

**Московский государственный технический  
университет им. Н.Э. Баумана**

Факультет Радиотехнический  
Кафедра “Системы обработки информации и управления”

Курс «Парадигмы и конструкции языков программирования»

Отчет по домашнему заданию

Выполнил:  
студент группы РТ5-31Б:  
Длютров Тимофей  
Олегович  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Юрий  
Евгеньевич  
Подпись и дата:

Москва, 2025

## Постановка задачи

Разработать приложение windows forms на языке программирования C# с использованием принципа IOC/DI. Программа должна уметь зашифровывать и расшифровывать текстовый файл тремя разными способами и предоставлять отчёты о работе в форме журнала и в форме отдельных отчётов. Способы шифрования и предоставления отчётов могут выбираться пользователем в окне исполняемой программы.

## Текст программы

### Файл Program.cs:

```
1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Linq;
5  using System.Runtime.InteropServices.WindowsRuntime;
6  using System.Threading.Tasks;
7  using System.Windows.Forms;
8
9  namespace Decoder
10 {
11     //Объявление интерфейсов
12     public interface IDecoder
13     {
14         void BeginWork(string filename);
15         char DecodeNext();
16         void EndWork();
17     }
18     public interface IEncoder
19     {
20         void BeginWork(string filename);
21         void SendNext(char smb);
22         void EndWork();
23     }
24     public interface ILogger
25     {
26         void SendReport(string filename, string message);
27     }
```

```

28 //Создание классов
29 Ссылка: 2
30 public class ClassicDecoder : IDecoder
31 {
32     Ссылка: 2
33     public void BeginWork(string filename)
34     {
35         f = File.OpenRead(filename);
36     }
37
38     public char DecodeNext()
39     {
40         if (f.Position <= f.Length - 1)
41         {
42             char to_return = Convert.ToChar(f.ReadByte());
43             return to_return;
44         }
45         else
46         {
47             return '\0';
48         }
49     }
50
51     Ссылка: 2
52     public void EndWork()
53     {
54         f.Close();
55     }
56 }
57
58 public class Crypt1Decoder : IDecoder
59 {
60     Ссылка: 2
61     public void BeginWork(string filename)
62     {
63         f = File.OpenRead(filename);
64     }
65
66     Ссылка: 3
67     public char DecodeNext()
68     {
69         if (f.Position <= f.Length - 1)
70         {
71             char to_return = Convert.ToChar(f.ReadByte() - 4);
72             return to_return;
73         }
74         else
75         {
76             return '\0';
77         }
78     }
79
80     Ссылка: 2
81     public void EndWork()
82     {
83         f.Close();
84     }
85 }
86
87 Ссылка: 1
88 public class Crypt2Decoder : IDecoder
89 {
90     Ссылка: 2
91     public void BeginWork(string filename)
92     {
93         f = File.OpenRead(filename);
94     }
95 }

```

```

91  > public char DecodeNext()
92  {
93  >     if (f.Position <= f.Length - 1)
94  {
95      char to_return = Convert.ToChar(f.ReadByte() - ((f.Position - 1) % 4));
96      return to_return;
97  }
98  >     else
99  {
100     return '\0';
101     }
102 }
103
104 > Ссылка: 2
105 public void EndWork()
106 {
107     f.Close();
108 }
109
110 > Ссылка: 2
111 public class ClassicEncoder : IEncoder
112 {
113     FileStream f;
114     > Ссылка: 2
115     public void BeginWork(string filename)
116     {
117         f = File.OpenWrite(filename);
118     }
119
120     public void SendNext(char smb)
121     {
122         f.WriteByte(Convert.ToByte(smb));
123     }
124
125     > Ссылка: 2
126     public void EndWork()
127     {
128         f.Close();
129     }
130 }
131
132 > Ссылка: 1
133 public class Crypt1Encoder : IEncoder
134 {
135     FileStream f;
136
137     > Ссылка: 2
138     public void BeginWork(string filename)
139     {
140         f = File.OpenWrite(filename);
141     }
142
143     > Ссылка: 2
144     public void SendNext(char smb)
145     {
146         f.WriteByte(Convert.ToByte(Convert.ToByte(smb) + 4));
147     }
148
149     > Ссылка: 2
150     public void EndWork()
151     {
152         f.Close();
153     }
154 }
155
156 > Ссылка: 2
157 public class Crypt2Encoder : IEncoder
158 {
159     FileStream f;
160     int current_pos = 0;
161     > Ссылка: 2
162     public void BeginWork(string filename)
163     {

```

```

154         f = File.OpenWrite(filename);
155     }
156
157     Ссылка: 2
158     public void SendNext(char smb)
159     {
160         f.WriteByte(Convert.ToByte(Convert.ToByte(smb) + (current_pos % 4)));
161         current_pos++;
162     }
163
164     Ссылка: 2
165     public void EndWork()
166     {
167         f.Close();
168     }
169
170     Ссылка: 2
171     public class LogLogger : ILogger
172     {
173         Ссылка: 5
174         public void SendReport(string path, string message)
175         {
176             string filename = path + "/MainLog.txt";
177             if (!File.Exists(filename))
178             {
179                 File.Create(filename).Close();
180                 File.AppendAllText(filename, "Some programm log");
181             }
182             File.AppendAllText(filename, "\n" + message + " --- " + DateTime.Now.ToString());
183         }
184     }
185
186     Ссылка: 1
187     public class ReportLogger : ILogger
188     {
189         Ссылка: 5
190         public void SendReport(string path, string message)
191         {
192             string filename = path + "/" + "Report " + DateTime.Now.ToString().Replace(':', '.').txt";
193             File.Create(filename).Close();
194             File.AppendAllText(filename, "Programm execution report\n");
195             File.AppendAllText(filename, DateTime.Now.ToString() + "\n");
196             File.AppendAllText(filename, "\nMessage:\n");
197             File.AppendAllText(filename, message);
198         }
199     }
200
201     //Основная программа
202     Ссылка: 0
203     internal static class Program
204     {
205         /// <summary>
206         /// Главная точка входа для приложения.
207         /// </summary>
208         [STAThread]
209         Ссылка: 0
210         static void Main()
211         {
212             Application.EnableVisualStyles();
213             Application.SetCompatibleTextRenderingDefault(false);
214             Application.Run(new Form1());
215         }
216     }
217 }

```

## Файл Form1.cs:

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.IO;
7  using System.Linq;
8  using System.Text;
9  using System.Threading.Tasks;
10 using System.Windows.Forms;
11
12 namespace Decoder
13 {
14     Ссылка: 3
15     public partial class Form1 : Form
16     {
17         ILogger logger = new LogLogger();
18         IDecoder decoder = new ClassicDecoder();
19         IEncoder encoder = new ClassicEncoder();
20
21         Ссылка: 1
22         public Form1()
23         {
24             InitializeComponent();
25         }
26
27         Ссылка: 1
28         private void button1_Click(object sender, EventArgs e)
29         {
30             if (openFileDialog1.ShowDialog() == DialogResult.OK)
31             {
32                 decoder.BeginWork(openFileDialog1.FileName);
33                 char nexts = decoder.DecodeNext();
34                 richTextBox1.Clear();
35                 while (nexts != '\0')
36                 {
37                     richTextBox1.Text += nexts;
38                     nexts = decoder.DecodeNext();
39                 }
40                 richTextBox1.Update();
41                 decoder.EndWork();
42                 logger.SendReport(Application.StartupPath, "Чтение файла завершено\n");
43             }
44             else
45             {
46                 logger.SendReport(Application.StartupPath, "Файл не найден\n");
47             }
48         }
49
50         Ссылка: 1
51         private void button1_Click_1(object sender, EventArgs e)
52         {
53             if (saveFileDialog1.ShowDialog() == DialogResult.OK)
54             {
55                 encoder.BeginWork(saveFileDialog1.FileName);
56                 for (int i = 0; i < richTextBox1.Text.Length; ++i)
57                 {
58                     encoder.SendNext(richTextBox1.Text[i]);
59                 }
60                 encoder.EndWork();
61                 logger.SendReport(Application.StartupPath, "Запись файла завершена\n");
62             }
63             else
64             {
65                 logger.SendReport(Application.StartupPath, "Файл не найден\n");
66             }
67         }
68
69         Ссылка: 1
70         private void Form1_Load(object sender, EventArgs e)
71         {
72             //Добавление кодеков
73             codeccb.Items.Add("Последовательный");
74             codeccb.Items.Add("Шифр1");
75             codeccb.Items.Add("Шифр2");
76         }
77     }
78 }
```

```

70 //Добавление логов
71 loggercb.Items.Add("Журнал");
72 loggercb.Items.Add("Отчёт");
73 }
74
75 Ссылка: 1
76 private void codeccb_SelectedValueChanged(object sender, EventArgs e)
77 {
78     switch (codeccb.SelectedIndex)
79     {
80         case 0: { decoder = new ClassicDecoder(); encoder = new ClassicEncoder(); break; }
81         case 1: { decoder = new Crypt1Decoder(); encoder = new Crypt1Encoder(); break; }
82         case 2: { decoder = new Crypt2Decoder(); encoder = new Crypt2Encoder(); break; }
83     }
84 }
85
86 Ссылка: 1
87 private void loggercb_SelectedIndexChanged(object sender, EventArgs e)
88 {
89     switch (loggercb.SelectedIndex)
90     {
91         case 0: { logger = new LogLogger(); break; }
92         case 1: { logger = new ReportLogger(); break; }
93     }
94 }
95

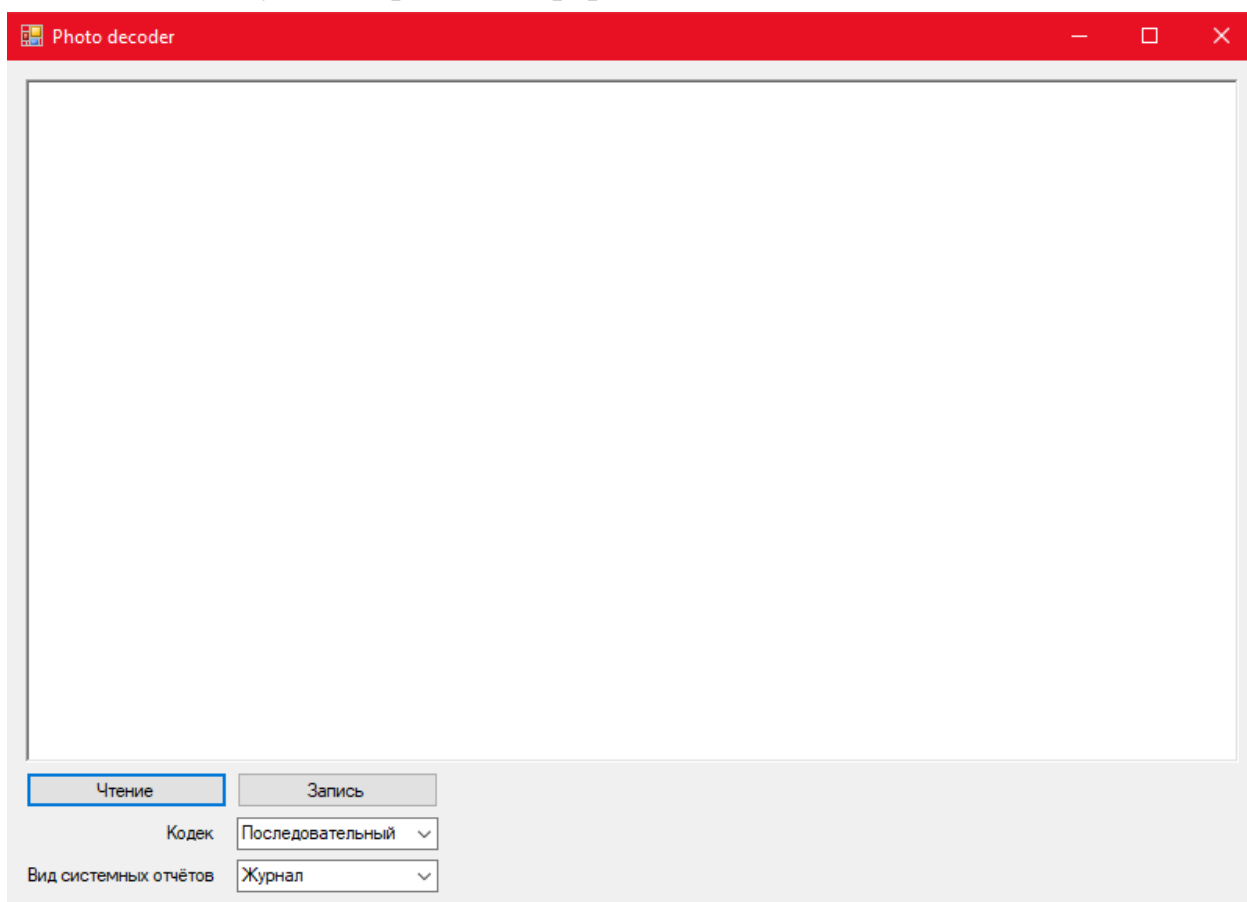
```

### Форма Form1.cs (режим конструктора):

The screenshot shows the Visual Studio Form Designer for a form named 'Form1.cs' in design mode. The window title is 'Photo decoder'. The form layout includes a large empty rectangular area at the top, likely for displaying a photo. Below this area are two buttons: 'Чтение' (Read) and 'Запись' (Write). At the bottom of the form, there are two dropdown menus: 'Кодек' (Codec) with 'Последовательный' (Sequential) selected, and 'Вид системных отчётов' (System report view) with 'Журнал' (Log) selected.

## Проверка работы программы

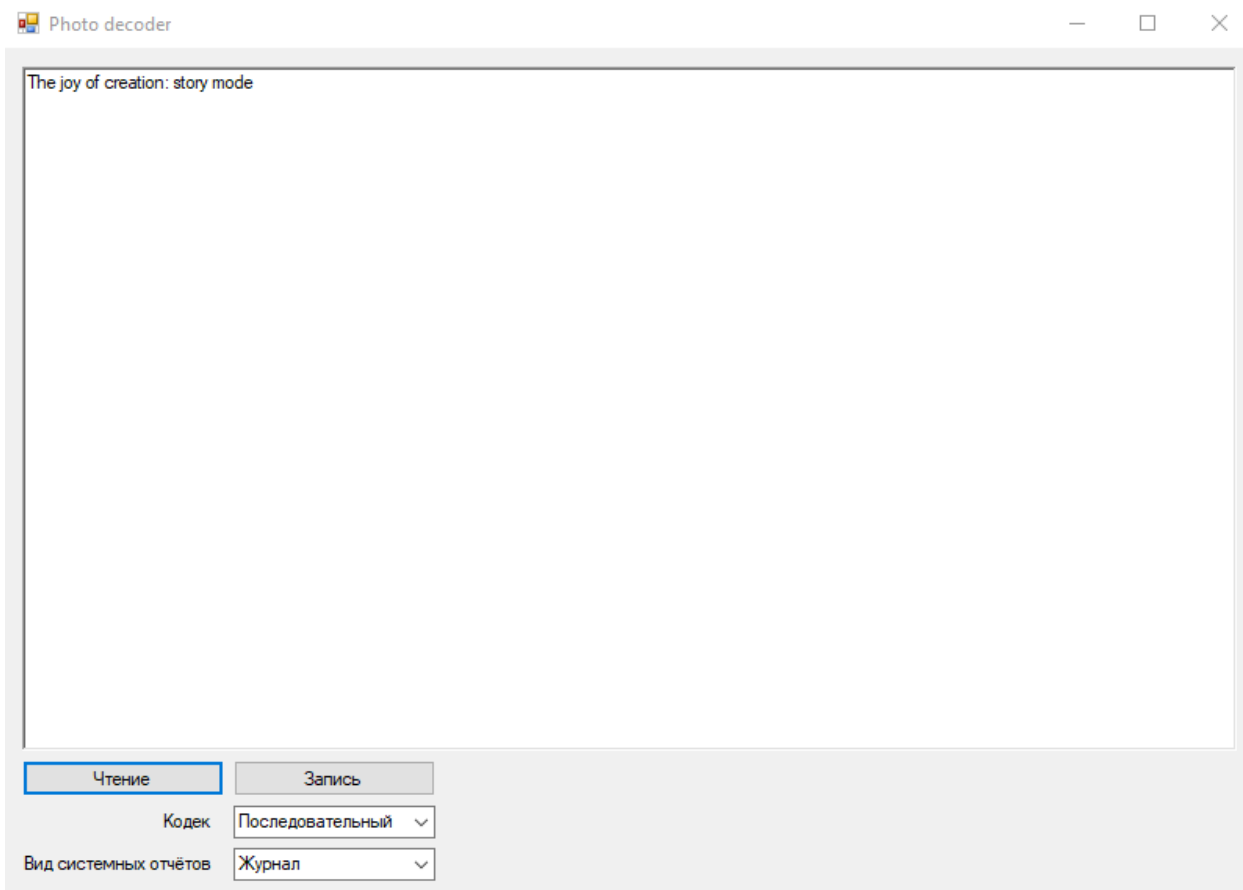
После запуска открывается форма:



Файл, зашифрованный последовательным кодеком:



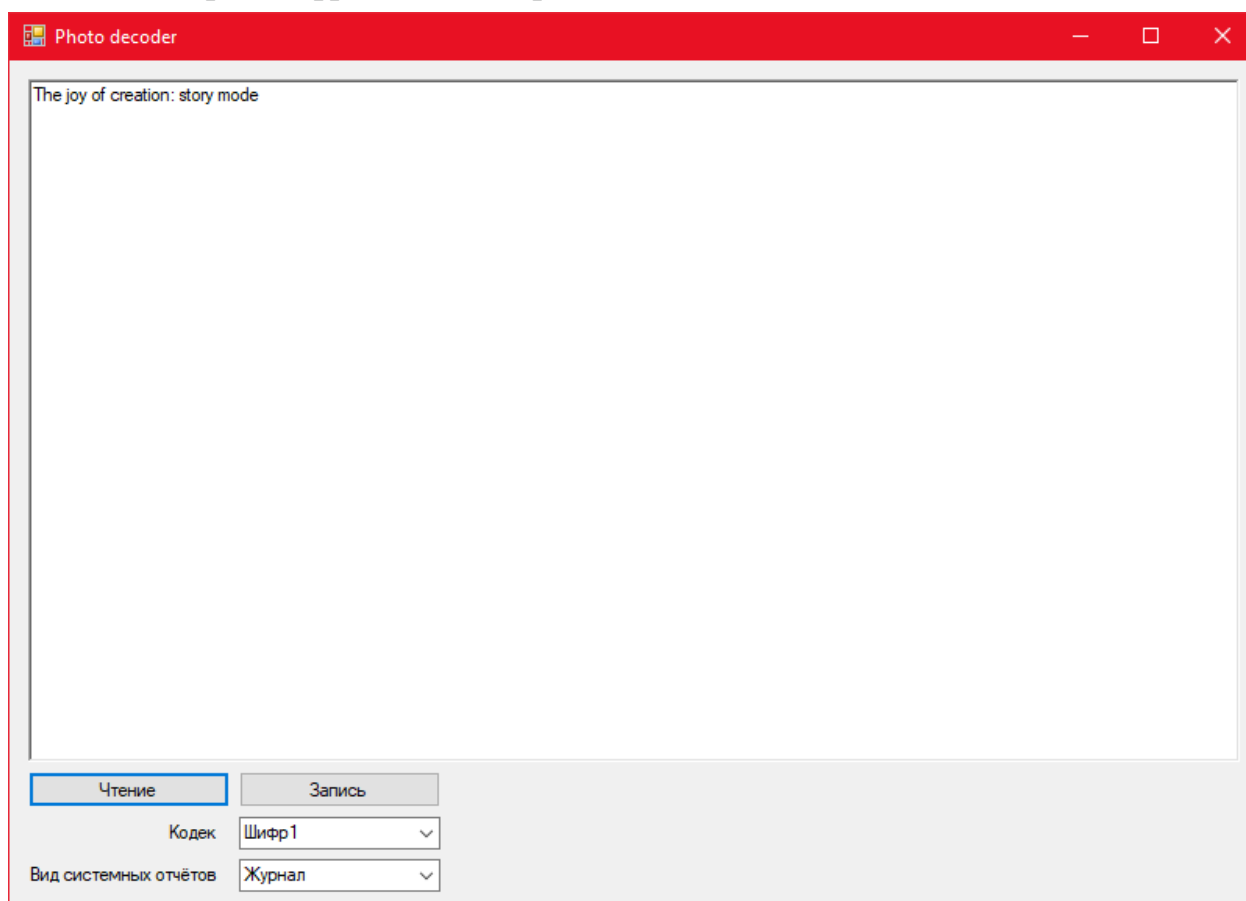
Файл, расшифрованный последовательным кодеком:



Файл, зашифрованный первым кодеком:



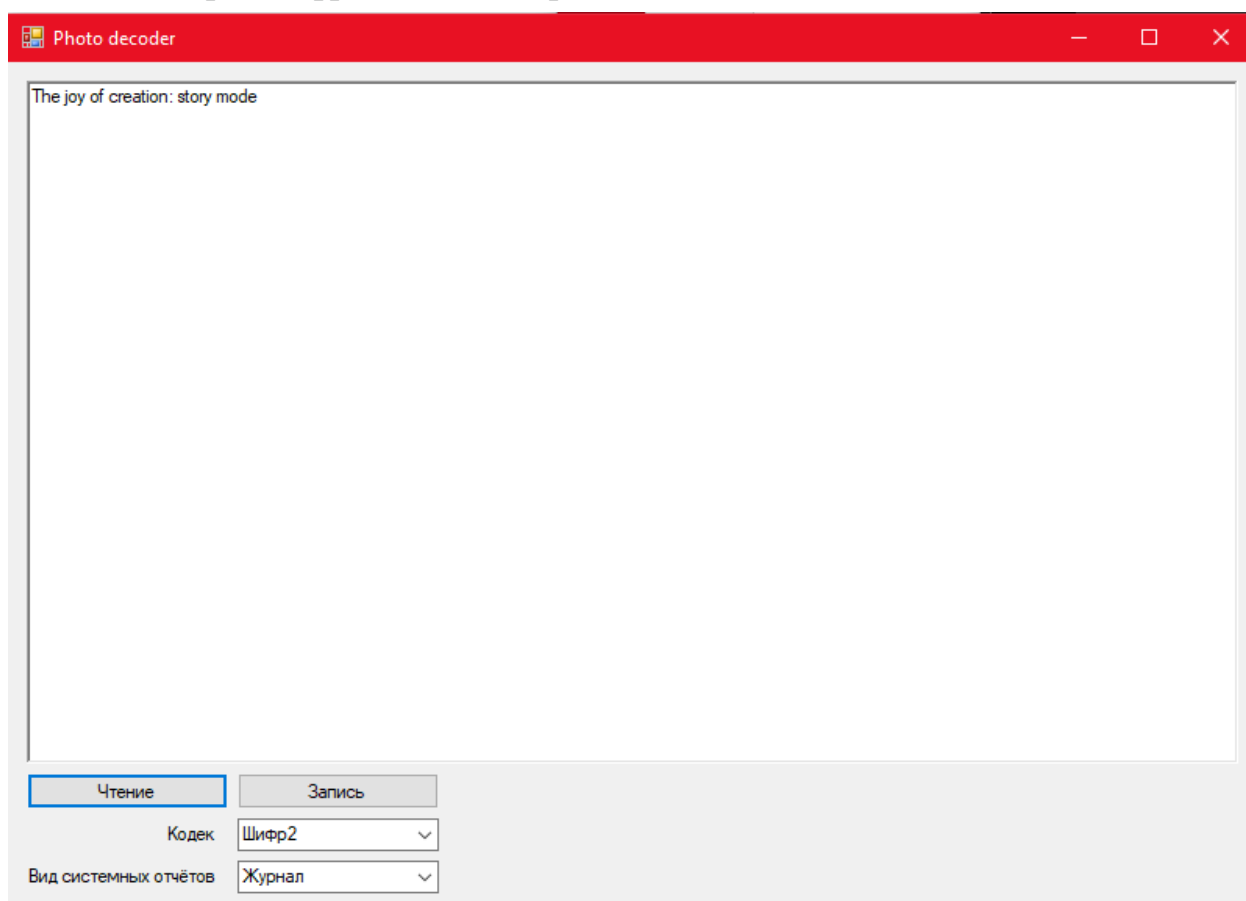
Файл, расшифрованный первым кодеком:



Файл, зашифрованный вторым кодеком:



Файл, расшифрованный вторым кодеком:



Журнал после всех действий:



Отчёт о проделанном действии:

