**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

СОГЛАСОВАНО УТВЕРЖДАЮ  
 Профессор департамента Академический руководитель

программной инженерии факультета образовательной программы

компьютерных наук «Программная инженерия»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Подбельский \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г. «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.



**Программа активации таблицы Д. И. Менделеева**

**Текст программы**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.15-01 12 01-1-ЛУ**

**Исполнитель**

Студент группы БПИ173

\_\_\_\_\_\_\_\_\_\_\_/К.Г. Кожакин/

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

**Москва 2019**

**УТВЕРЖДЕН**

**RU.17701729.04.15-01 12 01-1-ЛУ**

**Программа активации таблицы Д. И. Менделеева**

**Текст программы**

**RU.17701729.04.15-01 12 01-1**

**Листов 77**

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

**Москва 2019**

Оглавление

[**1.** **Класс MendeleevTable:** 4](#_Toc8818158)

[ **MendeleevTable.xaml** 4](#_Toc8818159)

[ **MendeleevTable.cs** 7](#_Toc8818160)

[**2.** **Класс ComposSystemContent:** 16](#_Toc8818161)

[ **ComposSystemContent.xaml** 16](#_Toc8818162)

[ **ComposSystemContent.cs** 18](#_Toc8818163)

[**3.** **Класс ElemContent:** 27](#_Toc8818164)

[ **ElemContent.xaml** 27](#_Toc8818165)

[ **ElemContent.cs** 29](#_Toc8818166)

[**4.** **Класс CreateComposSystem:** 35](#_Toc8818167)

[ **CreateComposSystem.xaml** 35](#_Toc8818168)

[ **CreateComposSystem.cs** 37](#_Toc8818169)

[**5.** **Класс ColumnAddWindow:** 38](#_Toc8818170)

[ **ColumnAddWindow.xaml** 38](#_Toc8818171)

[ **ColumnAddWindow.cs** 39](#_Toc8818172)

[**6.** **Класс FormulaInput:** 40](#_Toc8818173)

[ **FormulaInput.xaml** 40](#_Toc8818174)

[ **FormulaInput.cs** 42](#_Toc8818175)

[**7.** **Класс Help:** 43](#_Toc8818176)

[ **Help.xaml** 43](#_Toc8818177)

[ **Help.cs** 44](#_Toc8818178)

[**8.** **Классы HelperClasses:** 48](#_Toc8818179)

[**8.1** **Composition.cs** 48](#_Toc8818180)

[**8.2** **DataGridHelper.cs** 50](#_Toc8818181)

[**8.3** **HotKey.cs** 55](#_Toc8818182)

[**8.4** **MathParser.cs** 60](#_Toc8818183)

[**8.5** **Pair.cs** 69](#_Toc8818184)

[**8.6** **Strings.cs:** 70](#_Toc8818185)

[ **Strings** 70](#_Toc8818186)

[ **StringHelper** 71](#_Toc8818187)

[**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ** 76](#_Toc8818188)

[**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ** 77](#_Toc8818189)

1. **Класс MendeleevTable:**
   * **MendeleevTable.xaml**
2. **<Window** x:Name="MendeleevTable1" x:Class="Activision\_Mendeleyev\_table.MendeleevTable"
3. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6. **xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"**
7. mc:Ignorable="d"
8. Icon="icon.jpeg"
9. Title="Таблица Менделеева" Height="720" Width="1280" ScrollViewer.CanContentScroll="False" Loaded="Window\_Loaded" SizeChanged="Window\_SizeChanged" MinWidth="1280" MinHeight="720" FontSize="18" Closing="Window\_Closing"**>**
10. **<Grid** ScrollViewer.CanContentScroll="False"**>**
11. **<Grid.RowDefinitions>**
12. **<RowDefinition** Height="25" **/>**
13. **<RowDefinition** Height="\*"**/>**
14. **</Grid.RowDefinitions>**
15. **<Menu** x:Name="MainMenu" Grid.Row="0" Height="25" VerticalAlignment="Top"**>**
16. **<Button Name="Composition" Content="Работа с соединением" Click="Composition\_Click" />**
17. **<Button** Name="Compositions" Content="Работа с системой соединений" Click="Compositions\_Click" **/>**
18. **<Separator** Name="sep" Width="850" Opacity="0"**/>**
19. **<Button** Name="Help" Content="Справка" Click="Help\_Click"**/>**
20. **</Menu>**
21. **<DataGrid Grid.Row="1" x:Name="table" AutoGenerateColumns="False" ColumnWidth="2\*" VerticalGridLinesBrush="White"**
22. HorizontalGridLinesBrush="White" CanUserReorderColumns="False" CanUserSortColumns="False" CanUserAddRows="False"
23. CanUserDeleteRows="False" CanUserResizeColumns="False" CanUserResizeRows="False" IsReadOnly="True" SelectionUnit="Cell"
24. SelectionMode="Single" GridLinesVisibility="None" ScrollViewer.CanContentScroll="False"**>**
25. **<DataGrid.Columns>**
26. **<DataGridTextColumn Header="" Width="1\*"/>**
27. **<DataGridTextColumn** Header="Ia" Binding="{Binding Ia}"**/>**
28. **<DataGridTextColumn** Header="IIa" Binding="{Binding IIa}"**/>**
29. **<DataGridTextColumn** Header="IIIa" Binding="{Binding IIIa}"**/>**
30. **<DataGridTextColumn** Header="IVa" Binding="{Binding IVa}"**/>**
31. **<DataGridTextColumn Header="Va" Binding="{Binding Va}"/>**
32. **<DataGridTextColumn** Header="VIa" Binding="{Binding VIa}"**/>**
33. **<DataGridTextColumn** Header="VIIa" Binding="{Binding VIIa}"**/>**
34. **<DataGridTextColumn** Header="" Binding="{Binding VIIIa\_1}"**/>**
35. **<DataGridTextColumn** Header="VIIIa" Binding="{Binding VIIIa\_2}"**/>**
36. **<DataGridTextColumn Header="" Binding="{Binding VIIIa\_3}"/>**
37. **<DataGridTextColumn** Header="Iб" Binding="{Binding Ib}"**/>**
38. **<DataGridTextColumn** Header="IIб" Binding="{Binding IIb}"**/>**
39. **<DataGridTextColumn** Header="IIIб" Binding="{Binding IIIb}"**/>**
40. **<DataGridTextColumn** Header="IVб" Binding="{Binding IVb}"**/>**
41. **<DataGridTextColumn Header="Vб" Binding="{Binding Vb}"/>**
42. **<DataGridTextColumn** Header="VIб" Binding="{Binding VIb}"**/>**
43. **<DataGridTextColumn** Header="VIIб" Binding="{Binding VIIb}"**/>**
44. **<DataGridTextColumn** Header="VIIIб" Binding="{Binding VIIIb}"**/>**
45. **<DataGridTextColumn** Header="" Width="1\*"**/>**
46. **</DataGrid.Columns>**
47. **<DataGrid.ColumnHeaderStyle>**
48. **<Style** TargetType="{x:Type DataGridColumnHeader}"**>**
49. **<Setter** Property="SeparatorBrush" Value="Transparent"**/>**
50. **<Setter** Property="Thumb.HorizontalContentAlignment" Value="Stretch"**/>**
51. **<Setter Property="Thumb.VerticalContentAlignment" Value="Stretch"/>**
52. **</Style>**
53. **</DataGrid.ColumnHeaderStyle>**
54. **<DataGrid.CellStyle>**
55. **<Style** TargetType="DataGridCell"**>**
56. **<Setter Property="FrameworkElement.HorizontalAlignment" Value="Stretch"/>**
57. **<Setter** Property="FrameworkElement.VerticalAlignment" Value="Stretch"**/>**
58. **<EventSetter** Event="MouseDoubleClick" Handler="DataGridCell\_MouseDoubleClick"**/>**
59. **<Style.Triggers>**
60. **<Trigger** Property="IsSelected" Value="True"**>**
61. **<Setter Property="Background" Value="Transparent"/>**
62. **<Setter** Property="Foreground" Value="Black" **/>**
63. **<Setter** Property="BorderThickness" Value="0"**/>**
64. **</Trigger>**
65. **</Style.Triggers>**
66. **</Style>**
67. **</DataGrid.CellStyle>**
68. **<DataGrid.RowHeaderStyle>**
69. **<Style** TargetType="{x:Type DataGridRowHeader}"**>**
70. **<Setter** Property="Content" Value=""**/>**
71. **<Setter Property="Width" Value="0"/>**
72. **</Style>**
73. **</DataGrid.RowHeaderStyle>**
74. **</DataGrid>**
75. **<Label** Name="M" Content="M-Металлы" Foreground="HotPink" Grid.Row="1" Width="200" Height="35" RenderTransformOrigin="0.5,0.5" **>**
76. **<Label.RenderTransform>**
77. **<TransformGroup>**
78. **<ScaleTransform/>**
79. **<SkewTransform/>**
80. **<RotateTransform/>**
81. **<TranslateTransform X="-200" Y="-160"/>**
82. **</TransformGroup>**
83. **</Label.RenderTransform>**
84. **</Label>**
85. **<Label** Name="T" Content="T-Металлы" Foreground="CornflowerBlue" Grid.Row="1" Width="200" Height="35" RenderTransformOrigin="0.5,0.5" **>**
86. **<Label.RenderTransform>**
87. **<TransformGroup>**
88. **<ScaleTransform/>**
89. **<SkewTransform/>**
90. **<RotateTransform/>**
91. **<TranslateTransform X="-200" Y="160"/>**
92. **</TransformGroup>**
93. **</Label.RenderTransform>**
94. **</Label>**
95. **<Label** Name="B1" Content="B1-Металлы" Grid.Row="1" Width="200" Height="35" RenderTransformOrigin="0.5,0.5" **>**
96. **<Label.RenderTransform>**
97. **<TransformGroup>**
98. **<ScaleTransform/>**
99. **<SkewTransform/>**
100. **<RotateTransform/>**
101. **<TranslateTransform Y="160" X="230"/>**
102. **</TransformGroup>**
103. **</Label.RenderTransform>**
104. **</Label>**
105. **<Label** Name="B2" Content="B2-Металлы" Grid.Row="1" Width="200" Height="35" RenderTransformOrigin="0.5,0.5" **>**
106. **<Label.RenderTransform>**
107. **<TransformGroup>**
108. **<ScaleTransform/>**
109. **<SkewTransform/>**
110. **<RotateTransform/>**
111. **<TranslateTransform Y="160" X="477"/>**
112. **</TransformGroup>**
113. **</Label.RenderTransform>**
114. **</Label>**
115. **<Label** Name="R" Content="R-Лантаноиды" Grid.Row="1" Width="200" Height="35" RenderTransformOrigin="0.5,0.5" **>**
116. **<Label.RenderTransform>**
117. **<TransformGroup>**
118. **<ScaleTransform/>**
119. **<SkewTransform/>**
120. **<RotateTransform/>**
121. **<TranslateTransform Y="215" X="470"/>**
122. **</TransformGroup>**
123. **</Label.RenderTransform>**
124. **</Label>**
125. **<Label** Name="TR" Content="TR-Актиноиды" Grid.Row="1" Width="200" Height="35" RenderTransformOrigin="0.5,0.5" **>**
126. **<Label.RenderTransform>**
127. **<TransformGroup>**
128. **<ScaleTransform/>**
129. **<SkewTransform/>**
130. **<RotateTransform/>**
131. **<TranslateTransform X="470" Y="275"/>**
132. **</TransformGroup>**
133. **</Label.RenderTransform>**
134. **</Label>**
135. **</Grid>**
136. **</Window>**
     * **MendeleevTable.cs**
137. **using** Activision\_Mendeleyev\_table.HelperClasses;
138. **using** **static** Activision\_Mendeleyev\_table.HelperClasses.DataGridHelper;
139. **using** System.Collections.Generic;
140. **using** System.Windows;
141. **using System.Windows.Controls;**
142. **using** System.Windows.Input;
143. **using** System.Windows.Media;
145. **namespace** Activision\_Mendeleyev\_table
146. **{**
147. */// <summary>*
148. */// Логика взаимодействия для MendeleevTable.xaml*
149. */// </summary>*
150. **public** **partial** **class** MendeleevTable : Window
151. **{**
152. */// <summary>*
153. */// Лист элементов*
154. */// </summary>*
155. **public** **static** List<Composition> Elems = new List<Composition>();
157. */// <summary>*
158. */// Лист соединений*
159. */// </summary>*
160. **public** **static** List<Composition> Compos = new List<Composition>();
162. */// <summary>*
163. */// Лист систем соединений*
164. */// </summary>*
165. **public** **static** List<System.Data.DataTable> BinarySistem = new List<System.Data.DataTable>();
167. */// <summary>*
168. */// Конструктор главного окна*
169. */// </summary>*
170. **public** MendeleevTable()
171. **{**
172. InitializeComponent();
173. *//Привязка горячей клавише F1 к методу OnF1Handler*
174. new HotKey(Key.F1, KeyModifier.None, OnF1Handler);
176. **List<Strings> strings = new List<Strings>(10)**
177. {
178. new Strings(),
179. new Strings("Li", "Be"),
180. new Strings("Na", "Mg"),
181. **new Strings("K", "Ca", "Sc", "Ti", "V", "Cr", "Mn", "Fe", "Co", "Ni", "Cu", "Zn", "Ga", "Ge", "As", "Se", "Br", "Kr"),**
182. new Strings("Rb", "Sr", "Y", "Zr", "Nb", "Mo", "Tc", "Ru", "Rh", "Pd", "Ag", "Cd", "In", "Sn", "Sb", "Te", "I", "Xe"),
183. new Strings("Cs", "Ba", "\*La", "Hf", "Ta", "W", "Re", "Os", "Ir", "Pt", "Au", "Hg", "Tl", "Pb", "Bi", "Po", "At", "Rn"),
184. new Strings("Fr", "Ra", "\*\*Ac"),
185. new Strings(),
186. **new Strings("\*Ce", "Pr", "Nd", "Pm", "Sm", "Eu", "Gd", "Tb", "Dy", "Ho", "Er", "Tm", "Yb", "Lu"),**
187. new Strings("\*\*", "Th", "Pa", "U", "Np", "Pu", "Am", "Cm", "Bk", "Cf", "Es", "Fm", "Md", "-")
188. };
190. strings[0].VIIb = "H";
191. **strings[0].VIIIb = "He";**
192. strings[1].IIIb = "B";
193. strings[1].IVb = "C";
194. strings[1].Vb = "N";
195. strings[1].VIb = "O";
196. **strings[1].VIIb = "F";**
197. strings[1].VIIIb = "Ne";
198. strings[2].IIIb = "Al";
199. strings[2].IVb = "Si";
200. strings[2].Vb = "P";
201. **strings[2].VIb = "S";**
202. strings[2].VIIb = "Cl";
203. strings[2].VIIIb = "Ar";
204. strings[6].VIIIb = "I газ";
206. **table.Items.Clear();**
207. table.ItemsSource = strings;
209. Deserialize("BinarySistems.xml", **ref** BinarySistem);
210. Deserialize("Compositions.xml", **ref** Compos);
212. **if** (!Deserialize("Elems.xml", **ref** Elems))
213. {
214. Elems.**Add**(new Composition("H"));
215. Elems.**Add**(new Composition("Li"));
216. **Elems.Add(new Composition("Be"));**
217. Elems.**Add**(new Composition("Na"));
218. Elems.**Add**(new Composition("Mg"));
219. Elems.**Add**(new Composition("K"));
220. Elems.**Add**(new Composition("Ca"));
221. **Elems.Add(new Composition("Sc"));**
222. Elems.**Add**(new Composition("Ti"));
223. Elems.**Add**(new Composition("V"));
224. Elems.**Add**(new Composition("Cr"));
225. Elems.**Add**(new Composition("Mn"));
226. **Elems.Add(new Composition("Fe"));**
227. Elems.**Add**(new Composition("Co"));
228. Elems.**Add**(new Composition("Ni"));
229. Elems.**Add**(new Composition("Cu"));
230. Elems.**Add**(new Composition("Zn"));
231. **Elems.Add(new Composition("Ga"));**
232. Elems.**Add**(new Composition("Ge"));
233. Elems.**Add**(new Composition("As"));
234. Elems.**Add**(new Composition("Se"));
235. Elems.**Add**(new Composition("Br"));
236. **Elems.Add(new Composition("Kr"));**
237. Elems.**Add**(new Composition("Rb"));
238. Elems.**Add**(new Composition("Sr"));
239. Elems.**Add**(new Composition("Y"));
240. Elems.**Add**(new Composition("Zr"));
241. **Elems.Add(new Composition("Nb"));**
242. Elems.**Add**(new Composition("Mo"));
243. Elems.**Add**(new Composition("Tc"));
244. Elems.**Add**(new Composition("Ru"));
245. Elems.**Add**(new Composition("Ac"));
246. **Elems.Add(new Composition("Rh"));**
247. Elems.**Add**(new Composition("Pd"));
248. Elems.**Add**(new Composition("Ag"));
249. Elems.**Add**(new Composition("Cd"));
250. Elems.**Add**(new Composition("In"));
251. **Elems.Add(new Composition("Sn"));**
252. Elems.**Add**(new Composition("Sb"));
253. Elems.**Add**(new Composition("Te"));
254. Elems.**Add**(new Composition("I"));
255. Elems.**Add**(new Composition("Xe"));
256. **Elems.Add(new Composition("Cs"));**
257. Elems.**Add**(new Composition("Ba"));
258. Elems.**Add**(new Composition("Hf"));
259. Elems.**Add**(new Composition("Ta"));
260. Elems.**Add**(new Composition("W"));
261. **Elems.Add(new Composition("Re"));**
262. Elems.**Add**(new Composition("Os"));
263. Elems.**Add**(new Composition("Ir"));
264. Elems.**Add**(new Composition("Pt"));
265. Elems.**Add**(new Composition("Au"));
266. **Elems.Add(new Composition("Hg"));**
267. Elems.**Add**(new Composition("Tl"));
268. Elems.**Add**(new Composition("Pb"));
269. Elems.**Add**(new Composition("Bi"));
270. Elems.**Add**(new Composition("Po"));
271. **Elems.Add(new Composition("At"));**
272. Elems.**Add**(new Composition("Rn"));
273. Elems.**Add**(new Composition("Fr"));
274. Elems.**Add**(new Composition("Ra"));
275. Elems.**Add**(new Composition("He"));
276. **Elems.Add(new Composition("B"));**
277. Elems.**Add**(new Composition("C"));
278. Elems.**Add**(new Composition("N"));
279. Elems.**Add**(new Composition("O"));
280. Elems.**Add**(new Composition("F"));
281. **Elems.Add(new Composition("Ne"));**
282. Elems.**Add**(new Composition("Al"));
283. Elems.**Add**(new Composition("Si"));
284. Elems.**Add**(new Composition("P"));
285. Elems.**Add**(new Composition("S"));
286. **Elems.Add(new Composition("Cl"));**
287. Elems.**Add**(new Composition("Ar"));
288. Elems.**Add**(new Composition("I газ"));
289. Elems.**Add**(new Composition("Ce"));
290. Elems.**Add**(new Composition("Pr"));
291. **Elems.Add(new Composition("Nd"));**
292. Elems.**Add**(new Composition("Pm"));
293. Elems.**Add**(new Composition("Sm"));
294. Elems.**Add**(new Composition("Eu"));
295. Elems.**Add**(new Composition("Gd"));
296. **Elems.Add(new Composition("Tb"));**
297. Elems.**Add**(new Composition("Dy"));
298. Elems.**Add**(new Composition("Ho"));
299. Elems.**Add**(new Composition("Er"));
300. Elems.**Add**(new Composition("Tm"));
301. **Elems.Add(new Composition("Yb"));**
302. Elems.**Add**(new Composition("Lu"));
303. Elems.**Add**(new Composition("Th"));
304. Elems.**Add**(new Composition("Pa"));
305. Elems.**Add**(new Composition("U"));
306. **Elems.Add(new Composition("Np"));**
307. Elems.**Add**(new Composition("Pu"));
308. Elems.**Add**(new Composition("Am"));
309. Elems.**Add**(new Composition("Cm"));
310. Elems.**Add**(new Composition("Bk"));
311. **Elems.Add(new Composition("Cf"));**
312. Elems.**Add**(new Composition("Es"));
313. Elems.**Add**(new Composition("Fm"));
314. Elems.**Add**(new Composition("Md"));
315. Elems.**Add**(new Composition("La"));
317. }
318. }
