

Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации Сибирский Государственный Университет
Телекоммуникаций и Информатики СибГУТИ

Кафедра прикладной математики и кибернетики

Лабораторная работа №8
по дисциплине “Современные технологии программирования”
Редактор р-ичных чисел

Выполнил:
Студент группы ИП-916
Меньщиков Д.А.

Работу проверил:
Агалаков А. А.

Новосибирск, 2022

Задание

1. Разработать и реализовать класс TEditor «Редактор p-ичных чисел», используя класс C++.
2. Протестировать каждую операцию, определенную на типе данных, используя средства модульного тестирования по критерию С0.
3. Если необходимо, предусмотрите возбуждение исключительных ситуаций.

Выполнение лабораторной работы

Код программы:

```
8 namespace lab8
9 {
10     public class PNumberEditor
11     {
12         private string currentNumber;
13
14         public PNumberEditor()
15         {
16             clear();
17         }
18
19         public bool IsNull()
20         {
21             return currentNumber.Split(" ")[0] == "0";
22         }
23
24         public void toggleNegative()
25         {
26             var parsed = currentNumber.Split(" ");
27             if(parsed[0][0] == '-')
28             {
29                 currentNumber.Remove(0, 1);
30             }
31             else
32             {
33                 currentNumber.Insert(0, "-");
34             }
35         }
36
37         public string appendNumber(int num)
38         {
39             currentNumber += num.ToString();
40             return currentNumber;
41         }
42
43         public string appendZero()
44         {
45             return appendNumber(0);
46         }
47     }
48 }
```

```
47
48     public string popNumberBack()
49     {
50         currentNumber = currentNumber.Substring(0, currentNumber.Length - 1);
51         return currentNumber;
52     }
53
54     public string clear()
55     {
56         currentNumber = "0";
57         return currentNumber;
58     }
59
60     public string getNumber()
61     {
62         return currentNumber;
63     }
64
65     public string setNumber(string number)
66     {
67         bool isValid = Regex.Match(number, "[0-9]+").Success;
68         if (!isValid)
69         {
70             throw new Exception("Number is not valid");
71         }
72         currentNumber = number;
73         return currentNumber;
74     }
75 }
76 }
```

Модульные тесты:

```
4 namespace TestPNumberEditor
5 {
6     [TestClass]
7     public class UnitTest1
8     {
9         [TestMethod]
10        public void TestPNumberEditor()
11        {
12            var editor = new PNumberEditor();
13
14            var actual = editor.getNumber();
15            var expected = "0";
16
17            Assert.AreEqual(expected, actual);
18        }
19
20        [TestMethod]
21        public void TestPNumberEditorIsNull()
22        {
23            var editor = new PNumberEditor();
24
25            var actual = editor.IsNull();
26            var expected = true;
27
28            Assert.AreEqual(expected, actual);
29        }
30
31        [TestMethod]
32        public void TestPNumberAppendNumber()
33        {
34            var editor = new PNumberEditor();
35
36            var expected = editor.appendNumber(5);
37            var actual = "05";
38
39            Assert.IsTrue(expected == actual);
40        }
41    }
```

```
41
42     [TestMethod]
43     public void TestPNumberAppendZero()
44     {
45         var editor = new PNumberEditor();
46
47         var expected = editor.appendZero();
48         var actual = "00";
49
50         Assert.IsTrue(expected == actual);
51     }
52
53     [TestMethod]
54     public void TestPNumberPopNumberBack()
55     {
56         var editor = new PNumberEditor();
57         editor.appendNumber(5);
58         editor.appendNumber(8);
59         editor.popNumberBack();
60
61         var expected = editor.getNumber();
62         var actual = "05";
63
64         Assert.IsTrue(expected == actual);
65     }
66
67     [TestMethod]
68     public void TestPNumberClear()
69     {
70         var editor = new PNumberEditor();
71         editor.appendNumber(5);
72         editor.appendNumber(8);
73         editor.clear();
74
75         var expected = editor.getNumber();
76         var actual = "0";
77
78         Assert.IsTrue(expected == actual);
79     }
```

```

80
81     [TestMethod]
82     public void TestPNumberGetNumber()
83     {
84         var editor = new PNumberEditor();
85
86         var expected = editor.getNumber();
87         var actual = "0";
88
89         Assert.IsTrue(expected == actual);
90     }
91
92     [TestMethod]
93     public void TestPNumberSetNumber()
94     {
95         var editor = new PNumberEditor();
96         editor.setNumber("12");
97
98         var expected = editor.getNumber();
99         var actual = "12";
100
101         Assert.IsTrue(expected == actual);
102     }
103 }
104 }

```

Обозреватель тестов



Тестирование	Длительн...	Признаки
▲ ✓ TestPNumberEditor (8)	7 мс	
▲ ✓ TestPNumberEditor (8)	7 мс	
▲ ✓ UnitTest1 (8)	7 мс	
✓ TestPNumberAppendNumber	7 мс	
✓ TestPNumberAppendZero	< 1 мс	
✓ TestPNumberClear	< 1 мс	
✓ TestPNumberEditor	< 1 мс	
✓ TestPNumberEditorIsNull	< 1 мс	
✓ TestPNumberGetNumber	< 1 мс	
✓ TestPNumberPopNumberBack	< 1 мс	
✓ TestPNumberSetNumber	< 1 мс	

