

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО

Старший преподаватель программной
инженерии факультета компьютерных
наук

_____ Л.В. Дворянский
« ____ » _____ 2018 г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия»

_____ В.В. Шилов
« ____ » _____ 2018 г.

**Обучающий визуализатор с тестами: основные понятия теорий множеств и
отношений.**

Текст программы

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.15-01 12 01-1-ЛУ

Исполнитель

Студент группы БПИ173

_____ /К.Г. Кожакин/

« ____ » _____ 2018 г.

Москва 2018

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

УТВЕРЖДЕН

RU.17701729.04.15-01 12 01-1-ЛУ

Обучающий визуализатор с тестами: основные понятия теорий множеств и отношений.

Текст программы

RU.17701729.04.15-01 51 01-1

Листов 42

Инв. № подл	Подп. и дата
Взам. инв. №	Инв. № дубл.
Подп. и дата	

Москва 2018

СОДЕРЖАНИЕ

1. ТЕКСТ ПРОГРАММЫ.....	3
1.1. Класс Login.cs.....	3
1.2. Класс ChangeInfo.cs.....	4
1.3. Класс DBMySQLUtils.cs.....	6
1.4. Класс DBUtils.cs.....	7
1.5. Класс General.cs.....	8
1.6. Классы Graph.cs.....	12
1.6.1. Класс Graph.....	12
1.6.2. Класс Vertice.....	12
1.6.3. Класс Edge.....	13
1.6.4. Класс Origin.....	13
1.6.5. Класс Destination.....	13
1.7. Класс GraphLayout.xaml.cs.....	14
1.8. Класс Program.cs.....	15
1.9. Класс Reg.cs.....	16
1.10. Класс Results.cs.....	18
1.11. Классы Test.cs.....	21
1.11.1. Класс Test.....	22
1.11.2. Класс Task.....	23
1.12. Класс TestForm.cs.....	23
2. Текст скрипта базы данных MySQL.....	39
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	41
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	42

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

1. ТЕКСТ ПРОГРАММЫ

1.1. Класс Login.cs:

```

1. using MySql.Data.MySqlClient;
2. using System;
3. using System.Data.Common;
4. using System.IO;
5. using System.Windows.Forms;
6.
7. namespace Sets_RelationsLessons
8. {
9.     public partial class Login : Form
10.    {
11.        MySqlConnection conn;
12.
13.        public Login()
14.        {
15.            InitializeComponent();
16.            conn = DBUtils.GetDBConnection();
17.            if (Application.OpenForms.Count > 10)
18.            {
19.                Application.Restart();
20.            }
21.        }
22.        /// <summary>
23.        /// Обработчик события нажатия кнопки "Зарегистрироваться"(Перейти на форму рег
    истрации)
24.        /// </summary>
25.        private void Registration_Click(object sender, EventArgs e)
26.        {
27.            Reg_RegForm = new Reg(this);
28.            _RegForm.Show();
29.            this.Hide();
30.        }
31.        /// <summary>
32.        /// Обработчик события нажатия кнопки "Вход"
33.        /// </summary>
34.        private void Start_Click(object sender, EventArgs e)
35.        {
36.            bool flag = false;
37.            try
38.            {
39.                conn.Open();
40.
41.                using (MySqlCommand cmd = new MySqlCommand("Select idUsers, Login, Pass
    word from users", conn))
42.                {
43.                    using (DbDataReader reader = cmd.ExecuteReader())
44.                    {
45.                        if (reader.HasRows)
46.                        {
47.                            while (reader.Read())
48.                            {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

49.         if (Login_value.Text == reader.GetString(reader.GetOrdinal("Login")))
50.         {
51.             if (Password_value.Text.GetHashCode().ToString() ==
52.                 reader.GetString(reader.GetOrdinal("Password")))
53.             {
54.                 Properties.Settings.Default.login = Login_value
55.                 .Text;
56.                 Properties.Settings.Default.Save();
57.                 General _GeneralForm = new General();
58.                 _GeneralForm.Show();
59.                 this.Hide();
60.                 flag = true;
61.                 break;
62.             }
63.             else
64.             {
65.                 throw new Exception("Неверный пароль.");
66.             }
67.             if (!flag) throw new Exception("Такой логин не существует.");
68.         }
69.     };
70. }
71. }
72. }
73. catch (Exception ex)
74. {
75.     MessageBox.Show(ex.Message);
76. }
77. finally
78. {
79.     conn.Close();
80.     conn.Dispose();
81. }
82. }
83. /// <summary>
84. /// Обработчик события закрытия формы
85. /// </summary>
86. private void Login_FormClosing(object sender, FormClosingEventArgs e)
87. {
88.     foreach(string name in Directory.GetFiles(Directory.GetCurrentDirectory() +
89.         @"\Lessons\", "tmp*"))
90.     {
91.         File.Delete(name);
92.     }
93. }

```

1.2. Класс ChangeInfo.cs:

```

1. using MySql.Data.MySqlClient;
2. using System;
3. using System.Data.Common;
4. using System.Windows.Forms;
5.
6. namespace Sets_RelationsLessons

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

7. {
8.     public partial class ChangeInfo : Form
9.     {
10.         MySqlConnection conn;
11.         General _GenForm;
12.
13.         public ChangeInfo(General GenForm)
14.         {
15.             _GenForm = GenForm;
16.             InitializeComponent();
17.             conn = DBUtils.GetDBConnection();
18.         }
19.         /// <summary>
20.         /// Обработчик события нажатия на кнопку "Отмена"
21.         /// </summary>
22.         private void Cancel_Click(object sender, EventArgs e)
23.         {
24.             this.Close();
25.         }
26.         /// <summary>
27.         /// Обработчик события нажатия на кнопку "Изменить"
28.         /// </summary>
29.         private void Change_Click(object sender, EventArgs e)
30.         {
31.             int id = 0;
32.             try
33.             {
34.                 conn.Open();
35.                 if (Login_value.Text.Length < 3) throw new Exception("Слишком короткий
логин(должен содержать 3 или больше символов).");
36.                 if (Login_value.Text.Length > 15) throw new Exception("Слишком длинный
логин(должен содержать 15 или меньше символов).");
37.                 if (Login_value.Text.IndexOf(' ') > -
1) throw new Exception("Логин не должен содержать пробелы.");
38.                 if (Password_value2.Text.Length < 3) throw new Exception("Слишком корот
кий пароль(должен содержать 3 или больше символов).");
39.                 if (Password_value2.Text.Length > 20) throw new Exception("Слишком длин
ный пароль(должен содержать 20 или меньше символов).");
40.                 if (Password_value2.Text.IndexOf(' ') > -
1) throw new Exception("Пароль не должен содержать пробелы.");
41.
42.                 using (MySqlCommand cmd = new MySqlCommand("Select idUsers, Login, Pass
word from users", conn))
43.                 {
44.                     using (DbDataReader reader = cmd.ExecuteReader())
45.                     {
46.                         if (reader.HasRows)
47.                         {
48.                             while (reader.Read())
49.                             {
50.                                 if (Properties.Settings.Default.login == reader.GetStri
ng(reader.GetOrdinal("Login")))
51.                                     if (Password_value.Text.GetHashCode().ToString() !=
reader.GetString(reader.GetOrdinal("Password")))
52.                                         throw new Exception("Неверный пароль.");
53.                                 else id = reader.GetInt32(reader.GetOrdinal("idUser
s"));

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

54.         if (Login_value.Text == reader.GetString(reader.GetOrdinal("Login")))
55.             throw new Exception("Этот логин уже занят.");
56.         }
57.     }
58. }
59. }
60.
61.     using (MySqlCommand cmd = new MySqlCommand("Update users set Login = @login, Password = @password where idUsers = @id", conn))
62.     {
63.         cmd.Parameters.AddWithValue("@login", Login_value.Text);
64.         cmd.Parameters.AddWithValue("@id", id);
65.         cmd.Parameters.AddWithValue("@password", Password_value2.Text.GetHasCode());
66.
67.         cmd.ExecuteNonQuery();
68.     }
69.     MessageBox.Show("Изменения успешно сохранены!");
70.
71.     Properties.Settings.Default.login = Login_value.Text;
72.     Properties.Settings.Default.Save();
73.     _GenForm.Text = "Добро пожаловать, " + Login_value.Text;
74.     _GenForm.Show();
75.     conn.Close();
76.     this.Close();
77. }
78. catch (Exception ex)
79. {
80.     MessageBox.Show(ex.Message);
81. }
82. finally
83. {
84.     conn.Close();
85. }
86. }
87. /// <summary>
88. /// Обработчик события закрытия формы
89. /// </summary>
90. private void ChangeInfo_FormClosing(object sender, FormClosingEventArgs e)
91. {
92.     _GenForm.Show();
93. }
94. }
95. }

```

1.3 Класс DBMySQLUtils.cs:

```

1. using System;
2. using MySql.Data.MySqlClient;
3.
4. namespace Sets_RelationsLessons
5. {
6.     class DBMySQLUtils
7.     {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

8.      /// <summary>
9.      /// Получение соединения с базой данных
10.     /// </summary>
11.     /// <param name="host">хост</param>
12.     /// <param name="port">порт</param>
13.     /// <param name="database">название базы данных</param>
14.     /// <param name="username">имя пользователя</param>
15.     /// <param name="password">пароль</param>
16.     /// <returns>соединение</returns>
17.     public static MySqlConnection GetDBConnection(string host, uint port, string da
    tabase, string username, string password)
18.     {
19.         var builder = new MySqlConnectionStringBuilder
20.         {
21.             Database = database,
22.             Password = password,
23.             Server = host,
24.             UserID = username,
25.             SslMode = MySqlSslMode.None,
26.             Port = port
27.         };
28.
29.         MySqlConnection conn = new MySqlConnection
30.         {
31.             ConnectionString = builder.ConnectionString
32.         };
33.
34.         return conn;
35.     }
36. }
37. }