320. */// <summary>*
321. ***/// Открывает окно работы с системой соединений***
322. */// </summary>*
323. **private** **void** Compositions\_Click(**object** sender, RoutedEventArgs e)
324. {
325. new CreateComposSystem(**false**).ShowDialog();
326. **}**
328. */// <summary>*
329. */// Открывает окно работы с соединением*
330. */// </summary>*
331. **private void Composition\_Click(object sender, RoutedEventArgs e)**
332. {
333. new CreateComposSystem(**true**).ShowDialog();
334. }
336. **private void Window\_Loaded(object sender, RoutedEventArgs e)**
337. {
338. **for** (**int** i = 3; i < 12; i++)
339. **for** (**int** j = 3; j < 7; j++)
340. BrushCell(j, i, Brushes.LightBlue, table);
342. **for** (**int** i = 1; i <= 2; i++)
343. **for** (**int** j = 1; j < 7; j++)
344. BrushCell(j, i, Brushes.LightPink, table);
345. BrushCell(2, 13, Brushes.LightPink, table);
347. **for** (**int** i = 0; i <= 6; i++)
348. BrushCell(i, 18, Brushes.Orange, table);
350. **for** (**int** i = 12; i <= 14; i++)
351. **for (int j = 3; j < 7; j++)**
352. BrushCell(j, i, Brushes.AntiqueWhite, table);
353. BrushCell(3, 14, Brushes.White, table);
355. BrushCell(1, 13, Brushes.LightGreen, table);
356. **BrushCell(2, 14, Brushes.LightGreen, table);**
357. BrushCell(3, 15, Brushes.LightGreen, table);
358. BrushCell(4, 16, Brushes.LightGreen, table);
359. BrushCell(5, 17, Brushes.LightGreen, table);
361. **}**
363. **private** **void** Window\_SizeChanged(**object** sender, SizeChangedEventArgs e)
364. {
365. DataGridCell cell;
366. **double height = e.NewSize.Height;**
367. **double** width = e.NewSize.Width;
368. sep.Width = 850 + (width - 1280);
369. **for** (**int** i = 0; i <= 19; i++)
370. {
371. **for (int j = 0; j < 10; j++)**
372. {
373. cell = GetCell(j, i, table);
374. cell.Height = 68 + (height - 780) / 10;
375. cell.FontSize = 20 + (height - 780) / 100;
376. **}**
377. }
378. M.FontSize = 18 + (height - 780) / 100;
379. T.FontSize = 18 + (height - 780) / 100;
380. B1.FontSize = 18 + (height - 780) / 100;
381. **B2.FontSize = 18 + (height - 780) / 100;**
382. R.FontSize = 18 + (height - 780) / 100;
383. TR.FontSize = 18 + (height - 780) / 100;
384. TranslateTransform transform = new TranslateTransform
385. {
386. **X = -180 - (width - 1280) / 4.5,**
387. Y = -170 - (height - 780) / 4.5
388. };
389. M.RenderTransform = transform;
390. transform = new TranslateTransform
391. **{**
392. X = -180 - (width - 1280) / 4.5,
393. Y = 180 + (height - 780) / 4.5
394. };
395. T.RenderTransform = transform;
396. **transform = new TranslateTransform**
397. {
398. X = 260 + (width - 1280) / 6,
399. Y = 180 + (height - 780) / 4.5
400. };
401. **B1.RenderTransform = transform;**
402. transform = new TranslateTransform
403. {
404. X = 500 + (width - 1280) / 2.5,
405. Y = 180 + (height - 780) / 4.5
406. **};**
407. B2.RenderTransform = transform;
408. transform = new TranslateTransform
409. {
410. X = 490 + (width - 1280) / 2.5,
411. **Y = 230 + (height - 780) / 3**
412. };
413. R.RenderTransform = transform;
414. transform = new TranslateTransform
415. {
416. **X = 490 + (width - 1280) / 2.5,**
417. Y = 295 + (height - 780) / 2.5
418. };
419. TR.RenderTransform = transform;
420. }
422. */// <summary>*
423. */// Открывает окно работы с элементом*
424. */// </summary>*
425. **private** **void** DataGridCell\_MouseDoubleClick(**object** sender, MouseButtonEventArgs e)
426. **{**
427. **var** dataGridCellTarget = (DataGridCell)sender;
428. **if** (dataGridCellTarget != **null**)
429. {
430. TextBlock text = dataGridCellTarget.Content **as** TextBlock;
431. **if (text.Text != "" && text.Text != "-" && text.Text != "\*\*")**
432. new ElemContent(text.Text).ShowDialog();
433. }
434. }
436. **private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)**
437. {
438. **var** mbResult = MessageBox.Show("Вы точно хотите выйти из приложения?", "Выход", MessageBoxButton.YesNo, MessageBoxImage.Question);
439. **if** (mbResult == MessageBoxResult.Yes)
440. Application.Current.Shutdown();
441. **else**
442. e.Cancel = **true**;
443. }
445. */// <summary>*
446. ***/// Открывает окно справки***
447. */// </summary>*
448. **private** **void** Help\_Click(**object** sender, RoutedEventArgs e)
449. {
450. new Help().Show();
451. **}**
453. */// <summary>*
454. */// Открывает окно справки*
455. */// </summary>*
456. ***/// <param name="hotKey">горячая клавиша для вызова метода</param>***
457. **private** **static** **void** OnF1Handler(HotKey hotKey)
458. {
459. new Help().Show();
460. }
461. **}**
462. }
463. **Класс ComposSystemContent:**
     * **ComposSystemContent.xaml**
464. **<Window** x:Class="Activision\_Mendeleyev\_table.ComposSystemContent"
465. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
466. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
467. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
468. **xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"**
469. mc:Ignorable="d"
470. Icon="icon.jpeg"
471. Title="ComposSystemContent" Height="450" Width="1050" WindowStartupLocation="CenterScreen" MinHeight="450" MinWidth="1050" Closing="Window\_Closing" SizeChanged="Window\_SizeChanged"**>**
472. **<Grid>**
473. **<Grid.RowDefinitions>**
474. **<RowDefinition** Height="\*" **/>**
475. **<RowDefinition** Height="50"**/>**
476. **</Grid.RowDefinitions>**
477. **<DataGrid** x:Name="ComposSystemTable" Grid.Row="0" IsReadOnly="True" CanUserSortColumns="False" CanUserReorderColumns="False" CanUserResizeColumns="False" ColumnWidth="\*"
478. **AutoGenerateColumns="False" CanUserAddRows="False" CanUserDeleteRows="False" CanUserResizeRows="False" CellEditEnding="ComposSystemTable\_CellEditEnding"/>**
479. **<Button** Name="AddFormul" Grid.Row="1" Visibility="Hidden" Content="Добавить формулу" Click="AddFormul\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5"**>**
480. **<Button.RenderTransform>**
481. **<TransformGroup>**
482. **<ScaleTransform/>**
483. **<SkewTransform/>**
484. **<RotateTransform/>**
485. **<TranslateTransform** X="20"**/>**
486. **</TransformGroup>**
487. **</Button.RenderTransform>**
488. **</Button>**
489. **<Button** Name="AddColumn" Grid.Row="1" Visibility="Hidden" Content="Добавить столбец" Click="AddColumn\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5"**>**
490. **<Button.RenderTransform>**
491. **<TransformGroup>**
492. **<ScaleTransform/>**
493. **<SkewTransform/>**
494. **<RotateTransform/>**
495. **<TranslateTransform** X="190"**/>**
496. **</TransformGroup>**
497. **</Button.RenderTransform>**
498. **</Button>**
499. **<Button** Name="AddRow" Grid.Row="1" Visibility="Hidden" Content="Добавить строку" Click="AddRow\_Click" Width="150" Height="40" RenderTransformOrigin="0.5,0.5" HorizontalAlignment="Left"**>**
500. **<Button.RenderTransform>**
501. **<TransformGroup>**
502. **<ScaleTransform/>**
503. **<SkewTransform/>**
504. **<RotateTransform/>**
505. **<TranslateTransform** X="360"**/>**
506. **</TransformGroup>**
507. **</Button.RenderTransform>**
508. **</Button>**
509. **<Button** Name="DelColumn" Grid.Row="1" Visibility="Hidden" Content="Удалить столбец" Click="DelColumn\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" IsEnabled="False"**>**
510. **<Button.RenderTransform>**
511. **<TransformGroup>**
512. **<ScaleTransform/>**
513. **<SkewTransform/>**
514. **<RotateTransform/>**
515. **<TranslateTransform** X="530"**/>**
516. **</TransformGroup>**
517. **</Button.RenderTransform>**
518. **</Button>**
519. **<Button** Name="DelRow" Grid.Row="1" Visibility="Hidden" Content="Удалить строку" Click="DelRow\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" IsEnabled="False"**>**
520. **<Button.RenderTransform>**
521. **<TransformGroup>**
522. **<ScaleTransform/>**
523. **<SkewTransform/>**
524. **<RotateTransform/>**
525. **<TranslateTransform** X="700"**/>**
526. **</TransformGroup>**
527. **</Button.RenderTransform>**
528. **</Button>**
529. **<Button** Name="Calculate" Grid.Row="1" Visibility="Hidden" Content="Раcсчитать и сохранить" Click="Calculate\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5"**>**
530. **<Button.RenderTransform>**
531. **<TransformGroup>**
532. **<ScaleTransform/>**
533. **<SkewTransform/>**
534. **<RotateTransform/>**
535. **<TranslateTransform** X="870"**/>**
536. **</TransformGroup>**
537. **</Button.RenderTransform>**
538. **</Button>**
539. **<Button** Name="EditTable" Grid.Row="1" Content="Редактировать таблицу" Click="EditTable\_Click" Width="150" Height="40" **/>**
540. **</Grid>**
541. **</Window>**
     * **ComposSystemContent.cs**
542. **using** Activision\_Mendeleyev\_table.HelperClasses;
543. **using** System;
544. **using** System.Collections.Generic;
545. **using** System.Data;
546. **using System.Windows;**
547. **using** System.Windows.Controls;
548. **using** System.Windows.Data;
549. **using** System.Windows.Media;
551. **namespace Activision\_Mendeleyev\_table**
552. {
553. */// <summary>*
554. */// Логика взаимодействия для ComposSystemContent.xaml*
555. */// </summary>*
556. **public partial class ComposSystemContent : Window**
557. {
558. */// <summary>*
559. */// Таблица данных*
560. */// </summary>*
561. **DataTable dat;**
563. */// <summary>*
564. */// Обозначение соединения(системы соединений)*
565. */// </summary>*
566. **string elem;**
568. */// <summary>*
569. */// Флаг: true - соединение, false - система*
570. */// </summary>*
571. **bool f;**
573. */// <summary>*
574. */// Конструктор, инициализирующий окно таблицы соединения(системы соединений)*
575. */// </summary>*
576. ***/// <param name="elem">название соединения(системы соединений)</param>***
577. */// <param name="f">флаг: true - соединение, false - система</param>*
578. **public** ComposSystemContent(**string** elem, **bool** f)
579. {
580. InitializeComponent();
581. **this.f = f;**
582. **this**.elem = elem;
584. **if** (f)
585. {
586. **this.Title = "Таблица соединения " + elem;**
588. Composition comp = MendeleevTable.Compos.Find(x => x.Name == elem);
589. dat = new DataTable() { TableName = elem };
590. **if** (comp != **null**)
591. **{**
592. *//Заполнение столбцов*
593. **for** (**int** i = 0; i < comp.Properties.Count; i++)
594. dat.Columns.**Add**( new DataColumn() { ColumnName = comp.Properties[i].First.First, Caption = comp.Properties[i].First.Second });
596. ***//Заполнение строк***
597. **for** (**int** i = 0; i < comp.Properties.Count; i++)
598. **for** (**int** j = 0; j < dat.Columns.Count; j++)
599. **if** (dat.Columns[j].ColumnName == comp.Properties[i].First.First)
600. **for** (**int** k = 0; k < comp.Properties[i].Second.Count; k++)
601. **{**
602. **if** (dat.Rows.Count <= k)
603. dat.Rows.**Add**();
604. dat.Rows[k][j] = comp.Properties[i].Second[k];
605. }
607. *//Визуализация столбцов*
608. **foreach** (DataColumn i **in** dat.Columns)
609. ComposSystemTable.Columns.**Add**(new DataGridTextColumn()
610. {
611. **Header = (i.Caption == "" || i.Caption == " ") ? i.ColumnName : (i.ColumnName[0] == '=')? i.Caption + i.ColumnName : i.ColumnName + ", " + i.Caption,**
612. Binding = new Binding("[" + ComposSystemTable.Columns.Count + "]")
613. });
614. }
615. }
616. **else**
617. {
618. **this**.Title = "Таблица системы " + elem;
620. dat = MendeleevTable.BinarySistem.Find(x => x.TableName == elem);
622. **if** (dat == **null**)
623. {
624. dat = new DataTable() { TableName = elem };
625. dat.Columns.**Add**(new DataColumn("X") { Caption = "x"} );
626. **ComposSystemTable.Columns.Add(new DataGridTextColumn()**
627. {
628. Header = "X",
629. Binding = new Binding("[0]"),
630. IsReadOnly = **false**,
631. **Width = 100**
632. });
633. }
634. **else**
635. {
636. **foreach (DataColumn i in dat.Columns)**
637. {
638. ComposSystemTable.Columns.**Add**(new DataGridTextColumn()
639. {
640. Header = (i.Caption == "" || i.Caption == " ") ? i.ColumnName : (i.ColumnName[0] == '=') ? i.Caption + i.ColumnName : i.ColumnName + ", " + i.Caption,
641. **Binding = new Binding("[" + ComposSystemTable.Columns.Count + "]")**
642. });
643. }
644. }
645. }
646. **ComposSystemTable.ItemsSource = dat.DefaultView;**
648. **if** (f && dat.Columns.Count > 0 || !f && dat.Columns.Count > 1)
649. DelColumn.IsEnabled = **true**;
650. **if** (dat.Rows.Count > 0)
651. **DelRow.IsEnabled = true;**
652. }
654. */// <summary>*
655. */// Добавляет текстовый столбец в таблицу*
656. ***/// </summary>***
657. **private** **void** AddColumn\_Click(**object** sender, RoutedEventArgs e)
658. {
659. DelColumn.IsEnabled = DataGridHelper.AddColumn(**ref** ComposSystemTable, **ref** dat, f);
660. }
662. */// <summary>*
663. */// Добавляет строку в таблицу*
664. */// </summary>*
665. **private** **void** AddRow\_Click(**object** sender, RoutedEventArgs e)
666. **{**
667. dat.Rows.**Add**();
668. **if** (dat.Rows.Count > 0)
669. DelRow.IsEnabled = **true**;
670. **if** (!f)
671. **dat.Rows[dat.Rows.Count - 1][0] = 0;**
672. CollectionViewSource.GetDefaultView(ComposSystemTable.ItemsSource).Refresh();
673. }
675. */// <summary>*
676. ***/// Удаляет столбец в таблицу***
677. */// </summary>*
678. **private** **void** DelColumn\_Click(**object** sender, RoutedEventArgs e)
679. {
680. ComposSystemTable.Columns.RemoveAt(ComposSystemTable.Columns.Count - 1);
681. **if (f && dat.Columns.Count <= 1 || !f && dat.Columns.Count <= 2)**
682. DelColumn.IsEnabled = **false**;
683. dat.Columns.RemoveAt(dat.Columns.Count - 1);
684. }
686. ***/// <summary>***
687. */// Удаляет строку в таблицу*
688. */// </summary>*
689. **private** **void** DelRow\_Click(**object** sender, RoutedEventArgs e)
690. {
691. **if (dat.Rows.Count <= 1)**
692. DelRow.IsEnabled = **false**;
693. dat.Rows.RemoveAt(dat.Rows.Count - 1);
694. CollectionViewSource.GetDefaultView(ComposSystemTable.ItemsSource).Refresh();
695. }
697. */// <summary>*
698. */// Запускает раcчет формул и сохраняет данные в файл*
699. */// </summary>*
700. **private** **void** Calculate\_Click(**object** sender, RoutedEventArgs e)
701. **{**
702. ComposSystemTable.IsReadOnly = **true**;
703. EditTable.Visibility = Visibility.Visible;
704. AddColumn.Visibility = Visibility.Hidden;
705. AddRow.Visibility = Visibility.Hidden;
706. **DelColumn.Visibility = Visibility.Hidden;**
707. DelRow.Visibility = Visibility.Hidden;
708. Calculate.Visibility = Visibility.Hidden;
709. AddFormul.Visibility = Visibility.Hidden;
711. **try**
712. {
713. *//Рассчет формул*
714. **for** (**int** i = f?0:1; i < dat.Columns.Count; i++)
715. **for** (**int** u = 0; u < dat.Rows.Count; u++)
716. **if (dat.Columns[i].ColumnName[0] == '=')**
717. dat.Rows[u][i] = MathParser.Parse(dat.Columns[i].ColumnName.Substring(1), **ref** dat, u);
718. }
719. **catch** (Exception ex)
720. {
721. **if (ex.Message.Split(' ')[0] == "Отсутствует")**
722. MessageBox.Show(ex.Message, "Error", MessageBoxButton.OK, MessageBoxImage.Error);
723. **else**
724. MessageBox.Show("Неверный формат формулы!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
725. }
726. **dat.AcceptChanges();**
727. **if** (f)
728. {
729. List<Pair<Pair<**string**, **string**>, List<**string**>>> prop = new List<Pair<Pair<**string**, **string**>, List<**string**>>>();
731. **if (prop == null)**
732. prop = new List<Pair<Pair<**string**, **string**>, List<**string**>>>();
734. **for** (**int** i = 0; i < dat.Columns.Count; i++)
735. {
736. **prop.Add(new Pair<Pair<string, string>, List<string>>(new Pair<string, string>(dat.Columns[i].ColumnName, dat.Columns[i].Caption), new List<string>()));**
737. **for** (**int** j = 0; j < dat.Rows.Count; j++)
738. prop[i].Second.**Add**(dat.Rows[j][i].ToString());
739. }
741. **Composition el = MendeleevTable.Compos.Find(x => x.Name == elem);**
742. **if** (el != **null**)
743. MendeleevTable.Compos.Find(x => x.Name == elem).Properties = new List<Pair<Pair<**string**, **string**>, List<**string**>>>();
744. **else**
745. MendeleevTable.Compos.**Add**(new Composition() { Name = elem });
747. MendeleevTable.Compos.Find(x => x.Name == elem).Properties = prop;
749. DataGridHelper.Serialize("Compositions.xml", **ref** MendeleevTable.Compos);
750. }
751. **else**
752. {
753. MendeleevTable.BinarySistem.**Remove**(MendeleevTable.BinarySistem.Find(x => x.TableName == elem));
754. MendeleevTable.BinarySistem.**Add**(dat);
756. **DataGridHelper.Serialize("BinarySistems.xml", ref MendeleevTable.BinarySistem);**
757. }
758. }
760. */// <summary>*
761. ***/// Добавляет столбец-формулу в таблицу***
762. */// </summary>*
763. **private** **void** AddFormul\_Click(**object** sender, RoutedEventArgs e)
764. {
765. FormulaInput form = new FormulaInput();
766. **form.ShowDialog();**
767. **try**
768. {
769. **if** (form.formula != "")
770. {
771. **if (form.symbol != "" && form.symbol != " ")**
772. **foreach** (DataColumn v **in** dat.Columns)
773. **if** (v.Caption == form.symbol)
774. **throw** new DuplicateNameException();
776. **DataColumn col = new DataColumn('=' + form.formula) { Caption = form.symbol };**
777. dat.Columns.**Add**(col);
778. ComposSystemTable.Columns.**Add**(new DataGridTextColumn()
779. {
780. Header = form.symbol + '=' + form.formula,
781. **Binding = new Binding("[" + ComposSystemTable.Columns.Count + "]")**
782. });
783. }
784. **if** (f && dat.Columns.Count > 0 || !f && dat.Columns.Count > 1)
785. DelColumn.IsEnabled = **true**;
786. **for (int u = 0; u < dat.Rows.Count; u++)**
787. dat.Rows[u][dat.Columns.Count - 1] = MathParser.Parse(dat.Columns[dat.Columns.Count - 1].ColumnName.Substring(1), **ref** dat, u);
788. }
789. **catch** (DuplicateNameException)
790. {
791. **MessageBox.Show("Такая формула уже принадлежит данной таблице!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);**
792. }
793. **catch** (Exception ex)
794. {
795. **if** (ex.Message.Split(' ')[0] == "Отсутствует")
796. **MessageBox.Show(ex.Message, "Error", MessageBoxButton.OK, MessageBoxImage.Error);**
797. **else**
798. MessageBox.Show("Неверный формат формулы!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
799. }
800. }
802. **private** **void** Window\_Closing(**object** sender, System.ComponentModel.CancelEventArgs e)
803. {
804. **if** (MessageBox.Show("Вы точно хотите закрыть окно? Все несохраненные данные будут удалены!", "", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No)
805. e.Cancel = **true**;
806. **else**
807. dat.RejectChanges();
808. }
810. **private** **void** Window\_SizeChanged(**object** sender, SizeChangedEventArgs e)
811. **{**
812. **if** (e.PreviousSize.Width != 0)
813. {
814. AddRow.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
815. AddColumn.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
816. **DelColumn.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;**
817. DelRow.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
818. AddFormul.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
819. Calculate.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
820. AddRow.RenderTransform = new TranslateTransform(360 + (e.NewSize.Width - 1050) / 3, 0);
821. **AddColumn.RenderTransform = new TranslateTransform(190 + (e.NewSize.Width - 1050) / 6, 0);**
822. DelRow.RenderTransform = new TranslateTransform(700 + (e.NewSize.Width - 1050) / 1.5, 0);
823. DelColumn.RenderTransform = new TranslateTransform(530 + (e.NewSize.Width - 1050) / 2, 0);
824. Calculate.RenderTransform = new TranslateTransform(870 + (e.NewSize.Width - 1050) / 1.2, 0);
825. }
826. **}**
828. **private** **void** ComposSystemTable\_CellEditEnding(**object** sender, DataGridCellEditEndingEventArgs e)
829. {
830. **string** str1 = (e.EditingElement **as** TextBox).Text;
831. **double i = 0;**
832. str1 = str1.Replace('.', ',');
833. **if** (!**double**.TryParse(str1, **out** i))
834. str1 = StringHelper.DoString(str1);
836. **if (e.Column.DisplayIndex == 0)**
837. str1 = i.ToString();
839. dat.Rows[e.Row.GetIndex()][e.Column.DisplayIndex] = str1;
840. (e.EditingElement **as** TextBox).Text = str1;
841. **}**
843. */// <summary>*
844. */// Позволяет редактировать таблицу*
845. */// </summary>*
846. **private void EditTable\_Click(object sender, RoutedEventArgs e)**
847. {
848. ComposSystemTable.IsReadOnly = **false**;
849. EditTable.Visibility = Visibility.Hidden;
850. AddColumn.Visibility = Visibility.Visible;
851. **AddRow.Visibility = Visibility.Visible;**
852. DelColumn.Visibility = Visibility.Visible;
853. DelRow.Visibility = Visibility.Visible;
854. Calculate.Visibility = Visibility.Visible;
855. AddFormul.Visibility = Visibility.Visible;
856. **}**
857. }
858. }
859. **Класс ElemContent:**
     * **ElemContent.xaml**
860. **<Window**
861. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
862. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
863. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
864. **xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"**
865. x:Class="Activision\_Mendeleyev\_table.ElemContent"
866. mc:Ignorable="d"
867. Icon="icon.jpeg"
868. Title="ElemContent" Height="450" Width="880" Closing="Window\_Closing" WindowStartupLocation="CenterScreen" MinHeight="450" MinWidth="880" SizeChanged="Window\_SizeChanged"**>**
869. **<Grid>**
870. **<Grid.RowDefinitions>**
871. **<RowDefinition** Height="\*" **/>**
872. **<RowDefinition** Height="50"**/>**
873. **</Grid.RowDefinitions>**
874. **<DataGrid x:Name="ElemTable" Grid.Row="0" CanUserSortColumns="False" CanUserReorderColumns="False" CanUserResizeColumns="False" ColumnWidth="\*"**
875. AutoGenerateColumns="False" CanUserAddRows="False" CanUserDeleteRows="False" CanUserResizeRows="False" IsReadOnly="True" CellEditEnding="ElemContent\_CellEditEnding"**/>**
876. **<Button** Name="AddColumn" Grid.Row="1" Content="Добавить столбец" Click="AddColumn\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Visibility="Hidden"**>**
877. **<Button.RenderTransform>**
878. **<TransformGroup>**
879. **<ScaleTransform/>**
880. **<SkewTransform/>**
881. **<RotateTransform/>**
882. **<TranslateTransform** X="20"**/>**
883. **</TransformGroup>**
884. **</Button.RenderTransform>**
885. **</Button>**
886. **<Button** Name="AddRow" Grid.Row="1" Content="Добавить строку" Click="AddRow\_Click" Width="150" Height="40" RenderTransformOrigin="0.5,0.5" HorizontalAlignment="Left" Visibility="Hidden"**>**
887. **<Button.RenderTransform>**
888. **<TransformGroup>**
889. **<ScaleTransform/>**
890. **<SkewTransform/>**
891. **<RotateTransform/>**
892. **<TranslateTransform** X="190"**/>**
893. **</TransformGroup>**
894. **</Button.RenderTransform>**
895. **</Button>**
896. **<Button** Name="DelColumn" Grid.Row="1" Content="Удалить столбец" Click="DelColumn\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Visibility="Hidden" IsEnabled="False"**>**
897. **<Button.RenderTransform>**
898. **<TransformGroup>**
899. **<ScaleTransform/>**
900. **<SkewTransform/>**
901. **<RotateTransform/>**
902. **<TranslateTransform** X="360"**/>**
903. **</TransformGroup>**
904. **</Button.RenderTransform>**
905. **</Button>**
906. **<Button** Name="DelRow" Grid.Row="1" Content="Удалить строку" Click="DelRow\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Visibility="Hidden" IsEnabled="False"**>**
907. **<Button.RenderTransform>**
908. **<TransformGroup>**
909. **<ScaleTransform/>**
910. **<SkewTransform/>**
911. **<RotateTransform/>**
912. **<TranslateTransform** X="530"**/>**
913. **</TransformGroup>**
914. **</Button.RenderTransform>**
915. **</Button>**
916. **<Button** Name="Save" Grid.Row="1" Content="Сохранить изменения" Click="Save\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Visibility="Hidden"**>**
917. **<Button.RenderTransform>**
918. **<TransformGroup>**
919. **<ScaleTransform/>**
920. **<SkewTransform/>**
921. **<RotateTransform/>**
922. **<TranslateTransform** X="700"**/>**
923. **</TransformGroup>**
924. **</Button.RenderTransform>**
925. **</Button>**
926. **<Button** Name="EditTable" Grid.Row="1" Content="Редактировать таблицу" Click="EditTable\_Click" Width="150" Height="40"**/>**
927. **</Grid>**
928. **</Window>**
     * **ElemContent.cs**
929. **using** System.Collections.Generic;
930. **using** System.Data;
931. **using** System.Windows;
932. **using** System.Windows.Controls;
933. **using System.Windows.Data;**
934. **using** System.Windows.Media;
935. **using** Activision\_Mendeleyev\_table.HelperClasses;
937. **namespace** Activision\_Mendeleyev\_table
938. **{**
939. */// <summary>*
940. */// Логика взаимодействия для ElemContent.xaml*
941. */// </summary>*
942. **public** **partial** **class** ElemContent : Window
943. **{**
944. */// <summary>*
945. */// Таблица данных*
946. */// </summary>*
947. **private** DataTable dat;
949. */// <summary>*
950. */// Обозначение элемента*
951. */// </summary>*
952. **private** **string** elem;
954. */// <summary>*
955. */// Конструктор, инициализирующий окно таблицы элемента*
956. */// </summary>*
957. */// <param name="elem">название элемента</param>*
958. **public ElemContent(string elem)**
959. {
960. InitializeComponent();
961. **this**.Title = "Свойства элемента " + elem;
962. **this**.elem = elem;
964. Composition comp = MendeleevTable.Elems.Find(x => x.Name == elem);
965. **if** (comp != **null**)
966. {
967. *//Заполнение строк*
968. **dat = comp.DataTable;**
969. **for** (**int** i = 0; i < comp.Properties.Count; i++)
970. **for** (**int** j = 0; j < dat.Columns.Count; j++)
971. **if** (dat.Columns[j].ColumnName == comp.Properties[i].First.First)
972. **for** (**int** k = 0; k < comp.Properties[i].Second.Count; k++)
973. **{**
974. **if** (dat.Rows.Count <= k)
975. dat.Rows.**Add**();
976. dat.Rows[k][j] = comp.Properties[i].Second[k];
977. }
978. **}**
980. **if** (dat == **null**)
981. dat = new DataTable() { TableName = elem };
982. **else**
983. **{**
984. *//Визуализация столбцов*
985. **foreach** (DataColumn i **in** dat.Columns)
986. {
987. ElemTable.Columns.**Add**(new DataGridTextColumn()
988. **{**
989. Header = (i.Caption == "" || i.Caption == " ") ? i.ColumnName : i.ColumnName + ", " + i.Caption,
990. Binding = new Binding("[" + ElemTable.Columns.Count + "]")
991. });
992. }
993. **}**
995. ElemTable.ItemsSource = dat.DefaultView;
996. **if** (dat.Columns.Count > 0)
997. DelColumn.IsEnabled = **true**;
998. **if (dat.Rows.Count > 0)**
999. DelRow.IsEnabled = **true**;
1000. }
1002. */// <summary>*
1003. ***/// Добавляет текстовый столбец в таблицу***
1004. */// </summary>*
1005. **private** **void** AddColumn\_Click(**object** sender, RoutedEventArgs e)
1006. {
1007. DelColumn.IsEnabled = DataGridHelper.AddColumn(**ref** ElemTable, **ref** dat);
1008. **}**
1010. */// <summary>*
1011. */// Добавляет строку в таблицу*
1012. */// </summary>*
1013. **private void AddRow\_Click(object sender, RoutedEventArgs e)**
1014. {
1015. dat.Rows.**Add**();
1016. **if** (dat.Rows.Count > 0)
1017. DelRow.IsEnabled = **true**;
1018. **CollectionViewSource.GetDefaultView(ElemTable.ItemsSource).Refresh();**
1020. }
1022. */// <summary>*
1023. ***/// Удаляет столбец в таблицу***
1024. */// </summary>*
1025. **private** **void** DelColumn\_Click(**object** sender, RoutedEventArgs e)
1026. {
1027. ElemTable.Columns.RemoveAt(ElemTable.Columns.Count - 1);
1028. **if (dat.Columns.Count <= 1)**
1029. DelColumn.IsEnabled = **false**;
1030. dat.Columns.RemoveAt(dat.Columns.Count - 1);
1031. }
1033. ***/// <summary>***
1034. */// Удаляет строку в таблицу*
1035. */// </summary>*
1036. **private** **void** DelRow\_Click(**object** sender, RoutedEventArgs e)
1037. {
1038. **if (dat.Rows.Count <= 1)**
1039. DelRow.IsEnabled = **false**;
1040. dat.Rows.RemoveAt(dat.Rows.Count - 1);
1041. CollectionViewSource.GetDefaultView(ElemTable.ItemsSource).Refresh();
1042. }
1044. */// <summary>*
1045. */// Сохраняет данные в файл*
1046. */// </summary>*
1047. **private** **void** Save\_Click(**object** sender, RoutedEventArgs e)
1048. **{**
1049. ElemTable.IsReadOnly = **true**;
1050. EditTable.Visibility = Visibility.Visible;
1051. AddColumn.Visibility = Visibility.Hidden;
1052. AddRow.Visibility = Visibility.Hidden;
1053. **DelColumn.Visibility = Visibility.Hidden;**
1054. DelRow.Visibility = Visibility.Hidden;
1055. Save.Visibility = Visibility.Hidden;
1057. dat.AcceptChanges();
1058. **List<Pair<Pair<string, string>, List<string>>> prop = new List<Pair<Pair<string, string>, List<string>>>();**
1060. **if** (prop == **null**)
1061. prop = new List<Pair<Pair<**string**, **string**>, List<**string**>>>();
1063. **for (int i = 0; i < dat.Columns.Count; i++)**
1064. {
1065. prop.**Add**(new Pair<Pair<**string**, **string**>, List<**string**>>(new Pair<**string**, **string**>(dat.Columns[i].ColumnName, dat.Columns[i].Caption), new List<**string**>()));
1066. **for** (**int** j = 0; j < dat.Rows.Count; j++)
1067. prop[i].Second.**Add**(dat.Rows[j][i].ToString());
1068. **}**
1070. Composition el = MendeleevTable.Elems.Find(x => x.Name == elem);
1071. **if** (el != **null**)
1072. MendeleevTable.Elems.Find(x => x.Name == elem).Properties = new List<Pair<Pair<**string**, **string**>, List<**string**>>>();
1073. **else**
1074. MendeleevTable.Elems.**Add**(new Composition() { Name = elem });
1076. MendeleevTable.Elems.Find(x => x.Name == elem).Properties = prop;
1078. **DataTable d = dat.Copy();**
1079. d.Rows.Clear();
1080. **for** (**int** i = 0; i < MendeleevTable.Elems.Count; i++)
1081. MendeleevTable.Elems[i].DataTable = d;
1083. **DataGridHelper.Serialize("Elems.xml", ref MendeleevTable.Elems);**
1084. }
1086. **private** **void** ElemContent\_CellEditEnding(**object** sender, DataGridCellEditEndingEventArgs e)
1087. {
1088. **string str1 = StringHelper.DoString((e.EditingElement as TextBox).Text);**
1089. str1 = str1.Replace('.', ',');
1090. dat.Rows[e.Row.GetIndex()][e.Column.DisplayIndex] = str1;
1091. (e.EditingElement **as** TextBox).Text = str1;
1092. }
1094. **private** **void** Window\_Closing(**object** sender, System.ComponentModel.CancelEventArgs e)
1095. {
1096. **if** (MessageBox.Show("Вы точно хотите закрыть окно? Все несохраненные данные будут удалены!", "", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No)
1097. e.Cancel = **true**;
1098. **else**
1099. dat.RejectChanges();
1100. }
1102. */// <summary>*
1103. ***/// Позволяет редактировать таблицу***
1104. */// </summary>*
1105. **private** **void** EditTable\_Click(**object** sender, RoutedEventArgs e)
1106. {
1107. ElemTable.IsReadOnly = **false**;
1108. **EditTable.Visibility = Visibility.Hidden;**
1109. AddColumn.Visibility = Visibility.Visible;
1110. AddRow.Visibility = Visibility.Visible;
1111. DelColumn.Visibility = Visibility.Visible;
1112. DelRow.Visibility = Visibility.Visible;
1113. **Save.Visibility = Visibility.Visible;**
1114. }
1116. **private** **void** Window\_SizeChanged(**object** sender, SizeChangedEventArgs e)
1117. {
1118. **if (e.PreviousSize.Width != 0)**
1119. {
1120. AddRow.Width += (e.NewSize.Width - e.PreviousSize.Width) / 5;
1121. AddColumn.Width += (e.NewSize.Width - e.PreviousSize.Width) / 5;
1122. DelColumn.Width += (e.NewSize.Width - e.PreviousSize.Width) / 5;
1123. **DelRow.Width += (e.NewSize.Width - e.PreviousSize.Width) / 5;**
1124. Save.Width += (e.NewSize.Width - e.PreviousSize.Width) / 5;
1125. DelColumn.RenderTransform = new TranslateTransform(360 + (e.NewSize.Width - 880) / 2.5, 0);
1126. AddRow.RenderTransform = new TranslateTransform(190 + (e.NewSize.Width - 880) / 5, 0);
1127. Save.RenderTransform = new TranslateTransform(700 + (e.NewSize.Width - 880) / 1.25, 0);
1128. **DelRow.RenderTransform = new TranslateTransform(530 + (e.NewSize.Width - 880) / 1.66, 0);**
1129. }
1130. }
1131. }
1132. }
1133. **Класс CreateComposSystem:**
      * **CreateComposSystem.xaml**
1134. **<Window** x:Class="Activision\_Mendeleyev\_table.CreateComposSystem"
1135. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
1136. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
1137. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
1138. **xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"**
1139. mc:Ignorable="d"
1140. Icon="icon.jpeg"
1141. Title="Выберите соединение" Height="300" Width="300" ResizeMode="NoResize" WindowStartupLocation="CenterScreen"**>**
1142. **<Grid>**
1143. **<Button Name="Complete" Content="Готово" Click="Complete\_Click" Width="150" Height="40" RenderTransformOrigin="0.5,0.5" HorizontalAlignment="Center" VerticalAlignment="Bottom" FontSize="16" >**
1144. **<Button.RenderTransform>**
1145. **<TransformGroup>**
1146. **<ScaleTransform/>**
1147. **<SkewTransform/>**
1148. **<RotateTransform/>**
1149. **<TranslateTransform** Y="-15"**/>**
1150. **</TransformGroup>**
1151. **</Button.RenderTransform>**
1152. **</Button>**
1153. **<TextBox Name="NewComposSystem" Width="250" Height="30" RenderTransformOrigin="0.5,0.5" FontSize="20">**
1154. **<TextBox.RenderTransform>**
1155. **<TransformGroup>**
1156. **<ScaleTransform/>**
1157. **<SkewTransform/>**
1158. **<RotateTransform/>**
1159. **<TranslateTransform** Y="0"**/>**
1160. **</TransformGroup>**
1161. **</TextBox.RenderTransform>**
1162. **</TextBox>**
1163. **<Label Name="label" Height="30" RenderTransformOrigin="0.5,0.5" HorizontalAlignment="Center">**
1164. **<Label.RenderTransform>**
1165. **<TransformGroup>**
1166. **<ScaleTransform/>**
1167. **<SkewTransform/>**
1168. **<RotateTransform/>**
1169. **<TranslateTransform** Y="-30"**/>**
1170. **</TransformGroup>**
1171. **</Label.RenderTransform>**
1172. **</Label>**
1173. **<ComboBox Name="ComposSystem" Width="250" Height="40" FontSize="20" VerticalAlignment="Top" RenderTransformOrigin="0.5,0.5" SelectionChanged="Compos\_SelectionChanged" >**
1174. **<ComboBox.RenderTransform>**
1175. **<TransformGroup>**
1176. **<ScaleTransform/>**
1177. **<SkewTransform/>**
1178. **<RotateTransform/>**
1179. **<TranslateTransform** Y="15"**/>**
1180. **</TransformGroup>**
1181. **</ComboBox.RenderTransform>**
1182. **</ComboBox>**
1183. **</Grid>**
1184. **</Window>**
      * **CreateComposSystem.cs**
1185. **using** System.Windows;
1186. **using** Activision\_Mendeleyev\_table.HelperClasses;
1188. **namespace** Activision\_Mendeleyev\_table
1189. **{**
1190. */// <summary>*
1191. */// Логика взаимодействия для CreateComposSystem.xaml*
1192. */// </summary>*
1193. **public** **partial** **class** CreateComposSystem : Window
1194. **{**
1195. */// <summary>*
1196. */// Флаг: true - соединение, false - система*
1197. */// </summary>*
1198. **bool** f;
1200. */// <summary>*
1201. */// Конструктор, инициализирующий окно создания/выбора соединения(системы соединений)*
1202. */// </summary>*
1203. */// <param name="f">флаг: true - соединение, false - система</param>*
1204. **public CreateComposSystem(bool f)**
1205. {
1206. InitializeComponent();
1208. **this**.f = f;
1210. **if** (f)
1211. {
1212. **this**.Title = "Выберите/введите соединение";
1213. label.Content = "Введите новое соединения";
1214. **for (int i = 0; i < MendeleevTable.Compos.Count; i++)**
1215. ComposSystem.Items.**Add**(MendeleevTable.Compos[i].Name);
1216. }
1217. **else**
1218. {
1219. **label.Content = "Введите новую систему соединений";**
1220. **this**.Title = "Выберите/введите систему соединений";
1221. **for** (**int** i = 0; i < MendeleevTable.BinarySistem.Count; i++)
1222. ComposSystem.Items.**Add**(MendeleevTable.BinarySistem[i].TableName);
1223. }
1225. **if** (ComposSystem.Items.Count > 0)
1226. ComposSystem.SelectedIndex = 0;
1227. }
1229. ***/// <summary>***
1230. */// Открывает таблицу созданного/выбранного соединения(системы соединений)*
1231. */// </summary>*
1232. **private** **void** Complete\_Click(**object** sender, RoutedEventArgs e)
1233. {
1234. **this.Hide();**
1235. new ComposSystemContent(StringHelper.DoString(NewComposSystem.Text), f).ShowDialog();
1236. **this**.Close();
1237. }
1239. ***/// <summary>***
1240. */// Дублирует соединение(систему соединений) из ComboBox в TextBox*
1241. */// </summary>*
1242. **private** **void** Compos\_SelectionChanged(**object** sender, System.Windows.Controls.SelectionChangedEventArgs e)
1243. {
1244. **NewComposSystem.Text = e.AddedItems[0] as string;**
1245. }
1246. }
1247. }
1248. **Класс ColumnAddWindow:**
      * **ColumnAddWindow.xaml**
1249. **<Window** x:Class="Activision\_Mendeleyev\_table.ColumnAddWindow"
1250. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
1251. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
1252. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
1253. **xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"**
1254. mc:Ignorable="d"
1255. Icon="icon.jpeg"
1256. Title="Введите название столбца " Height="150" Width="265" ResizeMode="NoResize" WindowStartupLocation="CenterScreen"**>**
1257. **<Grid>**
1258. **<Button Name="Create" Content="Создать" HorizontalAlignment="Left" Margin="20,70,0,0" VerticalAlignment="Top" Width="105" Click="Create\_Click" Height="38"/>**
1259. **<TextBox** Name="ColumnName" HorizontalAlignment="Left" Height="20" Margin="20,10,0,0" TextWrapping="Wrap" Text="New Column" VerticalAlignment="Top" Width="215"**/>**
1260. **<TextBox** Name="ColumnSymbol" HorizontalAlignment="Left" Height="20" Margin="120,40,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="115" ToolTip="Если обозначение не требуется, оставьте поле пустым"**/>**
1261. **<Label** HorizontalAlignment="Left" Height="30" Margin="20,35,0,0" Content="Обозначение: " VerticalAlignment="Top"**/>**
1262. **<Button** Name="Cancel" Content="Отмена" HorizontalAlignment="Left" Margin="130,70,0,0" VerticalAlignment="Top" Width="105" Click="Cancel\_Click" Height="38"**/>**
1263. **</Grid>**
1264. **</Window>**
      * **ColumnAddWindow.cs**
1265. **using** System.Windows;
1267. **namespace** Activision\_Mendeleyev\_table
1268. {
1269. ***/// <summary>***
1270. */// Логика взаимодействия для ColumnAddWindow.xaml*
1271. */// </summary>*
1272. **public** **partial** **class** ColumnAddWindow : Window
1273. {
1274. ***/// <summary>***
1275. */// Название столбца*
1276. */// </summary>*
1277. **public** **string** name { **get**; **private** **set**; }
1279. ***/// <summary>***
1280. */// Обозначение свойства, значения которого будут находиться в данном столбце*
1281. */// </summary>*
1282. **public** **string** symbol { **get**; **private** **set**; }
1284. ***/// <summary>***
1285. */// Конструктор, инициализирующий окно создания текстового столбца*
1286. */// </summary>*
1287. **public** ColumnAddWindow()
1288. {
1289. **InitializeComponent();**
1290. name = "";
1291. symbol = "";
1292. }
1294. ***/// <summary>***
1295. */// Создание текстового столбца*
1296. */// </summary>*
1297. **private** **void** Create\_Click(**object** sender, RoutedEventArgs e)
1298. {
1299. **name = ColumnName.Text;**
1300. symbol = ColumnSymbol.Text;
1301. Close();
1302. }
1304. ***/// <summary>***
1305. */// Отмена изменений и закрытие окна*
1306. */// </summary>*
1307. **private** **void** Cancel\_Click(**object** sender, RoutedEventArgs e)
1308. {
1309. **Close();**
1310. }
1311. }
1312. }
1313. **Класс FormulaInput:**
      * **FormulaInput.xaml**
1314. **<Window** x:Class="Activision\_Mendeleyev\_table.FormulaInput"
1315. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
1316. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
1317. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
1318. **xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"**
1319. mc:Ignorable="d"
1320. Icon="icon.jpeg"
1321. Title="Введите формулу" Height="150" Width="300" ResizeMode="NoResize" WindowStartupLocation="CenterScreen"**>**
1322. **<Grid>**
1323. **<Button Name="Create" Content="Создать" Width="125" Click="Create\_Click" Height="38" VerticalAlignment="Bottom" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5">**
1324. **<Button.RenderTransform>**
1325. **<TransformGroup>**
1326. **<ScaleTransform/>**
1327. **<SkewTransform/>**
1328. **<RotateTransform/>**
1329. **<TranslateTransform** X="15" Y="-15"**/>**
1330. **</TransformGroup>**
1331. **</Button.RenderTransform>**
1332. **</Button>**
1333. **<Button Name="Cancel" Content="Отмена" Width="125" Click="Cancel\_Click" Height="38" VerticalAlignment="Bottom" HorizontalAlignment="Right" RenderTransformOrigin="0.5,0.5">**
1334. **<Button.RenderTransform>**
1335. **<TransformGroup>**
1336. **<ScaleTransform/>**
1337. **<SkewTransform/>**
1338. **<RotateTransform/>**
1339. **<TranslateTransform** X="-15" Y="-15"**/>**
1340. **</TransformGroup>**
1341. **</Button.RenderTransform>**
1342. **</Button>**
1343. **<TextBox Name="NewFormula" Width="205" Height="30" RenderTransformOrigin="0.5,0.5" FontSize="20" VerticalAlignment="Top">**
1344. **<TextBox.RenderTransform>**
1345. **<TransformGroup>**
1346. **<ScaleTransform/>**
1347. **<SkewTransform/>**
1348. **<RotateTransform/>**
1349. **<TranslateTransform** X="30" Y="20"**/>**
1350. **</TransformGroup>**
1351. **</TextBox.RenderTransform>**
1352. **</TextBox>**
1353. **<Label Content="=" Height="30" RenderTransformOrigin="0.5,0.5" HorizontalAlignment="Left" VerticalAlignment="Top">**
1354. **<Label.RenderTransform>**
1355. **<TransformGroup>**
1356. **<ScaleTransform/>**
1357. **<SkewTransform/>**
1358. **<RotateTransform/>**
1359. **<TranslateTransform** X="55" Y="20"**/>**
1360. **</TransformGroup>**
1361. **</Label.RenderTransform>**
1362. **</Label>**
1363. **<TextBox Name="FormulaSymbol" Width="40" Height="30" RenderTransformOrigin="0.5,0.5" FontSize="20" HorizontalAlignment="Left" VerticalAlignment="Top" >**
1364. **<TextBox.RenderTransform>**
1365. **<TransformGroup>**
1366. **<ScaleTransform/>**
1367. **<SkewTransform/>**
1368. **<RotateTransform/>**
1369. **<TranslateTransform** X="15" Y="20"**/>**
1370. **</TransformGroup>**
1371. **</TextBox.RenderTransform>**
1372. **</TextBox>**
1373. **</Grid>**
1374. **</Window>**
      * **FormulaInput.cs**
1375. **using** System.Windows;
1376. **using** Activision\_Mendeleyev\_table.HelperClasses;
1378. **namespace** Activision\_Mendeleyev\_table
1379. **{**
1380. */// <summary>*
1381. */// Логика взаимодействия для FormulaInput.xaml*
1382. */// </summary>*
1383. **public** **partial** **class** FormulaInput : Window
1384. **{**
1385. */// <summary>*
1386. */// Строка-формула*
1387. */// </summary>*
1388. **public** **string** formula { **get**; **set**; }
1390. */// <summary>*
1391. */// Обозначение формулы*
1392. */// </summary>*
1393. **public** **string** symbol { **get**; **set**; }
1395. */// <summary>*
1396. */// Конструктор, инициализирующий окно создания столбца-формулы*
1397. */// </summary>*
1398. **public** FormulaInput()
1399. **{**
1400. InitializeComponent();
1401. formula = "";
1402. symbol = "";
1403. }
1405. */// <summary>*
1406. */// Создание столбца-формулы*
1407. */// </summary>*
1408. **private** **void** Create\_Click(**object** sender, RoutedEventArgs e)
1409. **{**
1410. formula = StringHelper.DoString(NewFormula.Text);
1411. symbol = StringHelper.DoString(FormulaSymbol.Text);
1412. Close();
1413. }
1415. */// <summary>*
1416. */// Отмена изменений и закрытие окна*
1417. */// </summary>*
1418. **private** **void** Cancel\_Click(**object** sender, RoutedEventArgs e)
1419. **{**
1420. Close();
1421. }
1422. }
1423. }
1424. **Класс Help:**
      * **Help.xaml**
1425. **<Window** x:Class="Activision\_Mendeleyev\_table.Help"
1426. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
1427. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
1428. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
1429. **xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"**
1430. mc:Ignorable="d"
1431. Icon="icon.jpeg"
1432. Title="Help" Height="450" Width="800"**>**
1433. **<Grid>**
1434. **<WebBrowser Name="web" />**
1435. **</Grid>**
1436. **</Window>**
      * **Help.cs**
1437. **using** System.Windows;
1439. **namespace** Activision\_Mendeleyev\_table
1440. {
1441. ***///<summary>***
1442. */// Логика взаимодействия для Help.xaml*
1443. *///</summary>*
1444. **public** **partial** **class** Help : Window
1445. {
1446. ***/// <summary>***
1447. */// Конструктор справки, для упрощения изпользования программы*
1448. */// </summary>*
1449. **public** Help()
1450. {
1451. **InitializeComponent();**
1452. web.NavigateToString("<!DOCTYPE html>" +
1453. "<html lang = 'en' xmlns = 'http://www.w3.org/1999/xhtml**\'**>" +
1454. "<head>" +
1455. "<meta http-equiv='Content-Type' content='text/html;charset=UTF-8'>" +
1456. **"<title>Справка</title>" +**
1457. "</head>" +
1458. "<body>" +
1459. "<h1>Справка</h1>" +
1460. "<ol style = 'list-style-type: upper-roman;'>" +
1461. **"<li><a href = '#Работа с элементом'> Работа с элементом </a>" +**
1462. "<ol>" +
1463. "<li> Заполнение таблицы </li>" +
1464. "</ol>" +
1465. "</li>" +
1466. **"<li><a href = '#Работа с соединением'> Работа с соединением </a>" +**
1467. "<ol>" +
1468. "<li> Выбор(создание) соединения </li>" +
1469. "<li> Заполнение таблицы </li>" +
1470. "</ol>" +
1471. **"</li>" +**
1472. "<li><a href = '#Работа с системой соединений'> Работа с системой соединений </a>" +
1473. "<ol>" +
1474. "<li> Выбор(создание) системы соединений </li>" +
1475. "<li> Заполнение таблицы </li>" +
1476. **"</ol>" +**
1477. "</li>" +
1478. "<li><a href = '#Формат формул'> Формат формул </a></li>" +
1479. "<li><a href = '#Описание функций'> Описание функций </a></li>" +
1480. "</ol>" +
1481. **"<h3 style = 'text-align: center;'><a name = 'Работа с элементом'> Работа с элементом </a></h3>" +**
1482. "<p align = 'center'><b> Заполнение таблицы </b></p>" +
1483. "<ol>" +
1484. "<li> Для работы с элементом дважды кликните левой кнопкой мыши по соответствующему обозначению в Таблице Менделеева на стартовом окне;</li>" +
1485. "<li> Для начала изменения таблицы необходимо нажать кнопку &quot;Редактировать таблицу&quot; ;</li>" +
1486. **"<li> Для добавления(удаления) строки и столбца необходимо нажать, соответственно, кнопки &quot;Добавить строку&quot; (&quot;Удалить строку&quot;) и &quot;Добавить столбец&quot; (&quot;Удалить столбец&quot;);</li>" +**
1487. "<li> Чтобы сохранить изменения, которые вы произвели в таблице, необходимо нажать кнопку &quot;Сохранить&quot;(<b> !ВАЖНО: </b> заголовки таблицы сохраняются для всех элементов(удаление столбца может повлечь потерю данных));</li>" +
1488. "<li> Закрыть окно с таблицей можно нажав кнопку &quot;х&quot;.</li>" +
1489. "</ol>" +
1490. "<h3 style = 'text-align: center;'><a name = 'Работа с соединением'> Работа с соединением</a></h3>" +
1491. **"<p align = 'center'><b> Выбор(создание) соединения </b></p>" +**
1492. "<ol>" +
1493. "<li> Для работы с соединением нажмите кнопку &quot;Работа с соединением&quot; в верхнем меню стартового окна;</li>" +
1494. "<li> Если соединение уже создано, то вы можете выбрать его из выпадающего списка в данном окне или впишите его в текстовое поле и нажмите кнопку &quot;Готово&quot;;</li>" +
1495. "<li> Для создания нового соединения выпишите его в текстовое поле и нажмите кнопку &quot;Готово&quot;;</li>" +
1496. **"</ol>" +**
1497. "<p align = 'center'><b> Заполнение таблицы </b></p>" +
1498. "<ol>" +
1499. "<li> Для начала изменения таблицы необходимо нажать кнопку &quot;Редактировать таблицу&quot; ;</li>" +
1500. "<li> Для добавления(удаления) строки и столбца необходимо нажать, соответственно, кнопки &quot;Добавить строку&quot; (&quot;Удалить строку&quot;) и &quot;Добавить столбец&quot; (&quot;Удалить столбец&quot;);</li>" +
1501. **"<li> Для добавления формулы нужно нажать кнопку &quot;Добавить формулу&quot; ;</li>" +**
1502. "<li> Чтобы сохранить изменения, которые вы произвели в таблице, и пересчитать формулы, необходимо нажать кнопку &quot;Рассчитать и сохранить&quot;;</li>" +
1503. "<li> Закрыть окно с таблицей можно нажав кнопку &quot;х&quot;.</li>" +
1504. "</ol>" +
1505. "<h3 style = 'text-align: center;'><a name = 'Работа с системой соединений'> Работа с системой соединений </a></h3>" +
1506. **"<p align = 'center'><b> Выбор(создание) соединения </b></p>" +**
1507. "<ol>" +
1508. "<li> Для работы с системой соединений нажмите кнопку &quot;Работа с системой соединений&quot; в верхнем меню стартового окна;</li>" +
1509. "<li> Если система уже создана, то вы можете выбрать её из выпадающего списка в данном окне или впишите её в текстовое поле и нажмите кнопку &quot;Готово&quot;;</li>" +
1510. "<li> Для создания новой системы соединения выпишите её в текстовое поле и нажмите кнопку &quot;Готово&quot;;</li>" +
1511. **"</ol>" +**
1512. "<p align = 'center'><b> Заполнение таблицы </b></p>" +
1513. "<ol>" +
1514. "<li> Для начала изменения таблицы необходимо нажать кнопку &quot;Редактировать таблицу&quot; ;</li>" +
1515. "<li> Для добавления(удаления) строки и столбца необходимо нажать, соответственно, кнопки &quot;Добавить строку&quot; (&quot;Удалить строку&quot;) и&quot;Добавить столбец&quot; (&quot;Удалить столбец&quot;);</li>" +
1516. **"<li> Для добавления формулы нужно нажать кнопку &quot;Добавить формулу&quot; ;</li>" +**
1517. "<li> Столбец &quot;Х&quot; нельзя удалить, и можно заполнять только вещественными числами, в виде десятичных дробей;</li>" +
1518. "<li> Чтобы сохранить изменения, которые вы произвели в таблице, и пересчитать формулы, необходимо нажать кнопку &quot;Рассчитать и сохранить&quot;;</li>" +
1519. "<li> Закрыть окно с таблицей можно нажав кнопку &quot;х&quot;.</li>" +
1520. "</ol>" +
1521. **"<h3 style = 'text-align: center;'><a name = 'Формат формул'> Формат формул </a></h3>" +**
1522. "<p style='margin-left: 20px'>В формулу могут входить:</p>" +
1523. "<ol style = 'list-style-type: lower-alpha;'>" +
1524. "<li> Вещественные числа, в виде десятичных дробей;</li>" +
1525. "<li> ♣ Конструкции, по которым ищется значение в таблицах отделять с двух сторон символом пробел, а сами конструкции писать без каких - либо разделений( <обозначение искомого свойства>{<элемент/соединение>} – получает значение искомого свойства элемента/соединения находящегося в первой строке таблицы свойств; <обозначение искомого свойства>{<вспомогательное свойство, по которому ищем искомое>{<значение вспомогательного свойства>}<элемент/соединение>} - получает значение искомого свойства элемента/соединения соответствующего значению вспомогательного свойства из таблицы свойств);</li>" +
1526. **"<li> Скобки, обозначающие приоритет математических действий, отделенные символом пробел с ВНУТРЕННЕЙ стороны(&quot;(\_&quot;или&quot;\_)&quot;);</li>" +**
1527. "<li> Функции, описанные ниже(если аргументов большо одного - следует их разделять символом &quot;;&quot;. Пример: min( 32; sin( 4 ) ).</li>" +
1528. "<p>Необходимо наличие свойств и элементов(соединений), которые используются в формуле, в таблицах.</p>" +
1529. "</ol>" +
1530. "<h3 style = 'text-align: center;'><a name = 'Описание функций'> Описание функций </a></h3>" +
1531. **"<ol>" +**
1532. "<li> sin(a) - вычисляет синус а;</li>" +
1533. "<li> cos(a) - вычисляет косинус а;</li>" +
1534. "<li> tan(a) - вычисляет тангенс а;</li>" +
1535. "<li> ctan(a) - вычисляет котангенс а;</li>" +
1536. **"<li> abs(a) - вычисляет модуль а;</li>" +**
1537. "<li> ln(a) - вычисляет натуральный логарифм а;</li>" +
1538. "<li> exp(a) - вычисляет экспоненту степени а;</li>" +
1539. "<li> min(a; b) - находит минимум из a и b;</li>" +
1540. "<li> max(a; b) - находит максимум из a и b;</li>" +
1541. **"<li> pow(a; b) - возводит a в степень b;</li>" +**
1542. "</ol>" +
1543. "</body>" +
1544. "</html>");
1545. }
1546. **}**
1547. }
1548. **Классы HelperClasses:**
      1. **Composition.cs**
1549. **using** System;
1550. **using** System.Collections.Generic;
1551. **using** System.Data;
1552. **using** System.Xml.Serialization;
1554. **namespace** Activision\_Mendeleyev\_table.HelperClasses
1555. {
1556. */// <summary>*
1557. */// Класс, представляющий собой соединение(элемент)*
1558. ***/// </summary>***
1559. [Serializable]
1560. **public** **class** Composition
1561. {
1562. */// <summary>*
1563. ***/// Название соединения(элемента)***
1564. */// </summary>*
1565. [XmlElement("Name")]
1566. **public** **string** Name { **get**; **set**; }
1568. ***/// <summary>***
1569. */// Лист свойств и соответствующих им значений*
1570. */// </summary>*
1571. [XmlArray("Properties"), XmlArrayItem("Property")]
1572. **public** List<Pair<Pair<**string**, **string**>, List<**string**>>> Properties { **get**; **set**; }
1574. */// <summary>*
1575. */// Таблица свойств элементов*
1576. */// </summary>*
1577. [XmlElement("Table")]
1578. **public DataTable DataTable { get; set; }**
1580. */// <summary>*
1581. */// Создает объект типа Composition и инициализирует лист свойств*
1582. */// </summary>*
1583. **public Composition() { Properties = new List<Pair<Pair<string, string>, List<string>>>(); }**
1585. */// <summary>*
1586. */// Создает объект типа Composition и инициализирует все свойства класса*
1587. */// </summary>*
1588. ***/// <param name="name">название соединения(элемента)</param>***
1589. */// <param name="data">таблица свойств элементов</param>*
1590. */// <param name="prop">лист свойств и соответсвующих им значений</param>*
1591. **public** Composition(**string** name, DataTable data = **null**, List<Pair<Pair<**string**, **string**>, List<**string**>>> prop = **null**)
1592. {
1593. **Properties = prop;**
1594. DataTable = data;
1595. Name = name;
1596. **if** (data == **null**)
1597. DataTable = new DataTable() { TableName = name};
1598. **if (prop == null)**
1599. Properties = new List<Pair<Pair<**string**, **string**>, List<**string**>>>();
1600. }
1601. }
1602. }
      1. **DataGridHelper.cs**
1603. **using** System;
1604. **using** System.Data;
1605. **using** System.IO;
1606. **using** System.Windows;
1607. **using System.Windows.Controls;**
1608. **using** System.Windows.Data;
1609. **using** System.Windows.Media;
1610. **using** System.Xml.Serialization;
1612. **namespace Activision\_Mendeleyev\_table.HelperClasses**
1613. {
1614. */// <summary>*
1615. *///Вспомогательный статический класс для работы с DataGrid*
1616. */// </summary>*
1617. **public static class DataGridHelper**
1618. {
1619. */// <summary>*
1620. */// Вспомогательный метод для получения ячейки таблицы DataGrid*
1621. */// </summary>*
1622. ***/// <typeparam name="T">Visual type</typeparam>***
1623. */// <param name="parent">предок</param>*
1624. */// <returns>потомок</returns>*
1625. **private** **static** T GetVisualChild<T>(Visual parent) **where** T : Visual
1626. {
1627. **try**
1628. {
1629. T child = **default**(T);
1630. **int** numVisuals = VisualTreeHelper.GetChildrenCount(parent);
1631. **for** (**int** i = 0; i < numVisuals; i++)
1632. **{**
1633. Visual v = (Visual)VisualTreeHelper.GetChild(parent, i);
1634. child = v **as** T;
1635. **if** (child == **null**)
1636. child = GetVisualChild<T>(v);
1638. **if** (child != **null**)
1639. **break**;
1640. }
1641. **return** child;
1642. **}**
1643. **catch** (Exception exc)
1644. {
1645. MessageBox.Show("Ошибка получения данных из таблицы функция (GetVisualChild<T>)!**\n\n**" + exc.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
1646. **return** **null**;
1647. **}**
1649. }
1651. */// <summary>*
1652. ***/// Получает строку из DataGrid***
1653. */// </summary>*
1654. */// <param name="index">номер строки</param>*
1655. */// <param name="dg">DataGrid</param>*
1656. */// <returns>Строку DataGrid</returns>*
1657. **private static DataGridRow GetRow(int index, DataGrid dg)**
1658. {
1659. **try**
1660. {
1661. DataGridRow row = (DataGridRow)dg.ItemContainerGenerator.ContainerFromIndex(index);
1662. **if (row == null)**
1663. {
1664. dg.UpdateLayout();
1665. dg.ScrollIntoView(dg.Items[index]);
1666. row = (DataGridRow)dg.ItemContainerGenerator.ContainerFromIndex(index);
1668. }
1669. **return** row;
1670. }
1671. **catch** (Exception exc)
1672. **{**
1673. MessageBox.Show("Ошибка получения данных из таблицы функция (GetRow)!**\n\n**" + exc.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
1674. **return** **null**;
1675. }
1676. }
1678. */// <summary>*
1679. */// Добавляет текстовый столбец в таблицу*
1680. */// </summary>*
1681. */// <param name="dg">DataGrid</param>*
1682. ***/// <param name="dat">таблица данных</param>***
1683. */// <param name="f">флаг: true - соединение(элемент), false - система</param>*
1684. */// <returns>Можно ли удалять столбцы?</returns>*
1685. **public** **static** **bool** AddColumn(**ref** DataGrid dg, **ref** DataTable dat, **bool** f = **true**)
1686. {
1687. **ColumnAddWindow form = new ColumnAddWindow();**
1688. form.ShowDialog();
1689. **try**
1690. {
1691. **if** (form.name != "")
1692. **{**
1693. **if** (form.symbol != "" && form.symbol != " ")
1694. **foreach** (DataColumn v **in** dat.Columns)
1695. **if** (v.Caption == form.symbol)
1696. **throw** new DuplicateNameException();
1698. DataColumn col = new DataColumn(form.name) { Caption = form.symbol };
1699. dat.Columns.**Add**(col);
1700. dg.Columns.**Add**(new DataGridTextColumn()
1701. {
1702. **Header = (form.symbol == "" || form.symbol == " ") ? form.name : form.name + ", " + form.symbol,**
1703. Binding = new Binding("[" + dg.Columns.Count + "]")
1704. });
1705. }
1706. }
1707. **catch (DuplicateNameException)**
1708. {
1709. MessageBox.Show("Столбец с данным именем(обозначением) уже принадлежит данной таблице!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
1710. }
1712. **if (f && dat.Columns.Count > 0 || !f && dat.Columns.Count > 1)**
1713. **return** **true**;
1714. **return** **false**;
1715. }
1717. ***/// <summary>***
1718. */// Получает ячейку из DataGrid*
1719. */// </summary>*
1720. */// <param name="row">номер строки</param>*
1721. */// <param name="column">номер столбца</param>*
1722. ***/// <param name="dg">DataGrid</param>***
1723. */// <returns>Ячейка DataGrid</returns>*
1724. **public** **static** DataGridCell GetCell(**int** row, **int** column, DataGrid dg)
1725. {
1726. **try**
1727. **{**
1728. DataGridRow rowContainer = GetRow(row, dg);
1730. **if** (rowContainer != **null**)
1731. {
1732. **System.Windows.Controls.Primitives.DataGridCellsPresenter presenter = GetVisualChild<System.Windows.Controls.Primitives.DataGridCellsPresenter>(rowContainer);**
1734. DataGridCell cell = (DataGridCell)presenter.ItemContainerGenerator.ContainerFromIndex(column);
1735. **if** (cell == **null**)
1736. {
1737. **dg.ScrollIntoView(rowContainer, dg.Columns[column]);**
1738. cell = (DataGridCell)presenter.ItemContainerGenerator.ContainerFromIndex(column);
1739. }
1740. **return** cell;
1741. }
1742. **return null;**
1743. }
1744. **catch** (Exception exc)
1745. {
1746. MessageBox.Show("Ошибка получения данных из таблицы функция (GetCell)!**\n\n**" + exc.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
1747. **return null;**
1748. }
1749. }
1751. */// <summary>*
1752. ***/// Закрашивает ячейку DataGrid***
1753. */// </summary>*
1754. */// <param name="row">номер строки</param>*
1755. */// <param name="column">номер столбца</param>*
1756. */// <param name="color">цвет для закрашивания</param>*
1757. ***/// <param name="dg">DataGrid</param>***
1758. **public** **static** **void** BrushCell(**int** row, **int** column, Brush color, DataGrid dg)
1759. {
1760. DataGridCell cell = GetCell(row, column, dg);
1761. cell.Background = color;
1762. **cell.BorderBrush = color;**
1763. }
1765. */// <summary>*
1766. */// Метод сериализации соединений/элементов/систем соединений*
1767. ***/// </summary>***
1768. */// <typeparam name="T">тип: List<Composition> или List<DataTable></typeparam>*
1769. */// <param name="name">имя файла</param>*
1770. */// <param name="data">лист соединений/элементов/систем соединений</param>*
1771. **public** **static** **void** Serialize<T>(**string** name, **ref** System.Collections.Generic.List<T> data)
1772. **{**
1773. **using** (FileStream fs = new FileStream(name, FileMode.Create, FileAccess.Write))
1774. {
1775. XmlSerializer serializer = new XmlSerializer(data.GetType());
1776. serializer.Serialize(fs, data);
1777. **}**
1778. }
1780. */// <summary>*
1781. */// Метод десериализации соединений/элементов/систем соединений*
1782. ***/// </summary>***
1783. */// <typeparam name="T">тип: List<Composition> или List<DataTable></typeparam>*
1784. */// <param name="name">имя файла</param>*
1785. */// <param name="data">лист соединений/элементов/систем соединений</param>*
1786. */// <returns>Удалось ли десериализовать?</returns>*
1787. **public static bool Deserialize<T>(string name, ref System.Collections.Generic.List<T> data)**
1788. {
1789. **if** (!File.Exists(name))
1790. **return** **false**;
1792. **try**
1793. {
1794. **using** (FileStream fs = new FileStream(name, FileMode.Open, FileAccess.Read))
1795. {
1796. XmlSerializer serializer = new XmlSerializer(data.GetType());
1797. **data = (System.Collections.Generic.List<T>)serializer.Deserialize(fs);**
1798. }
1799. **return** **true**;
1800. }
1801. **catch** (FileLoadException)
1802. **{**
1803. **return** **false**;
1804. }
1805. }
1806. }
1807. **}**
      1. **HotKey.cs**
1808. **using** System;
1809. **using** System.Collections.Generic;
1810. **using** System.Diagnostics;
1811. **using** System.Runtime.InteropServices;
1812. **using System.Windows.Input;**
1813. **using** System.Windows.Interop;
1815. **namespace** Activision\_Mendeleyev\_table.HelperClasses
1816. {
1817. ***/// <summary>***
1818. */// Класс для привязки клавишь к действиям*
1819. */// </summary>*
1820. **public** **class** HotKey : IDisposable
1821. {
1822. ***/// <summary>***
1823. */// Словарь id и горячих клавиш для обратного вызова*
1824. */// </summary>*
1825. **private** **static** Dictionary<**int**, HotKey> \_dictHotKeyToCallBackProc;
1827. **[DllImport("user32.dll")]**
1828. **private** **static** **extern** **bool** RegisterHotKey(IntPtr hWnd, **int** id, **uint** fsModifiers, **uint** vlc);
1830. [DllImport("user32.dll")]
1831. **private** **static** **extern** **bool** UnregisterHotKey(IntPtr hWnd, **int** id);
1833. */// <summary>*
1834. */// Слово для генерации сообщений*
1835. */// </summary>*
1836. **public** **const** **int** WmHotKey = 0x0312;
1838. */// <summary>*
1839. */// Флаг для управления методоми Dispose() и Dispose(bool)*
1840. */// </summary>*
1841. **private** **bool** \_disposed = **false**;
1843. */// <summary>*
1844. */// Клавиша*
1845. */// </summary>*
1846. **public** Key Key { **get**; **private** **set**; }
1848. */// <summary>*
1849. */// Модификаторы*
1850. */// </summary>*
1851. **public** KeyModifier KeyModifiers { **get**; **private** **set**; }
1853. */// <summary>*
1854. */// Метод, который вызывается по нажатию сочетания клавиш*
1855. */// </summary>*
1856. **public** Action<HotKey> Action { **get**; **private** **set**; }
1858. */// <summary>*
1859. */// Уникальный идентификатор сочетания клавиш*
1860. */// </summary>*
1861. **public** **int** Id { **get**; **set**; }
1863. */// <summary>*
1864. */// Создает объект типа HotKey и регистрирует переданное сочетание клавиш(по умолчанию)*
1865. */// </summary>*
1866. */// <param name="k">клавиша</param>*
1867. ***/// <param name="keyModifiers">модификаторы</param>***
1868. */// <param name="action">метод</param>*
1869. */// <param name="register">Нужно ли регистрировать?</param>*
1870. **public** HotKey(Key k, KeyModifier keyModifiers, Action<HotKey> action, **bool** register = **true**)
1871. {
1872. **Key = k;**
1873. KeyModifiers = keyModifiers;
1874. Action = action;
1875. **if** (register)
1876. Register();
1877. **}**
1879. */// <summary>*
1880. */// Регистрирует сочетание клавиш*
1881. */// </summary>*
1882. ***/// <returns>Зарегистрирован или нет?</returns>***
1883. **public** **bool** Register()
1884. {
1885. **int** virtualKeyCode = KeyInterop.VirtualKeyFromKey(Key);
1886. Id = virtualKeyCode + ((**int**)KeyModifiers \* 0x10000);
1887. **bool result = RegisterHotKey(IntPtr.Zero, Id, (uint)KeyModifiers, (uint)virtualKeyCode);**
1889. **if** (\_dictHotKeyToCallBackProc == **null**)
1890. {
1891. \_dictHotKeyToCallBackProc = new Dictionary<**int**, HotKey>();
1892. **ComponentDispatcher.ThreadFilterMessage += new ThreadMessageEventHandler(ComponentDispatcherThreadFilterMessage);**
1893. }
1895. \_dictHotKeyToCallBackProc.**Add**(Id, **this**);
1897. **Debug.Print(result.ToString() + ", " + Id + ", " + virtualKeyCode);**
1898. **return** result;
1899. }
1901. */// <summary>*
1902. ***/// Утилизирует управляемые ресурсы***
1903. */// </summary>*
1904. **public** **void** Unregister()
1905. {
1906. HotKey hotKey;
1907. **if (\_dictHotKeyToCallBackProc.TryGetValue(Id, out hotKey))**
1908. UnregisterHotKey(IntPtr.Zero, Id);
1909. }
1911. */// <summary>*
1912. ***/// ???***
1913. */// </summary>*
1914. */// <param name="msg"></param>*
1915. */// <param name="handled"></param>*
1916. **private** **static** **void** ComponentDispatcherThreadFilterMessage(**ref** MSG msg, **ref** **bool** handled)
1917. **{**
1918. **if** (!handled && msg.message == WmHotKey)
1919. {
1920. HotKey hotKey;
1921. **if** (\_dictHotKeyToCallBackProc.TryGetValue((**int**)msg.wParam, **out** hotKey))
1922. **{**
1923. **if** (hotKey.Action != **null**)
1924. hotKey.Action.Invoke(hotKey);
1925. handled = **true**;
1926. }
1927. **}**
1928. }
1930. */// <summary>*
1931. */// Метод IDisposable, вызывает наш virtual Dispose(bool)*
1932. ***/// </summary>***
1933. **public** **void** Dispose()
1934. {
1935. Dispose(**true**);
1936. GC.SuppressFinalize(**this**);
1937. **}**
1939. */// <summary>*
1940. */// Утилизирует ресурсы*
1941. */// </summary>*
1942. ***/// <param name="disposing">Можно ли утилизировать управляемые ресурсы?</param>***
1943. **protected** **virtual** **void** Dispose(**bool** disposing)
1944. {
1945. **if** (!**this**.\_disposed)
1946. {
1947. **if (disposing)**
1948. Unregister();
1950. \_disposed = **true**;
1951. }
1952. **}**
1953. }
1955. */// <summary>*
1956. */// Модификаторы*
1957. ***/// </summary>***
1958. [Flags]
1959. **public** **enum** KeyModifier
1960. {
1961. None = 0x0000,
1962. **Alt = 0x0001,**
1963. Ctrl = 0x0002,
1964. NoRepeat = 0x4000,
1965. Shift = 0x0004,
1966. Win = 0x0008
1967. **}**
1968. }
      1. **MathParser.cs**
1969. **using** System;
1970. **using** System.Collections.Generic;
1971. **using** System.Data;
1972. **using** System.Globalization;
1973. **using System.Text.RegularExpressions;**
1975. **namespace** Activision\_Mendeleyev\_table.HelperClasses
1976. {
1977. */// <summary>*
1978. ***/// Класс, вычисляющей значение формулы по введенной строке по определенным правилам***
1979. */// </summary>*
1980. **public** **static** **class** MathParser
1981. {
1982. */// <summary>*
1983. ***/// Парсинг строки для вычисления значения формулы, если данная строка является формулой***
1984. */// </summary>*
1985. */// <param name="str">содержание ячайки таблицы</param>*
1986. */// <param name="dat">таблица, в которой производим вычисление</param>*
1987. */// <param name="u">номер строки</param>*
1988. ***/// <returns>результат парсинга</returns>***
1989. **public** **static** **double** Parse(**string** str, **ref** DataTable dat, **int** u)
1990. {
1991. str = str.Replace(',', '.');
1992. **string** left = "", right = "", inner = "";
1993. **int i = 0, j = 0, k = 0;**
1995. *// Парсинг функций*
1996. **string**[] func = { "sin", "cos", "tan", "ctan", "abs", "exp", "ln", "min", "max", "pow" };
1997. **for** (i = 0; i < func.Length; i++)
1998. **{**
1999. k = str.IndexOf(func[i]);
2000. **if** (k >= 0)
2001. {
2002. left = str.Substring(0, k);
2003. **k += func[i].Length;**
2004. j = 0;
2005. **bool** f = **false**;
2006. right = "";
2007. inner = "";
2008. **while (k < str.Length)**
2009. {
2010. **if** (!f && str[k] == '(' && k < str.Length - 1 && str[k + 1] == ' ')
2011. {
2012. j++;
2013. **if (j == 1)**
2014. {
2015. left += right;
2016. right = "";
2017. k++;
2018. **}**
2019. }
2021. right += str[k];
2023. **if (!f && str[k] == ' ' && k < str.Length - 1 && str[k + 1] == ')')**
2024. {
2025. j--;
2026. **if** (j == 0)
2027. {
2028. **inner += right;**
2029. right = "";
2030. f = **true**;
2031. k++;
2032. }
2033. **}**
2035. k++;
2036. }
2038. **switch (i)**
2039. {
2040. **case** 0:
2041. **return** Parse(left + Math.Sin(Parse(inner, **ref** dat, u)) + right, **ref** dat, u);
2043. **case 1:**
2044. **return** Parse(left + Math.Cos(Parse(inner, **ref** dat, u)) + right, **ref** dat, u);
2046. **case** 2:
2047. **return** Parse(left + Math.Tan(Parse(inner, **ref** dat, u)) + right, **ref** dat, u);
2049. **case** 3:
2050. **return** Parse(left + 1.0 / Math.Tan(Parse(inner, **ref** dat, u)) + right, **ref** dat, u);
2052. **case** 4:
2053. **return Parse(left + Math.Abs(Parse(inner, ref dat, u)) + right, ref dat, u);**
2055. **case** 5:
2056. **return** Parse(left + Math.Exp(Parse(inner, **ref** dat, u)) + right, **ref** dat, u);
2058. **case 6:**
2059. **return** Parse(left + Math.Log(Parse(inner, **ref** dat, u)) + right, **ref** dat, u);
2061. **case** 7:
2062. **string** inleft = inner.Substring(0, inner.IndexOf(';'));
2063. **string inright = inner.Substring(inner.IndexOf(';') + 1);**
2064. **return** Parse(left + Math.Min(Parse(inleft, **ref** dat, u), Parse(inright, **ref** dat, u)) + right, **ref** dat, u);
2066. **case** 8:
2067. inleft = inner.Substring(0, inner.IndexOf(';'));
2068. **inright = inner.Substring(inner.IndexOf(';') + 1);**
2069. **return** Parse(left + Math.Max(Parse(inleft, **ref** dat, u), Parse(inright, **ref** dat, u)) + right, **ref** dat, u);
2071. **case** 9:
2072. inleft = inner.Substring(0, inner.IndexOf(';'));
2073. **inright = inner.Substring(inner.IndexOf(';') + 1);**
2074. **return** Parse(left + Math.Pow(Parse(inleft, **ref** dat, u), Parse(inright, **ref** dat, u)) + right, **ref** dat, u);
2075. }
2076. }
2077. }
2079. *//Парсинг символа x*
2080. Match matchFuncx = Regex.Match(str, @"(x)");
2081. **if** (matchFuncx.Groups.Count > 1)
2082. {
2083. **left = str.Substring(0, matchFuncx.Index);**
2084. right = str.Substring(matchFuncx.Index + matchFuncx.Length);
2085. **return** Parse(left + **double**.Parse(dat.Rows[u]["X"].ToString()) + right, **ref** dat, u);
2086. }
2088. **string str1 = "";**
2089. *//Парсинг конструкций для получения значений из таблиц*
2090. **try**
2091. {
2092. Match matchElem = Regex.Match(str, @"([\w\[\]\.\+\-\\*\/%\^\_\(\)]\*){([\w\[\]\.\+\-\\*\/%\^\_\(\) ]\*){([\[\]\d\.\+\-\\*\/%\^ \_\(\) ]\*)}([\w\[\]\.\+\-\\*\/%\^ \_\(\) ]\*)}|([\w\[\]\.\+\-\\*\/%\^\_\(\)]\*){([\w\[\]\.\+\-\\*\/%\^\_\(\) ]\*)}");
2093. **if (matchElem.Groups.Count > 1)**
2094. {
2095. inner = StringHelper.DoString(str.Substring(matchElem.Index, matchElem.Length)).Trim(' ');
2096. left = str.Substring(0, matchElem.Index);
2097. right = str.Substring(matchElem.Index + matchElem.Length);
2098. **LinkedList<string> strs = new LinkedList<string>();**
2099. **string** symbol = "";
2100. k = 0;
2101. **while** (k < inner.Length)
2102. {
2103. **if (inner[k] == '{')**
2104. {
2105. k++;
2106. **while** (k < inner.Length && inner[k] != '}' && inner[k] != '{')
2107. {
2108. **str1 += inner[k];**
2109. k++;
2110. }
2111. strs.AddLast(str1);
2112. str1 = "";
2113. **}**
2114. **else** **if** (inner[k] == '}')
2115. {
2116. k++;
2117. **while** (k < inner.Length && inner[k] != '}' && inner[k] != '{')
2118. **{**
2119. str1 += inner[k];
2120. k++;
2121. }
2122. strs.AddFirst(str1);
2123. **str1 = "";**
2124. }
2125. **else**
2126. {
2127. symbol += inner[k];
2128. **k++;**
2129. }
2130. }
2131. strs.AddLast(symbol);
2132. strs.RemoveFirst();
2133. **return Parse(left + Find(strs).ToString() + right, ref dat, u);**
2134. }
2135. }
2136. **catch** (FormatException)
2137. {
2138. **throw new FormatException(string.Format("Неверная входная строка '{0}'", str));**
2139. }
2141. *// Парсинг скобок*
2142. i = 0;
2143. **j = 0;**
2144. left = "";
2145. right = "";
2146. inner = "";
2147. **while** (i < str.Length)
2148. **{**
2149. **if** (str[i] == '(' && i < str.Length - 1 && str[i + 1] == ' ')
2150. {
2151. j++;
2152. **if** (j == 1)
2153. **{**
2154. left += right;
2155. right = "";
2156. i++;
2157. }
2158. **}**
2160. **if** (str[i] == ' ' && i < str.Length - 1 && str[i + 1] == ')')
2161. {
2162. j--;
2163. **if (j == 0)**
2164. {
2165. inner += right;
2166. right = str.Substring(i + 2);
2167. **return** Parse(left + Parse(inner, **ref** dat, u) + right, **ref** dat, u);
2168. **}**
2169. }
2171. right += str[i];
2172. i++;
2173. **}**
2174. **if** (inner != "")
2175. **return** Parse(left + Parse(inner, **ref** dat, u) + right, **ref** dat, u);
2177. *// Парсинг действий*
2178. **Match matchMulOp = Regex.Match(str, string.Format(@"({0})\s?({1})\s?({0})\s?", RegexNum, RegexMulOp));**
2179. Match matchAddOp = Regex.Match(str, **string**.Format(@"({0})\s?({1})\s?({0})\s?", RegexNum, RegexAddOp));
2180. **var** match = (matchMulOp.Groups.Count > 1) ? matchMulOp : (matchAddOp.Groups.Count > 1) ? matchAddOp : **null**;
2181. **if** (match != **null**)
2182. {
2183. **left = str.Substring(0, match.Index);**
2184. right = str.Substring(match.Index + match.Length);
2185. **return** Parse(left + ParseAct(match).ToString(CultureInfo.InvariantCulture) + right, **ref** dat, u);
2186. }
2188. ***// Парсинг числа***
2189. str1 = "";
2190. **for** (i = 0; i < str.Length; i++)
2191. **if** (str[i] != ' ')
2192. **if** (str[i] == '.')
2193. **str1 += ',';**
2194. **else**
2195. str1 += str[i];
2196. **double**.TryParse(str1, **out** **double** num);
2197. **return** num;
2198. **}**
2200. */// <summary>*
2201. */// Форматная строка, соответствующая числу*
2202. */// </summary>*
2203. **private const string RegexNum = @"[-]?\d+\.?\d\*";**
2204. */// <summary>*
2205. */// Форматная строка, соответствующая операциям умножения и деления*
2206. */// </summary>*
2207. **private** **const** **string** RegexMulOp = @"[\\*\/%]";
2208. ***/// <summary>***
2209. */// Форматная строка, соответствующая операциям сложения и вычитания*
2210. */// </summary>*
2211. **private** **const** **string** RegexAddOp = @"[\+\-]";
2213. ***/// <summary>***
2214. */// Выполнение математических операций*
2215. */// </summary>*
2216. */// <param name="match">результаты вычисления регулярного выражения, определяющего мат. операции</param>*
2217. */// <returns>результат вычисления</returns>*
2218. **private static double ParseAct(Match match)**
2219. {
2220. **double** a = **double**.Parse(match.Groups[1].**Value**, CultureInfo.InvariantCulture);
2221. **double** b = **double**.Parse(match.Groups[3].**Value**, CultureInfo.InvariantCulture);
2223. **switch (match.Groups[2].Value)**
2224. {
2225. **case** "+":
2226. **return** a + b;
2228. **case "-":**
2229. **return** a - b;
2231. **case** "\*":
2232. **return** a \* b;
2234. **case** "/":
2235. **return** a / b;
2237. **case** "%":
2238. **return a % b;**
2240. **default**:
2241. **throw** new FormatException(**string**.Format("Неверная входная строка '{0}'", match.**Value**));
2242. }
2243. **}**
2245. */// <summary>*
2246. */// Поиск значения свойства в таблицах*
2247. */// </summary>*
2248. ***/// <param name="strs">лист, включающий название соединения(элемента), обозначение свойства и, если необходимо, доп. свойства и его значение</param>***
2249. */// <returns>значение свойсва</returns>*
2250. **private** **static** **double** Find(LinkedList<**string**> strs)
2251. {
2252. LinkedListNode<**string**> str = strs.First;
2253. **Composition e = MendeleevTable.Elems.Find(x => x.Name == str.Value);**
2255. **if** (e == **null**)
2256. e = MendeleevTable.Compos.Find(x => x.Name == str.**Value**);
2258. **if (e == null)**
2259. **throw** new Exception("Отсутствует соединение или элемент" + str.**Value** + ".");
2261. str = str.Next;
2262. List<Pair<Pair<**string**, **string**>, List<**string**>>> data = **null**;
2263. **data = e.Properties;**
2265. Pair<Pair<**string**, **string**>, List<**string**>> v = data.Find(x => x.First.Second == str.**Value**);
2267. **if** (v == **null**)
2268. **throw new Exception("Отсутствует свойство " + str.Value + " у данного соединения или элемента.");**
2270. str = str.Next;
2272. **if** (str == **null**)
2273. **return double.Parse(v.Second[0]);**
2275. **int** row = -1;
2276. **for** (**int** i = 0; i < v.Second.Count; i++)
2277. **if** (v.Second[i] == str.**Value**)
2278. **row = i;**
2280. **if** (row == -1)
2281. **throw** new Exception("Отсутствует значение параметра " + str.Previous.**Value** + " равного " + str.**Value** + " у данного соединения или элемента.");
2283. **v = data.Find(x => x.First.Second == strs.Last.Value);**
2285. **if** (v == **null**)
2286. **throw** new Exception("Отсутствует параметр " + strs.Last.**Value** + " у данного соединения или элемента.");
2288. **return double.Parse(v.Second[row]);**
2289. }
2290. }
2291. }
      1. **Pair.cs**
2292. **using** System;
2294. **namespace** Activision\_Mendeleyev\_table.HelperClasses
2295. {
2296. ***/// <summary>***
2297. */// Класс, который предоставляет возможность хранить два разнородных объекта как единое целое*
2298. */// </summary>*
2299. */// <typeparam name="T">тип первого аргумента</typeparam>*
2300. */// <typeparam name="U">тип второго аргумента</typeparam>*
2301. **[Serializable]**
2302. **public** **class** Pair<T, U>
2303. {
2304. */// <summary>*
2305. */// Первый аргумент*
2306. ***/// </summary>***
2307. **public** T First { **get**; **set**; }
2309. */// <summary>*
2310. */// Второй аргумент*
2311. ***/// </summary>***
2312. **public** U Second { **get**; **set**; }
2314. */// <summary>*
2315. */// Создает объект типа Pair*
2316. ***/// </summary>***
2317. **public** Pair() { }
2319. */// <summary>*
2320. */// Создает объект типа Pair и инициализирует значение аргументов*
2321. ***/// </summary>***
2322. */// <param name="first">первый аргумент</param>*
2323. */// <param name="second">второй аргумент</param>*
2324. **public** Pair(T first, U second)
2325. {
2326. **First = first;**
2327. Second = second;
2328. }
2329. };
2330. }
      1. **Strings.cs:**
         * **Strings**
2331. */// <summary>*
2332. */// Вспомогательный класс для заполнения Таблицы Менделеева*
2333. */// </summary>*
2334. **public** **class** Strings
2335. **{**
2336. */// <summary>*
2337. */// Задает один ряд элементов по группам и подгруппам*
2338. */// </summary>*
2339. **public** Strings(**string** ia = **null**, **string** iia = **null**, **string** iiia = **null**, **string** iva = **null**, **string** va = **null**, **string** via = **null**, **string** viia = **null**,
2340. **string viiia\_1 = null, string viiia\_2 = null, string viiia\_3 = null, string ib = null, string iib = null, string iiib = null, string ivb = null,**
2341. **string** vb = **null**, **string** vib = **null**, **string** viib = **null**, **string** viiib = **null**)
2342. {
2343. Ia = ia;
2344. IIa = iia;
2345. **IIIa = iiia;**
2346. IVa = iva;
2347. Va = va;
2348. VIa = via;
2349. VIIa = viia;
2350. **VIIIa\_1 = viiia\_1;**
2351. VIIIa\_2 = viiia\_2;
2352. VIIIa\_3 = viiia\_3;
2353. Ib = ib;
2354. IIb = iib;
2355. **IIIb = iiib;**
2356. IVb = ivb;
2357. Vb = vb;
2358. VIb = vib;
2359. VIIb = viib;
2360. **VIIIb = viiib;**
2361. }
2363. **public** **string** Ia { **get**; **set**; }
2364. **public** **string** IIa { **get**; **set**; }
2365. **public string IIIa { get; set; }**
2366. **public** **string** IVa { **get**; **set**; }
2367. **public** **string** Va { **get**; **set**; }
2368. **public** **string** VIa { **get**; **set**; }
2369. **public** **string** VIIa { **get**; **set**; }
2370. **public string VIIIa\_1 { get; set; }**
2371. **public** **string** VIIIa\_2 { **get**; **set**; }
2372. **public** **string** VIIIa\_3 { **get**; **set**; }
2373. **public** **string** Ib { **get**; **set**; }
2374. **public** **string** IIb { **get**; **set**; }
2375. **public string IIIb { get; set; }**
2376. **public** **string** IVb { **get**; **set**; }
2377. **public** **string** Vb { **get**; **set**; }
2378. **public** **string** VIb { **get**; **set**; }
2379. **public** **string** VIIb { **get**; **set**; }
2380. **public string VIIIb { get; set; }**
2381. }
      * + **StringHelper**
2382. */// <summary>*
2383. */// Класс для преобразования строк по определенным правилам*
2384. */// </summary>*
2385. **public** **static** **class** StringHelper
2386. **{**
2387. */// <summary>*
2388. */// Преобразует определённый символ в соответствующий подстрочный*
2389. */// </summary>*
2390. */// <param name="c">символ</param>*
2391. ***/// <returns>подстрочный символ</returns>***
2392. **private** **static** **char** Substring(**char** c)
2393. {
2394. **switch** (c)
2395. {
2396. **case '0':**
2397. **return** '₀';
2398. **case** '1':
2399. **return** '₁';
2400. **case** '2':
2401. **return '₂';**
2402. **case** '3':
2403. **return** '₃';
2404. **case** '4':
2405. **return** '₄';
2406. **case '5':**
2407. **return** '₅';
2408. **case** '6':
2409. **return** '₆';
2410. **case** '7':
2411. **return '₇';**
2412. **case** '8':
2413. **return** '₈';
2414. **case** '9':
2415. **return** '₉';
2416. **case '+':**
2417. **return** '₊';
2418. **case** '-':
2419. **return** '₋';
2420. **case** '=':
2421. **return '₌';**
2422. **case** '(':
2423. **return** '₍';
2424. **case** ')':
2425. **return** '₎';
2426. **case '.':**
2427. **case** ',':
2428. **return** '.';
2429. **default**:
2430. **return** c;
2431. **}**
2432. }
2434. */// <summary>*
2435. */// Преобразует определённый символ в соответствующий надстрочный*
2436. ***/// </summary>***
2437. */// <param name="c">символ</param>*
2438. */// <returns>надстрочный символ</returns>*
2439. **private** **static** **char** Superstring(**char** c)
2440. {
2441. **switch (c)**
2442. {
2443. **case** '0':
2444. **return** '⁰';
2445. **case** '1':
2446. **return '¹';**
2447. **case** '2':
2448. **return** '²';
2449. **case** '3':
2450. **return** '³';
2451. **case '4':**
2452. **return** '⁴';
2453. **case** '5':
2454. **return** '⁵';
2455. **case** '6':
2456. **return '⁶';**
2457. **case** '7':
2458. **return** '⁷';
2459. **case** '8':
2460. **return** '⁸';
2461. **case '9':**
2462. **return** '⁹';
2463. **case** '+':
2464. **return** '⁺';
2465. **case** '-':
2466. **return '⁻';**
2467. **case** '=':
2468. **return** '⁼';
2469. **case** '(':
2470. **return** '⁽';
2471. **case ')':**
2472. **return** '⁾';
2473. **case** '.':
2474. **case** ',':
2475. **return** '·';*//'ʼ';*
2476. **default:**
2477. **return** c;
2478. }
2479. }
2481. ***/// <summary>***
2482. */// Преобразует некоторые символы строки в над(под)строчные по заданным правилам*
2483. */// </summary>*
2484. */// <param name="str">исходная строка</param>*
2485. */// <returns>преобразованная строка</returns>*
2486. **public static string DoString(string str)**
2487. {
2488. **int** i = 0;
2489. **string** str1 = "";
2490. **while** (i < str.Length)
2491. **{**
2492. *// Преобразует символы после комбинации \_{ и до } в подстрочные*
2493. **if** (i < str.Length && str[i] == '\_')
2494. {
2495. i++;
2496. **if (str[i] != '{')**
2497. str1 += Substring(str[i]);
2498. **else**
2499. {
2500. i++;
2501. **while (i < str.Length && str[i] != '}')**
2502. {
2503. str1 += Substring(str[i]);
2504. i++;
2505. }
2506. **}**
2507. i++;
2508. }
2509. *// Преобразует символы после комбинации ^{ и до } в надстрочные*
2510. **if** (i < str.Length && str[i] == '^')
2511. **{**
2512. i++;
2513. **if** (str[i] != '{')
2514. str1 += Superstring(str[i]);
2515. **else**
2516. **{**
2517. i++;
2518. **while** (i < str.Length && str[i] != '}')
2519. {
2520. str1 += Superstring(str[i]);
2521. **i++;**
2522. }
2523. }
2524. i++;
2525. }
2526. **if (i < str.Length && str[i] != '\_')**
2527. {
2528. *// Преобразует символы после буквенных и подстрочных символов в подстрочные*
2529. **if** (i > 0 && "0123456789+-".Contains(str[i].ToString()) && "₀₁₂₃₄₅₆₇₈₉₊₋₌₍₎)QWERTYUIOPASDFGHJKLZXCVVBNMqwertyuiopasdfghjklzxcvbnm".Contains(str1[i - 1].ToString()))
2530. str1 += Substring(str[i]);
2531. **else**
2532. **if** (i > 1 && '.' == str[i - 1] && "₀₁₂₃₄₅₆₇₈₉₊₋₌₍₎()QWERTYUIOPASDFGHJKLZXCVVBNMqwertyuiopasdfghjklzxcvbnm".Contains(str1[i - 2].ToString()))
2533. str1 += Substring(str[i]);
2534. **else**
2535. str1 += str[i];
2536. **i++;**
2537. }
2538. }
2539. **return** str1;
2540. }
2541. **}**

**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.

# **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |