();

            List<int> conditions = new List<int>();

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            conditions.Add(automat.link());

            conditions.Add(automat.pull());

            conditions.Add(automat.link());

            conditions.Add(automat.pull());

            Debug.Assert(conditions.SequenceEqual(new List<int> { 0, 2, 3, 6, 4, 8, 1 }));

        }

**Четвертый тест:** Запуск автомата с условием для перехода по узлам со значениями {0, 2, 3, 4, 7, 10, 3, 5, 11}

*Листинг 8 – Код четвертого теста*

static void test4()

        {

            Automat automat = new Automat();

            automat.condition = new A();

            List<int> conditions = new List<int>();

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            conditions.Add(automat.pull());

            conditions.Add(automat.pull());

            conditions.Add(automat.pull());

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            Debug.Assert(conditions.SequenceEqual(new List<int> { 0, 2, 3, 4, 7, 10, 3, 5, 11 }));

        }

**Пятый тест:** Запуск автомата с условием для перехода по узлам со значениями {0, 2, 3, 4, 7, 9, 11}

*Листинг 9 – Код пятого теста*

static void test5()

        {

            Automat automat = new Automat();

            automat.condition = new A();

            List<int> conditions = new List<int>();

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            conditions.Add(automat.pull());

            conditions.Add(automat.pull());

            conditions.Add(automat.spin());

            conditions.Add(automat.spin());

            Debug.Assert(conditions.SequenceEqual(new List<int> { 0, 2, 3, 4, 7, 9, 11 }));

        }

**Вывод:** Программа работает корректно

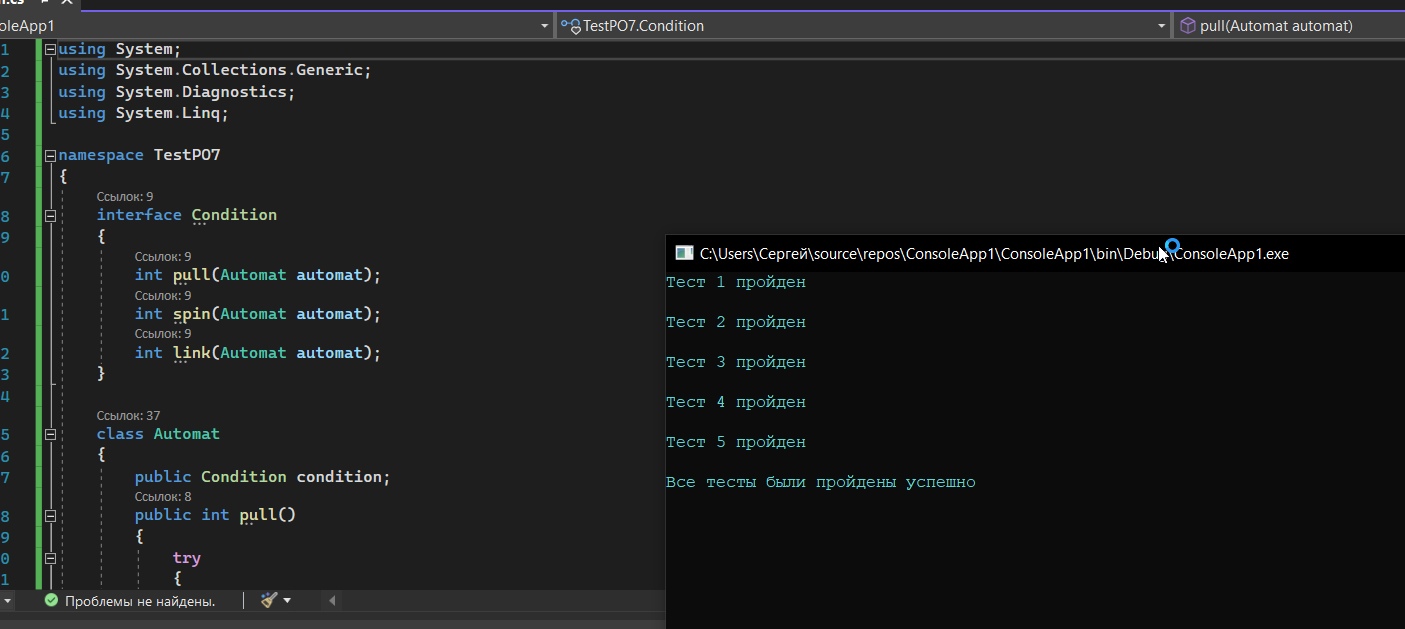


Рис. 15 – Результат теста

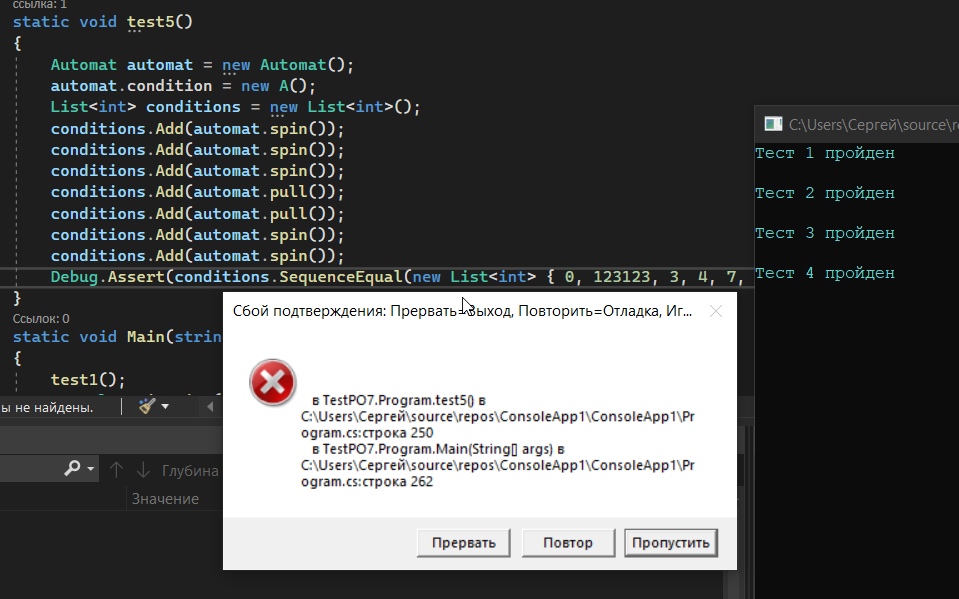


Рис.16 – Провал теста

**Вывод:** в ходе выполнения практической работы была разработана модель цифрового автомата и проведено ее тестирование, также были получены навыки в методологии тестирования с помощью модели.

## Список литературы

1. Алпатов, А. Н. Тестирование и отладка программного обеспечения : методические указания / А. Н. Алпатов. — Москва : РТУ МИРЭА, 2020. — 40 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/167578 (дата обращения: 24.09.2021). — Режим доступа: для авториз. пользователей.
2. Карпович, Е. Е. Методы тестирования и отладки программного обеспечения : учебник / Е. Е. Карпович. — Москва : МИСИС, 2020. — 136 с. — ISBN 978-5-907226-64-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/147965 (дата обращения: 24.09.2021). — Режим доступа: для авториз. пользователей.