```

1.4 Класс DBUtils.cs:

```

1. using MySql.Data.MySqlClient;
2.
3. namespace Sets_RelationsLessons
4. {
5.     class DBUtils
6.     {
7.         /// <summary>
8.         /// Получение соединения с базой данных
9.         /// </summary>
10.        /// <returns>содинение</returns>
11.        public static MySqlConnection GetDBConnection()
12.        {
13.            string host = "IP_Adress";
14.            uint port = 3306;
15.            string database = "DataBase_name";
16.            string username = "Username";
17.            string password = "Password";
18.
19.            return DBMySQLUtils.GetDBConnection(host, port, database, username, passwor
    d);
20.        }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата


```
21.     }
22. }
```

1.5 Класс General.cs:

```
1. using System;
2. using System.IO;
3. using System.Windows.Forms;
4.
5. namespace Sets_RelationsLessons
6. {
7.     public partial class General : Form
8.     {
9.         public General()
10.        {
11.            InitializeComponent();
12.            this.Text = "Добро пожаловать, " + Properties.Settings.Default.login;
13.            TextArea.ScriptErrorsSuppressed = true;
14.            for (int i = 1; i <= 10; i++)
15.            {
16.                if (File.Exists(Directory.GetCurrentDirectory() + @"\Lessons\lesson" +
17.                    i + ".html"))
18.                    выборТемыToolStripMenuItem.DropDownItems[i - 1].Visible = true;
19.                else
20.                    выборТемыToolStripMenuItem.DropDownItems[i - 1].Visible = false;
21.            }
22.            /// <summary>
23.            /// Обработчик события нажатия на пункт меню "Изменить персональные данные"
24.            /// </summary>
25.            private void изменитьДанныеToolStripMenuItem_Click(object sender, EventArgs e)
26.            {
27.                this.Hide();
28.                ChangeInfo form = new ChangeInfo(this);
29.                form.Show();
30.            }
31.            /// <summary>
32.            /// Обработчик события завершения загрузки html-документа
33.            /// </summary>
34.            private void TextArea_DocumentCompleted(object sender, WebBrowserDocumentComple
35.                tedEventArgs e)
36.            {
37.                foreach (HtmlElement he in TextArea.Document.GetElementsByTagName("input"))
38.                {
39.                    switch (he.GetAttribute("name"))
40.                    {
41.                        case "Next":
42.                            he.AttachEventHandler("onclick", Next_Click);
43.                            break;
44.                        case "Previous":
45.                            he.AttachEventHandler("onclick", Previous_Click);
46.                            break;
47.                        case "StartTest":
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

47.         he.AttachEventHandler("onclick", StartTest_Click);
48.         break;
49.         case "ContinueTest":
50.             he.AttachEventHandler("onclick", ContinueTest_Click);
51.             break;
52.         default:
53.             break;
54.     }
55. }
56. }
57. /// <summary>
58. /// Обработчик события нажатия кнопки "Следующий урок"
59. /// </summary>
60. private void Next_Click(object sender, EventArgs e)
61. {
62.     if (File.Exists(Directory.GetCurrentDirectory() + @"\Lessons\lesson" + (int
        .Parse(TextArea.DocumentTitle.Split()[1]) + 1) + ".html"))
63.         TextArea.Navigate(Directory.GetCurrentDirectory() + @"\Lessons\lesson"
        + (int.Parse(TextArea.DocumentTitle.Split()[1]) + 1) + ".html");
64.     }
65.     /// <summary>
66.     /// Обработчик события нажатия кнопки "Предыдущий урок"
67.     /// </summary>
68.     private void Previous_Click(object sender, EventArgs e)
69.     {
70.         if (File.Exists(Directory.GetCurrentDirectory() + @"\Lessons\lesson" + (int
            .Parse(TextArea.DocumentTitle.Split()[1]) - 1) + ".html"))
71.             TextArea.Navigate(Directory.GetCurrentDirectory() + @"\Lessons\lesson"
            + (int.Parse(TextArea.DocumentTitle.Split()[1]) - 1) + ".html");
72.         }
73.         /// <summary>
74.         /// Обработчик события нажатия кнопки "Начать тест"
75.         /// </summary>
76.         private void StartTest_Click(object sender, EventArgs e)
77.         {
78.             if (File.Exists(Directory.GetCurrentDirectory() + @"\Lessons\test" + int.Pa
                rse(TextArea.DocumentTitle.Split()[1]) + ".txt"))
79.             {
80.                 this.Hide();
81.                 TestForm form = new TestForm(Directory.GetCurrentDirectory() + @"\Lesso
                    ns\test" + int.Parse(TextArea.DocumentTitle.Split()[1]) + ".txt");
82.                 if (form.IsDisposed == false) form.Show();
83.             }
84.         }
85.         /// <summary>
86.         /// Обработчик события нажатия кнопки "Продолжить тест"
87.         /// </summary>
88.         private void ContinueTest_Click(object sender, EventArgs e)
89.         {
90.             if (File.Exists(Directory.GetCurrentDirectory() + @"\Lessons\test" + int.Pa
                rse(TextArea.DocumentTitle.Split()[1]) + ".txt"))
91.             {
92.                 this.Hide();
93.                 TestForm form = new TestForm(Directory.GetCurrentDirectory() + @"\Lesso
                    ns\test" + int.Parse(TextArea.DocumentTitle.Split()[1]) + ".txt", 1);
94.                 if (form.IsDisposed == false) form.Show();
95.             }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

96.     }
97.     /// <summary>
98.     /// Обработчик события нажатия на пункт меню "Сменить пользователя"
99.     /// </summary>
100.    private void сменитьПользователяToolStripMenuItem_Click(object sender, E
    ventArgs e)
101.    {
102.        DialogResult dialogResult = MessageBox.Show("Вы действительно хотите
        выйти?", "Смена пользователя", MessageBoxButtons.OKCancel);
103.        if (dialogResult == DialogResult.OK)
104.        {
105.            Login log = new Login();
106.            log.Show();
107.            this.Hide();
108.            Properties.Settings.Default.login = "";
109.            Properties.Settings.Default.Save();
110.        }
111.    }
112.    /// <summary>
113.    /// Обработчик события нажатия на пункт меню "Результаты тестов"
114.    /// </summary>
115.    private void результатыТестовToolStripMenuItem_Click(object sender, Even
        tArgs e)
116.    {
117.        Results res = new Results();
118.        res.Show();
119.    }
120.    /// <summary>
121.    /// Обработчик события нажатия на пункт меню "Урок 1"
122.    /// </summary>
123.    private void урок1ToolStripMenuItem_Click(object sender, EventArgs e)
124.    {
125.        TextArea.Navigate(Directory.GetCurrentDirectory() + @"\"Lessons\lesso
        n1.html");
126.    }
127.    /// <summary>
128.    /// Обработчик события нажатия на пункт меню "Урок 2"
129.    /// </summary>
130.    private void урок2ToolStripMenuItem_Click(object sender, EventArgs e)
131.    {
132.        TextArea.Navigate(Directory.GetCurrentDirectory() + @"\"Lessons\lesso
        n2.html");
133.    }
134.    /// <summary>
135.    /// Обработчик события нажатия на пункт меню "Урок 3"
136.    /// </summary>
137.    private void урок3ToolStripMenuItem_Click(object sender, EventArgs e)
138.    {
139.        TextArea.Navigate(Directory.GetCurrentDirectory() + @"\"Lessons\lesso
        n3.html");
140.    }
141.    /// <summary>
142.    /// Обработчик события нажатия на пункт меню "Урок 4"
143.    /// </summary>
144.    private void урок4ToolStripMenuItem_Click(object sender, EventArgs e)
145.    {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

146.                TextArea.Navigate(Directory.GetCurrentDirectory() + @"\\Lessons\lesso
n4.html");
147.            }
148.            /// <summary>
149.            /// Обработчик события нажатия на пункт меню "Урок 5"
150.            /// </summary>
151.            private void урок5ToolStripMenuItem_Click(object sender, EventArgs e)
152.            {
153.                TextArea.Navigate(Directory.GetCurrentDirectory() + @"\\Lessons\lesso
n5.html");
154.            }
155.            /// <summary>
156.            /// Обработчик события нажатия на пункт меню "Урок 6"
157.            /// </summary>
158.            private void урок6ToolStripMenuItem_Click(object sender, EventArgs e)
159.            {
160.                TextArea.Navigate(Directory.GetCurrentDirectory() + @"\\Lessons\lesso
n6.html");
161.            }
162.            /// <summary>
163.            /// Обработчик события нажатия на пункт меню "Урок 7"
164.            /// </summary>
165.            private void урок7ToolStripMenuItem_Click(object sender, EventArgs e)
166.            {
167.                TextArea.Navigate(Directory.GetCurrentDirectory() + @"\\Lessons\lesso
n7.html");
168.            }
169.            /// <summary>
170.            /// Обработчик события нажатия на пункт меню "Урок 8"
171.            /// </summary>
172.            private void урок8ToolStripMenuItem_Click(object sender, EventArgs e)
173.            {
174.                TextArea.Navigate(Directory.GetCurrentDirectory() + @"\\Lessons\lesso
n8.html");
175.            }
176.            /// <summary>
177.            /// Обработчик события нажатия на пункт меню "Урок 9"
178.            /// </summary>
179.            private void урок9ToolStripMenuItem_Click(object sender, EventArgs e)
180.            {
181.                TextArea.Navigate(Directory.GetCurrentDirectory() + @"\\Lessons\lesso
n9.html");
182.            }
183.            /// <summary>
184.            /// Обработчик события нажатия на пункт меню "Урок 10"
185.            /// </summary>
186.            private void урок10ToolStripMenuItem_Click(object sender, EventArgs e)
187.            {
188.                TextArea.Navigate(Directory.GetCurrentDirectory() + @"\\Lessons\lesso
n10.html");
189.            }
190.            /// <summary>
191.            /// Обработчик события нажатия на пункт меню "Выход"
192.            /// </summary>
193.            private void выходToolStripMenuItem_Click(object sender, EventArgs e)
194.            {
195.                this.Close();

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

196.     }
197.     /// <summary>
198.     /// Обработчик события закрытия формы
199.     /// </summary>
200.     private void General_FormClosing(object sender, FormClosingEventArgs e)
201.     {
202.         if (e.CloseReason.ToString() == "UserClosing")
203.         {
204.             DialogResult dialogResult = MessageBox.Show("Вы действительно хо
205. тите выйти?", "Выход", MessageBoxButtons.OKCancel);
206.             if (dialogResult == DialogResult.OK)
207.             {
208.                 foreach (string name in Directory.GetFiles(Directory.GetCurrent
209. Directory() + @"Lessons\", "tmp*"))
210.                 {
211.                     File.Delete(name);
212.                 }
213.             }
214.             else
215.             {
216.                 e.Cancel = true;
217.             }
218.         }
219.     }

```

1.6 Классы Graph.cs

1.6.1 Класс Graph:

```

1. using System.Collections.Generic;
2. using System.Xml.Serialization;
3.
4. namespace Sets_RelationsLessons
5. {
6.     /// <summary>
7.     /// Класс графа
8.     /// </summary>
9.     [XmlRoot("Diagram")]
10.    public class Graph
11.    {
12.        [XmlArray("Nodes")]
13.        [XmlArrayItem("Node")]
14.        public List<Vertice> vertices;
15.        [XmlArray("Links")]
16.        [XmlArrayItem("Link")]
17.        public List<Edge> edges;
18.    }
19. }

```

1.6.2 Класс Vertice:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

1. using System.Xml.Serialization;
2.
3. namespace Sets_RelationsLessons
4. {
5.     /// <summary>
6.     /// Класс вершины
7.     /// </summary>
8.     public class Vertice
9.     {
10.         [XmlAttribute("Id")]
11.         public string Id;
12.     }
13. }

```

1.6.3 Класс Edge:

```

1. using System.Xml.Serialization;
2.
3. namespace Sets_RelationsLessons
4. {
5.     /// <summary>
6.     /// Класс ребра
7.     /// </summary>
8.     public class Edge
9.     {
10.         [XmlElement("Origin")]
11.         public Origin OId;
12.         [XmlElement("Destination")]
13.         public Destination DId;
14.     }
15. }

```

1.6.4 Класс Origin:

```

1. using System.Xml.Serialization;
2.
3. namespace Sets_RelationsLessons
4. {
5.     /// <summary>
6.     /// Класс вершины, из которой выходит ребро
7.     /// </summary>
8.     public class Origin : Vertice
9.     {
10.         [XmlAttribute("Id")]
11.         public new string Id;
12.     }
13.     /// <summary>
14. }

```

1.6.5 Класс Destination:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

1. using System.Xml.Serialization;
2.
3. namespace Sets_RelationsLessons
4. {
5.     /// <summary>
6.     /// Класс вершины, в которую входит ребро
7.     /// </summary>
8.     public class Destination : Vertice
9.     {
10.         [XmlAttribute("Id")]
11.         public new string Id;
12.     }
13. }

```

1.7 Класс GraphLayout.xaml.cs:

```

1. using System;
2. using System.IO;
3. using System.Windows;
4. using System.Windows.Controls;
5. using MindFusion.Diagramming.Wpf;
6.
7. namespace Sets_RelationsLessons
8. {
9.     /// <summary>
10.    /// Логика взаимодействия для GraphLayout.xaml
11.    /// </summary>
12.    public partial class GraphLayout : UserControl
13.    {
14.        Test test;
15.        int num;
16.
17.        public GraphLayout(ref Test test, int num, string name, bool EdgeAuto)
18.        {
19.            InitializeComponent();
20.            this.test = test;
21.            this.num = num;
22.            diagram.DefaultShape = Shapes.Ellipse;
23.            try
24.            {
25.                if (name == "ans")
26.                    diagram.LoadFromXml(Directory.GetCurrentDirectory() + @"\Lessons\tm
pGraph" + num);
27.                else if (name == "true")
28.                    diagram.LoadFromXml(Directory.GetCurrentDirectory() + @"\Lessons\"
+ test.tasks[num].answer);
29.                else if (test.tasks[num].graph_name != "")
30.                    diagram.LoadFromXml(test.tasks[num].graph_name);
31.                else
32.                    diagram.LoadFromString("<diagram></diagram>");
33.            }
34.            catch (FileLoadException exp)
35.            {
36.                MessageBox.Show(exp.Message);
37.                return;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

38.     }
39.     catch (Exception exp)
40.     {
41.         MessageBox.Show(exp.Message);
42.     }
43.
44.     if (name == "DrawingGraph")
45.     {
46.         int m = diagram.Links.Count;
47.         for (int i = 0; i < m; i++)
48.         {
49.             diagram.Links.RemoveAt(0);
50.         }
51.     }
52.     if (name == "PictureGraph") diagram.EnableLanes = false;
53.
54.     DiagramLinkCollection diag = new DiagramLinkCollection();
55.     int n = diagram.Links.Count;
56.     for (int i = 0; i < n; i++)
57.     {
58.         diag.Add(new DiagramLink(diagram.Links[0].Parent, diagram.Links[0].Orig
in, diagram.Links[0].Destination));
59.         diag[i].AutoRoute = EdgeAuto;
60.         diagram.Links.RemoveAt(0);
61.     }
62.     foreach (var item in diag)
63.     {
64.         diagram.Links.Add(item);
65.     }
66. }
67. /// <summary>
68. /// Метод сохранения текущего ответа-графа во временный файл
69. /// </summary>
70. private void UserControl_IsEnabledChanged(object sender, DependencyPropertyChan
gedEventArgs e)
71. {
72.     if (test.tasks[num].choice_ans == 0 || test.tasks[num].choice_ans == 4) dia
gram.SaveToXml(Directory.GetCurrentDirectory() + @"\"Lessons\"tmpGraph" + num);
73. }
74. /// <summary>
75. /// Запрет на создание и удаление вершин
76. /// </summary>
77. private void diagram_NodeChanging(object sender, NodeValidationEventArgs e)
78. {
79.     e.Cancel = true;
80. }
81. /// <summary>
82. /// Позволяет удалять только петли
83. /// </summary>
84. private void diagram_LinkDeleting(object sender, LinkValidationEventArgs e)
85. {
86.     if (e.Origin != e.Destination) e.Cancel = true;
87. }
88. }
89. }

```

1.8 Класс Program.cs:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата


```

1. using System;
2. using System.Windows.Forms;
3.
4. namespace Sets_RelationsLessons
5. {
6.     static class Program
7.     {
8.         /// <summary>
9.         /// Главная точка входа для приложения.
10.        /// </summary>
11.        [STAThread]
12.        static void Main()
13.        {
14.            Application.EnableVisualStyles();
15.            Application.SetCompatibleTextRenderingDefault(false);
16.            if (Properties.Settings.Default.login == "")
17.            {
18.                Application.Run(new Login());
19.            }
20.            else
21.            {
22.                Application.Run(new General());
23.            }
24.        }
25.    }
26. }

```

1.6.1 Класс Reg.cs:

```

1. using MySql.Data.MySqlClient;
2. using System;
3. using System.Data.Common;
4. using System.Windows.Forms;
5.
6. namespace Sets_RelationsLessons
7. {
8.     public partial class Reg : Form
9.     {
10.        Login_LogForm;
11.        MySqlConnection conn;
12.
13.        public Reg(Login LogForm)
14.        {
15.            _LogForm = LogForm;
16.            InitializeComponent();
17.            conn = DBUtils.GetDBConnection();
18.        }
19.        /// <summary>
20.        /// Обработчик события нажатия кнопки "Отмена"
21.        /// </summary>
22.        private void Cancel_Click(object sender, EventArgs e)
23.        {
24.            this.Close();
25.        }
26.        /// <summary>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

27.      /// Обработчик события нажатия кнопки "Зарегистрироваться"(Создать пользователя
28.    )
29.      /// </summary>
30.      private void Registration_Click(object sender, EventArgs e)
31.      {
32.          int c = 1;
33.          try
34.          {
35.              if (Login_value.Text.Length < 3) throw new Exception("Слишком короткий
36.              логин(должен содержать 3 или больше символов).");
37.              if (Login_value.Text.Length > 15) throw new Exception("Слишком длинный
38.              логин(должен содержать 15 или меньше символов).");
39.              if (Login_value.Text.IndexOf(' ') > -
40.              1) throw new Exception("Логин не должен содержать пробелы.");
41.              if (Password_value.Text.Length < 3) throw new Exception("Слишком коротк
42.              ий пароль(должен содержать 3 или больше символов).");
43.              if (Password_value.Text.Length > 20) throw new Exception("Слишком длинн
44.              ый пароль(должен содержать 20 или меньше символов).");
45.              if (Password_value.Text.IndexOf(' ') > -
46.              1) throw new Exception("Пароль не должен содержать пробелы.");
47.              if (Password_value.Text.GetHashCode() != Password_value2.Text.GetHashCo
48.              de()) throw new Exception("Пароли не совпадают.");
49.
50.              conn.Open();
51.              using (MySqlCommand cmd = new MySqlCommand("Select idUsers, Login, Pass
52.              word from users", conn))
53.              {
54.                  using (DbDataReader reader = cmd.ExecuteReader())
55.                  {
56.                      if (reader.HasRows)
57.                      {
58.                          while (reader.Read())
59.                          {
60.                              c++;
61.                              if (Login_value.Text == reader.GetString(reader.GetOrdi
62.                              nal("Login")))
63.                                  throw new Exception("Этот логин уже занят.");
64.                          }
65.                      }
66.                  }
67.              }
68.              using (MySqlCommand cmd = new MySqlCommand("Insert into users(idUsers,
69.              Login, Password)" + "values (@id, @login, @password)", conn))
70.              {
71.                  MySqlParameter idParam = new MySqlParameter("@id", MySqlDbType.Int3
72.                  2)
73.                  {
74.                      Value = c
75.                  };
76.                  cmd.Parameters.Add(idParam);
77.                  MySqlParameter loginParam = new MySqlParameter("@login", MySqlDbType
78.                  e.VarChar)
79.                  {
80.                      Value = Login_value.Text
81.                  };

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

71.             cmd.Parameters.Add(loginParam);
72.
73.             MySqlParameter passwordParam = new MySqlParameter("@password", MySql
1DbType.Int32)
74.             {
75.                 Value = Password_value.Text.GetHashCode()
76.             };
77.             cmd.Parameters.Add(passwordParam);
78.
79.             cmd.ExecuteNonQuery();
80.         }
81.         MessageBox.Show("Регистрация прошла успешно!");
82.
83.         _LogForm.Show();
84.         conn.Close();
85.         this.Close();
86.     }
87.     catch (Exception ex)
88.     {
89.         MessageBox.Show(ex.Message);
90.     }
91.     finally
92.     {
93.         conn.Close();
94.     }
95. }
96. /// <summary>
97. /// Обработчик события закрытия формы
98. /// </summary>
99. private void Reg_FormClosing(object sender, FormClosingEventArgs e)
100.    {
101.        _LogForm.Show();
102.    }
103. }
104. }

```

1.9 Класс Results.cs:

```

1. using MySql.Data.MySqlClient;
2. using System;
3. using System.Data.Common;
4. using System.Drawing;
5. using System.Linq;
6. using System.Windows.Forms;
7.
8. namespace Sets_RelationsLessons
9. {
10.     public partial class Results : Form
11.     {
12.         MySqlConnection conn;
13.
14.         public Results()
15.         {
16.             InitializeComponent();
17.

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

18.         conn = DBUtils.GetDBConnection();
19.         ResultToTable();
20.     }
21.     /// <summary>
22.     /// Добавление столбца в таблицу результатов
23.     /// </summary>
24.     /// <param name="name">название столбца</param>
25.     private void AddColumn(string name)
26.     {
27.         var column = new DataGridViewColumn
28.         {
29.             HeaderText = name,
30.             ReadOnly = true,
31.             Frozen = false,
32.             CellTemplate = new DataGridViewTextBoxCell()
33.         };
34.         ResultTable.Columns.Add(column);
35.     }
36.     /// <summary>
37.     /// Внесение результатов в таблицу
38.     /// </summary>
39.     /// <param name="test">номер теста</param>
40.     private void ResultToTable(int test = -1)
41.     {
42.         ResultTable.Rows.Clear();
43.         ResultTable.Columns.Clear();
44.         int id = 0;
45.         try
46.         {
47.             conn.Open();
48.
49.             using (MySqlCommand cmd = new MySqlCommand("Select idUsers, Login from
users", conn))
50.             {
51.                 using (DbDataReader reader = cmd.ExecuteReader())
52.                 {
53.                     if (reader.HasRows)
54.                     {
55.                         while (reader.Read())
56.                         {
57.                             if (Properties.Settings.Default.login == reader.GetString(
reader.GetOrdinal("Login"))) id = reader.GetInt32(reader.GetOrdinal("idUsers"));
58.                         }
59.                     }
60.                 }
61.             }
62.
63.             using (MySqlCommand cmd = new MySqlCommand("Select idTests, Result, Use
rs_idUsers, Date, Time from tests", conn))
64.             {
65.                 if (test == -1)
66.                 {
67.                     AddColumn("Тест");
68.                     AddColumn("Процент верных ответов (%)");
69.                     AddColumn("Оценка первой попытки");
70.
71.                     ResultTable.AllowUserToAddRows = false;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

72.         string res;
73.         for (int i = 0; i < 10; i++)
74.         {
75.             using (DbDataReader reader = cmd.ExecuteReader())
76.             {
77.                 if (reader.HasRows)
78.                 {
79.                     while (reader.Read())
80.                     {
81.                         if (id == reader.GetInt32(reader.GetOrdinal("Us
ers_idUsers")) && i == reader.GetInt32(reader.GetOrdinal("idTests")))
82.                         {
83.                             res = reader.GetString(reader.GetOrdinal("R
esult"));
84.                             ResultTable.Rows.Add("Тест " + i, String.Fo
rmat("{0:0.00}", 1.0 * res.Count(f => f == 't') / res.Length * 100), String.Format("{0:
0}", 1.0 * res.Count(f => f == 't') / res.Length * 10));
85.                             if (double.Parse(ResultTable[2, ResultTable
.Rows.Count - 1].Value.ToString()) >= 8) ResultTable[2, ResultTable.Rows.Count - 1].Sty
le.BackColor = Color.Green;
86.                             else if (double.Parse(ResultTable[2, Result
Table.Rows.Count - 1].Value.ToString()) >= 6) ResultTable[2, ResultTable.Rows.Count - 1
].Style.BackColor = Color.GreenYellow;
87.                             else if (double.Parse(ResultTable[2, Result
Table.Rows.Count - 1].Value.ToString()) >= 4) ResultTable[2, ResultTable.Rows.Count - 1
].Style.BackColor = Color.Yellow;
88.                             else if (double.Parse(ResultTable[2, Result
Table.Rows.Count - 1].Value.ToString()) >= 1) ResultTable[2, ResultTable.Rows.Count - 1
].Style.BackColor = Color.Orange;
89.                             else ResultTable[2, ResultTable.Rows.Count
- 1].Style.BackColor = Color.Red;
90.                             break;
91.                         }
92.                     }
93.                 }
94.             }
95.         }
96.     }
97.     else
98.     {
99.         AddColumn("Тест " + (test + 1));
100.
101.         string res;
102.         bool f = true;
103.         using (DbDataReader reader = cmd.ExecuteReader())
104.         {
105.             if (reader.HasRows)
106.             {
107.                 while (reader.Read())
108.                 {
109.                     if (id == reader.GetInt32(reader.GetOrdinal(
"Users_idUsers")) && test + 1 == reader.GetInt32(reader.GetOrdinal("idTests")))
110.                     {
111.                         res = reader.GetString(reader.GetOrdinal
("Result"));
112.                         if (f)
113.                         {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

114.                                     for (int j = 0; j < res.Length; j++)
115.                                     {
116.                                         AddColumn("Вопрос " + (j + 1));
117.                                     }
118.                                     f = false;
119.                                 }
120.                                 if (ResultTable.Rows.Count == 0)
121.                                 {
122.                                     AddColumn("Процент верных ответов (%)");
123.                                     AddColumn("Оценка");
124.                                     AddColumn("Время сдачи");
125.                                     AddColumn("Время выполнения");
126.                                 }
127.                                 ResultTable.Rows.Add();
128.                                 ResultTable[0, ResultTable.Rows.Count -
129.                                     1].Value = "Попытка " + ResultTable.Rows.Count;
130.                                 for (int i = 0; i < res.Length; i++)
131.                                 {
132.                                     if (res[i] == 't')
133.                                     {
134.                                         ResultTable[i + 1, ResultTable.R
135.                                             ows.Count - 1].Value = "+";
136.                                         ResultTable[i + 1, ResultTable.R
137.                                             ows.Count - 1].Style.BackColor = Color.Green;
138.                                     }
139.                                     else
140.                                     {
141.                                         ResultTable[i + 1, ResultTable.R
142.                                             ows.Count - 1].Value = "-";
143.                                         ResultTable[i + 1, ResultTable.R
144.                                             ows.Count - 1].Style.BackColor = Color.Red;
145.                                     }
146.                                 }
147.                                 ResultTable[ResultTable.Columns.Count -
148.                                     4, ResultTable.Rows.Count - 1].Value = String.Format("{0:0.00}", 1.0 * res.Count(t => t
149.                                     == 't') / res.Length * 100);
150.                                 ResultTable[ResultTable.Columns.Count -
151.                                     3, ResultTable.Rows.Count - 1].Value = String.Format("{0:0}", 1.0 * res.Count(t => t ==
152.                                     't') / res.Length * 10);
153.                                 ResultTable[ResultTable.Columns.Count -
154.                                     2, ResultTable.Rows.Count - 1].Value = reader.GetDateTime(reader.GetOrdinal("Date"));
155.                                 ResultTable[ResultTable.Columns.Count -
156.                                     1, ResultTable.Rows.Count - 1].Value = reader.GetValue(reader.GetOrdinal("Time"));
157.                                 }
158.                             }
159.                         }
160.                         ResultTable.AllowUserToAddRows = false;
161.                     }
162.                 }
163.             }
164.         }
165.     }
166. }
167. catch (Exception)
168. {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

156.         MessageBox.Show("Соединение отсутствует. Попробуйте переподключи
        ться к Интернету.");
157.     }
158.     finally
159.     {
160.         conn.Close();
161.         conn.Dispose();
162.     }
163. }
164.     /// <summary>
165.     /// Метод, для осуществления перехода от таблицы с результатами всех тес
    тов к определённому с результатами попыток его решения
166.     /// </summary>
167.     private void ResultTable_CellDoubleClick(object sender, DataGridViewCell
    EventArgs e)
168.     {
169.         if (e.ColumnIndex == 0)
170.         {
171.             try
172.             {
173.                 if (ResultTable.Rows[e.RowIndex].Cells[e.ColumnIndex].Value.
    ToString().Split()[0] == "Тест")
174.                     ResultToTable(int.Parse(ResultTable.Rows[e.RowIndex].Cel
    ls[e.ColumnIndex].Value.ToString().Split()[1] - 1);
175.             }
176.             catch
177.             {
178.                 ResultToTable();
179.             }
180.         }
181.     }
182. }
183. }

```

1.10 Классы Test.cs

1.11.1 Класс Test:

```

1. using System.Xml.Serialization;
2.
3. namespace Sets_RelationsLessons
4. {
5.     /// <summary>
6.     /// Класс теста
7.     /// </summary>
8.     [XmlRoot("Test")]
9.     public class Test
10.    {
11.        [XmlArray("Tasks")]
12.        [XmlElement("Task")]
13.        public Task[] tasks;
14.        [XmlElement("Id")]
15.        public int id;
16.    }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```
17. }
```

1.11.2 Класс Task:

```
1. using System.IO;
2. using System.Xml.Serialization;
3.
4. namespace Sets_RelationsLessons
5. {
6.     /// <summary>
7.     /// Класс задания(вопроса)
8.     /// </summary>
9.     public class Task
10.    {
11.        [XmlElement("Graph")]
12.        public string graph_name;
13.        [XmlElement("Condition")]
14.        public string condition;
15.        [XmlElement("Choice")]
16.        public int choice_ans;
17.        [XmlArray("Answers")]
18.        [XmlArrayItem("Ans")]
19.        public string[] choices;
20.        [XmlElement("Answer")]
21.        public string answer;
22.        /// <summary>
23.        /// Получение графа из xml файла
24.        /// </summary>
25.        /// <param name="graph_name">название файла</param>
26.        /// <returns>граф</returns>
27.        public Graph GetGraph(string graph_name)
28.        {
29.            if (File.Exists(graph_name))
30.            {
31.                XmlSerializer serializer = new XmlSerializer(typeof(Graph));
32.                FileStream fs = new FileStream(graph_name, FileMode.Open);
33.                Graph graph = (Graph)serializer.Deserialize(fs);
34.                fs.Close();
35.                return graph;
36.            }
37.            else return new Graph();
38.        }
39.    }
40. }
```

1.11 Класс TestForm.cs:

```
1. using MySql.Data.MySqlClient;
2. using System;
3. using System.Collections.Generic;
4. using System.Data;
5. using System.Data.Common;
6. using System.Drawing;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата


```

7. using System.IO;
8. using System.Windows.Forms;
9. using System.Xml.Serialization;
10.
11. namespace Sets_RelationsLessons
12. {
13.
14.     public partial class TestForm : Form
15.     {
16.         Test test;
17.         string[] answers;
18.         string ans;
19.         bool EdgeAuto = true;
20.         MySqlConnection conn;
21.         DateTime time = new DateTime(0,0);
22.
23.         public TestForm(string testname, int t = 0)
24.         {
25.             InitializeComponent();
26.             автоToolStripMenuItem.BackColor = Color.Aqua;
27.             conn = DBUtils.GetDBConnection();
28.             DrawingGraph.Location = new Point(this.Width / 2 + 1, 39);
29.
30.             try
31.             {
32.                 XmlSerializer serializer = new XmlSerializer(typeof(Test));
33.                 using (FileStream fs = new FileStream(testname, FileMode.Open))
34.                 {
35.                     test = (Test)serializer.Deserialize(fs);
36.                 }
37.                 for (int i = 0; i < test.tasks.Length; i++)
38.                 {
39.                     if (File.Exists(Directory.GetCurrentDirectory() + @"Lessons\" + test.tasks[i].graph_name)) test.tasks[i].graph_name = Directory.GetCurrentDirectory() + @"Lessons\" + test.tasks[i].graph_name;
40.                     else
41.                     {
42.                         MessageBox.Show("Файл " + test.tasks[i].graph_name + " не найден!");
43.                         test.tasks[i].graph_name = "";
44.                     }
45.                 }
46.             }
47.             catch (Exception ex)
48.             {
49.                 MessageBox.Show(ex.Message);
50.                 this.Close();
51.                 return;
52.             }
53.
54.             answers = new string[test.tasks.Length];
55.             if (t == 1) LoadAns();
56.
57.             int id = 0;
58.             try
59.             {
60.                 conn.Open();

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

61.
62.         using (MySqlCommand cmd = new MySqlCommand("Select idUsers, Login from
        users", conn))
63.         {
64.             using (DbDataReader reader = cmd.ExecuteReader())
65.             {
66.                 if (reader.HasRows)
67.                 {
68.                     while (reader.Read())
69.                     {
70.                         if (Properties.Settings.Default.login == reader.GetStri
ng(reader.GetOrdinal("Login"))) id = reader.GetInt32(reader.GetOrdinal("idUsers"));
71.                     }
72.                 }
73.             }
74.         }
75.
76.         using (MySqlCommand cmd = new MySqlCommand("Delete from files where idT
est = @test and Users_idUsers = @id", conn))
77.         {
78.             cmd.Parameters.Add(new MySqlParameter { ParameterName = "@id", MySq
lDbType = MySqlDbType.Int32, Value = id });
79.             cmd.Parameters.Add(new MySqlParameter { ParameterName = "@test", My
SqlDbType = MySqlDbType.Int32, Value = test.id });
80.
81.             cmd.ExecuteNonQuery();
82.         }
83.     }
84.     catch (Exception e)
85.     {
86.         MessageBox.Show(e.Message);
87.     }
88.     finally
89.     {
90.         conn.Close();
91.         conn.Dispose();
92.     }
93.
94.     timer.Start();
95.     StartTask(0);
96. }
97. /// <summary>
98. /// Метод инициализации задания
99. /// </summary>
100. /// <param name="num">номер задания</param>
101. /// <param name="status">текущий статус</param>
102. private void StartTask(int num, int status = 0)
103. {
104.     if (num == 0) previous.Enabled = false;
105.     else previous.Enabled = true;
106.     if (num == test.tasks.Length - 1) next.Enabled = false;
107.     else next.Enabled = true;
108.     Condition.Text = test.tasks[num].condition;
109.     if (status == 0) this.Text = "Вопрос " + (num + 1) + " Время: " + ti
me.ToLongTimeString();
110.     else this.Text = "Вопрос " + (num + 1);
111.     if (status == 1)

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

112.         if (ans != null && ans[num] == 't')
113.         {
114.             this.Text += " - Верно";
115.             this.BackColor = Color.Green;
116.         }
117.     else
118.     {
119.         this.Text += " - Неверно";
120.         this.BackColor = Color.Red;
121.     }
122.
123.     try {
124.         switch (test.tasks[num].choice_ans)
125.         {
126.             case 1:
127.                 oneGraph.Child = null;
128.                 if (test.tasks[num].graph_name != "") oneGraph.Child = n
ew GraphLayout(ref test, num, "oneGraph", EdgeAuto);
129.                 pictureBox.Visible = false;
130.                 oneGraph.Visible = true;
131.                 PictureGraph.Visible = false;
132.                 DrawingGraph.Visible = false;
133.                 ChooseOneAnsPanel.Visible = true;
134.                 ChooseSomeAnsPanel.Visible = false;
135.                 WriteAnsPanel.Visible = false;
136.                 ChooseOneAnsPanel.Items.Clear();
137.                 foreach (string ans in test.tasks[num].choices)
138.                 {
139.                     ChooseOneAnsPanel.Items.Add(ans);
140.                 }
141.                 if (status == 1)
142.                 {
143.                     for (int i = 0; i < ChooseOneAnsPanel.Items.Count; i
++
144.                     {
145.                         if (ChooseOneAnsPanel.GetItemChecked(i)) ChooseO
neAnsPanel.SetItemChecked(i, false);
146.                     }
147.                     for (int i = 0; i < test.tasks[num].answer.Length; i
++
148.                     {
149.                         ChooseOneAnsPanel.SetItemChecked(int.Parse(test.
tasks[num].answer[i].ToString()) - 1, true);
150.                     }
151.                 } else if (answers[num] != null)
152.                 {
153.                     for (int i = 0; i < answers[num].Length; i++)
154.                     {
155.                         ChooseOneAnsPanel.SetItemChecked(int.Parse(answe
rs[num][i].ToString()) - 1, true);
156.                     }
157.                 }
158.                 break;
159.             case 2:
160.                 oneGraph.Child = null;
161.                 if (test.tasks[num].graph_name != "") oneGraph.Child = n
ew GraphLayout(ref test, num, "oneGraph", EdgeAuto);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

162.         pictureBox.Visible = false;
163.         oneGraph.Visible = true;
164.         PictureGraph.Visible = false;
165.         DrawingGraph.Visible = false;
166.         ChooseOneAnsPanel.Visible = false;
167.         ChooseSomeAnsPanel.Visible = true;
168.         WriteAnsPanel.Visible = false;
169.         ChooseSomeAnsPanel.Items.Clear();
170.         foreach (string ans in test.tasks[num].choices)
171.         {
172.             ChooseSomeAnsPanel.Items.Add(ans);
173.         }
174.         if (status == 1)
175.         {
176.             for (int i = 0; i < ChooseSomeAnsPanel.Items.Count;
177.                 i++)
178.             {
179.                 if (ChooseSomeAnsPanel.GetItemChecked(i)) Choose
180.                 SomeAnsPanel.SetItemChecked(i, false);
181.             }
182.             for (int i = 0; i < test.tasks[num].answer.Length; i
183.                 ++)
184.             {
185.                 ChooseSomeAnsPanel.SetItemChecked(int.Parse(test
186.                 .tasks[num].answer[i].ToString()) - 1, true);
187.             }
188.             else if (answers[num] != null)
189.             {
190.                 for (int i = 0; i < answers[num].Length; i++)
191.                 {
192.                     ChooseSomeAnsPanel.SetItemChecked(int.Parse(answ
193.                     ers[num][i].ToString()) - 1, true);
194.                 }
195.                 break;
196.             case 3:
197.                 oneGraph.Child = null;
198.                 if (test.tasks[num].graph_name != "") oneGraph.Child = n
199.                 ew GraphLayout(ref test, num, "oneGraph", EdgeAuto);
200.                 pictureBox.Visible = false;
201.                 oneGraph.Visible = true;
202.                 PictureGraph.Visible = false;
203.                 DrawingGraph.Visible = false;
204.                 ChooseOneAnsPanel.Visible = false;
205.                 ChooseSomeAnsPanel.Visible = false;
206.                 WriteAnsPanel.Visible = true;
207.                 WriteAnsPanel.Clear();
208.                 WriteAnsPanel.Text = answers[num] ?? "";
209.                 if (status == 1) WriteAnsPanel.Text = test.tasks[num].an
210.                 swer;
211.                 break;
212.             case 4:
213.                 DrawingGraph.Child = null;
214.                 if (test.tasks[num].graph_name != "")
215.                 {
216.                     if (status == 1)

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

212. DrawingGraph.Child = new GraphLayout(ref test, n
um, "true", EdgeAuto);
213. else if (answers[num] != null)
214. DrawingGraph.Child = new GraphLayout(ref test, n
um, "ans", EdgeAuto);
215. else
216. DrawingGraph.Child = new GraphLayout(ref test, n
um, "DrawingGraph", EdgeAuto);
217. PictureGraph.Child = new GraphLayout(ref test, num,
"PictureGraph", EdgeAuto);
218. }
219. pictureBox.Visible = false;
220. oneGraph.Visible = false;
221. PictureGraph.Visible = true;
222. DrawingGraph.Visible = true;
223. ChooseOneAnsPanel.Visible = false;
224. ChooseSomeAnsPanel.Visible = false;
225. WriteAnsPanel.Visible = false;
226. break;
227. case 5:
228. if (test.tasks[num].graph_name != "") pictureBox.Image =
Image.FromFile(test.tasks[num].graph_name);
229. pictureBox.Visible = true;
230. oneGraph.Visible = false;
231. PictureGraph.Visible = false;
232. DrawingGraph.Visible = false;
233. ChooseOneAnsPanel.Visible = true;
234. ChooseSomeAnsPanel.Visible = false;
235. WriteAnsPanel.Visible = false;
236. ChooseOneAnsPanel.Items.Clear();
237. foreach (string ans in test.tasks[num].choices)
238. {
239. ChooseOneAnsPanel.Items.Add(ans);
240. }
241. if (status == 1)
242. {
243. for (int i = 0; i < ChooseOneAnsPanel.Items.Count; i
++)
244. {
245. if (ChooseOneAnsPanel.GetItemChecked(i)) ChooseO
neAnsPanel.SetItemChecked(i, false);
246. }
247. for (int i = 0; i < test.tasks[num].answer.Length; i
++)
248. {
249. ChooseOneAnsPanel.SetItemChecked(int.Parse(test.
tasks[num].answer[i].ToString()) - 1, true);
250. }
251. } else if (answers[num] != null)
252. {
253. for (int i = 0; i < answers[num].Length; i++)
254. {
255. ChooseOneAnsPanel.SetItemChecked(int.Parse(answe
rs[num][i].ToString()) - 1, true);
256. }
257. }
258. break;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

259.         case 6:
260.             if (test.tasks[num].graph_name != "") pictureBox.Image =
Image.FromFile(test.tasks[num].graph_name);
261.             pictureBox.Visible = true;
262.             oneGraph.Visible = false;
263.             PictureGraph.Visible = false;
264.             DrawingGraph.Visible = false;
265.             ChooseOneAnsPanel.Visible = false;
266.             ChooseSomeAnsPanel.Visible = true;
267.             WriteAnsPanel.Visible = false;
268.             ChooseSomeAnsPanel.Items.Clear();
269.             foreach (string ans in test.tasks[num].choices)
270.             {
271.                 ChooseSomeAnsPanel.Items.Add(ans);
272.             }
273.             if (status == 1)
274.             {
275.                 for (int i = 0; i < ChooseSomeAnsPanel.Items.Count;
i++)
276.                 {
277.                     if (ChooseSomeAnsPanel.GetItemChecked(i)) Choose
SomeAnsPanel.SetItemChecked(i, false);
278.                 }
279.                 for (int i = 0; i < test.tasks[num].answer.Length; i
++)
280.                 {
281.                     ChooseSomeAnsPanel.SetItemChecked(int.Parse(test
.tasks[num].answer[i].ToString()) - 1, true);
282.                 }
283.             }
284.             else if (answers[num] != null)
285.             {
286.                 for (int i = 0; i < answers[num].Length; i++)
287.                 {
288.                     ChooseSomeAnsPanel.SetItemChecked(int.Parse(answ
ers[num][i].ToString()) - 1, true);
289.                 }
290.             }
291.             break;
292.         case 7:
293.             if (test.tasks[num].graph_name != "") pictureBox.Image =
Image.FromFile(test.tasks[num].graph_name);
294.             pictureBox.Visible = true;
295.             oneGraph.Visible = false;
296.             PictureGraph.Visible = false;
297.             DrawingGraph.Visible = false;
298.             ChooseOneAnsPanel.Visible = false;
299.             ChooseSomeAnsPanel.Visible = false;
300.             WriteAnsPanel.Visible = true;
301.             WriteAnsPanel.Clear();
302.             WriteAnsPanel.Text = answers[num] ?? "";
303.             if (status == 1) WriteAnsPanel.Text = test.tasks[num].an
swer;
304.             break;
305.         default:
306.             if (test.tasks[num].graph_name != "")
307.             {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

308.         if (status == 1)
309.             oneGraph.Child = new GraphLayout(ref test, num,
"true", EdgeAuto);
310.         else if (answers[num] != null)
311.             oneGraph.Child = new GraphLayout(ref test, num,
"ans", EdgeAuto);
312.         else
313.             oneGraph.Child = new GraphLayout(ref test, num,
"oneGraph", EdgeAuto);
314.     }
315.     pictureBox.Visible = false;
316.     ChooseOneAnsPanel.Visible = false;
317.     ChooseSomeAnsPanel.Visible = false;
318.     WriteAnsPanel.Visible = false;
319.     oneGraph.Visible = true;
320.     PictureGraph.Visible = false;
321.     DrawingGraph.Visible = false;
322.     break;
323.     }
324.     }
325.     catch (Exception ex)
326.     {
327.         MessageBox.Show(ex.Message);
328.     }
329. }
330. /// <summary>
331. /// Обработчик события нажатия кнопки "Сохранить результат"
332. /// </summary>
333. private void SaveAns_Click(object sender, EventArgs e)
334. {
335.     timer.Stop();
336.     SaveThisAnswer();
337.     int id = 0;
338.     try
339.     {
340.         conn.Open();
341.
342.         using (MySqlCommand cmd = new MySqlCommand("Select idUsers, Logi
n from users", conn))
343.         {
344.             using (DbDataReader reader = cmd.ExecuteReader())
345.             {
346.                 if (reader.HasRows)
347.                 {
348.                     while (reader.Read())
349.                     {
350.                         if (Properties.Settings.Default.login == reader.
GetString(reader.GetOrdinal("Login"))) id = reader.GetInt32(reader.GetOrdinal("idUsers"
));
351.                     }
352.                 }
353.             }
354.         }
355.
356.         for (int i = 0; i < answers.Length; i++)
357.         {
358.             if (answers[i] != null)

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

359.                {
360.                    using (MySQLCommand cmd = new MySQLCommand("Insert into
files(idTest, idTask, File, Users_idUsers) " + " values(@idTest, @idTask, @file, @id)",
conn))
361.                {
362.                    cmd.Parameters.Add(new MySQLParameter("@idTest", Sql
DbType.Int));
363.                    cmd.Parameters["@idTest"].Value = test.id;
364.                    cmd.Parameters.Add(new MySQLParameter("@idTask", Sql
DbType.Int));
365.                    cmd.Parameters["@idTask"].Value = i;
366.                    cmd.Parameters.Add(new MySQLParameter("@file", MySQL
DbType.LongText));
367.                    if (answers[i] == (@"Lessons\tmpGraph" + i))
368.                    {
369.                        FileInfo file = new FileInfo(Directory.GetCurren
tDirectory() + @"Lessons\tmpGraph" + i);
370.                        if (file.Length <= 1050000)
371.                            using (StreamReader sr = new StreamReader(ne
w FileStream(Directory.GetCurrentDirectory() + @"Lessons\tmpGraph" + i, FileMode.Open,
FileAccess.Read)))
372.                            {
373.                                cmd.Parameters["@file"].Value = sr.ReadT
oEnd();
374.                            }
375.                            File.Delete(Directory.GetCurrentDirectory() + @"
Lessons\tmpGraph" + i);
376.                        }
377.                        else
378.                        {
379.                            cmd.Parameters["@file"].Value = answers[i];
380.                        }
381.                        cmd.Parameters.Add(new MySQLParameter("@id", SqlDbTy
pe.Int));
382.                        cmd.Parameters["@id"].Value = id;
383.
384.                        cmd.ExecuteNonQuery();
385.                    }
386.                }
387.            }
388.        }
389.        catch (Exception ex)
390.        {
391.            MessageBox.Show(ex.Message);
392.        }
393.        finally
394.        {
395.            conn.Close();
396.            conn.Dispose();
397.        }
398.    }
399.    /// <summary>
400.    /// Загружает ответы пользователя из базы данных
401.    /// </summary>
402.    public void LoadAns()
403.    {
404.        int id = 0;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата


```

405.         try
406.         {
407.             conn.Open();
408.
409.             using (MySqlCommand cmd = new MySqlCommand("Select idUsers, Login
n from users", conn))
410.             {
411.                 using (DbDataReader reader = cmd.ExecuteReader())
412.                 {
413.                     if (reader.HasRows)
414.                     {
415.                         while (reader.Read())
416.                         {
417.                             if (Properties.Settings.Default.login == reader.
GetString(reader.GetOrdinal("Login"))) id = reader.GetInt32(reader.GetOrdinal("idUsers"
));
418.                         }
419.                     }
420.                 }
421.             }
422.
423.             using (MySqlCommand cmd = new MySqlCommand("Select idTest, idTask, File, Users_idUsers from files", conn))
424.             {
425.                 using (DbDataReader reader = cmd.ExecuteReader())
426.                 {
427.                     if (reader.HasRows)
428.                     {
429.                         while (reader.Read())
430.                         {
431.                             if (id == reader.GetInt32(reader.GetOrdinal("Users_idUsers")) && test.id == reader.GetInt32(reader.GetOrdinal("idTest")))
432.                             {
433.                                 if (test.tasks[reader.GetInt32(reader.GetOrdinal("idTask"))].choice_ans % 4 != 0) answers[reader.GetInt32(reader.GetOrdinal("idTask"))] = reader.GetString(reader.GetOrdinal("File"));
434.                                 else
435.                                 {
436.                                     answers[reader.GetInt32(reader.GetOrdinal("idTask"))] = @"Lessons\tmpGraph" + reader.GetInt32(reader.GetOrdinal("idTask"));
437.                                     using (FileStream fs = new FileStream(Directory.GetCurrentDirectory() + @"\Lessons\tmpGraph" + reader.GetInt32(reader.GetOrdinal("idTask")), FileMode.Create, FileAccess.Write))
438.                                     {
439.                                         using (StreamWriter sw = new StreamWriter(fs))
440.                                         {
441.                                             sw.WriteLine(reader.GetString(reader.GetOrdinal("File")));
442.                                         }
443.                                     }
444.                                 }
445.                             }
446.                         }
447.                     }
448.                 }
449.             }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

450.         catch (Exception ex)
451.         {
452.             MessageBox.Show(ex.Message);
453.         }
454.         finally
455.         {
456.             conn.Close();
457.             conn.Dispose();
458.         }
459.     }
460.     /// <summary>
461.     /// Обработчик события нажатия кнопки ">" (следующий вопрос)
462.     /// </summary>
463.     private void next_Click(object sender, EventArgs e)
464.     {
465.         if (ShowAns.Enabled == true)
466.         {
467.             if (ShowAns.Visible == false) SaveThisAnswer();
468.             StartTask(int.Parse(this.Text.Split()[1]));
469.         }
470.         else
471.             StartTask(int.Parse(this.Text.Split()[1]), 1);
472.     }
473.     /// <summary>
474.     /// Обработчик события нажатия кнопки "<" (предыдущий вопрос)
475.     /// </summary>
476.     private void previous_Click(object sender, EventArgs e)
477.     {
478.         if (ShowAns.Enabled == true)
479.         {
480.             if (ShowAns.Visible == false) SaveThisAnswer();
481.             StartTask(int.Parse(this.Text.Split()[1]) - 2);
482.         }
483.         else
484.             StartTask(int.Parse(this.Text.Split()[1]) - 2, 1);
485.     }
486.     /// <summary>
487.     /// Обработчик события нажатия кнопки "Ответить"
488.     /// </summary>
489.     private void answer_Click(object sender, EventArgs e)
490.     {
491.         timer.Stop();
492.         SaveThisAnswer();
493.         ans = "";
494.         Answer.Visible = false;
495.         SaveAns.Visible = false;
496.         Repeat.Visible = true;
497.         ShowAns.Visible = true;
498.         ChooseSomeAnsPanel.Enabled = false;
499.         ChooseOneAnsPanel.Enabled = false;
500.         WriteAnsPanel.Enabled = false;
501.         oneGraph.Enabled = false;
502.         DrawingGraph.Enabled = false;
503.         for (int i = 0; i < answers.Length; i++)
504.         {
505.             if (answers[i] == test.tasks[i].answer) ans += "t";
506.             else if (answers[i] == (@"Lessons\tmpGraph" + i))

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

507.         {
508.             if (Compare(test.tasks[i].GetGraph(Directory.GetCurrentDirec
tory() + @"\Lessons\" + test.tasks[i].answer), test.tasks[i].GetGraph(Directo
entDirectory() + @"\Lessons\tmpGraph" + i)))
509.                 ans += "t";
510.             else ans += "f";
511.         }
512.         else ans += "f";
513.     }
514.
515.     int id = 0;
516.     try
517.     {
518.         conn.Open();
519.
520.         using (MySqlCommand cmd = new MySqlCommand("Select idUsers, Logi
n from users", conn))
521.         {
522.             using (DbDataReader reader = cmd.ExecuteReader())
523.             {
524.                 if (reader.HasRows)
525.                 {
526.                     while (reader.Read())
527.                     {
528.                         if (Properties.Settings.Default.login == reader.
GetString(reader.GetOrdinal("Login"))) id = reader.GetInt32(reader.GetOrdinal("idUsers"
));
529.                     }
530.                 }
531.             }
532.         }
533.         using (MySqlCommand cmd = new MySqlCommand("Insert into tests(id
Tests, Result, Users_idUsers, Date, Time) " + " values(@idTests, @result, @id, @date, @
time)", conn))
534.         {
535.             cmd.Parameters.Add(new MySqlParameter("@idTests", SqlDbType.Int));
536.             cmd.Parameters["@idTests"].Value = test.id;
537.             cmd.Parameters.Add(new MySqlParameter("@result", SqlDbType.Type
.VarChar));
538.             cmd.Parameters["@result"].Value = ans;
539.             cmd.Parameters.Add(new MySqlParameter("@id", SqlDbType.Int));
540.             cmd.Parameters["@id"].Value = id;
541.             cmd.Parameters.Add(new MySqlParameter("@date", SqlDbType.Type.D
ateTime));
542.             cmd.Parameters["@date"].Value = DateTime.Now;
543.             cmd.Parameters.Add(new MySqlParameter("@time", SqlDbType.Type.T
ime));
544.             cmd.Parameters["@time"].Value = time.TimeOfDay;
545.
546.             cmd.ExecuteNonQuery();
547.         }
548.     }
549.     catch (Exception ex)
550.     {
551.         MessageBox.Show(ex.Message);

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

552.         MessageBox.Show("Результат не сохранен!");
553.     }
554.     finally
555.     {
556.         conn.Close();
557.         conn.Dispose();
558.     }
559. }
560. /// <summary>
561. /// Очистка выбранных пунктов при выборе нового
562. /// </summary>
563. private void ChooseOneAnsPanel_SelectedIndexChanged(object sender, EventArgs e)
564. {
565.     for (int i = 0; i < ChooseOneAnsPanel.Items.Count; i++)
566.         if (i != ChooseOneAnsPanel.SelectedIndex) ChooseOneAnsPanel.SetItemChecked(i, false);
567. }
568. /// <summary>
569. /// Сравнение двух графов
570. /// </summary>
571. /// <param name="x">первый граф</param>
572. /// <param name="y">второй граф</param>
573. /// <returns>равны ли графы</returns>
574. private bool Compare(Graph x, Graph y)
575. {
576.     if (x.edges.Count != y.edges.Count) return false;
577.     else
578.     {
579.         bool f;
580.         int j;
581.         List<int> ed = new List<int> { };
582.         for (int i = 0; i < x.edges.Count; i++)
583.         {
584.             f = false;
585.             j = 0;
586.             while (!f & j < y.edges.Count)
587.             {
588.                 if (!ed.Contains(j) && x.edges[i].OId.Id == y.edges[j].O
Id.Id &&
589.                     x.edges[i].DId.Id == y.edges[j].DId.Id) f = true;
590.                 else j++;
591.             }
592.             if (j == y.edges.Count) return false;
593.             ed.Add(j);
594.         }
595.         if (ed.Count == x.edges.Count) return true;
596.         else return false;
597.     }
598. }
599. /// <summary>
600. /// Сохраняет ответ на текущий вопрос
601. /// </summary>
602. public void SaveThisAnswer()
603. {
604.     int num = int.Parse(this.Text.Split()[1]) - 1;
605.     switch (test.tasks[num].choice_ans)

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

606.      {
607.          case 1:
608.              string ans = "";
609.              foreach (int i in ChooseOneAnsPanel.CheckedIndices) { ans +=
        (i + 1); }
610.              if (ChooseOneAnsPanel.CheckedItems != null) answers[num] = a
        ns;
611.              break;
612.          case 2:
613.              ans = "";
614.              foreach (int i in ChooseSomeAnsPanel.CheckedIndices) { ans +
        = (i + 1); }
615.              if (ChooseSomeAnsPanel.CheckedItems != null) answers[num] =
        ans;
616.              break;
617.          case 3:
618.              if (WriteAnsPanel.Text != null) answers[num] = WriteAnsPanel
        .Text;
619.              break;
620.          case 4:
621.              DrawingGraph.Enabled = false;
622.              DrawingGraph.Enabled = true;
623.              answers[num] = @"Lessons\tmpGraph" + num;
624.              break;
625.          case 5:
626.              ans = "";
627.              foreach (int i in ChooseOneAnsPanel.CheckedIndices) { ans +=
        (i + 1); }
628.              if (ChooseOneAnsPanel.CheckedItems != null) answers[num] = a
        ns;
629.              break;
630.          case 6:
631.              ans = "";
632.              foreach (int i in ChooseSomeAnsPanel.CheckedIndices) { ans +
        = (i + 1); }
633.              if (ChooseSomeAnsPanel.CheckedItems != null) answers[num] =
        ans;
634.              break;
635.          case 7:
636.              if (WriteAnsPanel.Text != null) answers[num] = WriteAnsPanel
        .Text;
637.              break;
638.          default:
639.
640.              oneGraph.Enabled = false;
641.              oneGraph.Enabled = true;
642.              answers[num] = @"Lessons\tmpGraph" + num;
643.              break;
644.      }
645.  }
646.  /// <summary>
647.  /// Обработчик события нажатия кнопки "Повторить тест"
648.  /// </summary>
649.  private void Repeat_Click(object sender, EventArgs e)
650.  {
651.      this.BackColor = DefaultBackColor;
652.      ChooseSomeAnsPanel.Enabled = true;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

653.         ChooseOneAnsPanel.Enabled = true;
654.         WriteAnsPanel.Enabled = true;
655.         oneGraph.Enabled = true;
656.         DrawingGraph.Enabled = true;
657.         Repeat.Visible = false;
658.         ShowAns.Visible = false;
659.         answers = new string[test.tasks.Length];
660.         Answer.Visible = true;
661.         SaveAns.Visible = true;
662.         time = new DateTime(0, 0);
663.         timer.Start();
664.         StartTask(0);
665.     }
666.     /// <summary>
667.     /// Обработчик события нажатия кнопки "Показать ответы"
668.     /// </summary>
669.     private void ShowAns_Click(object sender, EventArgs e)
670.     {
671.         ShowAns.Enabled = false;
672.         StartTask(0, 1);
673.     }
674.     /// <summary>
675.     /// Обработчик события закрытия формы
676.     /// </summary>
677.     private void TestForm_FormClosing(object sender, FormClosingEventArgs e)
678.     {
679.         if (e.CloseReason.ToString() == "UserClosing")
680.         {
681.             DialogResult dialogResult = MessageBox.Show("Все несохраненные о
тветы не будут учтены! Вы точно хотите прервать тест?", "Выход", MessageBoxButtons.OKCa
ncl);
682.             if (dialogResult == DialogResult.OK)
683.             {
684.                 Application.Restart();
685.             }
686.             else
687.             {
688.                 e.Cancel = true;
689.             }
690.         }
691.     }
692.     /// <summary>
693.     /// Обработчик события изменения размера формы
694.     /// </summary>
695.     private void TestForm_SizeChanged(object sender, EventArgs e)
696.     {
697.         DrawingGraph.Location = new Point(this.Width / 2 + 1, 39);
698.         DrawingGraph.Size = new Size(5 * this.Width / 14, this.Height / 2 -
20);
699.         PictureGraph.Size = new Size(5 * this.Width / 14, this.Height / 2 -
20);
700.     }
701.     /// <summary>
702.     /// Такт таймера
703.     /// </summary>
704.     private void timer_Tick(object sender, EventArgs e)

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

705.        {
706.            time = time.AddSeconds(1);
707.            this.Text = this.Text.Substring(0, this.Text.Length - 7) + time.ToLo
ngTimeString();
708.        }
709.        /// <summary>
710.        /// Открывает окно помощи по тесту
711.        /// </summary>
712.        private void помощьToolStripMenuItem_Click(object sender, EventArgs e)
713.        {
714.            timer.Stop();
715.            webBrowser.Navigate(Directory.GetCurrentDirectory() + @"Lessons\hel
p.html");
716.            webBrowser.Visible = true;
717.            вернутьяКТестуToolStripMenuItem.Visible = true;
718.            помощьToolStripMenuItem.Visible = false;
719.            форматРёберToolStripMenuItem.Visible = false;
720.            next.Visible = false;
721.            previous.Visible = false;
722.        }
723.        /// <summary>
724.        /// Возобновляет тест
725.        /// </summary>
726.        private void вернутьяКТестуToolStripMenuItem_Click(object sender, Event
Args e)
727.        {
728.            webBrowser.Visible = false;
729.            вернутьяКТестуToolStripMenuItem.Visible = false;
730.            помощьToolStripMenuItem.Visible = true;
731.            форматРёберToolStripMenuItem.Visible = true;
732.            next.Visible = true;
733.            previous.Visible = true;
734.            timer.Start();
735.        }
736.        /// <summary>
737.        /// Делает ребра графа прямыми
738.        /// </summary>
739.        private void прямыеToolStripMenuItem_Click(object sender, EventArgs e)
740.        {
741.            EdgeAuto = false;
742.            автоToolStripMenuItem.BackColor = прямыеToolStripMenuItem.BackColor;
743.            прямыеToolStripMenuItem.BackColor = Color.Aqua;
744.        }
745.        /// <summary>
746.        /// Делает ребра графа автовыстраиваемыми
747.        /// </summary>
748.        private void автоToolStripMenuItem_Click(object sender, EventArgs e)
749.        {
750.            EdgeAuto = true;
751.            прямыеToolStripMenuItem.BackColor = автоToolStripMenuItem.BackColor;
752.            автоToolStripMenuItem.BackColor = Color.Aqua;
753.        }
754.    }
755. }

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. ТЕКСТ СКРИПТА БАЗЫ ДАННЫХ MySQL

```

1. CREATE DATABASE IF NOT EXISTS `mydb` /*!40100 DEFAULT CHARACTER SET utf8 */;
2. USE `mydb`;
3. -- MySQL dump 10.13 Distrib 5.7.17, for Win64 (x86_64)
4. --
5. -- Host: 127.0.0.1 Database: mydb
6. --
7. -- Server version 5.6.25-log
8.
9. /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
10. /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
11. /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
12. /*!40101 SET NAMES utf8 */;
13. /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
14. /*!40103 SET TIME_ZONE='+00:00' */;
15. /*!40104 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
16. /*!40104 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
17. /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
18. /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
19.
20. --
21. -- Table structure for table `files`
22. --
23.
24. DROP TABLE IF EXISTS `files`;
25. /*!40101 SET @saved_cs_client = @@character_set_client */;
26. /*!40101 SET character_set_client = utf8 */;
27. CREATE TABLE `files` (
28.   `idTest` int(2) NOT NULL,
29.   `idTask` int(2) NOT NULL,
30.   `File` longtext NOT NULL,
31.   `Users_idUsers` int(11) NOT NULL,
32.   KEY `fk_Files_Users1_idx` (`Users_idUsers`),
33.   CONSTRAINT `fk_Files_Users1` FOREIGN KEY (`Users_idUsers`) REFERENCES `users` (`idUser
rs`) ON DELETE NO ACTION ON UPDATE NO ACTION
34. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
35. /*!40101 SET character_set_client = @saved_cs_client */;
36.
37. --
38. -- Dumping data for table `files`
39. --
40.
41. LOCK TABLES `files` WRITE;
42. /*!40000 ALTER TABLE `files` DISABLE KEYS */;
43. /*!40000 ALTER TABLE `files` ENABLE KEYS */;
44. UNLOCK TABLES;
45.
46. --
47. -- Table structure for table `tests`
48. --
49.
50. DROP TABLE IF EXISTS `tests`;
51. /*!40101 SET @saved_cs_client = @@character_set_client */;
52. /*!40101 SET character_set_client = utf8 */;
53. CREATE TABLE `tests` (

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата


```

54. `idTests` int(2) NOT NULL,
55. `Result` varchar(20) DEFAULT NULL,
56. `Users_idUsers` int(11) NOT NULL,
57. `Date` datetime NOT NULL,
58. `Time` time(6) NOT NULL,
59. KEY `fk_Tests_Users_idx` (`Users_idUsers`),
60. CONSTRAINT `fk_Tests_Users` FOREIGN KEY (`Users_idUsers`) REFERENCES `users` (`idUser
s`) ON DELETE NO ACTION ON UPDATE NO ACTION
61. ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
62. /*!40101 SET character_set_client = @saved_cs_client */;
63.
64. --
65. -- Dumping data for table `tests`
66. --
67.
68. LOCK TABLES `tests` WRITE;
69. /*!40000 ALTER TABLE `tests` DISABLE KEYS */;
70. /*!40000 ALTER TABLE `tests` ENABLE KEYS */;
71. UNLOCK TABLES;
72.
73. --
74. -- Table structure for table `users`
75. --
76.
77. DROP TABLE IF EXISTS `users`;
78. /*!40101 SET @saved_cs_client      = @@character_set_client */;
79. /*!40101 SET character_set_client = utf8 */;
80. CREATE TABLE `users` (
81.   `idUsers` int(11) NOT NULL AUTO_INCREMENT,
82.   `Login` varchar(15) NOT NULL,
83.   `Password` int(11) NOT NULL,
84.   PRIMARY KEY (`idUsers`)
85. ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
86. /*!40101 SET character_set_client = @saved_cs_client */;
87.
88. /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
89. /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
90. /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
91. /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
92. /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
93. /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
94. /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
95.
96. -- Dump completed on 2018-05-13 13:08:14

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.201-78. Техническое задание. Требование к содержанию и оформлению. // Единая система программной документации. – Москва: Издательство стандартов, 2005.
2. ГОСТ 19.404-79. Пояснительная записка. // Единая система программной документации. – Москва: Издательство стандартов, 2005.
3. ГОСТ 19.301-78. Программа и методика испытаний. // Единая система программной документации. – Москва: Издательство стандартов, 2005.
4. ГОСТ 19.505-79. Руководство оператора. // Единая система программной документации. – Москва: Издательство стандартов, 2005.
5. ГОСТ 19.401-78. Текст программы. // Единая система программной документации. – Москва: Издательство стандартов, 2005.
6. ГОСТ 19.106-78. Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – Москва: Издательство стандартов, 2005.
7. ГОСТ 19.101-77. Виды программ и программных документов. // Единая система программной документации. – Москва: Издательство стандартов, 2005.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата