**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

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.



**Программа для расчета и визуализации бинарных фазовых диаграмм в системе изоструктурных компонентов**

**Текст программы**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.15-01 12 01-1-ЛУ**

**Исполнитель**

Студент группы БПИ173

\_\_\_\_\_\_\_\_\_\_\_/К.Г. Кожакин/

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

**Москва 2020**

**УТВЕРЖДЕН**

**RU.17701729.04.15-01 12 01-1-ЛУ**

**Программа для расчета и визуализации бинарных фазовых диаграмм в системе изоструктурных компонентов**

**Текст программы**

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

**Листов 149**

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

**Москва 2020**

**Оглавление**

[1. Класс MendeleevTable: 4](#_Toc37846020)

[ MendeleevTable.xaml 4](#_Toc37846021)

[ MendeleevTable.cs 7](#_Toc37846022)

[2. Класс ElemContent: 15](#_Toc37846023)

[ ElemContent.xaml 15](#_Toc37846024)

[ ElemContent.cs 17](#_Toc37846025)

[3. Класс ComposSystemContent: 23](#_Toc37846026)

[ ComposSystemContent.xaml 24](#_Toc37846027)

[ ComposSystemContent.cs 26](#_Toc37846028)

[4. Класс CreateComposSystem: 37](#_Toc37846029)

[ CreateComposSystem.xaml 37](#_Toc37846030)

[ CreateComposSystem.cs 38](#_Toc37846031)

[5. Класс ColumnAddWindow: 40](#_Toc37846032)

[ ColumnAddWindow.xaml 40](#_Toc37846033)

[ ColumnAddWindow.cs 40](#_Toc37846034)

[6. Класс FormulaInput: 42](#_Toc37846035)

[ FormulaInput.xaml 42](#_Toc37846036)

[ FormulaInput.cs 44](#_Toc37846037)

[7. Класс FormulaList: 45](#_Toc37846038)

[ FormulaList.xaml 45](#_Toc37846039)

[ FormulaList.cs 46](#_Toc37846040)

[8. Класс DataSettings: 47](#_Toc37846041)

[ DataSettings.xaml 47](#_Toc37846042)

[ DataSettings.cs 56](#_Toc37846043)

[9. Класс DomeOfDecay: 58](#_Toc37846044)

[ DomeOfDecay.xaml 59](#_Toc37846045)

[ DomeOfDecay.cs 65](#_Toc37846046)

[10. Класс Help: 80](#_Toc37846047)

[ Help.xaml 80](#_Toc37846048)

[ Help.cs 80](#_Toc37846049)

[11. Классы Approximation: 86](#_Toc37846050)

[ Criterion.cs 86](#_Toc37846051)

[ Library.cs 88](#_Toc37846052)

[12. Классы DrawingClasses: 94](#_Toc37846053)

[ Collapse.cs 94](#_Toc37846054)

[ CollapseGraph.cs 96](#_Toc37846055)

[13. Классы HelperClasses: 103](#_Toc37846056)

[ BinSystem.cs 103](#_Toc37846057)

[ Composition.cs 113](#_Toc37846058)

[ DataGridHelper.cs 114](#_Toc37846059)

[ HotKey.cs 120](#_Toc37846060)

[ MathParser.cs 124](#_Toc37846061)

[ Pair.cs 140](#_Toc37846062)

[ Point.cs 141](#_Toc37846063)

[ RoundConverter.cs 141](#_Toc37846064)

[ Strings.cs: 142](#_Toc37846065)

[a. StringHelper 142](#_Toc37846066)

[b. Strings 145](#_Toc37846067)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 148](#_Toc37846068)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 149](#_Toc37846069)

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" WindowStartupLocation="CenterScreen" ScrollViewer.CanContentScroll="False" Loaded="Window\_Loaded" SizeChanged="Window\_SizeChanged" MinWidth="1280" MinHeight="720" FontSize="18">
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. */// <summary>*
437. */// Открывает окно справки*
438. */// </summary>*
439. private void Help\_Click(object sender, RoutedEventArgs e)
440. {
441. new Help().Show();
442. }
444. */// <summary>*
445. */// Открывает окно справки*
446. */// </summary>*
447. */// <param name="hotKey">горячая клавиша для вызова метода</param>*
448. private static void OnF1Handler(HotKey hotKey)
449. {
450. new Help().Show();
451. }
452. }
453. }
454. **Класс ElemContent:**
     * **ElemContent.xaml**
455. <Window
456. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
457. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
458. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
459. xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
460. x:Class="Activision\_Mendeleyev\_table.ElemContent"
461. mc:Ignorable="d"
462. Icon="icon.jpeg"
463. Title="ElemContent" Height="450" Width="880" Closing="Window\_Closing" WindowStartupLocation="CenterScreen" MinHeight="450" MinWidth="880" SizeChanged="Window\_SizeChanged">
464. <Grid>
465. <Grid.RowDefinitions>
466. <RowDefinition Height="\*" />
467. <RowDefinition Height="50"/>
468. </Grid.RowDefinitions>
469. <DataGrid x:Name="ElemTable" Grid.Row="0" CanUserSortColumns="False" CanUserReorderColumns="False" CanUserResizeColumns="False" ColumnWidth="\*"
470. AutoGenerateColumns="False" CanUserAddRows="False" CanUserDeleteRows="False" CanUserResizeRows="False" IsReadOnly="True" CellEditEnding="ElemContent\_CellEditEnding"/>
471. <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">
472. <Button.RenderTransform>
473. <TransformGroup>
474. <ScaleTransform/>
475. <SkewTransform/>
476. <RotateTransform/>
477. <TranslateTransform X="20"/>
478. </TransformGroup>
479. </Button.RenderTransform>
480. </Button>
481. <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">
482. <Button.RenderTransform>
483. <TransformGroup>
484. <ScaleTransform/>
485. <SkewTransform/>
486. <RotateTransform/>
487. <TranslateTransform X="190"/>
488. </TransformGroup>
489. </Button.RenderTransform>
490. </Button>
491. <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">
492. <Button.RenderTransform>
493. <TransformGroup>
494. <ScaleTransform/>
495. <SkewTransform/>
496. <RotateTransform/>
497. <TranslateTransform X="360"/>
498. </TransformGroup>
499. </Button.RenderTransform>
500. </Button>
501. <Button Name="DelSelectedRows" Grid.Row="1" Content="Удалить выделенные строки" Click="DelSelectedRows\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Visibility="Hidden" IsEnabled="False" FontSize="10">
502. <Button.RenderTransform>
503. <TransformGroup>
504. <ScaleTransform/>
505. <SkewTransform/>
506. <RotateTransform/>
507. <TranslateTransform X="530"/>
508. </TransformGroup>
509. </Button.RenderTransform>
510. </Button>
511. <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">
512. <Button.RenderTransform>
513. <TransformGroup>
514. <ScaleTransform/>
515. <SkewTransform/>
516. <RotateTransform/>
517. <TranslateTransform X="700"/>
518. </TransformGroup>
519. </Button.RenderTransform>
520. </Button>
521. <Button Name="EditTable" Grid.Row="1" Content="Редактировать таблицу" Click="EditTable\_Click" Width="150" Height="40"/>
522. </Grid>
523. </Window>
     * **ElemContent.cs**
524. using System.Collections.Generic;
525. using System.Data;
526. using System.Windows;
527. using System.Windows.Controls;
528. using System.Windows.Data;
529. using System.Windows.Media;
530. using Activision\_Mendeleyev\_table.HelperClasses;
532. namespace Activision\_Mendeleyev\_table
533. {
534. */// <summary>*
535. */// Логика взаимодействия для ElemContent.xaml*
536. */// </summary>*
537. public partial class ElemContent : Window
538. {
539. */// <summary>*
540. */// Таблица данных*
541. */// </summary>*
542. private DataTable dat;
544. */// <summary>*
545. */// Обозначение элемента*
546. */// </summary>*
547. private string elem;
549. */// <summary>*
550. */// Конструктор, инициализирующий окно таблицы элемента*
551. */// </summary>*
552. */// <param name="elem">название элемента</param>*
553. public ElemContent(string elem)
554. {
555. InitializeComponent();
556. this.Title = "Свойства элемента " + elem;
557. this.elem = elem;
559. Composition comp = MendeleevTable.Elems.Find(x => x.Name == elem);
560. if (comp != null)
561. {
562. *//Заполнение строк*
563. dat = comp.DataTable;
564. for (int i = 0; i < comp.Properties.Count; i++)
565. for (int j = 0; j < dat.Columns.Count; j++)
566. if (dat.Columns[j].ColumnName == comp.Properties[i].First.First)
567. for (int k = 0; k < comp.Properties[i].Second.Count; k++)
568. {
569. if (dat.Rows.Count <= k)
570. dat.Rows.Add();
571. dat.Rows[k][j] = comp.Properties[i].Second[k];
572. }
573. }
575. if (dat == null)
576. dat = new DataTable() { TableName = elem };
577. else
578. {
579. *//Визуализация столбцов*
580. foreach (DataColumn i in dat.Columns)
581. {
582. ElemTable.Columns.Add(new DataGridTextColumn()
583. {
584. Header = (i.Caption == "" || i.Caption == " ") ? i.ColumnName : i.ColumnName + ", " + i.Caption,
585. Binding = new Binding("[" + ElemTable.Columns.Count + "]")
586. });
587. }
588. }
590. ElemTable.ItemsSource = dat.DefaultView;
591. if (dat.Columns.Count > 0)
592. DelColumn.IsEnabled = true;
593. if (dat.Rows.Count > 0)
594. DelSelectedRows.IsEnabled = true;
595. }
597. */// <summary>*
598. */// Добавляет текстовый столбец в таблицу*
599. */// </summary>*
600. private void AddColumn\_Click(object sender, RoutedEventArgs e)
601. {
602. DelColumn.IsEnabled = DataGridHelper.AddColumn(ref ElemTable, ref dat);
603. }
605. */// <summary>*
606. */// Добавляет строку в таблицу*
607. */// </summary>*
608. private void AddRow\_Click(object sender, RoutedEventArgs e)
609. {
610. dat.Rows.Add();
611. if (dat.Rows.Count > 0)
612. DelSelectedRows.IsEnabled = true;
613. CollectionViewSource.GetDefaultView(ElemTable.ItemsSource).Refresh();
615. }
617. */// <summary>*
618. */// Удаляет столбец в таблицу*
619. */// </summary>*
620. private void DelColumn\_Click(object sender, RoutedEventArgs e)
621. {
622. ElemTable.Columns.RemoveAt(ElemTable.Columns.Count - 1);
623. if (dat.Columns.Count <= 1)
624. DelColumn.IsEnabled = false;
625. dat.Columns.RemoveAt(dat.Columns.Count - 1);
626. }
628. */// <summary>*
629. */// Удаляет строку в таблицу*
630. */// </summary>*
631. private void DelSelectedRows\_Click(object sender, RoutedEventArgs e)
632. {
633. try
634. {
635. while (ElemTable.SelectedItems.Count > 0)
636. {
637. int selectedIndex = ElemTable.SelectedIndex;
638. DrawingClasses.CollapseGraph.RemoveSelectedPoint(selectedIndex);
639. dat.Rows.RemoveAt(selectedIndex);
640. ElemTable.Items.Refresh();
641. }
642. }
643. catch (System.Exception)
644. {
645. MessageBox.Show("Невозможно удалить этот элемент!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
646. }
647. if (dat.Rows.Count == 0)
648. DelSelectedRows.IsEnabled = false;
649. }
651. */// <summary>*
652. */// Сохраняет данные в файл*
653. */// </summary>*
654. private void Save\_Click(object sender, RoutedEventArgs e)
655. {
656. ElemTable.IsReadOnly = true;
657. EditTable.Visibility = Visibility.Visible;
658. AddColumn.Visibility = Visibility.Hidden;
659. AddRow.Visibility = Visibility.Hidden;
660. DelColumn.Visibility = Visibility.Hidden;
661. DelSelectedRows.Visibility = Visibility.Hidden;
662. Save.Visibility = Visibility.Hidden;
664. dat.AcceptChanges();
665. List<Pair<Pair<string, string>, List<string>>> prop = new List<Pair<Pair<string, string>, List<string>>>();
667. if (prop == null)
668. prop = new List<Pair<Pair<string, string>, List<string>>>();
670. for (int i = 0; i < dat.Columns.Count; i++)
671. {
672. prop.Add(new Pair<Pair<string, string>, List<string>>(new Pair<string, string>(dat.Columns[i].ColumnName, dat.Columns[i].Caption), new List<string>()));
673. for (int j = 0; j < dat.Rows.Count; j++)
674. prop[i].Second.Add(dat.Rows[j][i].ToString());
675. }
677. Composition el = MendeleevTable.Elems.Find(x => x.Name == elem);
678. if (el != null)
679. MendeleevTable.Elems.Find(x => x.Name == elem).Properties = new List<Pair<Pair<string, string>, List<string>>>();
680. else
681. MendeleevTable.Elems.Add(new Composition() { Name = elem });
683. MendeleevTable.Elems.Find(x => x.Name == elem).Properties = prop;
685. DataTable d = dat.Copy();
686. d.Rows.Clear();
687. for (int i = 0; i < MendeleevTable.Elems.Count; i++)
688. MendeleevTable.Elems[i].DataTable = d;
690. DataGridHelper.Serialize("Elems.xml", ref MendeleevTable.Elems);
691. }
693. private void ElemContent\_CellEditEnding(object sender, DataGridCellEditEndingEventArgs e)
694. {
695. string str1 = StringHelper.DoString((e.EditingElement as TextBox).Text);
696. str1 = str1.Replace('.', ',');
697. dat.Rows[e.Row.GetIndex()][e.Column.DisplayIndex] = str1;
698. (e.EditingElement as TextBox).Text = str1;
699. }
701. private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)
702. {
703. if (EditTable.Visibility == Visibility.Hidden)
704. if (MessageBox.Show("Вы точно хотите закрыть окно? Все несохраненные данные будут удалены!", "", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No)
705. e.Cancel = true;
706. else
707. dat.RejectChanges();
708. }
710. */// <summary>*
711. */// Позволяет редактировать таблицу*
712. */// </summary>*
713. private void EditTable\_Click(object sender, RoutedEventArgs e)
714. {
715. ElemTable.IsReadOnly = false;
716. EditTable.Visibility = Visibility.Hidden;
717. AddColumn.Visibility = Visibility.Visible;
718. AddRow.Visibility = Visibility.Visible;
719. DelColumn.Visibility = Visibility.Visible;
720. DelSelectedRows.Visibility = Visibility.Visible;
721. Save.Visibility = Visibility.Visible;
722. }
724. private void Window\_SizeChanged(object sender, SizeChangedEventArgs e)
725. {
726. if (e.PreviousSize.Width != 0)
727. {
728. AddRow.Width += (e.NewSize.Width - e.PreviousSize.Width) / 5;
729. AddColumn.Width += (e.NewSize.Width - e.PreviousSize.Width) / 5;
730. DelColumn.Width += (e.NewSize.Width - e.PreviousSize.Width) / 5;
731. DelSelectedRows.Width += (e.NewSize.Width - e.PreviousSize.Width) / 5;
732. Save.Width += (e.NewSize.Width - e.PreviousSize.Width) / 5;
733. DelColumn.RenderTransform = new TranslateTransform(360 + (e.NewSize.Width - 880) / 2.5, 0);
734. AddRow.RenderTransform = new TranslateTransform(190 + (e.NewSize.Width - 880) / 5, 0);
735. Save.RenderTransform = new TranslateTransform(700 + (e.NewSize.Width - 880) / 1.25, 0);
736. DelSelectedRows.RenderTransform = new TranslateTransform(530 + (e.NewSize.Width - 880) / 1.66, 0);
737. }
738. }
739. }
740. }
742. **Класс ComposSystemContent:**
     * **ComposSystemContent.xaml**
743. <Window x:Class="Activision\_Mendeleyev\_table.ComposSystemContent"
744. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
745. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
746. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
747. xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
748. mc:Ignorable="d"
749. Icon="icon.jpeg"
750. Title="ComposSystemContent" Height="450" Width="1050" WindowStartupLocation="CenterScreen" MinHeight="450" MinWidth="1050" Closing="Window\_Closing" SizeChanged="Window\_SizeChanged">
751. <Grid>
752. <Grid.RowDefinitions>
753. <RowDefinition Height="\*" />
754. <RowDefinition Height="50"/>
755. </Grid.RowDefinitions>
756. <DataGrid x:Name="ComposSystemTable" Grid.Row="0" IsReadOnly="True" CanUserSortColumns="False" CanUserReorderColumns="False" CanUserResizeColumns="False" ColumnWidth="\*"
757. AutoGenerateColumns="False" CanUserAddRows="False" CanUserDeleteRows="False" CanUserResizeRows="False" CellEditEnding="ComposSystemTable\_CellEditEnding"/>
758. <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">
759. <Button.RenderTransform>
760. <TransformGroup>
761. <ScaleTransform/>
762. <SkewTransform/>
763. <RotateTransform/>
764. <TranslateTransform X="20"/>
765. </TransformGroup>
766. </Button.RenderTransform>
767. </Button>
768. <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">
769. <Button.RenderTransform>
770. <TransformGroup>
771. <ScaleTransform/>
772. <SkewTransform/>
773. <RotateTransform/>
774. <TranslateTransform X="190"/>
775. </TransformGroup>
776. </Button.RenderTransform>
777. </Button>
778. <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">
779. <Button.RenderTransform>
780. <TransformGroup>
781. <ScaleTransform/>
782. <SkewTransform/>
783. <RotateTransform/>
784. <TranslateTransform X="360"/>
785. </TransformGroup>
786. </Button.RenderTransform>
787. </Button>
788. <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">
789. <Button.RenderTransform>
790. <TransformGroup>
791. <ScaleTransform/>
792. <SkewTransform/>
793. <RotateTransform/>
794. <TranslateTransform X="530"/>
795. </TransformGroup>
796. </Button.RenderTransform>
797. </Button>
798. <Button Name="DelSelectedRows" Grid.Row="1" Visibility="Hidden" FontSize="10" Content="Удалить выделенные строки" Click="DelSelectedRows\_Click" Width="150" Height="40" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" IsEnabled="False">
799. <Button.RenderTransform>
800. <TransformGroup>
801. <ScaleTransform/>
802. <SkewTransform/>
803. <RotateTransform/>
804. <TranslateTransform X="700"/>
805. </TransformGroup>
806. </Button.RenderTransform>
807. </Button>
808. <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">
809. <Button.RenderTransform>
810. <TransformGroup>
811. <ScaleTransform/>
812. <SkewTransform/>
813. <RotateTransform/>
814. <TranslateTransform X="870"/>
815. </TransformGroup>
816. </Button.RenderTransform>
817. </Button>
818. <Button Name="EditTable" Grid.Row="1" Content="Редактировать таблицу" Click="EditTable\_Click" Width="150" Height="40" RenderTransformOrigin="0.5,0.5" >
819. <Button.RenderTransform>
820. <TransformGroup>
821. <ScaleTransform/>
822. <SkewTransform/>
823. <RotateTransform/>
824. <TranslateTransform X="-100"/>
825. </TransformGroup>
826. </Button.RenderTransform>
827. </Button>
828. <Button Name="DomeOfDecayWindowOpen" Grid.Row="1" Content="Купол распада" Click="DomeOfDecayWindowOpen\_Click" Width="150" Height="40" RenderTransformOrigin="0.5,0.5">
829. <Button.RenderTransform>
830. <TransformGroup>
831. <ScaleTransform/>
832. <SkewTransform/>
833. <RotateTransform/>
834. <TranslateTransform X="100"/>
835. </TransformGroup>
836. </Button.RenderTransform>
837. </Button>
838. </Grid>
839. </Window>
     * **ComposSystemContent.cs**
840. using Activision\_Mendeleyev\_table.HelperClasses;
841. using System;
842. using System.Collections.Generic;
843. using System.Data;
844. using System.Windows;
845. using System.Windows.Controls;
846. using System.Windows.Data;
847. using System.Windows.Media;
849. namespace Activision\_Mendeleyev\_table
850. {
851. */// <summary>*
852. */// Логика взаимодействия для ComposSystemContent.xaml*
853. */// </summary>*
854. public partial class ComposSystemContent : Window
855. {
856. */// <summary>*
857. */// Таблица данных*
858. */// </summary>*
859. private DataTable dat;
861. */// <summary>*
862. */// Обозначение соединения(системы соединений)*
863. */// </summary>*
864. private string elem;
866. */// <summary>*
867. */// Флаг: true - соединение, false - система*
868. */// </summary>*
869. private bool f;
871. */// <summary>*
872. */// Конструктор, инициализирующий окно таблицы соединения(системы соединений)*
873. */// </summary>*
874. */// <param name="elem">название соединения(системы соединений)</param>*
875. */// <param name="f">флаг: true - соединение, false - система</param>*
876. public ComposSystemContent(string elem, bool f)
877. {
878. InitializeComponent();
879. this.f = f;
880. this.elem = elem;
882. if (f)
883. {
884. this.Title = "Таблица соединения " + elem;
885. DomeOfDecayWindowOpen.Visibility = Visibility.Hidden;
886. EditTable.RenderTransform = new TranslateTransform();
888. Composition comp = MendeleevTable.Compos.Find(x => x.Name == elem);
889. dat = new DataTable() { TableName = elem };
890. if (comp != null)
891. {
892. *//Заполнение столбцов*
893. for (int i = 0; i < comp.Properties.Count; i++)
894. dat.Columns.Add( new DataColumn() { ColumnName = comp.Properties[i].First.First, Caption = comp.Properties[i].First.Second });
896. *//Заполнение строк*
897. for (int i = 0; i < comp.Properties.Count; i++)
898. for (int j = 0; j < dat.Columns.Count; j++)
899. if (dat.Columns[j].ColumnName == comp.Properties[i].First.First)
900. for (int k = 0; k < comp.Properties[i].Second.Count; k++)
901. {
902. if (dat.Rows.Count <= k)
903. dat.Rows.Add();
904. dat.Rows[k][j] = comp.Properties[i].Second[k];
905. }
907. *//Визуализация столбцов*
908. foreach (DataColumn i in dat.Columns)
909. {
910. bool r = (i.ColumnName[0] == '=') ? true : false;
911. ComposSystemTable.Columns.Add(new DataGridTextColumn()
912. {
913. Header = (i.Caption == "" || i.Caption == " ") ? i.ColumnName : (i.ColumnName[0] == '=') ? i.Caption + i.ColumnName : i.ColumnName + ", " + i.Caption,
914. Binding = new Binding("[" + ComposSystemTable.Columns.Count + "]"),
915. IsReadOnly = r
916. });
917. }
918. }
919. }
920. else
921. {
922. this.Title = "Таблица системы " + elem;
924. dat = MendeleevTable.BinarySistem.Find(x => x.TableName == elem);
926. if (dat == null)
927. {
928. dat = new DataTable() { TableName = elem };
929. dat.Columns.Add(new DataColumn("X") { Caption = "x"} );
930. ComposSystemTable.Columns.Add(new DataGridTextColumn()
931. {
932. Header = "X",
933. Binding = new Binding("[0]"),
934. IsReadOnly = false,
935. Width = 100
936. });
937. }
938. else
939. {
940. foreach (DataColumn i in dat.Columns)
941. {
942. bool r = (i.ColumnName[0] == '=') ? true : false;
943. ComposSystemTable.Columns.Add(new DataGridTextColumn()
944. {
945. Header = (i.Caption == "" || i.Caption == " ") ? i.ColumnName : (i.ColumnName[0] == '=') ? i.Caption + i.ColumnName : i.ColumnName + ", " + i.Caption,
946. Binding = new Binding("[" + ComposSystemTable.Columns.Count + "]"),
947. IsReadOnly = r
948. });
949. }
950. }
951. }
952. ComposSystemTable.ItemsSource = dat.DefaultView;
954. if (f && dat.Columns.Count > 0 || !f && dat.Columns.Count > 1)
955. DelColumn.IsEnabled = true;
956. if (dat.Rows.Count > 0)
957. DelSelectedRows.IsEnabled = true;
958. }
960. */// <summary>*
961. */// Добавляет текстовый столбец в таблицу*
962. */// </summary>*
963. private void AddColumn\_Click(object sender, RoutedEventArgs e)
964. {
965. DelColumn.IsEnabled = DataGridHelper.AddColumn(ref ComposSystemTable, ref dat, f);
966. }
968. */// <summary>*
969. */// Добавляет строку в таблицу*
970. */// </summary>*
971. private void AddRow\_Click(object sender, RoutedEventArgs e)
972. {
973. dat.Rows.Add();
974. if (dat.Rows.Count > 0)
975. DelSelectedRows.IsEnabled = true;
976. if (!f)
977. dat.Rows[dat.Rows.Count - 1][0] = 0;
978. CollectionViewSource.GetDefaultView(ComposSystemTable.ItemsSource).Refresh();
979. }
981. */// <summary>*
982. */// Удаляет столбец в таблице*
983. */// </summary>*
984. private void DelColumn\_Click(object sender, RoutedEventArgs e)
985. {
986. ComposSystemTable.Columns.RemoveAt(ComposSystemTable.Columns.Count - 1);
987. if (f && dat.Columns.Count <= 1 || !f && dat.Columns.Count <= 2)
988. DelColumn.IsEnabled = false;
989. dat.Columns.RemoveAt(dat.Columns.Count - 1);
990. }
992. */// <summary>*
993. */// Удаляет выделенные строки в таблице*
994. */// </summary>*
995. private void DelSelectedRows\_Click(object sender, RoutedEventArgs e)
996. {
997. try
998. {
999. while (ComposSystemTable.SelectedItems.Count > 0)
1000. {
1001. int selectedIndex = ComposSystemTable.SelectedIndex;
1002. DrawingClasses.CollapseGraph.RemoveSelectedPoint(selectedIndex);
1003. dat.Rows.RemoveAt(selectedIndex);
1004. ComposSystemTable.Items.Refresh();
1005. }
1006. }
1007. catch (Exception)
1008. {
1009. MessageBox.Show("Невозможно удалить этот элемент!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1010. }
1011. if (dat.Rows.Count == 0)
1012. DelSelectedRows.IsEnabled = false;
1013. }
1015. */// <summary>*
1016. */// Запускает раcчет формул и сохраняет данные в файл*
1017. */// </summary>*
1018. private void Calculate\_Click(object sender, RoutedEventArgs e)
1019. {
1020. ComposSystemTable.IsReadOnly = true;
1021. EditTable.Visibility = Visibility.Visible;
1022. AddColumn.Visibility = Visibility.Hidden;
1023. AddRow.Visibility = Visibility.Hidden;
1024. DelColumn.Visibility = Visibility.Hidden;
1025. DelSelectedRows.Visibility = Visibility.Hidden;
1026. Calculate.Visibility = Visibility.Hidden;
1027. AddFormul.Visibility = Visibility.Hidden;
1028. if (!f)
1029. DomeOfDecayWindowOpen.Visibility = Visibility.Visible;
1030. MathParser.f = f;
1032. try
1033. {
1034. *//Рассчет формул*
1035. for (int i = f?0:1; i < dat.Columns.Count; i++)
1036. for (int u = 0; u < dat.Rows.Count; u++)
1037. if (dat.Columns[i].ColumnName[0] == '=')
1038. dat.Rows[u][i] = String.Format("{0:f4}", MathParser.Parse(dat.Columns[i].ColumnName.Substring(1), ref dat, u));
1039. }
1040. catch (Exception ex)
1041. {
1042. if (ex.InnerException.Message == "MyException")
1043. MessageBox.Show(ex.Message, "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1044. else
1045. MessageBox.Show("Неверный формат формулы!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1046. }
1047. dat.AcceptChanges();
1049. if (f)
1050. {
1051. List<Pair<Pair<string, string>, List<string>>> prop = new List<Pair<Pair<string, string>, List<string>>>();
1053. if (prop == null)
1054. prop = new List<Pair<Pair<string, string>, List<string>>>();
1056. for (int i = 0; i < dat.Columns.Count; i++)
1057. {
1058. prop.Add(new Pair<Pair<string, string>, List<string>>(new Pair<string, string>(dat.Columns[i].ColumnName, dat.Columns[i].Caption), new List<string>()));
1059. for (int j = 0; j < dat.Rows.Count; j++)
1060. prop[i].Second.Add(dat.Rows[j][i].ToString());
1061. }
1063. Composition el = MendeleevTable.Compos.Find(x => x.Name == elem);
1064. if (el != null)
1065. MendeleevTable.Compos.Find(x => x.Name == elem).Properties = new List<Pair<Pair<string, string>, List<string>>>();
1066. else
1067. MendeleevTable.Compos.Add(new Composition() { Name = elem });
1069. MendeleevTable.Compos.Find(x => x.Name == elem).Properties = prop;
1071. DataGridHelper.Serialize("Compositions.xml", ref MendeleevTable.Compos);
1072. }
1073. else
1074. {
1075. MendeleevTable.BinarySistem.Remove(MendeleevTable.BinarySistem.Find(x => x.TableName == elem));
1076. MendeleevTable.BinarySistem.Add(dat);
1078. DataGridHelper.Serialize("BinarySistems.xml", ref MendeleevTable.BinarySistem);
1079. }
1080. }
1082. */// <summary>*
1083. */// Добавляет столбец-формулу в таблицу*
1084. */// </summary>*
1085. private void AddFormul\_Click(object sender, RoutedEventArgs e)
1086. {
1087. FormulaInput form = new FormulaInput();
1088. form.ShowDialog();
1089. try
1090. {
1091. if (form.formula != "")
1092. {
1093. if (form.symbol != "" && form.symbol != " ")
1094. foreach (DataColumn v in dat.Columns)
1095. if (v.Caption == form.symbol)
1096. throw new DuplicateNameException("Такая формула уже принадлежит данной таблице!", new Exception("MyException"));
1098. DataColumn col = new DataColumn('=' + form.formula) { Caption = form.symbol };
1099. dat.Columns.Add(col);
1100. ComposSystemTable.Columns.Add(new DataGridTextColumn()
1101. {
1102. Header = form.symbol + '=' + form.formula,
1103. Binding = new Binding("[" + ComposSystemTable.Columns.Count + "]"),
1104. IsReadOnly = true
1105. });
1106. for (int u = 0; u < dat.Rows.Count; u++)
1107. dat.Rows[u][dat.Columns.Count - 1] = MathParser.Parse(dat.Columns[dat.Columns.Count - 1].ColumnName.Substring(1), ref dat, u);
1108. }
1109. }
1110. catch (Exception ex)
1111. {
1112. if (ex.InnerException != null && ex.InnerException.Message == "MyException")
1113. MessageBox.Show(ex.Message, "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1114. else
1115. MessageBox.Show("Неверный формат формулы или она уже содержится в данной таблице!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1116. }
1117. if (f && dat.Columns.Count > 0 || !f && dat.Columns.Count > 1)
1118. DelColumn.IsEnabled = true;
1119. }
1121. private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)
1122. {
1123. if (EditTable.Visibility == Visibility.Hidden)
1124. if (MessageBox.Show("Вы точно хотите закрыть окно? Все несохраненные данные будут удалены!", "", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No)
1125. e.Cancel = true;
1126. else
1127. dat.RejectChanges();
1128. }
1130. private void Window\_SizeChanged(object sender, SizeChangedEventArgs e)
1131. {
1132. if (e.PreviousSize.Width != 0)
1133. {
1134. AddRow.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
1135. AddColumn.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
1136. DelColumn.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
1137. DelSelectedRows.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
1138. AddFormul.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
1139. Calculate.Width += (e.NewSize.Width - e.PreviousSize.Width) / 6;
1140. AddRow.RenderTransform = new TranslateTransform(360 + (e.NewSize.Width - 1050) / 3, 0);
1141. AddColumn.RenderTransform = new TranslateTransform(190 + (e.NewSize.Width - 1050) / 6, 0);
1142. DelSelectedRows.RenderTransform = new TranslateTransform(700 + (e.NewSize.Width - 1050) / 1.5, 0);
1143. DelColumn.RenderTransform = new TranslateTransform(530 + (e.NewSize.Width - 1050) / 2, 0);
1144. Calculate.RenderTransform = new TranslateTransform(870 + (e.NewSize.Width - 1050) / 1.2, 0);
1145. }
1146. }
1148. private void ComposSystemTable\_CellEditEnding(object sender, DataGridCellEditEndingEventArgs e)
1149. {
1150. string str1 = (e.EditingElement as TextBox).Text;
1151. double i = 0;
1152. str1 = str1.Replace('.', ',');
1153. if (!double.TryParse(str1, out i))
1154. str1 = StringHelper.DoString(str1);
1155. else
1156. str1 = String.Format("{0:f4}", i);
1158. if (e.Column.DisplayIndex == 0)
1159. str1 = i.ToString();
1161. dat.Rows[e.Row.GetIndex()][e.Column.DisplayIndex] = str1;
1162. (e.EditingElement as TextBox).Text = str1;
1163. }
1165. */// <summary>*
1166. */// Позволяет редактировать таблицу*
1167. */// </summary>*
1168. private void EditTable\_Click(object sender, RoutedEventArgs e)
1169. {
1170. ComposSystemTable.IsReadOnly = false;
1171. EditTable.Visibility = Visibility.Hidden;
1172. AddColumn.Visibility = Visibility.Visible;
1173. AddRow.Visibility = Visibility.Visible;
1174. DelColumn.Visibility = Visibility.Visible;
1175. DelSelectedRows.Visibility = Visibility.Visible;
1176. Calculate.Visibility = Visibility.Visible;
1177. AddFormul.Visibility = Visibility.Visible;
1178. if (!f)
1179. DomeOfDecayWindowOpen.Visibility = Visibility.Hidden;
1180. }
1182. */// <summary>*
1183. */// Открывает окно построения купола распада*
1184. */// </summary>*
1185. private void DomeOfDecayWindowOpen\_Click(object sender, RoutedEventArgs e)
1186. {
1187. new DomeOfDecay(elem).ShowDialog();
1188. }
1189. }
1190. }
1192. **Класс CreateComposSystem:**
      * **CreateComposSystem.xaml**
1193. <Window x:Class="Activision\_Mendeleyev\_table.CreateComposSystem"
1194. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
1195. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
1196. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
1197. xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
1198. mc:Ignorable="d"
1199. Icon="icon.jpeg"
1200. Title="Выберите соединение" Height="300" Width="300" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">
1201. <Grid>
1202. <Button Name="Complete" Content="Готово" Click="Complete\_Click" Width="150" Height="40" RenderTransformOrigin="0.5,0.5" HorizontalAlignment="Center" VerticalAlignment="Bottom" FontSize="16" >
1203. <Button.RenderTransform>
1204. <TransformGroup>
1205. <ScaleTransform/>
1206. <SkewTransform/>
1207. <RotateTransform/>
1208. <TranslateTransform Y="-15"/>
1209. </TransformGroup>
1210. </Button.RenderTransform>
1211. </Button>
1212. <TextBox Name="NewComposSystem" Width="250" Height="30" RenderTransformOrigin="0.5,0.5" FontSize="20">
1213. <TextBox.RenderTransform>
1214. <TransformGroup>
1215. <ScaleTransform/>
1216. <SkewTransform/>
1217. <RotateTransform/>
1218. <TranslateTransform Y="0"/>
1219. </TransformGroup>
1220. </TextBox.RenderTransform>
1221. </TextBox>
1222. <Label Name="label" Height="30" RenderTransformOrigin="0.5,0.5" HorizontalAlignment="Center">
1223. <Label.RenderTransform>
1224. <TransformGroup>
1225. <ScaleTransform/>
1226. <SkewTransform/>
1227. <RotateTransform/>
1228. <TranslateTransform Y="-30"/>
1229. </TransformGroup>
1230. </Label.RenderTransform>
1231. </Label>
1232. <ComboBox Name="ComposSystem" Width="250" Height="40" FontSize="20" VerticalAlignment="Top" RenderTransformOrigin="0.5,0.5" SelectionChanged="Compos\_SelectionChanged" >
1233. <ComboBox.RenderTransform>
1234. <TransformGroup>
1235. <ScaleTransform/>
1236. <SkewTransform/>
1237. <RotateTransform/>
1238. <TranslateTransform Y="15"/>
1239. </TransformGroup>
1240. </ComboBox.RenderTransform>
1241. </ComboBox>
1242. </Grid>
1243. </Window>
      * **CreateComposSystem.cs**
1244. using System.Windows;
1245. using Activision\_Mendeleyev\_table.HelperClasses;
1247. namespace Activision\_Mendeleyev\_table
1248. {
1249. */// <summary>*
1250. */// Логика взаимодействия для CreateComposSystem.xaml*
1251. */// </summary>*
1252. public partial class CreateComposSystem : Window
1253. {
1254. */// <summary>*
1255. */// Флаг: true - соединение, false - система*
1256. */// </summary>*
1257. private bool f;
1259. */// <summary>*
1260. */// Конструктор, инициализирующий окно создания/выбора соединения(системы соединений)*
1261. */// </summary>*
1262. */// <param name="f">флаг: true - соединение, false - система</param>*
1263. public CreateComposSystem(bool f)
1264. {
1265. InitializeComponent();
1267. this.f = f;
1269. if (f)
1270. {
1271. this.Title = "Выберите/введите соединение";
1272. label.Content = "Введите новое соединения";
1273. for (int i = 0; i < MendeleevTable.Compos.Count; i++)
1274. ComposSystem.Items.Add(MendeleevTable.Compos[i].Name);
1275. }
1276. else
1277. {
1278. label.Content = "Введите новую систему соединений";
1279. this.Title = "Выберите/введите систему соединений";
1280. for (int i = 0; i < MendeleevTable.BinarySistem.Count; i++)
1281. ComposSystem.Items.Add(MendeleevTable.BinarySistem[i].TableName);
1282. }
1284. if (ComposSystem.Items.Count > 0)
1285. ComposSystem.SelectedIndex = 0;
1286. }
1288. */// <summary>*
1289. */// Открывает таблицу созданного/выбранного соединения(системы соединений)*
1290. */// </summary>*
1291. private void Complete\_Click(object sender, RoutedEventArgs e)
1292. {
1293. this.Hide();
1294. new ComposSystemContent(StringHelper.DoString(NewComposSystem.Text), f).ShowDialog();
1295. this.Close();
1296. }
1298. */// <summary>*
1299. */// Дублирует соединение(систему соединений) из ComboBox в TextBox*
1300. */// </summary>*
1301. private void Compos\_SelectionChanged(object sender, System.Windows.Controls.SelectionChangedEventArgs e)
1302. {
1303. NewComposSystem.Text = e.AddedItems[0] as string;
1304. }
1305. }
1306. }
1308. **Класс ColumnAddWindow:**
      * **ColumnAddWindow.xaml**
1309. <Window x:Class="Activision\_Mendeleyev\_table.ColumnAddWindow"
1310. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
1311. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
1312. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
1313. xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
1314. mc:Ignorable="d"
1315. Icon="icon.jpeg"
1316. Title="Введите название столбца " Height="150" Width="265" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">
1317. <Grid>
1318. <Button Name="Create" Content="Создать" HorizontalAlignment="Left" Margin="20,70,0,0" VerticalAlignment="Top" Width="105" Click="Create\_Click" Height="38"/>
1319. <TextBox Name="ColumnName" HorizontalAlignment="Left" Height="20" Margin="20,10,0,0" TextWrapping="Wrap" Text="New Column" VerticalAlignment="Top" Width="215"/>
1320. <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"/>
1321. <Label HorizontalAlignment="Left" Height="30" Margin="20,35,0,0" Content="Обозначение: " VerticalAlignment="Top"/>
1322. <Button Name="Cancel" Content="Отмена" HorizontalAlignment="Left" Margin="130,70,0,0" VerticalAlignment="Top" Width="105" Click="Cancel\_Click" Height="38"/>
1323. </Grid>
1324. </Window>
1325. * **ColumnAddWindow.cs**
1326. using System.Windows;
1328. namespace Activision\_Mendeleyev\_table
1329. {
1330. */// <summary>*
1331. */// Логика взаимодействия для ColumnAddWindow.xaml*
1332. */// </summary>*
1333. public partial class ColumnAddWindow : Window
1334. {
1335. */// <summary>*
1336. */// Название столбца*
1337. */// </summary>*
1338. public string name { get; private set; }
1340. */// <summary>*
1341. */// Обозначение свойства, значения которого будут находиться в данном столбце*
1342. */// </summary>*
1343. public string symbol { get; private set; }
1345. */// <summary>*
1346. */// Конструктор, инициализирующий окно создания текстового столбца*
1347. */// </summary>*
1348. public ColumnAddWindow()
1349. {
1350. InitializeComponent();
1351. name = "";
1352. symbol = "";
1353. }
1355. */// <summary>*
1356. */// Создание текстового столбца*
1357. */// </summary>*
1358. private void Create\_Click(object sender, RoutedEventArgs e)
1359. {
1360. name = ColumnName.Text;
1361. symbol = ColumnSymbol.Text;
1362. Close();
1363. }
1365. */// <summary>*
1366. */// Отмена изменений и закрытие окна*
1367. */// </summary>*
1368. private void Cancel\_Click(object sender, RoutedEventArgs e)
1369. {
1370. Close();
1371. }
1372. }
1373. }
1375. **Класс FormulaInput:**
      * **FormulaInput.xaml**
1376. <Window x:Class="Activision\_Mendeleyev\_table.FormulaInput"
1377. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
1378. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
1379. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
1380. xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
1381. mc:Ignorable="d"
1382. Icon="icon.jpeg"
1383. Title="Введите формулу" Height="150" Width="370" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">
1384. <Grid>
1385. <Button Name="Create" Content="Создать" Width="100" Click="Create\_Click" Height="38" VerticalAlignment="Bottom" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5">
1386. <Button.RenderTransform>
1387. <TransformGroup>
1388. <ScaleTransform/>
1389. <SkewTransform/>
1390. <RotateTransform/>
1391. <TranslateTransform X="15" Y="-15"/>
1392. </TransformGroup>
1393. </Button.RenderTransform>
1394. </Button>
1395. <Button Name="Cancel" Content="Отмена" Width="100" Click="Cancel\_Click" Height="38" VerticalAlignment="Bottom" HorizontalAlignment="Right" RenderTransformOrigin="0.5,0.5">
1396. <Button.RenderTransform>
1397. <TransformGroup>
1398. <ScaleTransform/>
1399. <SkewTransform/>
1400. <RotateTransform/>
1401. <TranslateTransform X="-15" Y="-15"/>
1402. </TransformGroup>
1403. </Button.RenderTransform>
1404. </Button>
1405. <Button Name="OpenFormuls" Content="Список формул" Width="100" Click="OpenFormuls\_Click" Height="38" VerticalAlignment="Bottom" HorizontalAlignment="Center" RenderTransformOrigin="0.5,0.5">
1406. <Button.RenderTransform>
1407. <TransformGroup>
1408. <ScaleTransform/>
1409. <SkewTransform/>
1410. <RotateTransform/>
1411. <TranslateTransform Y="-15"/>
1412. </TransformGroup>
1413. </Button.RenderTransform>
1414. </Button>
1415. <TextBox Name="NewFormula" Width="270" Height="30" RenderTransformOrigin="0.5,0.5" FontSize="20" VerticalAlignment="Top">
1416. <TextBox.RenderTransform>
1417. <TransformGroup>
1418. <ScaleTransform/>
1419. <SkewTransform/>
1420. <RotateTransform/>
1421. <TranslateTransform X="30" Y="20"/>
1422. </TransformGroup>
1423. </TextBox.RenderTransform>
1424. </TextBox>
1425. <Label Content="=" Height="30" RenderTransformOrigin="0.5,0.5" HorizontalAlignment="Left" VerticalAlignment="Top">
1426. <Label.RenderTransform>
1427. <TransformGroup>
1428. <ScaleTransform/>
1429. <SkewTransform/>
1430. <RotateTransform/>
1431. <TranslateTransform X="55" Y="20"/>
1432. </TransformGroup>
1433. </Label.RenderTransform>
1434. </Label>
1435. <TextBox Name="FormulaSymbol" Width="40" Height="30" RenderTransformOrigin="0.5,0.5" FontSize="20" HorizontalAlignment="Left" VerticalAlignment="Top" >
1436. <TextBox.RenderTransform>
1437. <TransformGroup>
1438. <ScaleTransform/>
1439. <SkewTransform/>
1440. <RotateTransform/>
1441. <TranslateTransform X="15" Y="20"/>
1442. </TransformGroup>
1443. </TextBox.RenderTransform>
1444. </TextBox>
1445. </Grid>
1446. </Window>
      * **FormulaInput.cs**
1447. using System.Windows;
1448. using Activision\_Mendeleyev\_table.HelperClasses;
1450. namespace Activision\_Mendeleyev\_table
1451. {
1452. */// <summary>*
1453. */// Логика взаимодействия для FormulaInput.xaml*
1454. */// </summary>*
1455. public partial class FormulaInput : Window
1456. {
1457. */// <summary>*
1458. */// Строка-формула*
1459. */// </summary>*
1460. public string formula { get; private set; }
1462. */// <summary>*
1463. */// Обозначение формулы*
1464. */// </summary>*
1465. public string symbol { get; private set; }
1467. */// <summary>*
1468. */// Конструктор, инициализирующий окно создания столбца-формулы*
1469. */// </summary>*
1470. public FormulaInput()
1471. {
1472. InitializeComponent();
1473. formula = "";
1474. symbol = "";
1475. }
1477. */// <summary>*
1478. */// Создание столбца-формулы*
1479. */// </summary>*
1480. private void Create\_Click(object sender, RoutedEventArgs e)
1481. {
1482. formula = StringHelper.DoString(NewFormula.Text);
1483. symbol = StringHelper.DoString(FormulaSymbol.Text);
1484. Close();
1485. }
1487. */// <summary>*
1488. */// Отмена изменений и закрытие окна*
1489. */// </summary>*
1490. private void Cancel\_Click(object sender, RoutedEventArgs e)
1491. {
1492. Close();
1493. }
1495. */// <summary>*
1496. */// Просмотр списка формул*
1497. */// </summary>*
1498. private void OpenFormuls\_Click(object sender, RoutedEventArgs e)
1499. {
1500. new FormulaList().Show();
1501. }
1502. }
1503. }
1505. **Класс FormulaList:**
      * **FormulaList.xaml**
1506. <Window x:Class="Activision\_Mendeleyev\_table.FormulaList"
1507. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
1508. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
1509. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
1510. xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
1511. mc:Ignorable="d"
1512. Icon="icon.jpeg"
1513. Title="Формулы и столбцы" ResizeMode="NoResize" Height="450" Width="800" WindowStartupLocation="CenterScreen">
1514. <Grid>
1515. <DataGrid x:Name="formuls" CanUserSortColumns="False" CanUserReorderColumns="False" CanUserResizeColumns="False" ColumnWidth="\*"
1516. AutoGenerateColumns="False" CanUserAddRows="False" CanUserDeleteRows="False" CanUserResizeRows="False"/>
1517. </Grid>
1518. </Window>
      * **FormulaList.cs**
1519. using System.Data;
1520. using System.Windows;
1522. namespace Activision\_Mendeleyev\_table
1523. {
1524. */// <summary>*
1525. */// Логика взаимодействия для FormulaList.xaml*
1526. */// </summary>*
1527. public partial class FormulaList : Window
1528. {
1529. public FormulaList()
1530. {
1531. InitializeComponent();
1532. DataTable dat = new DataTable();
1533. dat.Columns.Add("Обозначение");
1534. dat.Columns.Add("Название/формула");
1535. foreach (DataColumn i in dat.Columns)
1536. {
1537. formuls.Columns.Add(new System.Windows.Controls.DataGridTextColumn()
1538. {
1539. Header = i.ColumnName,
1540. Binding = new System.Windows.Data.Binding("[" + formuls.Columns.Count + "]")
1541. });
1542. }
1543. for (int i = 0; i < MendeleevTable.Elems[0].DataTable.Columns.Count; i++)
1544. dat.Rows.Add(MendeleevTable.Elems[0].DataTable.Columns[i].Caption != null? MendeleevTable.Elems[0].DataTable.Columns[i].Caption:"", MendeleevTable.Elems[0].DataTable.Columns[i].ColumnName);
1545. for (int i = 0; i < MendeleevTable.Compos.Count; i++)
1546. for (int j = 0; j < MendeleevTable.Compos[i].Properties.Count; j++)
1547. dat.Rows.Add(MendeleevTable.Compos[i].Properties[j].First.First != null ? MendeleevTable.Compos[i].Properties[j].First.Second : "", (MendeleevTable.Compos[i].Properties[j].First.First[0] != '=')?MendeleevTable.Compos[i].Properties[j].First.First: MendeleevTable.Compos[i].Properties[j].First.First.Substring(1));
1548. for (int i = 0; i < MendeleevTable.BinarySistem.Count; i++)
1549. for (int j = 0; j < MendeleevTable.BinarySistem[i].Columns.Count; j++)
1550. dat.Rows.Add(MendeleevTable.BinarySistem[i].Columns[j].Caption != null ? MendeleevTable.BinarySistem[i].Columns[j].Caption : "", (MendeleevTable.BinarySistem[i].Columns[j].ColumnName[0] != '=')?MendeleevTable.BinarySistem[i].Columns[j].ColumnName: MendeleevTable.BinarySistem[i].Columns[j].ColumnName.Substring(1));
1552. formuls.ItemsSource = dat.DefaultView;
1553. }
1554. }
1555. }
1557. **Класс DataSettings:**
      * **DataSettings.xaml**
1558. <Window x:Class="Activision\_Mendeleyev\_table.DataSettings"
1559. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
1560. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
1561. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
1562. xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
1563. mc:Ignorable="d"
1564. Icon="icon.jpeg"
1565. Title="Настройка данных" WindowStartupLocation="CenterScreen" Height="600" Width="600" Closing="Window\_Closing" ResizeMode="NoResize" >
1566. <Grid>
1567. <TextBlock Text="Первый химический элемент:" TextWrapping="Wrap" RenderTransformOrigin="0.5,0.5" FontSize="20" VerticalAlignment="Top" HorizontalAlignment="Left" Width="300" Height="40" >
1568. <TextBlock.RenderTransform>
1569. <TransformGroup>
1570. <ScaleTransform/>
1571. <SkewTransform/>
1572. <RotateTransform/>
1573. <TranslateTransform X="20" Y="70"/>
1574. </TransformGroup>
1575. </TextBlock.RenderTransform>
1576. </TextBlock>
1577. <TextBlock Text="Символ химиче-ского элемента" TextWrapping="Wrap" RenderTransformOrigin="0.5,0.5" FontSize="20" VerticalAlignment="Top" HorizontalAlignment="Left" Width="160" Height="55" >
1578. <TextBlock.RenderTransform>
1579. <TransformGroup>
1580. <ScaleTransform/>
1581. <SkewTransform/>
1582. <RotateTransform/>
1583. <TranslateTransform X="310" Y="10"/>
1584. </TransformGroup>
1585. </TextBlock.RenderTransform>
1586. </TextBlock>
1587. <TextBlock Text="Номер строки" TextWrapping="Wrap" RenderTransformOrigin="0.5,0.5" FontSize="20" VerticalAlignment="Top" HorizontalAlignment="Left" Width="100" Height="55" >
1588. <TextBlock.RenderTransform>
1589. <TransformGroup>
1590. <ScaleTransform/>
1591. <SkewTransform/>
1592. <RotateTransform/>
1593. <TranslateTransform X="490" Y="10"/>
1594. </TransformGroup>
1595. </TextBlock.RenderTransform>
1596. </TextBlock>
1597. <TextBox x:Name="elemA" Text="A" FontSize="20" RenderTransformOrigin="0.5,0.5" Width="100" Height="30" HorizontalAlignment="Left" VerticalAlignment="Top">
1598. <TextBox.RenderTransform>
1599. <TransformGroup>
1600. <ScaleTransform/>
1601. <SkewTransform/>
1602. <RotateTransform/>
1603. <TranslateTransform X="340" Y="70"/>
1604. </TransformGroup>
1605. </TextBox.RenderTransform>
1606. </TextBox>
1607. <TextBox x:Name="numA" Text="0" FontSize="20" RenderTransformOrigin="0.5,0.5" Width="100" Height="30" HorizontalAlignment="Left" VerticalAlignment="Top">
1608. <TextBox.RenderTransform>
1609. <TransformGroup>
1610. <ScaleTransform/>
1611. <SkewTransform/>
1612. <RotateTransform/>
1613. <TranslateTransform X="470" Y="70"/>
1614. </TransformGroup>
1615. </TextBox.RenderTransform>
1616. </TextBox>
1617. <TextBlock Text="Второй химический элемент:" TextWrapping="Wrap" FontSize="20" VerticalAlignment="Top" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Width="300" Height="40">
1618. <TextBlock.RenderTransform>
1619. <TransformGroup>
1620. <ScaleTransform/>
1621. <SkewTransform/>
1622. <RotateTransform/>
1623. <TranslateTransform X="20" Y="110"/>
1624. </TransformGroup>
1625. </TextBlock.RenderTransform>
1626. </TextBlock>
1627. <TextBox x:Name="elemB" Text="B" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Left" Height="30" Width="100">
1628. <TextBox.RenderTransform>
1629. <TransformGroup>
1630. <ScaleTransform/>
1631. <SkewTransform/>
1632. <RotateTransform/>
1633. <TranslateTransform X="340" Y="110"/>
1634. </TransformGroup>
1635. </TextBox.RenderTransform>
1636. </TextBox>
1637. <TextBox x:Name="numB" Text="0" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Left" Height="30" Width="100">
1638. <TextBox.RenderTransform>
1639. <TransformGroup>
1640. <ScaleTransform/>
1641. <SkewTransform/>
1642. <RotateTransform/>
1643. <TranslateTransform X="470" Y="110"/>
1644. </TransformGroup>
1645. </TextBox.RenderTransform>
1646. </TextBox>
1647. <TextBlock Text="Общий химический элемент:" TextWrapping="Wrap" FontSize="20" VerticalAlignment="Top" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Width="300" Height="40">
1648. <TextBlock.RenderTransform>
1649. <TransformGroup>
1650. <ScaleTransform/>
1651. <SkewTransform/>
1652. <RotateTransform/>
1653. <TranslateTransform X="20" Y="150"/>
1654. </TransformGroup>
1655. </TextBlock.RenderTransform>
1656. </TextBlock>
1657. <TextBox x:Name="elemX" Text="X" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Left" Height="30" Width="100">
1658. <TextBox.RenderTransform>
1659. <TransformGroup>
1660. <ScaleTransform/>
1661. <SkewTransform/>
1662. <RotateTransform/>
1663. <TranslateTransform X="340" Y="150"/>
1664. </TransformGroup>
1665. </TextBox.RenderTransform>
1666. </TextBox>
1667. <TextBox x:Name="numX" Text="0" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Left" Height="30" Width="100">
1668. <TextBox.RenderTransform>
1669. <TransformGroup>
1670. <ScaleTransform/>
1671. <SkewTransform/>
1672. <RotateTransform/>
1673. <TranslateTransform X="470" Y="150"/>
1674. </TransformGroup>
1675. </TextBox.RenderTransform>
1676. </TextBox>
1677. <TextBlock Text="Обозначение радиуса иона элемента в таблице данных:" TextWrapping="Wrap" FontSize="20" VerticalAlignment="Top" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Width="347" Height="55">
1678. <TextBlock.RenderTransform>
1679. <TransformGroup>
1680. <ScaleTransform/>
1681. <SkewTransform/>
1682. <RotateTransform/>
1683. <TranslateTransform X="20" Y="190"/>
1684. </TransformGroup>
1685. </TextBlock.RenderTransform>
1686. </TextBlock>
1687. <TextBox x:Name="r" Text="R(i)" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Left" Height="30" Width="100">
1688. <TextBox.RenderTransform>
1689. <TransformGroup>
1690. <ScaleTransform/>
1691. <SkewTransform/>
1692. <RotateTransform/>
1693. <TranslateTransform X="400" Y="205"/>
1694. </TransformGroup>
1695. </TextBox.RenderTransform>
1696. </TextBox>
1697. <TextBlock Text="Обозначение электроотрицательности элемента в таблице данных:" TextWrapping="Wrap" FontSize="20" VerticalAlignment="Top" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Width="365" Height="55">
1698. <TextBlock.RenderTransform>
1699. <TransformGroup>
1700. <ScaleTransform/>
1701. <SkewTransform/>
1702. <RotateTransform/>
1703. <TranslateTransform X="20" Y="260"/>
1704. </TransformGroup>
1705. </TextBlock.RenderTransform>
1706. </TextBlock>
1707. <TextBlock Text="Обозначение формального заряда элемента в таблице данных:" TextWrapping="Wrap" FontSize="20" VerticalAlignment="Top" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Width="365" Height="55">
1708. <TextBlock.RenderTransform>
1709. <TransformGroup>
1710. <ScaleTransform/>
1711. <SkewTransform/>
1712. <RotateTransform/>
1713. <TranslateTransform X="20" Y="330"/>
1714. </TransformGroup>
1715. </TextBlock.RenderTransform>
1716. </TextBlock>
1717. <TextBox x:Name="x" Text="х" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Left" Height="30" Width="100">
1718. <TextBox.RenderTransform>
1719. <TransformGroup>
1720. <ScaleTransform/>
1721. <SkewTransform/>
1722. <RotateTransform/>
1723. <TranslateTransform X="400" Y="270"/>
1724. </TransformGroup>
1725. </TextBox.RenderTransform>
1726. </TextBox>
1727. <TextBox x:Name="FZ" Text="ФЗ" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Left" Height="30" Width="100">
1728. <TextBox.RenderTransform>
1729. <TransformGroup>
1730. <ScaleTransform/>
1731. <SkewTransform/>
1732. <RotateTransform/>
1733. <TranslateTransform X="400" Y="340"/>
1734. </TransformGroup>
1735. </TextBox.RenderTransform>
1736. </TextBox>
1737. <TextBlock Text="c = " FontSize="20" VerticalAlignment="Top" TextWrapping="Wrap" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Width="50" Height="40">
1738. <TextBlock.RenderTransform>
1739. <TransformGroup>
1740. <ScaleTransform/>
1741. <SkewTransform/>
1742. <RotateTransform/>
1743. <TranslateTransform X="20" Y="410"/>
1744. </TransformGroup>
1745. </TextBlock.RenderTransform>
1746. </TextBlock>
1747. <TextBox x:Name="c" Text="-1" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Left" Height="30" Width="100">
1748. <TextBox.RenderTransform>
1749. <TransformGroup>
1750. <ScaleTransform/>
1751. <SkewTransform/>
1752. <RotateTransform/>
1753. <TranslateTransform X="90" Y="410"/>
1754. </TransformGroup>
1755. </TextBox.RenderTransform>
1756. </TextBox>
1757. <TextBlock Text="A = " FontSize="20" VerticalAlignment="Top" TextWrapping="Wrap" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Width="50" Height="40">
1758. <TextBlock.RenderTransform>
1759. <TransformGroup>
1760. <ScaleTransform/>
1761. <SkewTransform/>
1762. <RotateTransform/>
1763. <TranslateTransform X="215" Y="435"/>
1764. </TransformGroup>
1765. </TextBlock.RenderTransform>
1766. </TextBlock>
1767. <TextBox x:Name="\_A" Text="-1" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Left" Height="30" Width="100">
1768. <TextBox.RenderTransform>
1769. <TransformGroup>
1770. <ScaleTransform/>
1771. <SkewTransform/>
1772. <RotateTransform/>
1773. <TranslateTransform X="265" Y="435"/>
1774. </TransformGroup>
1775. </TextBox.RenderTransform>
1776. </TextBox>
1777. <TextBlock Text="n = " FontSize="20" VerticalAlignment="Top" TextWrapping="Wrap" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Width="50" Height="40">
1778. <TextBlock.RenderTransform>
1779. <TransformGroup>
1780. <ScaleTransform/>
1781. <SkewTransform/>
1782. <RotateTransform/>
1783. <TranslateTransform X="20" Y="460"/>
1784. </TransformGroup>
1785. </TextBlock.RenderTransform>
1786. </TextBlock>
1787. <TextBox x:Name="n" Text="-1" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Left" Height="30" Width="100">
1788. <TextBox.RenderTransform>
1789. <TransformGroup>
1790. <ScaleTransform/>
1791. <SkewTransform/>
1792. <RotateTransform/>
1793. <TranslateTransform X="90" Y="460"/>
1794. </TransformGroup>
1795. </TextBox.RenderTransform>
1796. </TextBox>
1797. <TextBlock Text="m = " FontSize="20" TextWrapping="Wrap" VerticalAlignment="Top" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Width="50" Height="40">
1798. <TextBlock.RenderTransform>
1799. <TransformGroup>
1800. <ScaleTransform/>
1801. <SkewTransform/>
1802. <RotateTransform/>
1803. <TranslateTransform X="390" Y="410"/>
1804. </TransformGroup>
1805. </TextBlock.RenderTransform>
1806. </TextBlock>
1807. <TextBox x:Name="m" Text="-1" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Right" Height="30" Width="100" >
1808. <TextBox.RenderTransform>
1809. <TransformGroup>
1810. <ScaleTransform/>
1811. <SkewTransform/>
1812. <RotateTransform/>
1813. <TranslateTransform X="-25" Y="410"/>
1814. </TransformGroup>
1815. </TextBox.RenderTransform>
1816. </TextBox>
1817. <TextBlock Text="z = " FontSize="20" TextWrapping="Wrap" VerticalAlignment="Top" HorizontalAlignment="Left" RenderTransformOrigin="0.5,0.5" Width="50" Height="40">
1818. <TextBlock.RenderTransform>
1819. <TransformGroup>
1820. <ScaleTransform/>
1821. <SkewTransform/>
1822. <RotateTransform/>
1823. <TranslateTransform X="390" Y="460"/>
1824. </TransformGroup>
1825. </TextBlock.RenderTransform>
1826. </TextBlock>
1827. <TextBox x:Name="z" Text="-1" FontSize="20" RenderTransformOrigin="0.5,0.5" VerticalAlignment="Top" HorizontalAlignment="Right" Height="30" Width="100">
1828. <TextBox.RenderTransform>
1829. <TransformGroup>
1830. <ScaleTransform/>
1831. <SkewTransform/>
1832. <RotateTransform/>
1833. <TranslateTransform X="-25" Y="460"/>
1834. </TransformGroup>
1835. </TextBox.RenderTransform>
1836. </TextBox>
1837. <Button Name="Save" VerticalAlignment="Bottom" HorizontalAlignment="Center" Content="Сохранить изменения и закрыть окно" RenderTransformOrigin="0.5,0.5" Width="240" Height="35" Click="Save\_Click" >
1838. <Button.RenderTransform>
1839. <TransformGroup>
1840. <ScaleTransform/>
1841. <SkewTransform/>
1842. <RotateTransform/>
1843. <TranslateTransform Y="-20"/>
1844. </TransformGroup>
1845. </Button.RenderTransform>
1846. </Button>
1847. </Grid>
1848. </Window>
      * **DataSettings.cs**
1849. using Activision\_Mendeleyev\_table.HelperClasses;
1850. using System.Windows;
1852. namespace Activision\_Mendeleyev\_table
1853. {
1854. */// <summary>*
1855. */// Логика взаимодействия для DataSettings.xaml*
1856. */// </summary>*
1857. public partial class DataSettings : Window
1858. {
1859. */// <summary>*
1860. */// Система соединений*
1861. */// </summary>*
1862. private BinSystem sys;
1864. */// <summary>*
1865. */// Настройка некоторых параметров системы*
1866. */// </summary>*
1867. */// <param name="sys">система соединений</param>*
1868. public DataSettings(BinSystem sys)
1869. {
1870. InitializeComponent();
1871. this.sys = sys;
1872. if (sys != null)
1873. {
1874. elemA.Text = sys.ElementA;
1875. elemB.Text = sys.ElementB;
1876. elemX.Text = sys.ElementX;
1877. double[] par = sys.GetData();
1878. numX.Text = sys.numX.ToString();
1879. numB.Text = sys.numB.ToString();
1880. numA.Text = sys.numA.ToString();
1881. z.Text = par[3].ToString();
1882. c.Text = par[0].ToString();
1883. m.Text = par[1].ToString();
1884. n.Text = par[2].ToString();
1885. \_A.Text = sys.A.ToString();
1886. }
1887. }
1889. */// <summary>*
1890. */// Возвращает систему соединений*
1891. */// </summary>*
1892. public BinSystem GetBS() { return sys; }
1894. private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)
1895. {
1896. if (MessageBox.Show("Вы точно хотите закрыть окно? Все несохраненные данные будут удалены!", "", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No)
1897. e.Cancel = true;
1898. }
1900. */// <summary>*
1901. */// Сохраняет параметры в системе и закрывает окно*
1902. */// </summary>*
1903. private void Save\_Click(object sender, RoutedEventArgs e)
1904. {
1906. Composition A = MendeleevTable.Elems.Find(x => x.Name == elemA.Text);
1907. Composition B = MendeleevTable.Elems.Find(x => x.Name == elemB.Text);
1908. Composition X = MendeleevTable.Elems.Find(x => x.Name == elemX.Text);
1909. if (A == null || B == null || X == null)
1910. MessageBox.Show("Неверные заданы названия элементов входящих в систему! Измените их в меню настроек!", "Error",
1911. MessageBoxButton.OK, MessageBoxImage.Error);
1912. else
1913. {
1914. sys = new BinSystem(elemA.Text + elemX.Text + '-' + elemB.Text + elemX.Text, A, B, X);
1915. sys.symbols[0] = r.Text;
1916. sys.symbols[1] = x.Text;
1917. sys.symbols[2] = FZ.Text;
1918. \_A.Text = \_A.Text.Replace('.', ',');
1919. double q = -1;
1920. int w = -1;
1921. if (!double.TryParse(z.Text, out q) || q < 0)
1922. MessageBox.Show("Поле z - неотрицательное число!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1923. else if (!int.TryParse(numA.Text, out w) || w < 0)
1924. MessageBox.Show("Поле numA - целое неотрицательное число!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1925. else if (!int.TryParse(numB.Text, out w) || w < 0)
1926. MessageBox.Show("Поле numB - целое неотрицательное число!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1927. else if (!int.TryParse(numX.Text, out w) || w < 0)
1928. MessageBox.Show("Поле numX - целое неотрицательное число!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1929. else if (!double.TryParse(m.Text, out q) || q < 0)
1930. MessageBox.Show("Поле m - неотрицательное число!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1931. else if (!double.TryParse(c.Text, out q) || q < 0)
1932. MessageBox.Show("Поле c - неотрицательное число!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1933. else if (!double.TryParse(n.Text, out q) || q < 0)
1934. MessageBox.Show("Поле n - неотрицательное число!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1935. else if (!double.TryParse(\_A.Text, out q) || q < 0)
1936. MessageBox.Show("Поле A - неотрицательное число!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
1937. else
1938. {
1939. sys.SetData(double.Parse(c.Text), double.Parse(m.Text), double.Parse(n.Text), double.Parse(z.Text));
1940. sys.numA = int.Parse(numA.Text);
1941. sys.numB = int.Parse(numB.Text);
1942. sys.numX = int.Parse(numX.Text);
1943. sys.A = double.Parse(\_A.Text);
1944. this.Close();
1945. }
1946. }
1947. }
1948. }
1949. }
1951. **Класс DomeOfDecay:**
      * **DomeOfDecay.xaml**
1952. <Window x:Class="Activision\_Mendeleyev\_table.DomeOfDecay"
1953. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
1954. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
1955. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
1956. xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
1957. xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"
1958. xmlns:wfi="clr-namespace:System.Windows.Forms.Integration;assembly=WindowsFormsIntegration"
1959. xmlns:local="clr-namespace:Activision\_Mendeleyev\_table.HelperClasses"
1960. mc:Ignorable="d"
1961. Icon="icon.jpeg"
1962. Title="Купол распада" Height="465" Width="557" WindowStartupLocation="CenterScreen" MinHeight="765" MinWidth="857" Closing="Window\_Closing" SizeChanged="Window\_SizeChanged" SourceInitialized="Window\_SourceInitialized">
1963. <Window.Resources>
1964. <local:RoundConverter x:Key="RoundConverter"></local:RoundConverter>
1965. </Window.Resources>
1966. <Grid>
1967. <Grid.RowDefinitions>
1968. <RowDefinition Height="20"/>
1969. <RowDefinition Height="\*" />
1970. <RowDefinition Height="50"/>
1971. </Grid.RowDefinitions>
1972. <Grid.ColumnDefinitions>
1973. <ColumnDefinition Width="21\*" />
1974. <ColumnDefinition Width="628\*"/>
1975. <ColumnDefinition Width="200"/>
1976. </Grid.ColumnDefinitions>
1977. <Menu Grid.Row="0" Grid.ColumnSpan="2">
1978. <MenuItem Header="Графическое меню">
1979. <MenuItem Header="Настройка цвета">
1980. <MenuItem Header="Теория">
1981. <xctk:ColorPicker Name="Theory" Width="50" SelectedColor="Black" ShowTabHeaders="False" />
1982. </MenuItem>
1983. <MenuItem Header="Эксперимент">
1984. <xctk:ColorPicker Name="Experiment" Width="50" SelectedColor="Red" ShowTabHeaders="False" />
1985. </MenuItem>
1986. <MenuItem Header="Аппроксимация/чувствительность">
1987. <xctk:ColorPicker Name="Approximation" Width="50" SelectedColor="Green" ShowTabHeaders="False" />
1988. </MenuItem>
1989. </MenuItem>
1990. <MenuItem Header="Настройка границ">
1991. <MenuItem Header="Верхняя температура" >
1992. <TextBox Name="UpT" Text="-1" Width="50" />
1993. </MenuItem>
1994. <MenuItem Header="Нижняя температура" >
1995. <TextBox Name="DownT" Text="-1" Width="50" />
1996. </MenuItem>
1997. <MenuItem Header="Верхняя граница параметра Rmin" >
1998. <TextBox Name="UpR" Text="10" Width="50" IsEnabled="False"/>
1999. </MenuItem>
2000. <MenuItem Header="Нижняя граница параметра Rmin" >
2001. <TextBox Name="DownR" Text="0.01" Width="50" IsEnabled="False" />
2002. </MenuItem>
2003. <MenuItem Header="Верхняя граница параметра delEps" >
2004. <TextBox Name="UpdE" Text="1" Width="50" IsEnabled="False" />
2005. </MenuItem>
2006. <MenuItem Header="Нижняя граница параметра delEps" >
2007. <TextBox Name="DowndE" Text="0.01" Width="50" IsEnabled="False" />
2008. </MenuItem>
2009. <MenuItem Header="Верхняя граница параметра c" >
2010. <TextBox Name="Upc" Text="100" Width="50" IsEnabled="False" />
2011. </MenuItem>
2012. <MenuItem Header="Нижняя граница параметра c" >
2013. <TextBox Name="Downc" Text="0.01" Width="50" IsEnabled="False" />
2014. </MenuItem>
2015. </MenuItem>
2016. <MenuItem Name="IsExpPoints" Header="Эксперимент точками" IsCheckable="True" IsChecked="True" Checked="MenuItem\_Checked" Unchecked="MenuItem\_Unchecked" />
2017. </MenuItem>
2018. <MenuItem Header="Настройка данных" Click="DataSettings\_Click" />
2019. <MenuItem Header="Оценка чувствительности" Click="Sensitivity\_Click" />
2020. </Menu>
2021. <wfi:WindowsFormsHost Name="host" Grid.Column="0" Grid.Row="1" Grid.ColumnSpan="2"/>
2022. <DataGrid Name="Points" Grid.Column="2" Grid.Row="1" AutoGenerateColumns="False" CanUserReorderColumns="False" CanUserSortColumns="False" CanUserResizeRows="False" CanUserResizeColumns="False" CanUserAddRows="True" CanUserDeleteRows="True" ColumnWidth="\*" CellEditEnding="Points\_CellEditEnding" RowEditEnding="Points\_RowEditEnding"/>
2023. <Label Name="dE\_label" Content="delEps" Grid.Column="2" Grid.Row="1" HorizontalAlignment="Center" Visibility="Hidden" VerticalAlignment="Top" Margin="78,0" />
2024. <Label Name="R\_label" Content="R" Grid.Column="2" Grid.Row="1" HorizontalAlignment="Left" VerticalAlignment="Top" Visibility="Hidden" RenderTransformOrigin="0.5,0.5" >
2025. <Label.RenderTransform>
2026. <TransformGroup>
2027. <ScaleTransform/>
2028. <SkewTransform/>
2029. <RotateTransform/>
2030. <TranslateTransform X="30"/>
2031. </TransformGroup>
2032. </Label.RenderTransform>
2033. </Label>
2034. <Label Name="c\_label" Content="c" Grid.Column="2" Grid.Row="1" HorizontalAlignment="Right" VerticalAlignment="Top" Visibility="Hidden" RenderTransformOrigin="0.5,0.5" >
2035. <Label.RenderTransform>
2036. <TransformGroup>
2037. <ScaleTransform/>
2038. <SkewTransform/>
2039. <RotateTransform/>
2040. <TranslateTransform X="-20"/>
2041. </TransformGroup>
2042. </Label.RenderTransform>
2043. </Label>
2044. <TextBox Name="dE\_text" Grid.Column="2" Grid.Row="1" Text="{Binding ElementName=dE, Path=Value, UpdateSourceTrigger=PropertyChanged, Converter={StaticResource RoundConverter}}" HorizontalAlignment="Center" Visibility="Hidden" VerticalAlignment="Top" Height="20" Width="50" IsReadOnly="True" RenderTransformOrigin="0.5,0.5">
2045. <TextBox.RenderTransform>
2046. <TransformGroup>
2047. <ScaleTransform/>
2048. <SkewTransform/>
2049. <RotateTransform/>
2050. <TranslateTransform Y="35"/>
2051. </TransformGroup>
2052. </TextBox.RenderTransform>
2053. </TextBox>
2054. <TextBox Name="R\_text" Grid.Column="2" Grid.Row="1" Text="{Binding ElementName=R, Path=Value, UpdateSourceTrigger=PropertyChanged, Converter={StaticResource RoundConverter}}" HorizontalAlignment="Left" VerticalAlignment="Top" Visibility="Hidden" Height="20" Width="50" RenderTransformOrigin="0.5,0.5" IsReadOnly="True">
2055. <TextBox.RenderTransform>
2056. <TransformGroup>
2057. <ScaleTransform/>
2058. <SkewTransform/>
2059. <RotateTransform/>
2060. <TranslateTransform X="20" Y="35"/>
2061. </TransformGroup>
2062. </TextBox.RenderTransform>
2063. </TextBox>
2064. <TextBox Name="c\_text" Grid.Column="2" Grid.Row="1" Text="{Binding ElementName=c, Path=Value, UpdateSourceTrigger=PropertyChanged, Converter={StaticResource RoundConverter}}" HorizontalAlignment="Right" VerticalAlignment="Top" Visibility="Hidden" Height="20" Width="50" RenderTransformOrigin="0.5,0.5" IsReadOnly="True">
2065. <TextBox.RenderTransform>
2066. <TransformGroup>
2067. <ScaleTransform/>
2068. <SkewTransform/>
2069. <RotateTransform/>
2070. <TranslateTransform X="-20" Y="35"/>
2071. </TransformGroup>
2072. </TextBox.RenderTransform>
2073. </TextBox>
2074. <Slider Name="R" Grid.Column="2" Grid.Row="1" Orientation="Vertical" HorizontalAlignment="Left" VerticalAlignment="Top" Height="600" Visibility="Hidden" RenderTransformOrigin="0.5,0.5" ValueChanged="R\_ValueChanged">
2075. <Slider.RenderTransform>
2076. <TransformGroup>
2077. <ScaleTransform/>
2078. <SkewTransform/>
2079. <RotateTransform/>
2080. <TranslateTransform X="20" Y="70"/>
2081. </TransformGroup>
2082. </Slider.RenderTransform>
2083. </Slider>
2084. <Slider Name="dE" Grid.Column="2" Grid.Row="1" Orientation="Vertical" HorizontalAlignment="Center" VerticalAlignment="Top" Height="600" Visibility="Hidden" RenderTransformOrigin="0.5,0.5" ValueChanged="dE\_ValueChanged">
2085. <Slider.RenderTransform>
2086. <TransformGroup>
2087. <ScaleTransform/>
2088. <SkewTransform/>
2089. <RotateTransform/>
2090. <TranslateTransform Y="70"/>
2091. </TransformGroup>
2092. </Slider.RenderTransform>
2093. </Slider>
2094. <Slider Name="c" Grid.Column="2" Grid.Row="1" Orientation="Vertical" HorizontalAlignment="Right" VerticalAlignment="Top" Height="600" Visibility="Hidden" RenderTransformOrigin="0.5,0.5" ValueChanged="c\_ValueChanged">
2095. <Slider.RenderTransform>
2096. <TransformGroup>
2097. <ScaleTransform/>
2098. <SkewTransform/>
2099. <RotateTransform/>
2100. <TranslateTransform X="-20" Y="70"/>
2101. </TransformGroup>
2102. </Slider.RenderTransform>
2103. </Slider>
2104. <Button Name="Build" Grid.Column="0" Grid.Row="2" Content="Построить купол" HorizontalAlignment="Left" VerticalAlignment="Center" Width="150" Height="34" RenderTransformOrigin="0.5,0.5" Click="Build\_Click" Grid.ColumnSpan="2" Margin="0,8">
2105. <Button.RenderTransform>
2106. <TransformGroup>
2107. <ScaleTransform/>
2108. <SkewTransform/>
2109. <RotateTransform/>
2110. <TranslateTransform X="20"/>
2111. </TransformGroup>
2112. </Button.RenderTransform>
2113. </Button>
2114. <Button Name="Approxi" Grid.Column="0" Grid.Row="2" Content="Построить функцию Hsm и аппроксимировать" HorizontalAlignment="Left" VerticalAlignment="Center" Width="285" Height="34" RenderTransformOrigin="0.5,0.5" Click="Approxi\_Click" Grid.ColumnSpan="2" Margin="0,8">
2115. <Button.RenderTransform>
2116. <TransformGroup>
2117. <ScaleTransform/>
2118. <SkewTransform/>
2119. <RotateTransform/>
2120. <TranslateTransform X="190"/>
2121. </TransformGroup>
2122. </Button.RenderTransform>
2123. </Button>
2124. <Button Name="Back" Grid.Column="0" Grid.Row="2" Content="Назад" HorizontalAlignment="Center" VerticalAlignment="Center" Width="100" Height="34" Click="Back\_Click" Grid.ColumnSpan="2" Visibility="Hidden" />
2125. <Button Name="Save" Grid.Column="1" Grid.Row="2" Content="Сохранить точки" HorizontalAlignment="Right" VerticalAlignment="Center" Height="34" Width="150" Click="Save\_Click" RenderTransformOrigin="0.5,0.5" Margin="0,8">
2126. <Button.RenderTransform>
2127. <TransformGroup>
2128. <ScaleTransform/>
2129. <SkewTransform/>
2130. <RotateTransform/>
2131. <TranslateTransform X="10"/>
2132. </TransformGroup>
2133. </Button.RenderTransform>
2134. </Button>
2135. <Button Name="Load" Grid.Column="2" Grid.Row="2" Content="Загрузить точки" HorizontalAlignment="Right" VerticalAlignment="Center" Width="150" Height="34" RenderTransformOrigin="0.5,0.5" Click="Load\_Click" Margin="0,8">
2136. <Button.RenderTransform>
2137. <TransformGroup>
2138. <ScaleTransform/>
2139. <SkewTransform/>
2140. <RotateTransform/>
2141. <TranslateTransform X="-20"/>
2142. </TransformGroup>
2143. </Button.RenderTransform>
2144. </Button>
2145. </Grid>
2146. </Window>
      * **DomeOfDecay.cs**
2147. using Activision\_Mendeleyev\_table.Approximation;
2148. using Activision\_Mendeleyev\_table.HelperClasses;
2149. using System;
2150. using System.Collections.Generic;
2151. using System.Drawing;
2152. using System.IO;
2153. using System.Runtime.InteropServices;
2154. using System.Text.RegularExpressions;
2155. using System.Windows;
2156. using System.Windows.Controls;
2157. using System.Windows.Data;
2159. namespace Activision\_Mendeleyev\_table
2160. {
2161. */// <summary>*
2162. */// Логика взаимодействия для DomeOfDecay.xaml*
2163. */// </summary>*
2164. public partial class DomeOfDecay : Window
2165. {
2166. */// <summary>*
2167. */// Лист точек, представленных в DataGrid*
2168. */// </summary>*
2169. private List<List<double>> dat = new List<List<double>>();
2170. */// <summary>*
2171. */// Системы соединений(базовая и аппроксимированная)*
2172. */// </summary>*
2173. private BinSystem sys, sys\_ap = null;
2174. */// <summary>*
2175. */// Графики(купол распада/функция смешения и аппроксимированная функция смешения)*
2176. */// </summary>*
2177. private DrawingClasses.CollapseGraph graph, graph\_ap;
2178. */// <summary>*
2179. */// Флаг: true - купол распада, false - функция смешения*
2180. */// </summary>*
2181. private bool f = true;
2182. private System.Windows.Forms.PictureBox diag = new System.Windows.Forms.PictureBox();
2184. [DllImport("user32.dll")]
2185. private static extern int GetWindowLong(IntPtr hWnd, int nIndex);
2186. [DllImport("user32.dll")]
2187. private static extern int SetWindowLong(IntPtr hWnd, int nIndex, int dwNewLong);
2189. private const int GWL\_STYLE = -16;
2190. private const int WS\_MAXIMIZEBOX = 0x10000;
2192. */// <summary>*
2193. */// Инициализация элемента host*
2194. */// </summary>*
2195. private void Window\_SourceInitialized(object sender, EventArgs e)
2196. {
2197. var hwnd = new System.Windows.Interop.WindowInteropHelper((Window)sender).Handle;
2198. var value = GetWindowLong(hwnd, GWL\_STYLE);
2199. SetWindowLong(hwnd, GWL\_STYLE, (int)(value & ~WS\_MAXIMIZEBOX));
2200. }
2202. */// <summary>*
2203. */// Первоначальные настройки и построение купола распада*
2204. */// </summary>*
2205. */// <param name="name">обозначение системы</param>*
2206. public DomeOfDecay(string name)
2207. {
2208. InitializeComponent();
2210. *//Привязка горячей клавише Delete к методу DeleteSelectedRows*
2211. new HotKey(System.Windows.Input.Key.Delete, KeyModifier.None, DeleteSelectedRows);
2212. string[] elems = Parse(name);
2213. Composition A = MendeleevTable.Elems.Find(x => x.Name == elems[0]);
2214. Composition B = MendeleevTable.Elems.Find(x => x.Name == elems[1]);
2215. Composition X = MendeleevTable.Elems.Find(x => x.Name == elems[2]);
2216. if (A == null || B == null || X == null)
2217. MessageBox.Show("Неверно заданы названия элементов входящих в систему! Измените их в меню настроек!", "Error",
2218. MessageBoxButton.OK, MessageBoxImage.Error);
2219. else
2220. sys = new BinSystem(name, A, B, X);
2222. DataSettings ds = new DataSettings(sys);
2223. ds.ShowDialog();
2224. sys = ds.GetBS();
2226. host.Child = diag;
2227. diag.Paint += new System.Windows.Forms.PaintEventHandler(diag\_Paint);
2229. Points.Columns.Add(new DataGridTextColumn()
2230. {
2231. Header = "x",
2232. Binding = new Binding("[0]")
2233. });
2234. Points.Columns.Add(new DataGridTextColumn()
2235. {
2236. Header = "y",
2237. Binding = new Binding("[1]")
2238. });
2239. Points.ItemsSource = dat;
2240. }
2242. */// <summary>*
2243. */// Получение химических элементов из обозначения системы*
2244. */// </summary>*
2245. */// <param name="s">обозначение системы</param>*
2246. */// <returns>массив обозначений химических элементов</returns>*
2247. public static string[] Parse(string s)
2248. {
2249. string[] names = new string[] { "", "", ""};
2251. s = s.Replace(" ", "");
2253. if (new Regex(@"[A-Z]{1}[a-z]?[₀₁₂₃₄₅₆₇₈₉]\*[A-Z]{1}[a-z]?[₀₁₂₃₄₅₆₇₈₉]\*[-]{1}[A-Z]{1}[a-z]?[₀₁₂₃₄₅₆₇₈₉]\*[A-Z]{1}[a-z]?[₀₁₂₃₄₅₆₇₈₉]\*").IsMatch(s))
2254. {
2255. for (int i = 0; i < 3; i++)
2256. {
2257. Regex myReg = new Regex(@"[A-Z]{1}[a-z]?"); *//шаблон элемента*
2258. Match match = myReg.Match(s);
2260. s = s.Replace(match.Value, "");
2261. names[i] = match.Value;
2262. }
2264. *// поставка элемента Х на третье место в массиве*
2265. string z = names[2];
2266. names[2] = names[1];
2267. names[1] = z;
2268. }
2270. return names;
2271. }
2273. private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)
2274. {
2275. if (MessageBox.Show("Вы точно хотите закрыть окно? Все несохраненные данные будут удалены!", "", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No)
2276. e.Cancel = true;
2277. else
2278. DrawingClasses.CollapseGraph.ClearExperiment();
2279. }
2281. private void Window\_SizeChanged(object sender, SizeChangedEventArgs e)
2282. {
2283. MaxWidth = e.NewSize.Height + 92;
2284. Width = e.NewSize.Height + 92;
2285. }
2287. */// <summary>*
2288. */// Построение графиков*
2289. */// </summary>*
2290. private void diag\_Paint(object sender, System.Windows.Forms.PaintEventArgs e)
2291. {
2292. Graphics g = e.Graphics;
2293. g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.HighQuality;
2295. try
2296. {
2297. graph = new DrawingClasses.CollapseGraph(g, sys, diag.Width);
2299. if (f)
2300. graph.DrawCollapse();
2301. else
2302. {
2303. if (sys\_ap != null)
2304. {
2305. *//MessageBox.Show(sys\_ap.Tmax.ToString());*
2306. graph\_ap = new DrawingClasses.CollapseGraph(g, sys\_ap, diag.Width);
2307. graph\_ap.DrawDH();
2308. }
2309. *//else*
2310. *// MessageBox.Show(sys.Tmax.ToString());*
2311. graph.DrawDH(false);
2312. }
2314. graph.DrawAxes();
2315. graph.DrawExperiment();
2316. }
2317. catch (Exception ex)
2318. {
2319. if (ex.InnerException != null && ex.InnerException.Message == "MyException")
2320. MessageBox.Show(ex.Message, "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2321. else
2322. MessageBox.Show("Неверные данные для построения купола! Измените их в таблицах или в меню настроек!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2323. }
2324. }
2326. private void Points\_CellEditEnding(object sender, DataGridCellEditEndingEventArgs e)
2327. {
2328. (e.EditingElement as TextBox).Text = (e.EditingElement as TextBox).Text.Replace(',', '.');
2329. if (dat[e.Row.GetIndex()].Capacity == 0)
2330. {
2331. dat[e.Row.GetIndex()].Add(0);
2332. dat[e.Row.GetIndex()].Add(0);
2333. }
2334. if (!float.TryParse((e.EditingElement as TextBox).Text.Replace('.', ','), out float p) || p < 0)
2335. {
2336. MessageBox.Show("Координаты точки должны быть неотрицательным числом!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2337. (e.EditingElement as TextBox).Text = "";
2339. e.Cancel = true;
2340. }
2341. if (e.Column.DisplayIndex == 0)
2342. dat[e.Row.GetIndex()][0] = p;
2343. else
2344. dat[e.Row.GetIndex()][1] = p;
2345. }
2347. */// <summary>*
2348. */// Построение купола распада*
2349. */// </summary>*
2350. private void Build\_Click(object sender, RoutedEventArgs e)
2351. {
2352. if (!f && sys\_ap != null)
2353. if (MessageBox.Show("Использовать при построении купола новые значения параметров?", "", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)
2354. sys = sys\_ap.Clone();
2356. f = true;
2358. SetColor();
2359. SetBorders();
2361. diag.Refresh();
2362. }
2364. */// <summary>*
2365. */// Запускает оценку чувствительности*
2366. */// </summary>*
2367. private void Sensitivity\_Click(object sender, RoutedEventArgs e)
2368. {
2369. f = false;
2370. sys\_ap = sys.Clone();
2371. Points.Visibility = Visibility.Hidden;
2372. Build.Visibility = Visibility.Hidden;
2373. Approxi.Visibility = Visibility.Hidden;
2374. Save.Visibility = Visibility.Hidden;
2375. Load.Visibility = Visibility.Hidden;
2376. R.Visibility = Visibility.Visible;
2377. dE.Visibility = Visibility.Visible;
2378. c.Visibility = Visibility.Visible;
2379. R\_label.Visibility = Visibility.Visible;
2380. dE\_label.Visibility = Visibility.Visible;
2381. c\_label.Visibility = Visibility.Visible;
2382. R\_text.Visibility = Visibility.Visible;
2383. dE\_text.Visibility = Visibility.Visible;
2384. c\_text.Visibility = Visibility.Visible;
2385. Back.Visibility = Visibility.Visible;
2386. DownR.IsEnabled = true;
2387. UpdE.IsEnabled = true;
2388. Downc.IsEnabled = true;
2389. UpR.IsEnabled = true;
2390. DowndE.IsEnabled = true;
2391. Upc.IsEnabled = true;
2392. IsExpPoints.IsEnabled = false;
2394. R.Value = Math.Min(sys\_ap.R(0), sys\_ap.R(1));
2395. c.Value = sys\_ap.GetData()[0];
2396. dE.Value = sys\_ap.delEps;
2398. SetColor();
2399. SetBorders();
2401. diag.Refresh();
2402. }
2404. */// <summary>*
2405. */// Сохранение экспериментальных точек в файл*
2406. */// </summary>*
2407. private void Save\_Click(object sender, RoutedEventArgs e)
2408. {
2409. Microsoft.Win32.SaveFileDialog dlg = new Microsoft.Win32.SaveFileDialog
2410. {
2411. FileName = "Points",
2412. DefaultExt = ".txt",
2413. Filter = "Text files (.txt)|\*.txt"
2414. };

2417. if (dlg.ShowDialog() == true)
2418. using (FileStream fs = new FileStream(dlg.FileName, FileMode.Create, FileAccess.Write))
2419. using (StreamWriter sw = new StreamWriter(fs))
2420. for (int i = 0; i < dat.Count; i++)
2421. sw.WriteLine(dat[i][0] + " " + dat[i][1]);
2422. }
2424. */// <summary>*
2425. */// Загрузка экспериментальных точек из файла*
2426. */// </summary>*
2427. private void Load\_Click(object sender, RoutedEventArgs e)
2428. {
2429. Microsoft.Win32.OpenFileDialog dlg = new Microsoft.Win32.OpenFileDialog
2430. {
2431. Filter = "Text files (.txt)|\*.txt",
2432. CheckFileExists = true
2433. };
2435. try
2436. {
2437. if (dlg.ShowDialog() == true)
2438. {
2439. dat.Clear();
2440. using (FileStream fs = new FileStream(dlg.FileName, FileMode.Open, FileAccess.Read))
2441. using (StreamReader sr = new StreamReader(fs))
2442. while (!sr.EndOfStream)
2443. {
2444. string s = sr.ReadLine();
2445. dat.Add(new List<double> { double.Parse(s.Split()[0]), double.Parse(s.Split()[1]) });
2446. }
2447. }
2448. }
2449. catch (Exception)
2450. {
2451. MessageBox.Show("Неверный формат файла!");
2452. }
2453. Points.Items.Refresh();
2454. Points\_RowEditEnding(this, new DataGridRowEditEndingEventArgs(new DataGridRow(), DataGridEditAction.Commit));
2455. }
2457. */// <summary>*
2458. */// Открывает окно настройки параметров*
2459. */// </summary>*
2460. private void DataSettings\_Click(object sender, RoutedEventArgs e)
2461. {
2462. DataSettings ds = new DataSettings(sys);
2463. ds.ShowDialog();
2464. sys = ds.GetBS();
2466. diag.Refresh();
2467. }
2469. private void Points\_RowEditEnding(object sender, DataGridRowEditEndingEventArgs e)
2470. {
2471. DrawingClasses.CollapseGraph.ClearExperiment();
2472. for (int i = 0; i < dat.Count; i++)
2473. DrawingClasses.CollapseGraph.AddExperimentalPoint(dat[i][0], dat[i][1]);
2474. diag.Refresh();
2475. }
2477. */// <summary>*
2478. */// Удаляет выделенные строки из таблицы*
2479. */// </summary>*
2480. */// <param name="hotKey">горячая клавиша для вызова метода</param>*
2481. private void DeleteSelectedRows(HotKey hotKey)
2482. {
2483. try
2484. {
2485. while (Points.SelectedItems.Count > 0)
2486. {
2487. int selectedIndex = Points.SelectedIndex;
2488. DrawingClasses.CollapseGraph.RemoveSelectedPoint(selectedIndex);
2489. (Points.ItemsSource as List<List<double>>).RemoveAt(selectedIndex);
2490. Points.Items.Refresh();
2491. }
2492. diag.Refresh();
2493. }
2494. catch (Exception)
2495. {
2496. MessageBox.Show("Невозможно удалить этот элемент!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2497. }
2498. }
2500. */// <summary>*
2501. */// Задает флаг, определяющий формат отображения эксперимента точками*
2502. */// </summary>*
2503. private void MenuItem\_Checked(object sender, RoutedEventArgs e)
2504. {
2505. DrawingClasses.CollapseGraph.ExperimentIsPoints = true;
2506. }
2507. */// <summary>*
2508. */// Задает флаг, определяющий формат отображения эксперимента ломанными*
2509. */// </summary>*
2510. private void MenuItem\_Unchecked(object sender, RoutedEventArgs e)
2511. {
2512. DrawingClasses.CollapseGraph.ExperimentIsPoints = false;
2513. }
2515. */// <summary>*
2516. */// Запускает аппроксимацию функции смешения*
2517. */// </summary>*
2518. private void Approxi\_Click(object sender, RoutedEventArgs e)
2519. {
2520. f = false;
2522. SetColor();
2523. SetBorders();
2525. if (sys != null)
2526. Approximate(new double[] { Math.Min(sys.R(1), sys.R(0)), sys.delEps, sys.GetData()[0] });
2528. diag.Refresh();
2529. }
2531. */// <summary>*
2532. */// Задает границы параметров*
2533. */// </summary>*
2534. private void SetBorders()
2535. {
2536. int t = -1;
2537. if (!int.TryParse(DownT.Text, out t))
2538. MessageBox.Show("Неправильно установленно нижняя граница температуры!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2539. DrawingClasses.CollapseGraph.DownTemp = t;
2541. t = -1;
2542. if (!int.TryParse(UpT.Text, out t))
2543. MessageBox.Show("Неправильно установленно верхняя граница температуры!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2544. DrawingClasses.CollapseGraph.UpTemp = t;
2546. if (!f && sys\_ap != null)
2547. {
2548. if (!double.TryParse(UpR.Text.Replace('.', ','), out double b) && b <= 0)
2549. MessageBox.Show("Неправильно установленно верхняя граница параметра R!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2550. R.Maximum = b;
2552. b = 0.01;
2553. if (!double.TryParse(DownR.Text.Replace('.', ','), out b) && b <= 0)
2554. MessageBox.Show("Неправильно установленно нижняя граница параметра R!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2555. R.Minimum = b;
2557. b = 0.01;
2558. if (!double.TryParse(Upc.Text.Replace('.', ','), out b) && b <= 0)
2559. MessageBox.Show("Неправильно установленно верхняя граница параметра c!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2560. c.Maximum = b;
2562. b = 0.01;
2563. if (!double.TryParse(Downc.Text.Replace('.', ','), out b) && b <= 0)
2564. MessageBox.Show("Неправильно установленно нижняя граница параметра c!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2565. c.Minimum = b;
2567. b = 0.01;
2568. if (!double.TryParse(UpdE.Text.Replace('.', ','), out b) && b <= 0)
2569. MessageBox.Show("Неправильно установленно верхняя граница параметра dE!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2570. dE.Maximum = b;
2572. b = 0.01;
2573. if (!double.TryParse(DowndE.Text.Replace('.', ','), out b) && b <= 0)
2574. MessageBox.Show("Неправильно установленно нижняя граница параметра dE!", "Error", MessageBoxButton.OK, MessageBoxImage.Error);
2575. dE.Minimum = b;
2576. }
2577. }
2579. */// <summary>*
2580. */// Задает цвета отображения графиков*
2581. */// </summary>*
2582. private void SetColor()
2583. {
2584. byte[] bytes = BitConverter.GetBytes(Convert.ToInt64(Experiment.SelectedColor.Value.B \* (Math.Pow(256, 0)) +
2585. Experiment.SelectedColor.Value.G \* (Math.Pow(256, 1)) + Experiment.SelectedColor.Value.R \* (Math.Pow(256, 2))));
2586. DrawingClasses.CollapseGraph.ExperimentColor = Color.FromArgb(255, bytes[2], bytes[1], bytes[0]);
2588. bytes = BitConverter.GetBytes(Convert.ToInt64(Theory.SelectedColor.Value.B \* (Math.Pow(256, 0)) +
2589. Theory.SelectedColor.Value.G \* (Math.Pow(256, 1)) + Theory.SelectedColor.Value.R \* (Math.Pow(256, 2))));
2590. DrawingClasses.CollapseGraph.Color = Color.FromArgb(255, bytes[2], bytes[1], bytes[0]);
2592. bytes = BitConverter.GetBytes(Convert.ToInt64(Approximation.SelectedColor.Value.B \* (Math.Pow(256, 0)) +
2593. Approximation.SelectedColor.Value.G \* (Math.Pow(256, 1)) + Approximation.SelectedColor.Value.R \* (Math.Pow(256, 2))));
2594. DrawingClasses.CollapseGraph.ApproximationColor = Color.FromArgb(255, bytes[2], bytes[1], bytes[0]);
2595. }
2597. private void c\_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
2598. {
2599. double[] dat = sys\_ap.GetData();
2600. sys\_ap.SetData(c.Value, dat[1], dat[2], dat[3]);
2602. SetColor();
2603. SetBorders();
2605. diag.Refresh();
2606. }
2608. private void dE\_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
2609. {
2610. sys\_ap.delEps = dE.Value;
2612. SetColor();
2613. SetBorders();
2615. diag.Refresh();
2616. }
2618. private void R\_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
2619. {
2620. sys\_ap.R\_const = R.Value;
2622. SetColor();
2623. SetBorders();
2625. diag.Refresh();
2626. }
2628. */// <summary>*
2629. */// Возвращает к построению купола распада*
2630. */// </summary>*
2631. private void Back\_Click(object sender, RoutedEventArgs e)
2632. {
2633. f = true;
2634. sys\_ap = null;
2635. Points.Visibility = Visibility.Visible;
2636. Build.Visibility = Visibility.Visible;
2637. Approxi.Visibility = Visibility.Visible;
2638. Save.Visibility = Visibility.Visible;
2639. Load.Visibility = Visibility.Visible;
2640. R.Visibility = Visibility.Hidden;
2641. dE.Visibility = Visibility.Hidden;
2642. c.Visibility = Visibility.Hidden;
2643. R\_label.Visibility = Visibility.Hidden;
2644. dE\_label.Visibility = Visibility.Hidden;
2645. c\_label.Visibility = Visibility.Hidden;
2646. R\_text.Visibility = Visibility.Hidden;
2647. dE\_text.Visibility = Visibility.Hidden;
2648. c\_text.Visibility = Visibility.Hidden;
2649. Back.Visibility = Visibility.Hidden;
2650. DownR.IsEnabled = false;
2651. UpdE.IsEnabled = false;
2652. Downc.IsEnabled = false;
2653. UpR.IsEnabled = false;
2654. DowndE.IsEnabled = false;
2655. Upc.IsEnabled = false;
2656. IsExpPoints.IsEnabled = true;
2658. SetColor();
2659. SetBorders();
2661. diag.Refresh();
2662. }
2664. */// <summary>*
2665. */// Аппроксимация функции*
2666. */// </summary>*
2667. */// <param name="par">набор изменяемых параметров</param>*
2668. private void Approximate(double[] par)
2669. {
2670. List<HelperClasses.Point> Dots = new List<HelperClasses.Point>();
2671. foreach (List<double> point in dat)
2672. Dots.Add(new HelperClasses.Point(point[0], point[1]));
2673. double[] data = sys.GetData();
2675. Func<double, double[], double> Function = new Func<double, double[], double>((double x, double[] PP)
2676. => 1000 \* x \* (1 - x) \* ((332 \* sys.A / PP[0] \* PP[1] \* PP[1] + PP[2] \* data[1] \* data[2] \* data[3] \* sys.zX \*
2677. (Math.Abs(sys.r1 - sys.r2) / PP[0] \* Math.Abs(sys.r1 - sys.r2) / PP[0]))));
2679. try
2680. {
2681. double[] par\_ap = Library.AproxiTab(Dots, Function, par, Criterion.Criterion\_CKO); *//TODO min of max and sko*
2682. MessageBox.Show(String.Format("R\_min = {0:f4}; delta E = {1:f4}; c = {2:f4}", par\_ap[0], par\_ap[1], par\_ap[2]));
2683. sys\_ap = sys.Clone();
2684. sys\_ap.R\_const = par\_ap[0];
2685. sys\_ap.delEps = par\_ap[1];
2686. sys\_ap.SetData(par\_ap[2], data[1], data[2], data[3]);
2687. }
2688. catch (ArgumentNullException)
2689. {
2690. MessageBox.Show("Точки не заданы! Аппроксимация невозможна!", "Warning", MessageBoxButton.OK, MessageBoxImage.Warning);
2691. }
2692. }
2693. }
2694. }
2696. **Класс Help:**
      * **Help.xaml**
2697. <Window x:Class="Activision\_Mendeleyev\_table.Help"
2698. xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
2699. xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
2700. xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
2701. xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
2702. mc:Ignorable="d"
2703. Icon="icon.jpeg"
2704. Title="Help" Height="450" Width="800" WindowStartupLocation="CenterScreen">
2705. <Grid>
2706. <WebBrowser Name="web" />
2707. </Grid>
2708. </Window>
      * **Help.cs**
2709. using System.Windows;
2711. namespace Activision\_Mendeleyev\_table
2712. {
2713. *///<summary>*
2714. */// Логика взаимодействия для Help.xaml*
2715. *///</summary>*
2716. public partial class Help : Window
2717. {
2718. */// <summary>*
2719. */// Конструктор справки, для упрощения изпользования программы*
2720. */// </summary>*
2721. public Help()
2722. {
2723. InitializeComponent();
2724. web.NavigateToString("<!DOCTYPE html>" +
2725. "<html lang = 'en' xmlns = 'http://www.w3.org/1999/xhtml\'>" +
2726. "<head>" +
2727. "<meta http-equiv='Content-Type' content='text/html;charset=UTF-8'>" +
2728. "<title>Справка</title>" +
2729. "</head>" +
2730. "<body>" +
2731. "<h1>Справка</h1>" +
2732. "<ol style = 'list-style-type: upper-roman;'>" +
2733. "<li><a href = '#Работа с элементом'> Работа с элементом </a>" +
2734. "<ol>" +
2735. "<li> Заполнение таблицы </li>" +
2736. "</ol>" +
2737. "</li>" +
2738. "<li><a href = '#Работа с соединением'> Работа с соединением </a>" +
2739. "<ol>" +
2740. "<li> Выбор(создание) соединения </li>" +
2741. "<li> Заполнение таблицы </li>" +
2742. "</ol>" +
2743. "</li>" +
2744. "<li><a href = '#Работа с системой соединений'> Работа с системой соединений </a>" +
2745. "<ol>" +
2746. "<li> Выбор(создание) системы соединений </li>" +
2747. "<li> Заполнение таблицы </li>" +
2748. "</ol>" +
2749. "</li>" +
2750. "<li><a href = '#Формат формул'> Формат формул </a></li>" +
2751. "<li><a href = '#Описание функций'> Описание функций </a></li>" +
2752. "<li><a href = '#Работа с фазовыми диаграммами'> Работа с фазовыми диаграммами </a></li>" +
2753. "</ol>" +
2754. "<h3 style = 'text-align: center;'><a name = 'Работа с элементом'> Работа с элементом </a></h3>" +
2755. "<p align = 'center'><b> Заполнение таблицы </b></p>" +
2756. "<ol>" +
2757. "<li> Для работы с элементом дважды кликните левой кнопкой мыши по соответствующему обозначению в Таблице Менделеева на стартовом окне;</li>" +
2758. "<li> Для начала изменения таблицы необходимо нажать кнопку &quot;Редактировать таблицу&quot; ;</li>" +
2759. "<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>" +
2760. "<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>" +
2761. "<li> Закрыть окно с таблицей можно нажав кнопку &quot;х&quot;.</li>" +
2762. "</ol>" +
2763. "<h3 style = 'text-align: center;'><a name = 'Работа с соединением'> Работа с соединением</a></h3>" +
2764. "<p align = 'center'><b> Выбор(создание) соединения </b></p>" +
2765. "<ol>" +
2766. "<li> Для работы с соединением нажмите кнопку &quot;Работа с соединением&quot; в верхнем меню стартового окна;</li>" +
2767. "<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>" +
2768. "<li> Для создания нового соединения выпишите его в текстовое поле и нажмите кнопку &quot;Готово&quot;.</li>" +
2769. "</ol>" +
2770. "<p align = 'center'><b> Заполнение таблицы </b></p>" +
2771. "<ol>" +
2772. "<li> Для начала изменения таблицы необходимо нажать кнопку &quot;Редактировать таблицу&quot; ;</li>" +
2773. "<li> Для добавления(удаления) строки и столбца необходимо нажать, соответственно, кнопки &quot;Добавить строку&quot; (&quot;Удалить выделенные строки&quot;) и &quot;Добавить столбец&quot; (&quot;Удалить столбец&quot;);</li>" +
2774. "<li> Для добавления формулы нужно нажать кнопку &quot;Добавить формулу&quot; ;</li>" +
2775. "<li> Чтобы сохранить изменения, которые вы произвели в таблице, и пересчитать формулы, необходимо нажать кнопку &quot;Рассчитать и сохранить&quot;;</li>" +
2776. "<li> Закрыть окно с таблицей можно нажав кнопку &quot;х&quot;.</li>" +
2777. "</ol>" +
2778. "<h3 style = 'text-align: center;'><a name = 'Работа с системой соединений'> Работа с системой соединений </a></h3>" +
2779. "<p align = 'center'><b> Выбор(создание) соединения </b></p>" +
2780. "<ol>" +
2781. "<li> Для работы с системой соединений нажмите кнопку &quot;Работа с системой соединений&quot; в верхнем меню стартового окна;</li>" +
2782. "<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>" +
2783. "<li> Для создания новой системы соединения выпишите её в текстовое поле и нажмите кнопку &quot;Готово&quot;.</li>" +
2784. "</ol>" +
2785. "<p align = 'center'><b> Заполнение таблицы </b></p>" +
2786. "<ol>" +
2787. "<li> Для начала изменения таблицы необходимо нажать кнопку &quot;Редактировать таблицу&quot; ;</li>" +
2788. "<li> Для добавления(удаления) строки и столбца необходимо нажать, соответственно, кнопки &quot;Добавить строку&quot; (&quot;Удалить выделенные строки&quot;) и&quot;Добавить столбец&quot; (&quot;Удалить столбец&quot;);</li>" +
2789. "<li> Для добавления формулы нужно нажать кнопку &quot;Добавить формулу&quot; ;</li>" +
2790. "<li> Столбец &quot;Х&quot; нельзя удалить, и можно заполнять только вещественными числами, в виде десятичных дробей;</li>" +
2791. "<li> Чтобы сохранить изменения, которые вы произвели в таблице, и пересчитать формулы, необходимо нажать кнопку &quot;Рассчитать и сохранить&quot;;</li>" +
2792. "<li> Закрыть окно с таблицей можно нажав кнопку &quot;х&quot;.</li>" +
2793. "</ol>" +
2794. "<h3 style = 'text-align: center;'><a name = 'Формат формул'> Формат формул </a></h3>" +
2795. "<p style='margin-left: 20px'>В формулу могут входить:</p>" +
2796. "<ol style = 'list-style-type: lower-alpha;'>" +
2797. "<li> Вещественные числа, в виде десятичных дробей;</li>" +
2798. "<li> Конструкции, по которым ищется значение в таблицах:" +
2799. "<ol style = 'list-style-type: disc;'>" +
2800. "<li> <обозначение искомого свойства>{<элемент/соединение/система>} – получает значение искомого свойства элемента/соединения/системы соединений находящегося в первой строке таблицы свойств;</li>" +
2801. "<li> <обозначение искомого свойства>{<вспомогательное свойство, по которому ищем искомое>{<значение вспомогательного свойства>}<элемент/соединение/система>} - получает значение искомого свойства элемента/соединения/системы соединений соответствующего значению вспомогательного свойства из таблицы свойств);</li>" +
2802. "<li> <элемент/соединение/система>{<номер строки>;<номер столбца>} – получает значение ячейки таблицы свойств элемента/соединения/системы соединений на пересечении заданной строки и столбца(нумерация строк и столбцов начинается с нуля);</li>" +
2803. "<li> <элемент/соединение/система>{<номер строки>;<обозначение искомого свойства>} – получает значение искомого свойства элемента/соединения/системы соединений на заданной строке(нумерация строк начинается с нуля);</li>" +
2804. "</ol>" +
2805. "<li> Символы математических операций(&quot;+&quot;, &quot;-&quot;, &quot;\*&quot;, &quot;/&quot;, &quot;%&quot; - остаток от деления);</li>" +
2806. "<li> Скобки, обозначающие приоритет математических действий(&quot;(&quot; и &quot;)&quot;);</li>" +
2807. "<li> Функции, описанные ниже(если аргументов большо одного - следует их разделять символом &quot;;&quot;. Пример: min(32; sin(4)).</li>" +
2808. "<p>Необходимо наличие элементов/соединений/систем соединений и свойств им принадлежащих, которые используются в формуле, в таблицах.</p>" +
2809. "</ol>" +
2810. "<h3 style = 'text-align: center;'><a name = 'Описание функций'> Описание функций </a></h3>" +
2811. "<ol>" +
2812. "<li> sin(a) - вычисляет синус а;</li>" +
2813. "<li> cos(a) - вычисляет косинус а;</li>" +
2814. "<li> tan(a) - вычисляет тангенс а;</li>" +
2815. "<li> ctan(a) - вычисляет котангенс а;</li>" +
2816. "<li> abs(a) - вычисляет модуль а;</li>" +
2817. "<li> ln(a) - вычисляет натуральный логарифм а;</li>" +
2818. "<li> exp(a) - вычисляет экспоненту степени а;</li>" +
2819. "<li> min(a; b) - находит минимум из a и b;</li>" +
2820. "<li> max(a; b) - находит максимум из a и b;</li>" +
2821. "<li> pow(a; b) - возводит a в степень b.</li>" +
2822. "</ol>" +
2823. "<h3 style = 'text-align: center;'><a name = 'Работа с фазовыми диаграммами'> Работа с фазовыми диаграммами </a></h3>" +
2824. "<p align = 'center'><b> Купол распада </b></p>" +
2825. "<ol>" +
2826. "<li> Для начала построения купола распада необходимо нажать кнопку &quot;Купол распада&quot; в окне редактирования таблицы соответствующей системы соединений;</li>" +
2827. "<li> Далее в открывшемся окне необходимо задать некоторые параметры системы необходимые для построения купола(это же окно можно вызвать нажав на пункт меню &quotНастройка данных&quot);</li>" +
2828. "<li> Для построения купола распада необходимо нажать кнопку &quot;Построить купол&quot;;</li>" +
2829. "<li> Для настройки предельных значений температуры(оси Y) необходимо выбрать пункт меню &quot;Графическое меню&quot;->&quot;Настройка границ&quot; и ввести значения верхних и нижних границ температуры;</li>" +
2830. "<li> Добавление экспирементальных данных происходит путем ввода точек в таблицу в правой части окна или загруки из файли при нажатии на кнопку&quot;Загрузить точки&quot;(чтобы сохранить точки из таблицы в файл необходимо нажать кнопку &quot;Сохранить точки&quot;);</li>" +
2831. "<li> Удалить точки можно выделив их в таблице и нажав после клавишу DELETE.</li>" +
2832. "<li> Для изменения формата отображения экспирементальных данных необходимо поставить/снять отметку пункта меню &quot;Графическое меню&quot;->&quot;Эксперимент точками&quot;;</li>" +
2833. "<li> Также в пункте меню &quot;Графическое меню&quot;->&quot;Настройка цвета&quot; можно выбрать цвет каждого графика.</li>" +
2834. "</ol>" +
2835. "<p align = 'center'><b> Аппроксимация функции </b></p>" +
2836. "<ol>" +
2837. "<li> Для аппроксимации термодинамической функции смешения необходимо нажать кнопку &quot;Построить функцию Hsm и аппроксимировать&quot; в окне построения купола распада;</li>" +
2838. "<li> Если не были заданы точки, то аппроксимация не будет проведена;</li>" +
2839. "<li> Для настройки предельных значений температуры(оси Y) необходимо выбрать пункт меню &quot;Графическое меню&quot;->&quot;Настройка границ&quot; и ввести значения верхних и нижних границ температуры;</li>" +
2840. "<li> После аппроксимации, если нажать на кнопку &quot;Построить купол&quot;, будет предложено использовать новые значения параметров для его построения.</li>" +
2841. "</ol>" +
2842. "<p align = 'center'><b> Оценка чувствительности </b></p>" +
2843. "<ol>" +
2844. "<li> Для начала работы с оценкой чувствительности термодинамической функции смешения необходимо нажать пункт меню &quot;Оценка чувствительности&quot; в окне построения купола распада;</li>" +
2845. "<li> При изменении слайдеров в правой половине окна будет меняться функция в зависимоти от выбранных значений(текущие значения отображаются над слайдерами);</li>" +
2846. "<li> Для настройки предельных значений слайдеров необходимо выбрать пункт меню &quot;Графическое меню&quot;->&quot;Настройка границ&quot; и ввести значения верхних и нижних границ параметров;</li>" +
2847. "<li> Для восстановления начальных значений параметров необходимо снова нажать пункт меню &quot;Оценка чувствительности&quot;.</li>" +
2848. "</ol>" +
2849. "</body>" +
2850. "</html>");
2851. }
2852. }
2853. }
2855. **Классы Approximation:**
      * **Criterion.cs**
2856. using Activision\_Mendeleyev\_table.HelperClasses;
2857. using System;
2858. using System.Collections.Generic;
2860. namespace Activision\_Mendeleyev\_table.Approximation
2861. {
2862. */// <summary>*
2863. */// Класс, содержащий различные критерии оценки отклонения функции*
2864. */// </summary>*
2865. public static class Criterion
2866. {
2867. */// <summary>*
2868. */// Критерий max|f-y| оценки отклонения F(x) от точек*
2869. */// </summary>*
2870. */// <param name="tab">лист точек</param>*
2871. */// <param name="F">функция</param>*
2872. */// <param name="par">начальное значение параметров функции</param>*
2873. */// <returns>значение отклонения</returns>*
2874. public static double Criterion\_MAX(List<Point> tab, Func<double, double[], double> F, double[] par)
2875. {
2876. double max = 0, f, s = 0;
2877. foreach (Point mp in tab)
2878. {
2879. f = F(mp.X, par);
2880. s = Math.Abs(f - mp.Y);
2881. max = max > s ? max : s;
2882. }
2884. return max;
2885. }
2887. */// <summary>*
2888. */// Критерий (sum|f-y|^2)/N оценки отклонения F(x) от точек*
2889. */// </summary>*
2890. */// <param name="tab">лист точек</param>*
2891. */// <param name="F">функция</param>*
2892. */// <param name="par">начальное значение параметров функции</param>*
2893. */// <returns>значение отклонения</returns>*
2894. public static double Criterion\_CKO(List<Point> tab, Func<double, double[], double> F, double[] par)
2895. {
2896. double f, sum = 0;
2897. foreach (Point mp in tab)
2898. {
2899. f = F(mp.X, par);
2900. sum += Math.Pow((f - mp.Y), 2);
2901. }
2903. return Math.Sqrt(sum / tab.Count);
2904. }
2905. }
2906. }
2907. * **Library.cs**
2908. using Activision\_Mendeleyev\_table.HelperClasses;
2909. using System;
2910. using System.Collections.Generic;
2912. namespace Activision\_Mendeleyev\_table.Approximation
2913. {
2914. */// <summary>*
2915. */// Класс, содержащий методы аппроксимации функции*
2916. */// </summary>*
2917. public class Library
2918. {
2919. */// <summary>*
2920. */// Метод аппроксимации набора точек заданной функцией (Function)*
2921. */// </summary>*
2922. */// <param name="tab">лист точек</param>*
2923. */// <param name="Function">аппроксимирующая функция</param>*
2924. */// <param name="Par">начальное значение параметров функции</param>*
2925. */// <param name="ApproxiAccuracy">метод оценки точности аппроксимации</param>*
2926. */// <returns>новый нобор параметров функции</returns>*
2927. public static double[] AproxiTab(List<Point> tab, Func<double, double[], double> Function, double[] Par,
2928. Func<List<Point>, Func<double, double[], double>, double[], double> ApproxiAccuracy)
2929. {
2930. *// Локальный метод*
2931. double funN(double[] par)
2932. {
2933. if (tab.Count == 0)
2934. throw new ArgumentNullException("", new Exception("MyException"));
2935. double result = ApproxiAccuracy(tab, Function, par);
2936. return result;
2937. }
2939. double[] res = GradientMinimization(funN, Par, 1E-8, 1E-11, 10000);
2941. return res;
2942. }
2944. */// <summary>*
2945. */// Вычисление градиента и направляющих вектора перемещения*
2946. */// </summary>*
2947. */// <param name="funN">исследуемая функция</param>*
2948. */// <param name="X0">вектор параметров - иследуемая точка</param>*
2949. */// <param name="del">относительная ??вариация??? каждого параметра</param>*
2950. */// <returns>вектор перемещения вдоль градиента</returns>*
2951. public static double[] Gradient(Func<double[], double> funN, double[] X0, double del = 0.001)
2952. *//double del=0.00001)*
2953. *//double del = 0.01)*
2954. {
2955. int NC = X0.Length;
2956. double[] G = new double[NC];
2957. double[] dx = new double[NC];
2958. for (int j = 0; j < NC; j++)
2959. if (X0[j] == 0)
2960. dx[j] = del;
2961. else
2962. dx[j] = Math.Abs(X0[j] \* del);
2964. double[] V = new double[NC];
2965. X0.CopyTo(V, 0);
2966. double Fma, Fmi, dFi;
2967. for (int j = 0; j < NC; j++)
2968. {
2969. V[j] = X0[j] + dx[j];
2970. Fma = funN(V);
2971. V[j] = X0[j] - dx[j];
2972. Fmi = funN(V);
2973. dFi = Fma - Fmi;
2974. G[j] = dFi / (2 \* dx[j]);
2975. V[j] = X0[j];
2976. }
2978. *// unit vector along gradient:*
2979. double[] S = new double[NC];
2980. double len = 0;
2981. for (int j = 0; j < NC; j++)
2982. len += Math.Pow(G[j], 2);
2984. len = Math.Sqrt(len);
2985. for (int j = 0; j < NC; j++)
2986. S[j] = -G[j] / len;
2988. return S;
2989. }
2991. */// <summary>*
2992. */// Вычисление минимума вдоль градиента*
2993. */// </summary>*
2994. */// <param name="funN">исследуемая функция</param>*
2995. */// <param name="X0">начальный вектор параметров</param>*
2996. */// <param name="Delta">допустимое отклонение для ширины интервала</param>*
2997. */// <param name="Epsilon">допустимое отклонение для |f(b) - f(a)|</param>*
2998. */// <returns>параметры (точка) минимума вдоль градиента</returns>*
2999. private static double[] Quadmin(Func<double[], double> funN, double[] X0, double Delta = 1E-5, double Epsilon = 1E-7)
3000. {
3001. int NC = X0.Length;
3002. double Y0 = funN(X0);
3003. double[] P0 = new double[NC];
3004. X0.CopyTo(P0, 0);
3005. double[] P1 = new double[NC];
3006. double[] P2 = new double[NC];
3007. double[] S = new double[NC];
3008. double H = 1.0;
3009. double Err = 1.0;
3010. int Jmax = 20;
3011. double H0, H1, H2, Hmin, E0, E1, E2, Y1, Y2, D, Ymin;
3012. int i;
3013. int Cond = 0;
3014. int J = 0;
3016. S = Gradient(funN, X0);
3017. for (i = 0; i < NC; i++)
3018. {
3019. P1[i] = P0[i] + H \* S[i];
3020. P2[i] = P0[i] + 2.0 \* H \* S[i];
3021. }
3023. Y1 = funN(P1);
3024. Y2 = funN(P2);
3025. double[] Pmin = new double[NC];
3026. while ((J < Jmax) & (Cond == 0))
3027. {
3028. if (Y0 < Y1) */\* Make H smaller \*/*
3029. {
3030. Y2 = Y1;
3031. H = H / 2.0;
3032. for (i = 0; i < NC; i++)
3033. {
3034. P2[i] = P1[i];
3035. P1[i] = P0[i] + H \* S[i];
3036. }
3037. Y1 = funN(P1);
3038. }
3039. else
3040. {
3041. if (Y2 < Y1) */\* Make H larger \*/*
3042. {
3043. Y1 = Y2;
3044. H = 2.0 \* H;
3045. for (i = 0; i < NC; i++)
3046. {
3047. P1[i] = P2[i];
3048. P2[i] = P0[i] + 2.0 \* H \* S[i];
3049. }
3050. Y2 = funN(P2);
3051. }
3052. else Cond = -1;
3053. }
3054. }
3056. if (H < Delta) Cond = 1;
3057. D = 4.0 \* Y1 - 2.0 \* Y0 - 2.0 \* Y2;
3058. */\* Quadratic interpolation to find Hmin \*/*
3059. if (D < 0)
3060. Hmin = H \* (4.0 \* Y1 - 3.0 \* Y0 - Y2) / D;
3061. else */\* check division by zero \*/*
3062. {
3063. Cond = 4;
3064. Hmin = H / 3.0;
3065. }
3067. for (i = 0; i < NC; i++)
3068. Pmin[i] = P0[i] + Hmin \* S[i];
3070. Ymin = funN(Pmin);
3072. */\* Convergence test for the points \*/*
3073. H0 = Math.Abs(Hmin);
3074. H1 = Math.Abs(Hmin - H);
3075. H2 = Math.Abs(Hmin - 2.0 \* H);
3076. if (H0 < H) H = H0;
3077. if (H1 < H) H = H1;
3078. if (H2 < H) H = H2;
3079. if (H < Delta) Cond = 1;
3081. */\* Convergence test for the function values \*/*
3082. E0 = Math.Abs(Y0 - Ymin);
3083. E1 = Math.Abs(Y1 - Ymin);
3084. E2 = Math.Abs(Y2 - Ymin);
3085. if (E0 < Err)
3086. Err = E0;
3087. else if (E1 < Err)
3088. Err = E1;
3089. else if (E2 < Err)
3090. Err = E2;
3091. else if ((E0 == 0) && (E1 == 0) && (E2 == 0))
3092. Err = 0;
3094. if (Err < Epsilon)
3095. Cond = 2;
3096. if ((Cond == 2) && (H < Delta))
3097. Cond = 3;
3099. J++;
3100. return Pmin;
3101. }
3103. */// <summary>*
3104. */// Метод наискорейшего спуска (метод градиентной минимизации)*
3105. */// </summary>*
3106. */// <param name="funN">исследуемая функция</param>*
3107. */// <param name="X0">начальный вектор параметров</param>*
3108. */// <param name="Delta">допустимое отклонение для ширины интервала</param>*
3109. */// <param name="Epsilon">допустимое отклонение для |f(b) - f(a)|</param>*
3110. */// <param name="Max">максимальное число итераций</param>*
3111. */// <returns>оптимальный вектор параметров</returns>*
3112. public static double[] GradientMinimization(Func<double[], double> funN, double[] X0, double Delta = 1E-8, double Epsilon = 1E-11, int Max = 100)
3113. {
3114. int NC = X0.Length;
3115. double[] Q1 = new double[NC];
3116. double[] Q2 = new double[NC];
3117. double F1, F2;
3118. int iter = 0;
3119. double deltaX = 0;
3120. X0.CopyTo(Q2, 0);
3122. do
3123. {
3124. iter++;
3125. Q2.CopyTo(Q1, 0);
3126. F1 = funN(Q1);
3127. Q2 = Quadmin(funN, Q1);
3128. F2 = funN(Q2);
3129. deltaX = 0;
3130. for (int k = 0; k < NC; k++)
3131. {
3132. double ZN = Math.Abs(Q2[k]);
3133. if (ZN > 0.0)
3134. {
3135. double DR = Math.Abs((Q2[k] - Q1[k]) / ZN);
3136. deltaX = DR > deltaX ? DR : deltaX;
3137. }
3138. }
3139. }
3140. while (iter < Max & Math.Abs(F1 - F2) > Epsilon & deltaX > Delta);
3141. double deltaF = (F1 - F2);
3142. return Q2;
3143. }
3144. }
3145. }
3147. **Классы DrawingClasses:**
      * **Collapse.cs**
3148. using Activision\_Mendeleyev\_table.HelperClasses;
3149. using System;
3150. using System.Linq;
3151. using System.Xml.Linq;
3153. namespace Activision\_Mendeleyev\_table.DrawingClasses
3154. {
3155. */// <summary>*
3156. */// Класс, содержащий набор точек фазовой диаграммы и методы их получения*
3157. */// </summary>*
3158. class Collapse
3159. {
3160. */// <summary>*
3161. */// Набор точек правого соединения*
3162. */// </summary>*
3163. public Point[] right;
3165. */// <summary>*
3166. */// Набор точек левого соединения*
3167. */// </summary>*
3168. public Point[] left;
3170. */// <summary>*
3171. */// Получает точки для фазовой диаграммы*
3172. */// </summary>*
3173. */// <param name="system">система соединений</param>*
3174. public Collapse(BinSystem system)
3175. {
3176. string r = GetRatio(system.delR / Math.Min(system.R1, system.R2));
3178. System.Windows.Resources.StreamResourceInfo ri = System.Windows.Application.GetResourceStream(new Uri("DrawingClasses/Collapse.xml", UriKind.Relative));
3179. System.IO.Stream data = ri.Stream;
3181. XDocument doc = XDocument.Load(data);
3182. string[] x1values = doc.Root.Elements().First(
3183. p => p.Attribute("ratio").Value == r).Element("x1").Value.Split(';');
3184. string[] x2values = doc.Root.Elements().First(
3185. p => p.Attribute("ratio").Value == r).Element("x2").Value.Split(';');
3186. string[] y1values = doc.Root.Elements().First(
3187. p => p.Attribute("ratio").Value == r).Element("y1").Value.Split(';');
3188. string[] y2values = doc.Root.Elements().First(
3189. p => p.Attribute("ratio").Value == r).Element("y2").Value.Split(';');
3191. right = new Point[x1values.Length];
3192. left = new Point[x2values.Length];
3194. for (int i = 0; i < x1values.Length; i++)
3195. right[i] = new Point(double.Parse(x1values[i]), double.Parse(y1values[i]));
3197. for (int i = 0; i < x2values.Length; i++)
3198. left[i] = new Point(double.Parse(x2values[i]), double.Parse(y2values[i]));
3199. }
3201. */// <summary>*
3202. */// Получает соотношение радиусов*
3203. */// </summary>*
3204. */// <param name="ratio">delR/Rmin</param>*
3205. */// <returns>соотношение радиусов</returns>*
3206. private string GetRatio(double ratio)
3207. {
3208. ratio = Math.Round(ratio, 3);
3209. if ((ratio >= 0) && (ratio < 0.025)) return "0,00";
3210. else if ((ratio <= 0.05) || ((ratio > 0.05) && (ratio < 0.075))) return "0,05";
3211. else if ((ratio <= 0.1) || ((ratio > 0.1) && (ratio < 0.125))) return "0,10";
3212. else if ((ratio <= 0.15) || ((ratio > 0.15) && (ratio < 0.175))) return "0,15";
3213. else if ((ratio <= 0.20) || ((ratio > 0.20) && (ratio < 0.225))) return "0,20";
3214. else if ((ratio <= 0.25) || ((ratio > 0.25) && (ratio < 0.275))) return "0,25";
3215. else if ((ratio <= 0.30) || ((ratio > 0.30) && (ratio < 0.325))) return "0,30";
3216. else throw new Exception("Недопустимое отношение радиусов!", new Exception("MyException"));
3217. }
3218. }
3219. }
3220. * **CollapseGraph.cs**
3221. using Activision\_Mendeleyev\_table.HelperClasses;
3222. using System;
3223. using System.Collections.Generic;
3224. using System.Drawing;
3225. using Point = System.Drawing.Point;
3227. namespace Activision\_Mendeleyev\_table.DrawingClasses
3228. {
3229. */// <summary>*
3230. */// Класс, отрисовывающий фазовую диаграмму*
3231. */// </summary>*
3232. class CollapseGraph
3233. {
3234. */// <summary>*
3235. */// Карандаш теоритического соотношения*
3236. */// </summary>*
3237. private static Pen pen = Pens.Black;
3238. */// <summary>*
3239. */// Карандаш экспериментального соотношения*
3240. */// </summary>*
3241. private static Pen penExp = Pens.Red;
3242. */// <summary>*
3243. */// Карандаш аппроксримированного соотношения*
3244. */// </summary>*
3245. private static Pen penApp = Pens.Green;
3246. */// <summary>*
3247. */// Точки эксперимента*
3248. */// </summary>*
3249. private static List<PointF> experiment = new List<PointF>();
3250. */// <summary>*
3251. */// Порверхность для рисования*
3252. */// </summary>*
3253. private Graphics g;
3254. */// <summary>*
3255. */// Система соединений*
3256. */// </summary>*
3257. private BinSystem system;
3258. */// <summary>*
3259. */// Ширина поля для диаграммы*
3260. */// </summary>*
3261. private readonly int width;
3262. */// <summary>*
3263. */// Точки правого соединения(теория)*
3264. */// </summary>*
3265. private Point[] right;
3266. */// <summary>*
3267. */// Точки левого соединения(теория)*
3268. */// </summary>*
3269. private Point[] left;
3271. */// <summary>*
3272. */// Конструктор*
3273. */// </summary>*
3274. */// <param name="g">поверхность для рисования</param>*
3275. */// <param name="system">система соединений</param>*
3276. */// <param name="width">ширина поля для диаграммы</param>*
3277. public CollapseGraph(Graphics g, BinSystem system, int width)
3278. {
3279. this.width = width - 80;
3280. this.g = g;
3281. this.system = system;
3282. }
3284. */// <summary>*
3285. */// Свойство цвета теоритического соотношения*
3286. */// </summary>*
3287. public static Color Color
3288. {
3289. get { return pen.Color; }
3290. set { pen = new Pen(value); }
3291. }
3293. */// <summary>*
3294. */// Свойство цвета экспериментального соотношения*
3295. */// </summary>*
3296. public static Color ExperimentColor
3297. {
3298. get { return penExp.Color; }
3299. set { penExp = new Pen(value); }
3300. }
3302. */// <summary>*
3303. */// Свойство цвета аппроксимированного соотношения*
3304. */// </summary>*
3305. public static Color ApproximationColor
3306. {
3307. get { return penApp.Color; }
3308. set { penApp = new Pen(value); }
3309. }
3311. */// <summary>*
3312. */// Рисует купол распада*
3313. */// </summary>*
3314. public void DrawCollapse()
3315. {
3316. Collapse collapse = new Collapse(system);
3317. UpTemp = UpTemp == -1 ? (int)system.Tmax : UpTemp;
3318. DownTemp = DownTemp == -1 ? (int)(0.20 \* system.Tmax) : DownTemp;
3320. right = new Point[collapse.right.Length];
3321. for (int i = 0; i < right.Length; i++)
3322. {
3323. int x = 80 + (int)(width \* (1 - collapse.right[i].X));
3324. int y = width - 40 - (int)(width \* ((collapse.right[i].Y \* system.Tmax - DownTemp) / (UpTemp - DownTemp)));
3325. y = y > width ? width : y;
3327. right[i] = new Point(x, y);
3328. }
3330. left = new Point[collapse.left.Length];
3331. for (int i = 0; i < left.Length; i++)
3332. {
3333. int x = 80 + (int)(width \* collapse.left[i].X);
3334. int y = width - 40 - (int)(width \* ((collapse.left[i].Y \* system.Tmax - DownTemp) / (UpTemp - DownTemp)));
3335. y = y > width ? width : y;
3337. left[i] = new Point(x, y);
3338. }
3340. g.DrawString("T, °C", new Font("X", 14), Brushes.Black, new Point(80, 0));
3341. g.DrawLines(pen, right);
3342. g.DrawLines(pen, left);
3343. }
3345. */// <summary>*
3346. */// Рисует термодинамическую функцию смешения*
3347. */// </summary>*
3348. */// <param name="f">флаг: true - аппроксимация, false - теория</param>*
3349. public void DrawDH(bool f = true)
3350. {
3351. DownTemp = DownTemp == -1 ? 0 : DownTemp;
3352. UpTemp = UpTemp == -1 ? (int)(system.Hsm(0.5) \* 1000) + 100 : UpTemp;
3354. Point[] dh = new Point[21];
3355. for (double i = 0; i < 1; i += 0.05)
3356. dh[(int)Math.Round(i \* 20)] = new Point(80 + (int)(i \* width), width - 40 - (int)((system.Hsm(i) \* 1000 - DownTemp) / (UpTemp - DownTemp) \* width));
3358. dh[20] = new Point(80 + width, width - 40);
3359. g.DrawString("ΔHcм, ккал/моль", new Font("X", 14), Brushes.Black, new Point(80, 0));
3360. if (f)
3361. g.DrawLines(penApp, dh);
3362. else
3363. g.DrawLines(pen, dh);
3364. }
3366. */// <summary>*
3367. */// Рисует оси координат*
3368. */// </summary>*
3369. public void DrawAxes()
3370. {
3371. g.DrawString(system.ElementA + system.ElementX, new Font("X", 14), Brushes.Black, new Point(80, width + 30));
3372. g.DrawString(system.ElementB + system.ElementX, new Font("X", 14), Brushes.Black, new Point(width - 100, width + 30));
3374. g.DrawLine(Pens.Black, 80, 0, 80, width + 30);
3375. g.DrawLine(Pens.Black, 30, width - 40, width + 80, width - 40);
3377. for (double x = 0; x <= 1; x += 0.1)
3378. {
3379. g.DrawString(x <= 0.5 ? x.ToString() : (1 - x).ToString(), new Font("X", 12), Brushes.Black,
3380. 80 + (float)(width \* x), width - 30);
3381. g.DrawLine(Pens.Black, 80 + (int)(width \* x), width - 45, 80 + (int)(width \* x), width - 35);
3382. }
3383. g.DrawString("0", new Font("X", 12), Brushes.Black, width + 30, width - 30);
3385. double c = Math.Round((UpTemp - DownTemp) / 100.0);
3386. if (c < (UpTemp - DownTemp) / 100)
3387. c = (c + 1) \* 10;
3388. else
3389. c \*= 10;
3391. for (double x = DownTemp; x <= UpTemp; x += c)
3392. {
3393. g.DrawString(x.ToString(), new Font("X", 12), Brushes.Black, 0, width - 40 - (int)((x - DownTemp) / (UpTemp - DownTemp) \* width));
3394. g.DrawLine(Pens.Black, 75, width - 40 - (int)((x - DownTemp) / (UpTemp - DownTemp) \* width), 85, width - 40 - (int)((x - DownTemp) / (UpTemp - DownTemp) \* width));
3395. }
3396. }
3398. */// <summary>*
3399. */// Рисует эксперимент*
3400. */// </summary>*
3401. public void DrawExperiment()
3402. {
3403. List<Point> left = new List<Point>();
3404. List<Point> right = new List<Point>();
3406. foreach (var item in experiment)
3407. {
3409. int x = 80 + (int)(item.X \* width);
3410. int y = width - 40 - (int)((item.Y - DownTemp) / (UpTemp - DownTemp) \* width);
3412. y = y > width ? width : y;
3414. if (item.X <= 0.5)
3415. left.Add(new Point(x, y));
3416. else
3417. right.Add(new Point(x, y));
3418. }
3420. if (!ExperimentIsPoints)
3421. {
3422. Point[] arrLeft = left.ToArray();
3423. Point[] arrRight = right.ToArray();
3425. Array.Sort(arrLeft, (x, y) => x.X.CompareTo(y.X));
3426. Array.Sort(arrRight, (x, y) => x.X.CompareTo(y.X));
3428. if (left.Count > 1)
3429. g.DrawLines(penExp, arrLeft);
3431. if (right.Count > 1)
3432. g.DrawLines(penExp, arrRight);
3433. }
3434. else
3435. {
3436. foreach (var item in left)
3437. {
3438. g.FillEllipse(penExp.Brush, item.X - 6, item.Y - 6, 12, 12);
3439. }
3440. foreach (var item in right)
3441. {
3442. g.FillEllipse(penExp.Brush, item.X - 6, item.Y - 6, 12, 12);
3443. }
3444. }
3445. }
3447. */// <summary>*
3448. */// Свойство, определяющее вид отрисовки эксперимента*
3449. */// </summary>*
3450. public static bool ExperimentIsPoints { get; set; } = true;
3452. */// <summary>*
3453. */// Свойство верхней границы температуры(графика по Y)*
3454. */// </summary>*
3455. public static int UpTemp { get; set; } = -1;
3457. */// <summary>*
3458. */// Свойство нижней границы температуры(графика по Y)*
3459. */// </summary>*
3460. public static int DownTemp { get; set; } = -1;
3462. */// <summary>*
3463. */// Добавляет точку в эксперимент*
3464. */// </summary>*
3465. */// <param name="x1">координата X</param>*
3466. */// <param name="t">координата Y</param>*
3467. public static void AddExperimentalPoint(double x1, double t)
3468. {
3469. experiment.Add(new PointF((float)x1, (float)t));
3470. }
3472. */// <summary>*
3473. */// Удаляет выбранную точку из эксперимента*
3474. */// </summary>*
3475. public static void RemoveSelectedPoint(int i)
3476. {
3477. if (experiment.Count > i)
3478. experiment.RemoveAt(i);
3479. }
3481. */// <summary>*
3482. */// Удаляет все точки из эксперимента*
3483. */// </summary>*
3484. public static void ClearExperiment()
3485. {
3486. experiment.Clear();
3487. }
3488. }
3489. }
3491. **Классы HelperClasses:**
      * **BinSystem.cs**
3492. using System;
3494. namespace Activision\_Mendeleyev\_table.HelperClasses
3495. {
3496. */// <summary>*
3497. */// Класс, представляющий бинарную систему соединений*
3498. */// </summary>*
3499. public class BinSystem
3500. {
3501. */// <summary>*
3502. */// Массив обозначений в таблицах данных*
3503. */// </summary>*
3504. public string[] symbols = new string[3] { "R(i)", "х", "ФЗ" };
3505. */// <summary>*
3506. */// Универсальная газовая постоянная*
3507. */// </summary>*
3508. private const double kN = 1.9844 \* 0.001;
3509. */// <summary>*
3510. */// Константа Моделунга*
3511. */// </summary>*
3512. public double A = -1;
3513. */// <summary>*
3514. */// Формальный заряд общего химического элемента*
3515. */// </summary>*
3516. public double zX = -1;
3517. */// <summary>*
3518. */// Номер строки из таблицы данных первого химического элемента*
3519. */// </summary>*
3520. public int numA = 0;
3521. */// <summary>*
3522. */// Номер строки из таблицы данных второго химического элемента*
3523. */// </summary>*
3524. public int numB = 0;
3525. */// <summary>*
3526. */// Номер строки из таблицы данных общего химического элемента*
3527. */// </summary>*
3528. public int numX = 0;
3529. */// <summary>*
3530. */// Обозначение системы соединений*
3531. */// </summary>*
3532. private readonly string sourceString;
3533. */// <summary>*
3534. */// Первый химический элемент*
3535. */// </summary>*
3536. private Composition elemA;
3537. */// <summary>*
3538. */// Второй химический элемент*
3539. */// </summary>*
3540. private Composition elemB;
3541. */// <summary>*
3542. */// Общий химический элемент*
3543. */// </summary>*
3544. private Composition elemX;
3546. */// <summary>*
3547. */// Эмпирический параметр*
3548. */// </summary>*
3549. private double c = -1;
3550. */// <summary>*
3551. */// Число структурных единиц*
3552. */// </summary>*
3553. private double m = -1;
3554. */// <summary>*
3555. */// Координационное число*
3556. */// </summary>*
3557. private double n = -1;
3558. */// <summary>*
3559. */// Формальный заряд*
3560. */// </summary>*
3561. private double z = -1;
3562. */// <summary>*
3563. */// Ионный радиус первого элемента*
3564. */// </summary>*
3565. private double r\_1 = -1;
3566. */// <summary>*
3567. */// Ионный радиус второго элемента*
3568. */// </summary>*
3569. private double r\_2 = -1;
3570. */// <summary>*
3571. */// Ионный радиус общего элемента*
3572. */// </summary>*
3573. private double r\_3 = -1;
3574. */// <summary>*
3575. */// Межатомное расстояние*
3576. */// </summary>*
3577. public double R\_const = -1;
3578. */// <summary>*
3579. */// Разность степеней ионности*
3580. */// </summary>*
3581. private double deleps = -1;
3583. */// <summary>*
3584. */// Возвращает ионный радиус первого элемента*
3585. */// </summary>*
3586. public double r1
3587. {
3588. get
3589. {
3590. if (r\_1 == -1)
3591. double.TryParse(elemA.Properties.Find(x => x.First.Second == symbols[0]).Second[numA], out r\_1);
3593. return r\_1;
3594. }
3595. }
3597. */// <summary>*
3598. */// Возвращает ионный радиус второго элемента*
3599. */// </summary>*
3600. public double r2
3601. {
3602. get
3603. {
3604. if (r\_2 == -1)
3605. double.TryParse(elemB.Properties.Find(x => x.First.Second == symbols[0]).Second[numB], out r\_2);
3607. return r\_2;
3608. }
3609. }
3611. */// <summary>*
3612. */// Возвращает ионный радиус общего элемента*
3613. */// </summary>*
3614. public double r3
3615. {
3616. get
3617. {
3618. if (r\_3 == -1)
3619. double.TryParse(elemX.Properties.Find(x => x.First.Second == symbols[0]).Second[numX], out r\_3);
3621. return r\_3;
3622. }
3623. }
3625. */// <summary>*
3626. */// Межатомное расстояние первого соединения*
3627. */// </summary>*
3628. public double R1
3629. {
3630. get
3631. {
3632. if (r\_1 == -1)
3633. double.TryParse(elemA.Properties.Find(x => x.First.Second == symbols[0]).Second[numA], out r\_1);
3634. if (r\_3 == -1)
3635. double.TryParse(elemX.Properties.Find(x => x.First.Second == symbols[0]).Second[numX], out r\_3);
3637. if (r\_3 != -1 && r\_1 != -1)
3638. return r\_1 + r\_3;
3639. else
3640. return -1;
3641. }
3642. }
3644. */// <summary>*
3645. */// Межатомное расстояние второго соединения*
3646. */// </summary>*
3647. public double R2
3648. {
3649. get
3650. {
3651. if (r\_2 == -1)
3652. double.TryParse(elemB.Properties.Find(x => x.First.Second == symbols[0]).Second[numB], out r\_2);
3653. if (r\_3 == -1)
3654. double.TryParse(elemX.Properties.Find(x => x.First.Second == symbols[0]).Second[numX], out r\_3);
3656. if (r\_3 != -1 && r\_2 != -1)
3657. return r\_2 + r\_3;
3658. else
3659. return -1;
3660. }
3661. }
3663. */// <summary>*
3664. */// Обозначение первого химического элемента*
3665. */// </summary>*
3666. public string ElementA
3667. {
3668. get { return elemA.Name; }
3669. }
3671. */// <summary>*
3672. */// Обозначение второго химического элемента*
3673. */// </summary>*
3674. public string ElementB
3675. {
3676. get { return elemB.Name; }
3677. }
3679. */// <summary>*
3680. */// Обозначение общего химического элемента*
3681. */// </summary>*
3682. public string ElementX
3683. {
3684. get { return elemX.Name; }
3685. }
3687. */// <summary>*
3688. */// Конструктор системы*
3689. */// </summary>*
3690. */// <param name="source">обозначение системы</param>*
3691. */// <param name="A">первый химичский элемент</param>*
3692. */// <param name="B">второй химичский элемент</param>*
3693. */// <param name="X">общий химичский элемент</param>*
3694. public BinSystem(string source, Composition A, Composition B, Composition X)
3695. {
3696. sourceString = source;
3697. elemA = A;
3698. elemB = B;
3699. elemX = X;
3700. }
3702. */// <summary>*
3703. */// Позволяет задать параметры системы*
3704. */// </summary>*
3705. */// <param name="c">эмпирический параметр</param>*
3706. */// <param name="m">число структурных единиц</param>*
3707. */// <param name="n">координационное число</param>*
3708. */// <param name="z">формальный заряд</param>*
3709. public void SetData(double c, double m, double n, double z)
3710. {
3711. this.c = c;
3712. this.m = m;
3713. this.n = n;
3714. this.z = z;
3715. }
3717. */// <summary>*
3718. */// Позволяет получить параметры системы*
3719. */// </summary>*
3720. */// <returns>массив параметров</returns>*
3721. public double[] GetData() { return new double[] { c, m, n, z, numA, numB, numX }; }
3723. */// <summary>*
3724. */// Энтропия смешения*
3725. */// </summary>*
3726. public double Ssm(double x1)
3727. {
3728. double x2 = 1 - x1;
3729. double Skon = (-1) \* kN \* (x1 \* Math.Log(x1) + x2 \* Math.Log(x2));
3730. double Skol = 2.7252 \* x1 \* x2 \* (delR / R1) \* 0.001;
3732. return Skon + Skol;
3733. }
3735. */// <summary>*
3736. */// Среднее межатомное расстояние*
3737. */// </summary>*
3738. public double R(double x1)
3739. {
3740. if (R\_const != -1)
3741. return R\_const;
3743. if (r\_2 == -1)
3744. double.TryParse(elemB.Properties.Find(x => x.First.Second == symbols[0]).Second[numB], out r\_2);
3745. if (r\_3 == -1)
3746. double.TryParse(elemX.Properties.Find(x => x.First.Second == symbols[0]).Second[numX], out r\_3);
3747. if (r\_1 == -1)
3748. double.TryParse(elemA.Properties.Find(x => x.First.Second == symbols[0]).Second[numA], out r\_1);
3750. if (r\_1 != -1 && r\_2 != -1 && r\_3 != -1)
3751. return x1 \* r\_1 + (1 - x1) \* r\_2 + r\_3;
3752. else
3753. return -1;
3754. }
3756. */// <summary>*
3757. */// Разность радиусов*
3758. */// </summary>*
3759. public double delR
3760. {
3761. get
3762. {
3763. if (r1 != -1 && r2 != -1)
3764. return Math.Abs(r\_1 - r\_2);
3765. else
3766. return -1;
3767. }
3768. }
3770. */// <summary>*
3771. */// Степень ионности*
3772. */// </summary>*
3773. */// <param name="i">Флаг: 1 - элемент A, 2 - элемент B</param>*
3774. public double Eps(int i)
3775. {
3777. Composition temp;
3778. int num;
3780. switch (i)
3781. {
3782. case 1:
3783. temp = elemA;
3784. num = numA;
3785. break;
3786. case 2:
3787. temp = elemB;
3788. num = numB;
3789. break;
3790. default:
3791. return 0;
3792. }
3794. if (double.TryParse(temp.Properties.Find(x => x.First.Second == symbols[1]).Second[num], out double k)
3795. && double.TryParse(elemX.Properties.Find(x => x.First.Second == symbols[1]).Second[numX], out double j))
3796. return 1 - (z / n) \* Math.Exp((k - j) \* (k - j) \* -0.25);
3797. else
3798. return -1;
3799. }
3801. */// <summary>*
3802. */// Разность степеней ионности*
3803. */// </summary>*
3804. public double delEps
3805. {
3806. get
3807. {
3808. if (deleps < 0)
3809. deleps = Math.Abs(Eps(1) - Eps(2));
3811. return deleps;
3812. }
3814. set
3815. {
3816. deleps = value;
3817. }
3818. }
3820. */// <summary>*
3821. */// Теплота смешения*
3822. */// </summary>*
3823. public double Hsm(double x1)
3824. {
3825. if (zX == -1)
3826. double.TryParse(elemX.Properties.Find(x => x.First.Second == symbols[2]).Second[numX], out zX);
3828. return x1 \* (1 - x1) \* (322 \* A / R(x1) \* (delEps \* delEps) + c \* m \* n \* z \* zX \* (delR / R(x1) \* delR / R(x1)));
3829. }
3831. */// <summary>*
3832. */// Свободная энергия Гиббса*
3833. */// </summary>*
3834. */// <param name="T">температура</param>*
3835. public double Gsm(double x1, double T)
3836. {
3837. return Hsm(x1) - T \* Ssm(x1);
3838. }
3840. */// <summary>*
3841. */// Критическая температура*
3842. */// </summary>*
3843. public double Tmax
3844. {
3845. get
3846. {
3847. if (zX == -1)
3848. double.TryParse(elemX.Properties.Find(x => x.First.Second == symbols[2]).Second[numX], out zX);
3850. return (c \* m \* n \* z \* zX \* ((delR / Math.Min(R1, R2)) \* (delR / Math.Min(R1, R2)))) / (2 \* kN);
3851. }
3852. }
3854. */// <summary>*
3855. */// Возвращает обозначение системы*
3856. */// </summary>*
3857. */// <returns>обозначение системы</returns>*
3858. public override string ToString()
3859. {
3860. return sourceString;
3861. }
3863. */// <summary>*
3864. */// Создает копию системы*
3865. */// </summary>*
3866. */// <returns>копия данной системы</returns>*
3867. public BinSystem Clone()
3868. {
3869. BinSystem toClone = new BinSystem(sourceString, new Composition(elemA.Name, elemA.DataTable, elemA.Properties), new Composition(elemB.Name, elemB.DataTable, elemB.Properties),
3870. new Composition(elemX.Name, elemX.DataTable, elemX.Properties));
3871. toClone.SetData(c, m, n, z);
3872. toClone.numA = numA;
3873. toClone.numB = numB;
3874. toClone.numX = numX;
3875. toClone.A = A;
3877. return toClone;
3878. }
3879. }
3880. }
3881. * **Composition.cs**
3882. using System;
3883. using System.Collections.Generic;
3884. using System.Data;
3885. using System.Xml.Serialization;
3887. namespace Activision\_Mendeleyev\_table.HelperClasses
3888. {
3889. */// <summary>*
3890. */// Класс, представляющий собой соединение(элемент)*
3891. */// </summary>*
3892. [Serializable]
3893. public class Composition
3894. {
3895. */// <summary>*
3896. */// Название соединения(элемента)*
3897. */// </summary>*
3898. [XmlElement("Name")]
3899. public string Name { get; set; }
3901. */// <summary>*
3902. */// Лист свойств и соответствующих их значений*
3903. */// </summary>*
3904. [XmlArray("Properties"), XmlArrayItem("Property")]
3905. public List<Pair<Pair<string, string>, List<string>>> Properties { get; set; }
3907. */// <summary>*
3908. */// Таблица свойств элементов*
3909. */// </summary>*
3910. [XmlElement("Table")]
3911. public DataTable DataTable { get; set; }
3913. */// <summary>*
3914. */// Создает объект типа Composition и инициализирует лист свойств*
3915. */// </summary>*
3916. public Composition() { Properties = new List<Pair<Pair<string, string>, List<string>>>(); }
3918. */// <summary>*
3919. */// Создает объект типа Composition и инициализирует все свойства класса*
3920. */// </summary>*
3921. */// <param name="name">название соединения(элемента)</param>*
3922. */// <param name="data">таблица свойств элементов</param>*
3923. */// <param name="prop">лист свойств и соответсвующих им значений</param>*
3924. public Composition(string name, DataTable data = null, List<Pair<Pair<string, string>, List<string>>> prop = null)
3925. {
3926. Properties = prop;
3927. DataTable = data;
3928. Name = name;
3929. if (data == null)
3930. DataTable = new DataTable() { TableName = name};
3931. if (prop == null)
3932. Properties = new List<Pair<Pair<string, string>, List<string>>>();
3933. }
3934. }
3935. }
3936. * **DataGridHelper.cs**
3937. using System;
3938. using System.Data;
3939. using System.IO;
3940. using System.Windows;
3941. using System.Windows.Controls;
3942. using System.Windows.Data;
3943. using System.Windows.Media;
3944. using System.Xml.Serialization;
3946. namespace Activision\_Mendeleyev\_table.HelperClasses
3947. {
3948. */// <summary>*
3949. *///Вспомогательный статический класс для работы с DataGrid*
3950. */// </summary>*
3951. public static class DataGridHelper
3952. {
3953. */// <summary>*
3954. */// Вспомогательный метод для получения ячейки таблицы DataGrid*
3955. */// </summary>*
3956. */// <typeparam name="T">Visual type</typeparam>*
3957. */// <param name="parent">предок</param>*
3958. */// <returns>потомок</returns>*
3959. private static T GetVisualChild<T>(Visual parent) where T : Visual
3960. {
3961. try
3962. {
3963. T child = default(T);
3964. int numVisuals = VisualTreeHelper.GetChildrenCount(parent);
3965. for (int i = 0; i < numVisuals; i++)
3966. {
3967. Visual v = (Visual)VisualTreeHelper.GetChild(parent, i);
3968. child = v as T;
3969. if (child == null)
3970. child = GetVisualChild<T>(v);
3972. if (child != null)
3973. break;
3974. }
3975. return child;
3976. }
3977. catch (Exception ex)
3978. {
3979. MessageBox.Show("Ошибка получения данных из таблицы функция (GetVisualChild<T>)!\n\n" + ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
3980. return null;
3981. }
3983. }
3985. */// <summary>*
3986. */// Получает строку из DataGrid*
3987. */// </summary>*
3988. */// <param name="index">номер строки</param>*
3989. */// <param name="dg">DataGrid</param>*
3990. */// <returns>Строку DataGrid</returns>*
3991. private static DataGridRow GetRow(int index, DataGrid dg)
3992. {
3993. try
3994. {
3995. DataGridRow row = (DataGridRow)dg.ItemContainerGenerator.ContainerFromIndex(index);
3996. if (row == null)
3997. {
3998. dg.UpdateLayout();
3999. dg.ScrollIntoView(dg.Items[index]);
4000. row = (DataGridRow)dg.ItemContainerGenerator.ContainerFromIndex(index);
4002. }
4003. return row;
4004. }
4005. catch (Exception ex)
4006. {
4007. MessageBox.Show("Ошибка получения данных из таблицы функция (GetRow)!\n\n" + ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
4008. return null;
4009. }
4010. }
4012. */// <summary>*
4013. */// Добавляет текстовый столбец в таблицу*
4014. */// </summary>*
4015. */// <param name="dg">DataGrid</param>*
4016. */// <param name="dat">таблица данных</param>*
4017. */// <param name="f">флаг: true - соединение(элемент), false - система</param>*
4018. */// <returns>Можно ли удалять столбцы?</returns>*
4019. public static bool AddColumn(ref DataGrid dg, ref DataTable dat, bool f = true)
4020. {
4021. ColumnAddWindow form = new ColumnAddWindow();
4022. form.ShowDialog();
4023. try
4024. {
4025. if (form.name != "")
4026. {
4027. if (form.symbol != "" && form.symbol != " ")
4028. foreach (DataColumn v in dat.Columns)
4029. if (v.Caption == form.symbol)
4030. throw new DuplicateNameException();
4032. DataColumn col = new DataColumn(form.name) { Caption = form.symbol };
4033. dat.Columns.Add(col);
4034. dg.Columns.Add(new DataGridTextColumn()
4035. {
4036. Header = (form.symbol == "" || form.symbol == " ") ? form.name : form.name + ", " + form.symbol,
4037. Binding = new Binding("[" + dg.Columns.Count + "]")
4038. });
4039. }
4040. }
4041. catch (DuplicateNameException)
4042. {
4043. MessageBox.Show("Столбец с данным именем(обозначением) уже принадлежит данной таблице!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
4044. }
4046. if (f && dat.Columns.Count > 0 || !f && dat.Columns.Count > 1)
4047. return true;
4048. return false;
4049. }
4051. */// <summary>*
4052. */// Получает ячейку из DataGrid*
4053. */// </summary>*
4054. */// <param name="row">номер строки</param>*
4055. */// <param name="column">номер столбца</param>*
4056. */// <param name="dg">DataGrid</param>*
4057. */// <returns>Ячейка DataGrid</returns>*
4058. public static DataGridCell GetCell(int row, int column, DataGrid dg)
4059. {
4060. try
4061. {
4062. DataGridRow rowContainer = GetRow(row, dg);
4064. if (rowContainer != null)
4065. {
4066. System.Windows.Controls.Primitives.DataGridCellsPresenter presenter = GetVisualChild<System.Windows.Controls.Primitives.DataGridCellsPresenter>(rowContainer);
4068. DataGridCell cell = (DataGridCell)presenter.ItemContainerGenerator.ContainerFromIndex(column);
4069. if (cell == null)
4070. {
4071. dg.ScrollIntoView(rowContainer, dg.Columns[column]);
4072. cell = (DataGridCell)presenter.ItemContainerGenerator.ContainerFromIndex(column);
4073. }
4074. return cell;
4075. }
4076. return null;
4077. }
4078. catch (Exception ex)
4079. {
4080. MessageBox.Show("Ошибка получения данных из таблицы функция (GetCell)!\n\n" + ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
4081. return null;
4082. }
4083. }
4085. */// <summary>*
4086. */// Закрашивает ячейку DataGrid*
4087. */// </summary>*
4088. */// <param name="row">номер строки</param>*
4089. */// <param name="column">номер столбца</param>*
4090. */// <param name="color">цвет для закрашивания</param>*
4091. */// <param name="dg">DataGrid</param>*
4092. public static void BrushCell(int row, int column, Brush color, DataGrid dg)
4093. {
4094. DataGridCell cell = GetCell(row, column, dg);
4095. cell.Background = color;
4096. cell.BorderBrush = color;
4097. }
4099. */// <summary>*
4100. */// Метод сериализации соединений/элементов/систем соединений*
4101. */// </summary>*
4102. */// <typeparam name="T">тип: List<Composition> или List<DataTable></typeparam>*
4103. */// <param name="name">имя файла</param>*
4104. */// <param name="data">лист соединений/элементов/систем соединений</param>*
4105. public static void Serialize<T>(string name, ref System.Collections.Generic.List<T> data)
4106. {
4107. using (FileStream fs = new FileStream(name, FileMode.Create, FileAccess.Write))
4108. {
4109. XmlSerializer serializer = new XmlSerializer(data.GetType());
4110. serializer.Serialize(fs, data);
4111. }
4112. }
4114. */// <summary>*
4115. */// Метод десериализации соединений/элементов/систем соединений*
4116. */// </summary>*
4117. */// <typeparam name="T">тип: List<Composition> или List<DataTable></typeparam>*
4118. */// <param name="name">имя файла</param>*
4119. */// <param name="data">лист соединений/элементов/систем соединений</param>*
4120. */// <returns>Удалось ли десериализовать?</returns>*
4121. public static bool Deserialize<T>(string name, ref System.Collections.Generic.List<T> data)
4122. {
4123. if (!File.Exists(name))
4124. return false;
4126. try
4127. {
4128. using (FileStream fs = new FileStream(name, FileMode.Open, FileAccess.Read))
4129. {
4130. XmlSerializer serializer = new XmlSerializer(data.GetType());
4131. data = (System.Collections.Generic.List<T>)serializer.Deserialize(fs);
4132. }
4133. return true;
4134. }
4135. catch (FileLoadException)
4136. {
4137. return false;
4138. }
4139. }
4140. }
4141. }
4142. * **HotKey.cs**
4143. using System;
4144. using System.Collections.Generic;
4145. using System.Diagnostics;
4146. using System.Runtime.InteropServices;
4147. using System.Windows.Input;
4148. using System.Windows.Interop;
4150. namespace Activision\_Mendeleyev\_table.HelperClasses
4151. {
4152. */// <summary>*
4153. */// Класс для привязки клавишь к действиям*
4154. */// </summary>*
4155. public class HotKey : IDisposable
4156. {
4157. */// <summary>*
4158. */// Словарь id и горячих клавиш для обратного вызова*
4159. */// </summary>*
4160. private static Dictionary<int, HotKey> \_dictHotKeyToCallBackProc;
4162. [DllImport("user32.dll")]
4163. private static extern bool RegisterHotKey(IntPtr hWnd, int id, uint fsModifiers, uint vlc);
4165. [DllImport("user32.dll")]
4166. private static extern bool UnregisterHotKey(IntPtr hWnd, int id);
4168. */// <summary>*
4169. */// Слово для генерации сообщений*
4170. */// </summary>*
4171. public const int WmHotKey = 0x0312;
4173. */// <summary>*
4174. */// Флаг для управления методоми Dispose() и Dispose(bool)*
4175. */// </summary>*
4176. private bool \_disposed = false;
4178. */// <summary>*
4179. */// Клавиша*
4180. */// </summary>*
4181. public Key Key { get; private set; }
4183. */// <summary>*
4184. */// Модификаторы*
4185. */// </summary>*
4186. public KeyModifier KeyModifiers { get; private set; }
4188. */// <summary>*
4189. */// Метод, который вызывается по нажатию сочетания клавиш*
4190. */// </summary>*
4191. public Action<HotKey> Action { get; private set; }
4193. */// <summary>*
4194. */// Уникальный идентификатор сочетания клавиш*
4195. */// </summary>*
4196. public int Id { get; set; }
4198. */// <summary>*
4199. */// Создает объект типа HotKey и регистрирует переданное сочетание клавиш(по умолчанию)*
4200. */// </summary>*
4201. */// <param name="k">клавиша</param>*
4202. */// <param name="keyModifiers">модификаторы</param>*
4203. */// <param name="action">метод</param>*
4204. */// <param name="register">Нужно ли регистрировать?</param>*
4205. public HotKey(Key k, KeyModifier keyModifiers, Action<HotKey> action, bool register = true)
4206. {
4207. Key = k;
4208. KeyModifiers = keyModifiers;
4209. Action = action;
4210. if (register)
4211. Register();
4212. }
4214. */// <summary>*
4215. */// Регистрирует сочетание клавиш*
4216. */// </summary>*
4217. */// <returns>Зарегистрирован или нет?</returns>*
4218. public bool Register()
4219. {
4220. int virtualKeyCode = KeyInterop.VirtualKeyFromKey(Key);
4221. Id = virtualKeyCode + ((int)KeyModifiers \* 0x10000);
4222. bool result = RegisterHotKey(IntPtr.Zero, Id, (uint)KeyModifiers, (uint)virtualKeyCode);
4224. if (\_dictHotKeyToCallBackProc == null)
4225. {
4226. \_dictHotKeyToCallBackProc = new Dictionary<int, HotKey>();
4227. ComponentDispatcher.ThreadFilterMessage += new ThreadMessageEventHandler(ComponentDispatcherThreadFilterMessage);
4228. }
4230. \_dictHotKeyToCallBackProc.Add(Id, this);
4232. Debug.Print(result.ToString() + ", " + Id + ", " + virtualKeyCode);
4233. return result;
4234. }
4236. */// <summary>*
4237. */// Утилизирует управляемые ресурсы*
4238. */// </summary>*
4239. public void Unregister()
4240. {
4241. HotKey hotKey;
4242. if (\_dictHotKeyToCallBackProc.TryGetValue(Id, out hotKey))
4243. UnregisterHotKey(IntPtr.Zero, Id);
4244. }
4246. private static void ComponentDispatcherThreadFilterMessage(ref MSG msg, ref bool handled)
4247. {
4248. if (!handled && msg.message == WmHotKey)
4249. {
4250. HotKey hotKey;
4251. if (\_dictHotKeyToCallBackProc.TryGetValue((int)msg.wParam, out hotKey))
4252. {
4253. if (hotKey.Action != null)
4254. hotKey.Action.Invoke(hotKey);
4255. handled = true;
4256. }
4257. }
4258. }
4260. */// <summary>*
4261. */// Метод IDisposable, вызывает наш virtual Dispose(bool)*
4262. */// </summary>*
4263. public void Dispose()
4264. {
4265. Dispose(true);
4266. GC.SuppressFinalize(this);
4267. }
4269. */// <summary>*
4270. */// Утилизирует ресурсы*
4271. */// </summary>*
4272. */// <param name="disposing">Можно ли утилизировать управляемые ресурсы?</param>*
4273. protected virtual void Dispose(bool disposing)
4274. {
4275. if (!this.\_disposed)
4276. {
4277. if (disposing)
4278. Unregister();
4280. \_disposed = true;
4281. }
4282. }
4283. }
4285. */// <summary>*
4286. */// Модификаторы*
4287. */// </summary>*
4288. [Flags]
4289. public enum KeyModifier
4290. {
4291. None = 0x0000,
4292. Alt = 0x0001,
4293. Ctrl = 0x0002,
4294. NoRepeat = 0x4000,
4295. Shift = 0x0004,
4296. Win = 0x0008
4297. }
4298. }
4299. * **MathParser.cs**
4300. using System;
4301. using System.Collections.Generic;
4302. using System.Data;
4303. using System.Globalization;
4304. using System.Text.RegularExpressions;
4306. namespace Activision\_Mendeleyev\_table.HelperClasses
4307. {
4308. */// <summary>*
4309. */// Класс, вычисляющей значение формулы по введенной строке по определенным правилам*
4310. */// </summary>*
4311. public static class MathParser
4312. {
4313. */// <summary>*
4314. */// Флаг: true - соединений, false - система*
4315. */// </summary>*
4316. public static bool f = true;
4318. */// <summary>*
4319. */// Парсинг строки для вычисления значения формулы, если данная строка является формулой*
4320. */// </summary>*
4321. */// <param name="str">содержание ячайки таблицы</param>*
4322. */// <param name="dat">таблица, в которой производим вычисление</param>*
4323. */// <param name="u">номер строки</param>*
4324. */// <returns>результат парсинга</returns>*
4325. public static double Parse(string str, ref DataTable dat, int u)
4326. {
4327. str = str.Replace(',', '.');
4328. string left = "", right = "", inner = "";
4329. int i = 0, j = 0, k = 0;
4331. *// Парсинг функций*
4332. string[] func = { "sin", "cos", "tan", "ctan", "abs", "exp", "ln", "min", "max", "pow" };
4333. for (i = 0; i < func.Length; i++)
4334. {
4335. k = str.IndexOf(func[i]);
4336. if (k >= 0)
4337. {
4338. left = str.Substring(0, k);
4339. k += func[i].Length;
4340. j = 0;
4341. bool f = false;
4342. right = "";
4343. inner = "";
4344. while (k < str.Length)
4345. {
4346. if (!f && str[k] == '(' && k < str.Length)
4347. {
4348. j++;
4349. if (j == 1)
4350. {
4351. left += right;
4352. right = "";
4353. k++;
4354. }
4355. }
4357. right += str[k];
4359. if (!f && k < str.Length - 1 && str[k + 1] == ')')
4360. {
4361. j--;
4362. if (j == 0)
4363. {
4364. inner += right;
4365. right = "";
4366. f = true;
4367. k++;
4368. }
4369. }
4371. k++;
4372. }
4374. switch (i)
4375. {
4376. case 0:
4377. return Parse(left + Math.Sin(Parse(inner, ref dat, u)) + right, ref dat, u);
4379. case 1:
4380. return Parse(left + Math.Cos(Parse(inner, ref dat, u)) + right, ref dat, u);
4382. case 2:
4383. return Parse(left + Math.Tan(Parse(inner, ref dat, u)) + right, ref dat, u);
4385. case 3:
4386. return Parse(left + 1.0 / Math.Tan(Parse(inner, ref dat, u)) + right, ref dat, u);
4388. case 4:
4389. return Parse(left + Math.Abs(Parse(inner, ref dat, u)) + right, ref dat, u);
4391. case 5:
4392. return Parse(left + Math.Exp(Parse(inner, ref dat, u)) + right, ref dat, u);
4394. case 6:
4395. return Parse(left + Math.Log(Parse(inner, ref dat, u)) + right, ref dat, u);
4397. case 7:
4398. int a = 0, b = 0, v = 0, g = 0;
4399. string inleft = "";
4400. while (inner[g] != ';' || a != 0 || b != 0 || v != 0)
4401. {
4402. inleft += inner[g];
4403. if (inner[g] == '(')
4404. a++;
4405. else if (inner[g] == ')')
4406. a--;
4407. else if (inner[g] == '[')
4408. b++;
4409. else if (inner[g] == ']')
4410. b--;
4411. else if (inner[g] == '{')
4412. v++;
4413. else if (inner[g] == '}')
4414. v--;
4415. g++;
4416. }
4418. string inright = inner.Substring(g + 1);
4419. return Parse(left + Math.Min(Parse(inleft, ref dat, u), Parse(inright, ref dat, u)) + right, ref dat, u);
4421. case 8:
4422. a = 0; b = 0; v = 0; g = 0;
4423. inleft = "";
4424. while (inner[g] != ';' || a != 0 || b != 0 || v != 0)
4425. {
4426. inleft += inner[g];
4427. if (inner[g] == '(')
4428. a++;
4429. else if (inner[g] == ')')
4430. a--;
4431. else if (inner[g] == '[')
4432. b++;
4433. else if (inner[g] == ']')
4434. b--;
4435. else if (inner[g] == '{')
4436. v++;
4437. else if (inner[g] == '}')
4438. v--;
4439. g++;
4440. }
4442. inright = inner.Substring(g + 1);
4443. return Parse(left + Math.Max(Parse(inleft, ref dat, u), Parse(inright, ref dat, u)) + right, ref dat, u);
4445. case 9:
4446. a = 0; b = 0; v = 0; g = 0;
4447. inleft = "";
4448. while (inner[g] != ';' || a != 0 || b != 0 || v != 0)
4449. {
4450. inleft += inner[g];
4451. if (inner[g] == '(')
4452. a++;
4453. else if (inner[g] == ')')
4454. a--;
4455. else if (inner[g] == '[')
4456. b++;
4457. else if (inner[g] == ']')
4458. b--;
4459. else if (inner[g] == '{')
4460. v++;
4461. else if (inner[g] == '}')
4462. v--;
4463. g++;
4464. }
4466. inright = inner.Substring(g + 1);
4467. return Parse(left + Math.Pow(Parse(inleft, ref dat, u), Parse(inright, ref dat, u)) + right, ref dat, u);
4468. }
4469. }
4470. }
4472. *//Парсинг символа x*
4473. Match matchFuncx = Regex.Match(str, @"(x)");
4474. if (matchFuncx.Groups.Count > 1)
4475. {
4476. left = str.Substring(0, matchFuncx.Index);
4477. right = str.Substring(matchFuncx.Index + matchFuncx.Length);
4478. return Parse(left + double.Parse(dat.Rows[u]["X"].ToString()) + right, ref dat, u);
4479. }
4481. *//Парсинг бесконечности*
4482. Match matchinf = Regex.Match(str, @"(∞)");
4483. if (matchinf.Groups.Count > 1)
4484. {
4485. left = str.Substring(0, matchinf.Index);
4486. right = str.Substring(matchinf.Index + matchinf.Length);
4487. return Parse(left + "10000000000000000000" + right, ref dat, u);
4488. }
4490. string str1 = "";
4491. int r = -1, c = -1;
4492. *//Парсинг конструкций для получения значений из таблиц*
4493. try
4494. {
4495. Match matchElem = Regex.Match(str, @"([\w\[\]\.\+\-\\*\/%\^\_\(\)₀₁₂₃₄₅₆₇₈₉]\*){([\w\[\]\.\+\-\\*\/%\^\_\(\)₀₁₂₃₄₅₆₇₈₉ ]\*){([\[\]\d\.\+\-\\*\/%\^\_\(\) ]\*)}([\w\[\]\.\+\-\\*\/%\^\_\(\)₀₁₂₃₄₅₆₇₈₉ ]\*)}|([\w\[\]\.\+\-\\*\/%\^\_\(\)₀₁₂₃₄₅₆₇₈₉]\*){([\w\[\]\.\+\-\\*\/%\^\_\(\)₀₁₂₃₄₅₆₇₈₉ ]\*)}");
4496. if (matchElem.Groups.Count > 1)
4497. {
4498. inner = StringHelper.DoString(str.Substring(matchElem.Index, matchElem.Length)).Trim(' ');
4499. left = str.Substring(0, matchElem.Index);
4500. right = str.Substring(matchElem.Index + matchElem.Length);
4501. LinkedList<string> strs = new LinkedList<string>();
4502. string symbol = "";
4503. k = 0;
4504. while (k < inner.Length)
4505. {
4506. if (inner[k] == '{')
4507. {
4508. k++;
4509. while (k < inner.Length && inner[k] != '}' && inner[k] != '{')
4510. {
4511. str1 += inner[k];
4512. k++;
4513. }
4514. strs.AddLast(str1);
4515. str1 = "";
4516. }
4517. else if (inner[k] == '}')
4518. {
4519. k++;
4520. while (k < inner.Length && inner[k] != '}' && inner[k] != '{')
4521. {
4522. str1 += inner[k];
4523. k++;
4524. }
4525. strs.AddFirst(str1);
4526. str1 = "";
4527. }
4528. else
4529. {
4530. symbol += inner[k];
4531. k++;
4532. }
4533. }
4534. strs.AddLast(symbol);
4535. strs.RemoveFirst();
4536. return Parse(left + Find(strs, u).ToString() + right, ref dat, u);
4537. }
4539. matchElem = Regex.Match(str, @"([\w\[\]\.\+\-\\*\/%\^\_\(\)₀₁₂₃₄₅₆₇₈₉]\*){([\d ]\*;[\d ]\*)}");
4540. if (matchElem.Groups.Count > 1)
4541. {
4542. inner = StringHelper.DoString(str.Substring(matchElem.Index, matchElem.Length)).Trim(' ');
4543. left = str.Substring(0, matchElem.Index);
4544. right = str.Substring(matchElem.Index + matchElem.Length);
4545. str1 = "";
4546. k = 0;
4547. Composition e = null;
4548. DataTable d = null;
4550. while (k < inner.Length)
4551. {
4552. if (inner[k] == '{')
4553. {
4554. k++;
4555. e = MendeleevTable.Elems.Find(x => x.Name == str1);
4556. if (e == null)
4557. e = MendeleevTable.Compos.Find(x => x.Name == str1);
4559. if (e == null)
4560. d = MendeleevTable.BinarySistem.Find(x => x.TableName == str1);
4562. if (d == null & e == null)
4563. throw new Exception("Отсутствует соединение, элемент или система" + str1 + ".", new Exception("MyException"));
4564. str1 = "";
4565. }
4566. else if (inner[k] == ';')
4567. {
4568. k++;
4569. int.TryParse(str1, out r);
4570. str1 = "";
4571. }
4572. else if (inner[k] == '}')
4573. {
4574. k++;
4575. int.TryParse(str1, out c);
4576. str1 = "";
4577. }
4578. else
4579. {
4580. str1 += inner[k];
4581. k++;
4582. }
4583. }
4584. if (e != null)
4585. return Parse(left + e.Properties[c].Second[r] + right, ref dat, u);
4586. return Parse(left + d.Rows[r][c] + right, ref dat, u);
4587. }
4589. matchElem = Regex.Match(str, @"([\w\[\]\.\+\-\\*\/%\^\_\(\)₀₁₂₃₄₅₆₇₈₉]\*){([\d ]\*;[\w\[\]\.\+\-\\*\/%\^\_\(\)₀₁₂₃₄₅₆₇₈₉ ]\*)}");
4590. if (matchElem.Groups.Count > 1)
4591. {
4592. inner = StringHelper.DoString(str.Substring(matchElem.Index, matchElem.Length)).Trim(' ');
4593. left = str.Substring(0, matchElem.Index);
4594. right = str.Substring(matchElem.Index + matchElem.Length);
4595. str1 = "";
4596. k = 0;
4597. Composition e = null;
4598. DataTable d = null;
4600. while (k < inner.Length)
4601. {
4602. if (inner[k] == '{')
4603. {
4604. k++;
4605. e = MendeleevTable.Elems.Find(x => x.Name == str1);
4606. if (e == null)
4607. e = MendeleevTable.Compos.Find(x => x.Name == str1);
4609. if (e == null)
4610. d = MendeleevTable.BinarySistem.Find(x => x.TableName == str1);
4612. if (d == null & e == null)
4613. throw new Exception("Отсутствует соединение, элемент или система" + str1 + ".", new Exception("MyException"));
4614. str1 = "";
4615. }
4616. else if (inner[k] == ';')
4617. {
4618. k++;
4619. int.TryParse(str1, out r);
4620. str1 = "";
4621. }
4622. else if (inner[k] == '}')
4623. {
4624. k++;
4625. if (e != null)
4626. {
4627. for (c = 0; c < e.Properties.Count; c++)
4628. if (e.Properties[c].First.Second == str1)
4629. break;
4630. }
4631. else
4632. for (c = 0; c < d.Columns.Count; c++)
4633. if (d.Columns[c].Caption == str1)
4634. break;
4635. str1 = "";
4636. }
4637. else
4638. {
4639. str1 += inner[k];
4640. k++;
4641. }
4642. }
4643. if (e != null)
4644. return Parse(left + e.Properties[c].Second[r] + right, ref dat, u);
4645. return Parse(left + d.Rows[r][c] + right, ref dat, u);
4646. }
4647. }
4648. catch (FormatException)
4649. {
4650. throw new FormatException(string.Format("Неверная входная строка '{0}'", str), new Exception("MyException"));
4651. }
4652. catch (ArgumentOutOfRangeException)
4653. {
4654. throw new FormatException(string.Format("Отсутствует строка {0} или стобец {1} у данного элемента/соединения.", r, c), new Exception("MyException"));
4655. }
4656. catch (NullReferenceException)
4657. {
4658. throw new FormatException("Отсутствует содержание ячейки, к которой вы обратились.", new Exception("MyException"));
4659. }
4661. *// Парсинг скобок*
4662. Match brackets = Regex.Match(str, @"\(([\w\d\[\]\.\+\-\\*\/%\^\_\(\) ]\*)\)");
4663. if (brackets.Groups.Count > 1)
4664. {
4665. i = 0;
4666. j = 0;
4667. left = "";
4668. right = "";
4669. inner = "";
4670. while (i < str.Length)
4671. {
4672. if (str[i] == '(' && i < str.Length)
4673. {
4674. j++;
4675. if (j == 1)
4676. {
4677. left += right;
4678. right = "";
4679. i++;
4680. }
4681. }
4683. if (i < str.Length && str[i] == ')')
4684. {
4685. j--;
4686. if (j == 0)
4687. {
4688. inner += right;
4689. right = str.Substring(i + 1);
4690. return Parse(left + Parse(inner, ref dat, u) + right, ref dat, u);
4691. }
4692. }
4694. right += str[i];
4695. i++;
4696. }
4697. if (inner != "")
4698. return Parse(left + Parse(inner, ref dat, u) + right, ref dat, u);
4699. }
4701. *// Парсинг действий*
4702. Match matchMulOp = Regex.Match(str, string.Format(@"({0})\s?({1})\s?({0})\s?", RegexNum, RegexMulOp));
4703. Match matchAddOp = Regex.Match(str, string.Format(@"({0})\s?({1})\s?({0})\s?", RegexNum, RegexAddOp));
4704. var match = (matchMulOp.Groups.Count > 1) ? matchMulOp : (matchAddOp.Groups.Count > 1) ? matchAddOp : null;
4705. if (match != null)
4706. {
4707. left = str.Substring(0, match.Index);
4708. right = str.Substring(match.Index + match.Length);
4709. return Parse(left + ParseAct(match).ToString(CultureInfo.InvariantCulture) + right, ref dat, u);
4710. }
4712. *// Парсинг числа*
4713. str1 = "";
4714. for (i = 0; i < str.Length; i++)
4715. if (str[i] != ' ')
4716. if (str[i] == '.')
4717. str1 += ',';
4718. else
4719. str1 += str[i];
4720. if (double.TryParse(str1, out double num))
4721. return num;
4722. else
4723. throw new FormatException(string.Format("Неверная входная строка '{0}'", str), new Exception("MyException"));
4724. }
4726. */// <summary>*
4727. */// Форматная строка, соответствующая числу*
4728. */// </summary>*
4729. private const string RegexNum = @"[-]?\d+\.?\d\*";
4730. */// <summary>*
4731. */// Форматная строка, соответствующая операциям умножения и деления*
4732. */// </summary>*
4733. private const string RegexMulOp = @"[\\*\/%]";
4734. */// <summary>*
4735. */// Форматная строка, соответствующая операциям сложения и вычитания*
4736. */// </summary>*
4737. private const string RegexAddOp = @"[\+\-]";
4739. */// <summary>*
4740. */// Выполнение математических операций*
4741. */// </summary>*
4742. */// <param name="match">результаты вычисления регулярного выражения, определяющего мат. операции</param>*
4743. */// <returns>результат вычисления</returns>*
4744. private static double ParseAct(Match match)
4745. {
4746. double a = double.Parse(match.Groups[1].Value, CultureInfo.InvariantCulture);
4747. double b = double.Parse(match.Groups[3].Value, CultureInfo.InvariantCulture);
4749. switch (match.Groups[2].Value)
4750. {
4751. case "+":
4752. return a + b;
4754. case "-":
4755. return a - b;
4757. case "\*":
4758. return a \* b;
4760. case "/":
4761. return a / b;
4763. case "%":
4764. return a % b;
4766. default:
4767. throw new FormatException(string.Format("Неверная входная строка '{0}'", match.Value), new Exception("MyException"));
4768. }
4769. }
4771. */// <summary>*
4772. */// Поиск значения свойства в таблицах*
4773. */// </summary>*
4774. */// <param name="strs">лист, включающий название соединения(элемента), обозначение свойства и, если необходимо, доп. свойства и его значение</param>*
4775. */// <returns>значение свойсва</returns>*
4776. private static double Find(LinkedList<string> strs, int u)
4777. {
4778. LinkedListNode<string> str = strs.First;
4779. DataTable d = MendeleevTable.BinarySistem.Find(x => x.TableName == str.Value);
4780. Composition e = MendeleevTable.Elems.Find(x => x.Name == str.Value);
4782. if (e == null)
4783. e = MendeleevTable.Compos.Find(x => x.Name == str.Value);
4785. if (d == null & e == null)
4786. throw new Exception("Отсутствует соединение или элемент" + str.Value + ".", new Exception("MyException"));
4788. str = str.Next;
4789. int r = -1;
4790. if (e != null)
4791. {
4792. List<Pair<Pair<string, string>, List<string>>> data = null;
4793. data = e.Properties;
4795. Pair<Pair<string, string>, List<string>> v = data.Find(x => x.First.Second == str.Value);
4797. if (v == null)
4798. throw new Exception("Отсутствует свойство " + str.Value + " у данного соединения или элемента.", new Exception("MyException"));
4800. str = str.Next;
4802. if (str == null)
4803. if (f)
4804. return double.Parse(v.Second[v.Second.Count > u ? u : 0]);
4805. else
4806. return double.Parse(v.Second[0]);
4808. for (int i = 0; i < v.Second.Count; i++)
4809. if (v.Second[i] == str.Value)
4810. r = i;
4812. if (r == -1)
4813. throw new Exception("Отсутствует значение параметра " + str.Previous.Value + " равного " + str.Value + " у данного соединения или элемента.", new Exception("MyException"));
4815. v = data.Find(x => x.First.Second == strs.Last.Value);
4817. if (v == null)
4818. throw new Exception("Отсутствует параметр " + strs.Last.Value + " у данного соединения или элемента.", new Exception("MyException"));
4820. return double.Parse(v.Second[r]);
4821. }
4822. int c = -1;
4824. for (int i = 0; i < d.Columns.Count; i++)
4825. if (d.Columns[i].Caption == str.Value)
4826. c = i;
4828. if (c == -1)
4829. throw new Exception("Отсутствует свойство " + str.Value + " у данной системы соединений.", new Exception("MyException"));
4831. str = str.Next;
4833. if (str == null)
4834. return double.Parse(d.Rows[u][c].ToString());
4836. for (int i = 0; i < d.Rows.Count; i++)
4837. if (d.Rows[i][c].ToString().Replace(',', '.') == str.Value)
4838. r = i;
4840. if (r == -1)
4841. throw new Exception("Отсутствует значение параметра " + str.Previous.Value + " равного " + str.Value + " у данной системы соединений.", new Exception("MyException"));
4843. str = str.Next;
4844. c = -1;
4845. for (int i = 0; i < d.Columns.Count; i++)
4846. if (d.Columns[i].Caption == str.Value)
4847. c = i;
4849. if (c == -1)
4850. throw new Exception("Отсутствует параметр " + strs.Last.Value + " у данной системы соединений.", new Exception("MyException"));
4852. return double.Parse(d.Rows[r][c].ToString());
4853. }
4854. }
4855. }
4856. * **Pair.cs**
4857. using System;
4859. namespace Activision\_Mendeleyev\_table.HelperClasses
4860. {
4861. */// <summary>*
4862. */// Класс, который предоставляет возможность хранить два разнородных объекта как единое целое*
4863. */// </summary>*
4864. */// <typeparam name="T">тип первого аргумента</typeparam>*
4865. */// <typeparam name="U">тип второго аргумента</typeparam>*
4866. [Serializable]
4867. public class Pair<T, U>
4868. {
4869. */// <summary>*
4870. */// Первый аргумент*
4871. */// </summary>*
4872. public T First { get; set; }
4874. */// <summary>*
4875. */// Второй аргумент*
4876. */// </summary>*
4877. public U Second { get; set; }
4879. */// <summary>*
4880. */// Создает объект типа Pair*
4881. */// </summary>*
4882. public Pair() { }
4884. */// <summary>*
4885. */// Создает объект типа Pair и инициализирует значение аргументов*
4886. */// </summary>*
4887. */// <param name="first">первый аргумент</param>*
4888. */// <param name="second">второй аргумент</param>*
4889. public Pair(T first, U second)
4890. {
4891. First = first;
4892. Second = second;
4893. }
4894. };
4895. }
4896. * **Point.cs**
4897. namespace Activision\_Mendeleyev\_table.HelperClasses
4898. {
4899. */// <summary>*
4900. */// Класс, который представляет собой точку с двумя численными координатами*
4901. */// </summary>*
4902. public class Point
4903. {
4904. */// <summary>*
4905. */// Конструктор точки*
4906. */// </summary>*
4907. */// <param name="x">координата X</param>*
4908. */// <param name="y">координата Y</param>*
4909. public Point(double x, double y)
4910. {
4911. this.X = x;
4912. this.Y = y;
4913. }
4915. */// <summary>*
4916. */// Свойство координаты Y*
4917. */// </summary>*
4918. public double Y { get; set; }
4919. */// <summary>*
4920. */// Свойство координаты X*
4921. */// </summary>*
4922. public double X { get; set; }
4923. }
4924. }
4925. * **RoundConverter.cs**
4926. using System;
4928. namespace Activision\_Mendeleyev\_table.HelperClasses
4929. {
4930. */// <summary>*
4931. */// Класс, позволяющий округлять значения в таблицах*
4932. */// </summary>*
4933. public class RoundConverter : System.Windows.Data.IValueConverter
4934. {
4935. */// <summary>*
4936. */// Округляет значение до 4 знака после запятой*
4937. */// </summary>*
4938. */// <param name="value">значение</param>*
4939. */// <returns>округленное значение</returns>*
4940. public object Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
4941. {
4942. if (value != null)
4943. {
4944. var val = (double)value;
4945. return Math.Round(val, 4).ToString(System.Globalization.CultureInfo.InvariantCulture);
4946. }
4947. return null;
4948. }
4950. public object ConvertBack(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
4951. {
4952. throw new NotImplementedException("", new Exception("MyException"));
4953. }
4954. }
4955. }
4956. * **Strings.cs:**
        1. **StringHelper**
4957. */// <summary>*
4958. */// Класс для преобразования строк по определенным правилам*
4959. */// </summary>*
4960. public static class StringHelper
4961. {
4962. */// <summary>*
4963. */// Преобразует определённый символ в соответствующий подстрочный*
4964. */// </summary>*
4965. */// <param name="c">символ</param>*
4966. */// <returns>подстрочный символ</returns>*
4967. private static char Substring(char c)
4968. {
4969. switch (c)
4970. {
4971. case '0':
4972. return '₀';
4973. case '1':
4974. return '₁';
4975. case '2':
4976. return '₂';
4977. case '3':
4978. return '₃';
4979. case '4':
4980. return '₄';
4981. case '5':
4982. return '₅';
4983. case '6':
4984. return '₆';
4985. case '7':
4986. return '₇';
4987. case '8':
4988. return '₈';
4989. case '9':
4990. return '₉';
4991. case '+':
4992. return '₊';
4993. case '-':
4994. return '₋';
4995. case '=':
4996. return '₌';
4997. case '(':
4998. return '₍';
4999. case ')':
5000. return '₎';
5001. case '.':
5002. case ',':
5003. return '.';
5004. default:
5005. return c;
5006. }
5007. }
5009. */// <summary>*
5010. */// Преобразует определённый символ в соответствующий надстрочный*
5011. */// </summary>*
5012. */// <param name="c">символ</param>*
5013. */// <returns>надстрочный символ</returns>*
5014. private static char Superstring(char c)
5015. {
5016. switch (c)
5017. {
5018. case '0':
5019. return '⁰';
5020. case '1':
5021. return '¹';
5022. case '2':
5023. return '²';
5024. case '3':
5025. return '³';
5026. case '4':
5027. return '⁴';
5028. case '5':
5029. return '⁵';
5030. case '6':
5031. return '⁶';
5032. case '7':
5033. return '⁷';
5034. case '8':
5035. return '⁸';
5036. case '9':
5037. return '⁹';
5038. case '+':
5039. return '⁺';
5040. case '-':
5041. return '⁻';
5042. case '=':
5043. return '⁼';
5044. case '(':
5045. return '⁽';
5046. case ')':
5047. return '⁾';
5048. case '.':
5049. case ',':
5050. return '·';*//'ʼ';*
5051. default:
5052. return c;
5053. }
5054. }
5056. */// <summary>*
5057. */// Преобразует числовые символы после буквенных и подстрочных символов в подстрочные*
5058. */// </summary>*
5059. */// <param name="str">исходная строка</param>*
5060. */// <returns>преобразованная строка</returns>*
5061. public static string DoString(string str)
5062. {
5063. int i = 0;
5064. string str1 = "";
5065. while (i < str.Length)
5066. {
5067. if (i > 0 && "0123456789".Contains(str[i].ToString()) && "₀₁₂₃₄₅₆₇₈₉)QWERTYUIOPASDFGHJKLZXCVVBNMqwertyuiopasdfghjklzxcvbnm".Contains(str1[i - 1].ToString()))
5068. str1 += Substring(str[i]);
5069. else
5070. if (i > 1 && '.' == str[i - 1] && "₀₁₂₃₄₅₆₇₈₉)QWERTYUIOPASDFGHJKLZXCVVBNMqwertyuiopasdfghjklzxcvbnm".Contains(str1[i - 2].ToString()))
5071. str1 += Substring(str[i]);
5072. else
5073. str1 += str[i];
5074. i++;
5075. }
5076. return str1;
5077. }
5078. }
5079. * 1. **Strings**
5080. */// <summary>*
5081. */// Вспомогательный класс для заполнения Таблицы Менделеева*
5082. */// </summary>*
5083. public class Strings
5084. {
5085. */// <summary>*
5086. */// Задает один ряд элементов по группам и подгруппам*
5087. */// </summary>*
5088. public Strings(string ia = null, string iia = null, string iiia = null, string iva = null, string va = null, string via = null, string viia = null,
5089. string viiia\_1 = null, string viiia\_2 = null, string viiia\_3 = null, string ib = null, string iib = null, string iiib = null, string ivb = null,
5090. string vb = null, string vib = null, string viib = null, string viiib = null)
5091. {
5092. Ia = ia;
5093. IIa = iia;
5094. IIIa = iiia;
5095. IVa = iva;
5096. Va = va;
5097. VIa = via;
5098. VIIa = viia;
5099. VIIIa\_1 = viiia\_1;
5100. VIIIa\_2 = viiia\_2;
5101. VIIIa\_3 = viiia\_3;
5102. Ib = ib;
5103. IIb = iib;
5104. IIIb = iiib;
5105. IVb = ivb;
5106. Vb = vb;
5107. VIb = vib;
5108. VIIb = viib;
5109. VIIIb = viiib;
5110. }
5112. public string Ia { get; set; }
5113. public string IIa { get; set; }
5114. public string IIIa { get; set; }
5115. public string IVa { get; set; }
5116. public string Va { get; set; }
5117. public string VIa { get; set; }
5118. public string VIIa { get; set; }
5119. public string VIIIa\_1 { get; set; }
5120. public string VIIIa\_2 { get; set; }
5121. public string VIIIa\_3 { get; set; }
5122. public string Ib { get; set; }
5123. public string IIb { get; set; }
5124. public string IIIb { get; set; }
5125. public string IVb { get; set; }
5126. public string Vb { get; set; }
5127. public string VIb { get; set; }
5128. public string VIIb { get; set; }
5129. public string VIIIb { get; set; }
5130. }

**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 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |