

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»,
профессор департамента программной
инженерии, канд. техн. наук

«__» _____ 2021 г. В.В. Шилов

Выпускная квалификационная работа

на тему «Программа определения границ растворимости твердых растворов в зависимости от температуры»

по направлению подготовки 09.03.04 «Программная инженерия»

Научный руководитель
профессор департамента
программной инженерии
факультета компьютерных наук
док. техн. наук
В.В. Подбельский

02 июня 2021_____
Подпись, Дата

Выполнил
студент группы БПИ 173
4 курса бакалавриата
образовательной программы
«Программная инженерия»

К. Г. Кожакин

02 июня 2021_____
Подпись, Дата

Аннотация

В данном отчете предоставлены результаты выпускной-квалификационной работы по теме «Программа определения границ растворимости твердых растворов в зависимости от температуры», выполненной в соответствии с Приказом Национального исследовательского университета "Высшая школа экономики" № 2.3-02/1412-06 от 14.12.20 «Об утверждении тем, руководителей выпускных квалификационных работ студентов образовательной программы Программная инженерия факультета компьютерных наук».

В данной выпускной-квалификационной работе описывается разработанное приложение для расчёта областей смесимости изовалентных бинарных соединений. Результаты работы программы сопоставлены с известными экспериментальными данными в системе NaCl-AgCl и $\text{SnO}_2\text{-TiO}_2$ соответственно с непрерывными и ограниченными твердыми растворами. Программа полезна для предварительной оценки зависимости границ твердых растворов от температуры.

Работа содержит 58 страниц, 3 главы, 18 рисунка, 9 таблиц и 22 источников.

Ключевые слова: бинарные твердые растворы, изовалентные компоненты, оптимизация.

Abstract

This graduation and qualification work describes the developed application for calculating the miscibility regions of isovalent binary compounds. The results of the program will be compared with the known experimental data in the NaCl-AgCl and SnO₂-TiO₂ systems with continuous and limited solid solutions. The program will be useful for a preliminary assessment of the dependence of the boundaries of solid solutions on temperature.

The paper contains 58 pages, 3 chapters, 18 illustrations, 9 tables and 22 bibliography items.

Keywords: binary solid solutions, isovalent components, optimization.

Оглавление

Аннотация	2
Abstract.....	3
Основные определения, термины и сокращения	6
Введение.....	7
Глава 1. Обзор предметной области и существующих решений	9
1.1 Предметная область	9
1.2 Существующие решения	12
1.3 Обоснование разработки нового приложения	13
1.4 Выводы по главе	13
Глава 2. Используемые методы, модели и инструменты.....	14
2.1 Теоретическая информация	14
2.2 Использованные инструменты	16
2.2.1. Язык программирования C#.....	16
2.2.2. Microsoft Visual Studio	16
2.2.3. Windows Presentation Foundation	17
2.2.4. Xml.....	17
2.2.5. Itextsharp.....	17
2.3 Выводы по главе	18
Глава 3. Разработка и архитектура приложения.....	19
3.1 Особенности реализации программы	19
3.2 Описание классов.....	22
3.2.1 Интерактивная таблица Менделеева	22
3.2.1.1 Окна программы	22
3.2.1.1.1 ColumnAddWindow	22
3.2.1.1.2 ComposSystemContent	22
3.2.1.1.3 CreateComposSystem	23
3.2.1.1.4 ElemContent.....	24
3.2.1.1.5 FormulaInput.....	25
3.2.1.1.6 FormulaList	26
3.2.1.1.7 MendeleevTable	27
3.2.1.2 Composition	27
3.2.1.3 DataGridHelper	28
3.2.1.4 MathParser.....	30
3.2.1.5 RoundConverter	36
3.2.1.6 StringHelper	36
3.2.1.7 Strings.....	37
3.2.2 Построение графиков.....	37

3.2.2.1	Окна программы	37
3.2.2.1.1	DataSettings	37
3.2.2.1.2	dG_Temp	38
3.2.2.1.3	DomeOfDecay.....	38
3.2.2.2	BinSystem.....	40
3.2.2.3	Collapse	41
3.2.2.4	CollapseGraph	42
3.2.2.5	Report	45
3.2.3	Аппроксимация	46
3.2.3.1	Criterion.....	46
3.2.3.2	Library	46
3.2.4	Help	50
3.2.5	HotKey	50
3.2.6	Pair<T, U>.....	51
3.2.7	Point	51
3.3	Тестирование	51
3.4	Выводы по главе	56
Заключение.....		57
Список использованных источников		58
Приложение А. Техническое задание.....		59
Приложение Б. Текст программы.....		75
Приложение В. Программа и методика испытаний.....		80
Приложение Г. Руководство оператора.....		90

Основные определения, термины и сокращения

1. *Диаграмма состояния* – графическое отображение равновесного состояния физико-химической системы при определенных условиях.
2. *Изоморфное замещение* – мощный и гибкий способ реализации конкретного свойства с направленным изменением состава в рамках желаемой кристаллической структуры.
3. *Изовалентный изоморфизм* – тип изоморфизма, при котором происходит замещение ионов с одинаковой валентностью.
4. *Купол распада* – изображение двух кривых замещения, полученных экспериментальным образом.
5. *Твердые растворы замещения* – однородные системы, в которых атомы одного компонента неупорядоченно замещают атомы другого на всем интервале концентраций – от 0 до 1 (от 0 до 100%).
6. *Энтальпия* – термодинамический потенциал, характеризующий состояние системы в термодинамическом равновесии при выборе в качестве независимых переменных давления, энтропии и числа частиц.
7. *Энтропия* – широко используемый в естественных и точных науках термин, обозначающий меру необратимого рассеивания энергии или бесполезности энергии.

Введение

Современное научное и прикладное материаловедение требует появления как новых материалов, обладающих необходимым сочетанием функциональных свойств, так и оптимизации свойств известных соединений, уже зарекомендовавших себя на практике, с дальнейшим управлением ими с помощью внешних (ростовые или пост-ростовые условия обработки) или внутренних (изоморфное замещение) воздействий. Изоморфное замещение - мощный и гибкий способ реализации конкретного свойства с направленным изменением состава в рамках желаемой кристаллической структуры.

Для ускорения процесса получения материалов, обоснования возможности реализации твердых растворов с нужными составами и конкретизации температурного режима требуется соответствующее программное обеспечение. Основой для создания такого обеспечения должны служить уже существующие известные аналитические расчетные формулы, значения структурных параметров (кристаллохимические данные), табличные, рассчитанные или экспериментальные, эмпирические коэффициенты [1, 2].

В связи с тем, что программа должна быть создана не только на основе теоретических предпосылок, но в ней должны максимально использоваться эмпирические табличные данные, получаемые из разных источников (в том числе и экспериментальные), важным этапом создания такой программы является проверка ее работоспособности на примере известных систем с уточнением вводимых параметров. Только после такой проверки и анализа полученных результатов, представляется возможным использование разработанной программы для новых систем с отсутствующими экспериментальными данными.

Цель работы – создание программного продукта для определения границ растворимости твердых растворов замещения с изовалентными компонентами, доступной для работы с ней широкого круга исследователей, и проверка его работоспособности на примере известных систем с уточнением вводимых параметров.

Для достижения поставленной цели был составлен список задач:

1. Проанализировать существующие решения;
2. Составить функциональные требования к программе;
3. Написать техническое задание;
4. Реализовать возможность получения информации об атомах, химических соединениях и бинарных системах соединений;
5. Реализовать возможность изменения (добавления) данных об атомах, ионах

и соединениях, а также добавления новых химических соединений/систем соединений;

6. Решить задачу аппроксимации табличной зависимости (полученной при экспериментальной оценке границы фаз бинарной системы) функциональной зависимостью, теоретически определяющей термодинамическую функцию смешения;
7. Решить задачу аппроксимации табличной зависимости (полученной при экспериментальной оценке границы фаз бинарной системы) зависимостью, теоретически определяющей купол распада;
8. Реализовать возможность проведения оптимизации параметров функции купола распада по заданным экспериментальным точкам и критической температуре;
9. Реализовать возможность проведения оценки чувствительности (влияния) параметров функции смешения на конечный результат;
10. Реализовать возможность, на основании полученных данных путем оптимального выбора параметров функций, построения графика свободной энергии Гиббса в заданном температурном диапазоне;
11. Разработать и протестировать программу;
12. Подготовить техническую документацию.

Работа структурирована следующим образом:

Первая глава состоит из описания предметной области и анализа существующих аналогов и решений.

Во второй главе содержится описание теоретических предпосылок к написанию приложения и формул, которые в нем используются. Также данная глава включает в себя обзор использованных инструментов разработки.

В третьей главе описываются особенности реализации программы и её классы. Также в данной главе содержатся особенности тестирования.

Заключение работы содержит анализ полученных результатов и вектор дальнейшего развития.

Глава 1. Обзор предметной области и существующих решений

Текущая глава состоит из описания предметной области и анализа имеющихся аналогов на предмет выявления их отличий от данного программного продукта

1.1 Предметная область

Современное материаловедение, опираясь на ряд основных типов материалов (металлы, органические или неорганические полимеры, керамика и композиционные материалы на их основе), представляет собой комплексную (междисциплинарную) науку и учебную дисциплину. Она основана на симбиозе, как минимум, четырех наук: химии, физики, механики и технологии. При этом вклад четырех вышеперечисленных дисциплин естественно различен. Химия — это, прежде всего, теоретическое и практическое изучение и конструирование структур и свойств конкретных химических веществ и материалов на их основе. Первостепенная практическая важность химии в материаловедении определяется тем, что все виды синтетических и искусственных материалов получают путем химических превращений только из химических веществ. Например, металлы (за исключением самородных) типа железа получают восстановлением оксидов железа, бытовую керамику — спеканием («отверждением») слоистых полиалюмосиликатов, органические полимеры — полимеризацией или поликонденсацией мономерных углеводородов. Физика — это, в первую очередь, методы исследования структуры и свойств материалов, моделирование и теоретические обобщения. Механика — это, прежде всего, методы исследования механических и деформационных свойств материалов и их интерпретация. Технология — это методы или последовательность операций и поддержание условий для изменения структуры и свойств вещества и материала, его модификации (легирования), а также способы их переработки в конечные изделия с комплексом заданных свойств.

Главным объектом исследования и изучения в науке и учебной дисциплине, именуемой «Материаловедение», является материал. Именно четкое и правильное понимание существа этого основного объекта исследования материаловедения и его главного фундаментального понятия позволяет наиболее точно раскрыть индивидуальность и специфические особенности предмета этой науки.

Общепринятое определение материаловедения как «науки о связи состава, структуры (строения) и свойств материалов (или в значительной части дефиниций — металлов)» устарело в настоящее время в силу целого ряда факторов.

Во-первых, в настоящее время отсутствует достаточно строгая дефиниция самой главной компоненты приведенного выше определения — понятия «материал». Ответ на вопрос, что такое материал, также далеко не прост. При этом часто не делают различие между понятиями «материал», «вещество», хотя очевидно, что это не одно и то же.

Во-вторых, состав вещества и материала в целом далеко не всегда определяет структуру и свойства материала.

Понятие материал является производным от главного естественного фундаментального понятия окружающего нас мира — материи. В рамках парадигмы многоуровневой организации материи вещество подразделяется на целый ряд уровней и подуровней (физические: элементарные и атомные вещества химические: молекулярные, надмолекулярные и немолекулярные вещества, биологические и т.д.).

Поэтому в широком смысле слова материал — это любой вид (форма или уровень организации вещества) материи, который может быть использован или используется природой или человеком для получения других его разновидностей.

Однако сегодня необходимо понимать разницу между понятием «вещество» и «материал», так как исследование различных конкретных видов веществ (их структуры, свойств и превращений) составляет специфику предмета таких естественных наук, как физика (элементарные и атомные вещества) и химия (химические молекулярные и немолекулярные вещества), и т.д. В известной триаде «вещество — материал — изделие (конструкция)» разница между материалом и изделием достаточно очевидна: из материала изготавливается конечная вещь в виде изделия для конкретного практического использования в повседневной жизни в течение определенного срока его эксплуатации. А вот разницу между веществом и материалом можно установить на уровне оценки реальной практической (чаще всего технической) значимости первого (когда вещество становится реальным материалом для изготовления чего-либо) и возможности изготовления из данного вещества конкретного изделия и конструкции.

В более узком, практическом (прикладном), смысле материал — это тот вид вещества или совокупность нескольких его типов в виде одной материальной системы (материального тела), который наиболее востребован в науке, технике и быту для производства изделий, конструкций и жизнеобеспечения в целом. Отсюда очевидно, что не любая разновидность вещества и даже химического вещества может служить материалом, применяемым на практике.

Предмет же материаловедения раскрывается через универсальную методологическую цепочку: «состав и тип связи элементов — специфика многоуровневой структуры — свойства», характеризующую специфику материала и систему базовых (основных и производных) понятий материаловедения. Давая современное определение (дефиницию) материаловедения и учитывая, что состав не определяет однозначно структуру материала, необходимо учесть в нем главенствующее влияние на структуру и на конечные свойства материала природы взаимодействия элементов, его составляющих. В результате материаловедение — это наука о связи состава и природы (а также типа) взаимодействия элементов (химической, физической, механической или смешанной), представляющих материал с его строением (структурой) и свойствами, определяющими области его практического применения. При этом понимается, что изучение этой связи в материаловедении направлено, прежде всего, на научное прогнозирование структуры и свойств конкретных материалов в целях эффективного их практического использования, а также методов и технологий управления их структурой и переработки в конкретные изделия.

Специфика предмета материаловедения основывается на единстве химической природы веществ, составляющих основной объект этой дисциплины — материал (металла или неметалла). Единство и специфика предмета материаловедения сегодня раскрывается при опоре на систему взаимосвязанных базисных инноваций: универсальной системы понятий материаловедения, унифицированной классификации уровней структурной организации материалов, единой модели химической связи элементов и системы химических связей и соединений (СХСС), образующих металлические и неметаллические материалы. Кроме того, эти инновации позволяют вскрыть причины различий структуры и свойств металлических и неметаллических материалов, закладывая основы универсальной методологии конструирования структуры материалов.

Сегодня под инновацией понимаются продукты творческого труда, имеющие законченный вид товара, готового к применению и распространению. А под базисными инновациями понимаются такие инновации, в основе которых лежат фундаментальные научные достижения, позволяющие создавать качественно новые системы (товары, технологии, учебники и т.д.) следующих поколений. По мнению автора, ярким примером базисной инновации может служить Периодическая система элементов Д. И. Менделеева, позволившая спрогнозировать наличие в природе и свойства атомов, не известных ученым в тот период времени. Кроме того, именно она позволила сегодня понять, что системы, объединяющие различные виды веществ и полей, должны иметь место для других

разновидностей вещества (химического, биологического, элементарного физического и т.д.). [14]

1.2 Существующие решения

На данный момент существует только одно программное решение, которое выполняет некоторые функции, необходимые для достижения цели данной работы: «Программа для расчета областей смесимости неорганических соединений в бинарной системе» В. А. Миллера [21].

Программа производит расчет энтропии смешения, теплоты смешения и константы Маделунга, строит купола распада и диаграммы состояния, позволяет работать с экспериментальными данными.

Функции выполняемые программой:

- вычисление различных параметров бинарной системы (значение энтропии и энтальпии смешения, теплоты смешения и константы Маделунга для различных значений замещения);
- построение таблиц по рассчитанным параметрам;
- построение таблицы энергии Гиббса;
- построение купола распада по вспомогательной таблице [1, с. 79] и нанесение на него экспериментальных данных;
- построение диаграммы состояния;
- сохранение таблиц (формат csv) и изображения купола распада (формат bmp).

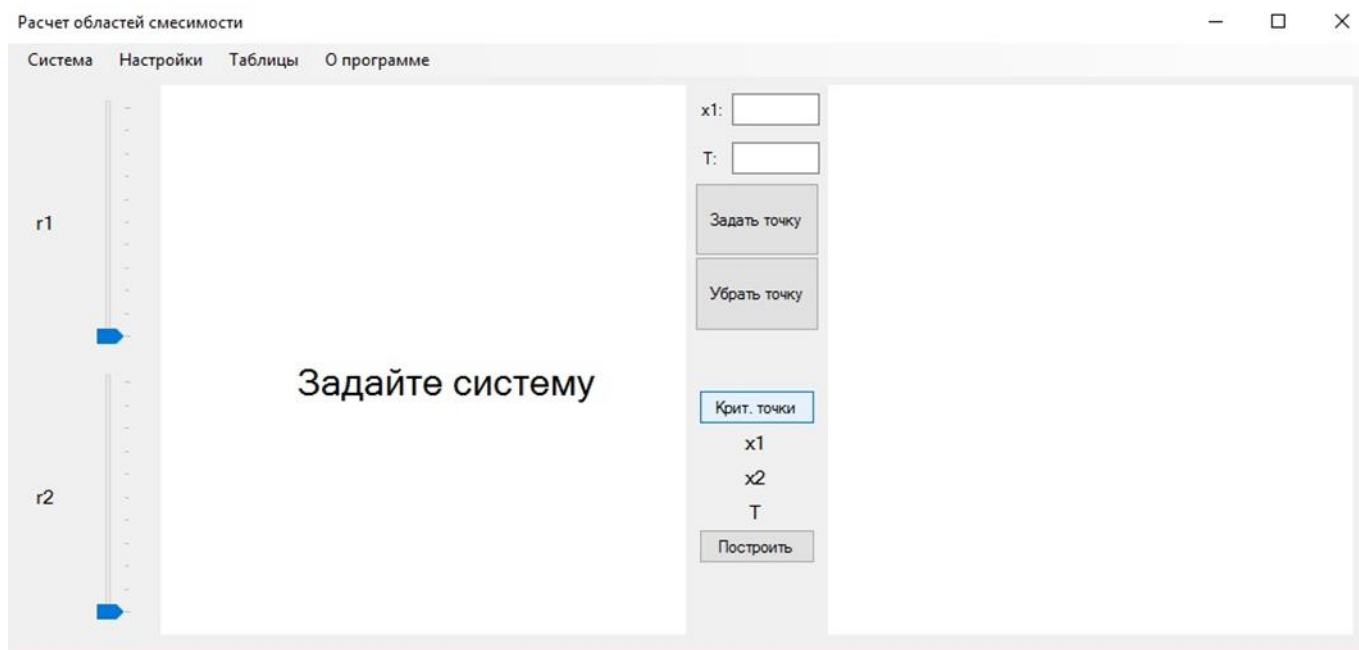


Рисунок 1 – Интерфейс программы для расчета областей смесимости неорганических соединений в бинарной системе

1.3 Обоснование разработки нового приложения

Программа для расчета областей смесимости неорганических соединений в бинарной системе имеет недостаточный функционал для определения границ растворимости твердых растворов. В первую очередь, это из-за несовпадения теоретических и экспериментальных расчётов. Разрабатываемое приложение должно будет изменять формульные коэффициенты для приближения результата к эксперименту. Также данное приложение может предоставлять справочные материалы по химическим элементам, соединениям и системам соединений.

1.4 Выводы по главе

В первой главе была рассмотрена предметная область данной работы и единственный существующий аналог. Также было приведено обоснование разработки нового приложения.

Глава 2. Используемые методы, модели и инструменты

В данной главе представлены формулы и зависимости, по которым производятся расчеты, а также описаны инструменты, используемые при разработке.

2.1 Теоретическая информация

Твердые растворы замещения – фазы переменного состава, в которых атомы (ионы) одного компонента замещают в структуре атомы (ионы) другого компонента, при этом соблюдаются правила образования непрерывных твердых растворов замещения:

- оба компонента должны принадлежать к одному и тому же или близким структурным типам (структурный фактор);
- разница атомных (ионных) радиусов замещаемого и замещающего атома (иона) в компонентах не должны превышать 15% (размерный фактор);
- у замещаемого и замещающего атома (иона) в компонентах должны быть близкие поляризационные свойства и величины электроотрицательности (разница не должна превышать 0.4 отн. ед.) или потенциалов ионизации (т.е. одинаковое строение электронных оболочек); одинаковый формальный заряд (фактор химической связи).

Равновесное состояние изоморфной смеси определяется стремлением к минимуму свободной энергии Гиббса:

$$\Delta G_{\text{см}} = \Delta H_{\text{см}} - T\Delta S_{\text{см}}, \text{ ккал/моль} \quad (1)$$

где $\Delta H_{\text{см}}$ и $\Delta S_{\text{см}}$ теплота и энтропия смешения.

В зависимости от соотношения $\Delta H_{\text{см}}$ и $T\Delta S_{\text{см}}$ возможна реализация трех видов диаграмм состояния:

$\Delta H_{\text{см}} > 0$, $\Delta H_{\text{см}} < T\Delta S_{\text{см}}$, $\Delta G_{\text{см}} < 0$ (случай 1) - непрерывные твердые растворы между компонентами (соединениями, фазами) при повышенных температурах и область несмесимости компонентов при низких температурах;

$\Delta H_{\text{см}} > 0$, $\Delta H_{\text{см}} > T\Delta S_{\text{см}}$, $\Delta G_{\text{см}} > 0$ (случай 2) – отсутствие взаимной растворимости компонентов в твердом состоянии;

$\Delta H_{\text{см}} > 0$, кривая $\Delta G_{\text{см}}$ с двумя перегибами ($\Delta G_{\text{см}} < 0$, $\Delta G_{\text{см}} > 0$, $\Delta G_{\text{см}} < 0$) (случай 3) свидетельствует о распаде твердого раствора при температуре эвтектики, т.е. образуются твердые растворы с ограниченной растворимостью на основе одного и другого компонента.

Значения $\Delta S_{\text{см}}$ и $\Delta H_{\text{см}}$ определяются по формулам, приведенным в работах [1, 2] для изовалентных систем AX-BX (в нашем случае) или AX-AY (A, B и X, Y –

изовалентные замещаемые/замещающие катионы или анионы в компонентах системы) при условии выполнения правила Вегарда:

$$\Delta S_{\text{см}} = \Delta S_{\text{конф}} + \Delta S_{\text{кол}} = -kN(x_1 \ln x_1 + x_2 \ln x_2) + 2.725x_1x_2(\Delta R/R_{\text{мин}}), \text{ ккал/моль} \quad (2)$$

где $\Delta S_{\text{конф}}$ и $\Delta S_{\text{кол}}$ конфигурационная и колебательная энтропия, k и N – постоянная Больцмана и число Авогадро (kN – универсальная газовая постоянная), x_1 и x_2 – содержание компонентов в системе: $x_1 + x_2 = 1$.

$$\Delta H_{\text{см}} = x_1x_2[(332A/R)\Delta\varepsilon^2 + \text{cmnz}_{\text{AZX}}(\Delta R/R)^2], \text{ ккал/моль} \quad (3)$$

В уравнении (3)

- размерный фактор:

$$R = x_1R_1 + x_2R_2 + R_{\text{общ}}, \quad (4)$$

где R – межатомное расстояние катион–анион: R_1 и R_2 – радиусы замещающего/замещаемого иона, $R_{\text{общ}}$ – радиус общей структурной единицы в компонентах системы [3]

или

$$R = x_1R_1 + x_2R_2, \quad (5)$$

где R – межатомное расстояние катион–анион: R_1 и R_2 – межатомные расстояния первого и второго компонентов системы;

- структурный фактор:

A – константа Маделунга, m – сумма формульных коэффициентов в компонентах (соединениях), n – координационное число (одинаковое для замещаемого и замещающего иона);

- фактор химической связи:

z_A и z_X – формальные заряды катиона (A) и аниона (X) в компонентах системы, c – эмпирический коэффициент, зависящий от степени ионности системы: $c = 33.33\varepsilon + 8.83$ (зависимость получена на основании данных, приведенных в [1,2])

Степень ионности оценивается по формуле:

$$\varepsilon = 1 - z/n \times \exp(-\Delta\chi^2 \times 0.25), \quad (6)$$

где z и n – соответственно формальный заряд и координационное число, одинаковые для замещаемого/замещающего иона, $\Delta\chi = |\chi_A - \chi_X|$ – разность величин электроотрицательностей катиона (A) и аниона (X) для каждого компонента системы, $\Delta\varepsilon = |\varepsilon_1 - \varepsilon_2|$ – разность степеней ионности первого и второго компонентов системы.

Энергия смешения (энергия взаимообмена, параметр взаимодействия), которая

вычисляется в субрегулярном приближении:

$$Q_{\max} = \text{cmnz}_{AZX}(\Delta R/R_{\min})^2, \text{ ккал/моль} \quad (7)$$

связана с критической температурой купола распада ($T_{\text{кр}}$, К) простым соотношением:

$$T_{\text{кр}} = Q_{\max}/2kN, \text{ К} \quad (8)$$

2.2 Используемые инструменты

2.2.1. Язык программирования C#

C# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML [22].

2.2.2. Microsoft Visual Studio

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения

функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server) [22].

2.2.3. Windows Presentation Foundation

Windows Presentation Foundation (WPF) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML [22].

2.2.4. Xml

XML (eXtensible Markup Language) — расширяемый язык разметки. Рекомендован Консорциумом Всемирной паутины (W3C). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов как программами так и человеком, с акцентом на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Расширение XML — это конкретная грамматика, созданная на базе XML и представленная словарём тегов и их атрибутов, а также набором правил, определяющих, какие атрибуты и элементы могут входить в состав других элементов. Сочетание простого формального синтаксиса, удобства для человека, расширяемости, а также базирование на кодировках Юникод для представления содержания документов привело к широкому использованию как, собственно, XML, так и множества производных специализированных языков на базе XML в самых разнообразных программных средствах [22].

2.2.5. Itextsharp

iText — это библиотека классов для генерации, анализа и изменения документов в форматах Portable Document Format (PDF), а также XML, HTML и RTF.

Начиная с версии 5 библиотека имеет двойную лицензию — AGPL (вынуждающую предоставлять пользователям полный исходный текст приложения) либо коммерческую.

2.3 Выводы по главе

В данной главе были рассмотрены инструменты, которые использовались для разработки приложения, и формульные зависимости, используемые в программе.

Глава 3. Разработка и архитектура приложения

Эта глава состоит из описания классов и особенностей реализации программы. Также в ней описывается, как проходило тестирование её работоспособности.

3.1 Особенности реализации программы

Исходные данные с характеристиками отдельных атомов (ионов) и соединений, необходимые в процессе вычислений, выбираются из информационно-расчетной системы «Интерактивная таблица Д. И. Менделеева» [15]. Представляемая в тексте программа и названная информационно-расчетная система (ИРС «Таблица Менделеева») входит в единый комплекс под названием «Программа определения границ растворимости твердых растворов в зависимости от температуры».

Конечному пользователю указанный комплекс позволяет выполнять построение трех графиков: купола распада, термодинамической функции смещения и свободной энергии Гиббса. Кроме того, пользователь имеет возможность в диалоговом режиме наносить на эти графики экспериментальные данные (в виде точек) и выполнять приближение графиков (теплоты смещения и купола распада) к этим данным вариацией параметров (радиусов и электроотрицательностей элементов).

Основой для построения в программном комплексе названных графиков послужили (приводимые в тексте) аналитические выражения, описывающие форму купола распада, термодинамическую функцию смещения и функцию изменения свободной энергии Гиббса. Особенностью существующих в литературе аналитических выражений, которые можно использовать при разработке программных средств, является наличие в этих выражениях большого количества величин, которые необходимо выбирать из справочной литературы и из публикуемых результатов экспериментальных исследований.

Для величин, значения которых уже вошли в справочники, удобно иметь специализированную базу данных, позволяющую избавить пользователя от поиска и ручного ввода нужных для конкретного расчета значений. Специализированную базу данных со справочными величинами, необходимыми для решения задач данного проекта, возможно создать автономно. Однако ранее была создана достаточно универсальная информационно-расчетная система (Интерактивная таблица Менделеева), позволяющая оперативно получать большинство значений параметров для тех функций, на основе которых выполняются расчеты и строятся графики описываемого проекта. Данная ИРС была разработана для того, чтобы можно было собрать все необходимые для исследования данные в одном месте и не нагружать пользователя лишними поисками информации в разных источниках. Она представляет собой набор таблиц с текстовыми значениями данных и/или аналитическими выражениями. ИРС достаточно универсальна и

может заполняться информацией из разных предметных областей материаловедения. Тем самым на основе ИРС может быть построен специализированный программный комплекс, одним из которых является примененная в данной работе «Программа определения границ растворимости твердых растворов в зависимости от температуры». Пользователь при работе с названной программой имеет возможность дополнять или заменять значения сохраняемых констант, уточнять как запись аналитических выражений, так и числовые значения коэффициентов, входящих в них. Рассмотрим, что дает такая возможность.

Особенностью ИРС «Таблица Менделеева» является хранение в ее таблицах как точных значений данных, так и данных, полученных или получаемых экспериментально, точность которых нельзя гарантировать. Аналогичная ситуация и с сохраняемыми в таблицах ИРС записями функций. Они могут быть взяты из публикаций, где не указаны точности представления коэффициентов, входящих в аналитические выражения.

В тех случаях, когда планируется активное применение ИРС «Таблица Менделеева» для многоразового решения определенного круга задач в конкретной области материаловедения, целесообразно перед внедрением построенного на основе ИРС программного комплекса провести исследование и выполнить его «настройку» с помощью имеющихся экспериментальных данных, не входящих непосредственно в таблицы ИРС. Особенно важно – выполнить верификацию существующих (опубликованных) аналитических выражений, теоретически описывающих те или иные функциональные зависимости выбранной предметной области.

В данной работе: выполнялась верификация теоретических соотношений, описывающих термодинамическую функцию смещения (3) и аналитических выражений, позволяющих построить купола распада. Верификация выполнялась за счет решения задач аппроксимации вводимых пользователем экспериментальных значений указанными теоретическими выражениями. При аппроксимации функции смещения (3) меняются R_1 , R_2 , $R_{\text{общ}}$ (4) и χ_1 , χ_2 , $\chi_{\text{общ}}$ (6), а при уточнении аналитических зависимостей купола распада варьируются значения R_1 , R_2 , $R_{\text{общ}}$ и χ_1 , $\chi_{\text{общ}}$. Начальные значения всех перечисленных параметров выбираются из Интерактивной таблицы Менделеева. Результаты решения аппроксимационных задач – графики соответствующих функциональных зависимостей и уточненные значения параметров, входящих в их аналитические описания.

При исследовании графика термодинамической функции смещения критерием оптимизации служит среднеквадратичное отклонение аналитического описания функции от набора экспериментальных точек.

Для купола распада вместо аналитических выражений, теоретически описывающих графики, использовались опубликованные в литературе номограммы [12]. Для

использования номограмм в программе они были переведены в наборы точек, сохраненных в виде xml-файла, который непосредственно включен в исполнимый модуль. Задача аппроксимации в конкретном случае требует задания не только набора экспериментальных точек, но и критической температуры. По значению критической температуры из файла с данными о номограмме выбирается нужные данные, приведенные к размерности набора экспериментальных точек. Далее из номограммы выбирается кривая, точки которой наименее удалены от набора экспериментальных точек. Критерий оценки удаленности - сумма расстояний между всеми точками эксперимента и точками кривой, выбранной из номограммы. Выбранная из номограммы кривая позволяет получить требуемое отношение радиусов. Следующий шаг – поиск набора значений параметров (R_1 , R_2 , $R_{\text{общ}}$ и χ_1 , $\chi_{\text{общ}}$), при котором выполняются следующие соотношения:

1. $x - 0.025 \leq \Delta R / R_{\min} < x + 0.025$, где x – отношение, полученное из номограмм;
2. $i^* \in [i - 0.02, i + 0.02]$, где i – начальные значения параметров, i^* - конечные;
3. $|T_{\text{кр}} - T^*| \leq 10^\circ$, где T^* - водимая пользователем критическая температура.

В результате сеансов работы с программой пользователь может построить графики купола распада и термодинамической функции смешения и получить новые значения параметров, соответствующих построенным графикам.

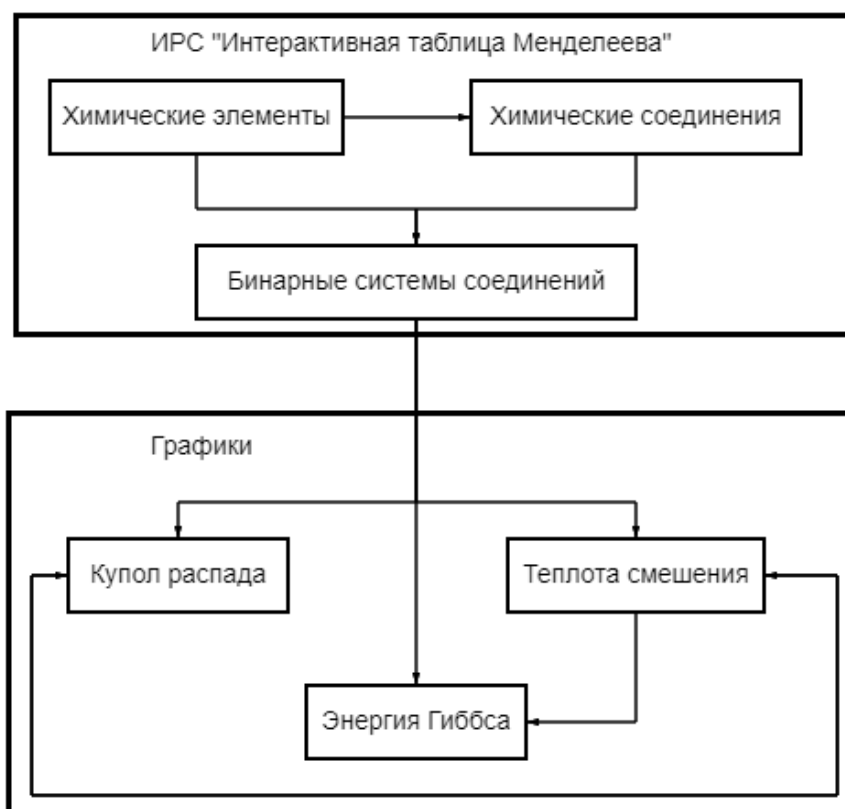


Рисунок 2 – Схема взаимодействия компонентов программы

3.2 Описание классов

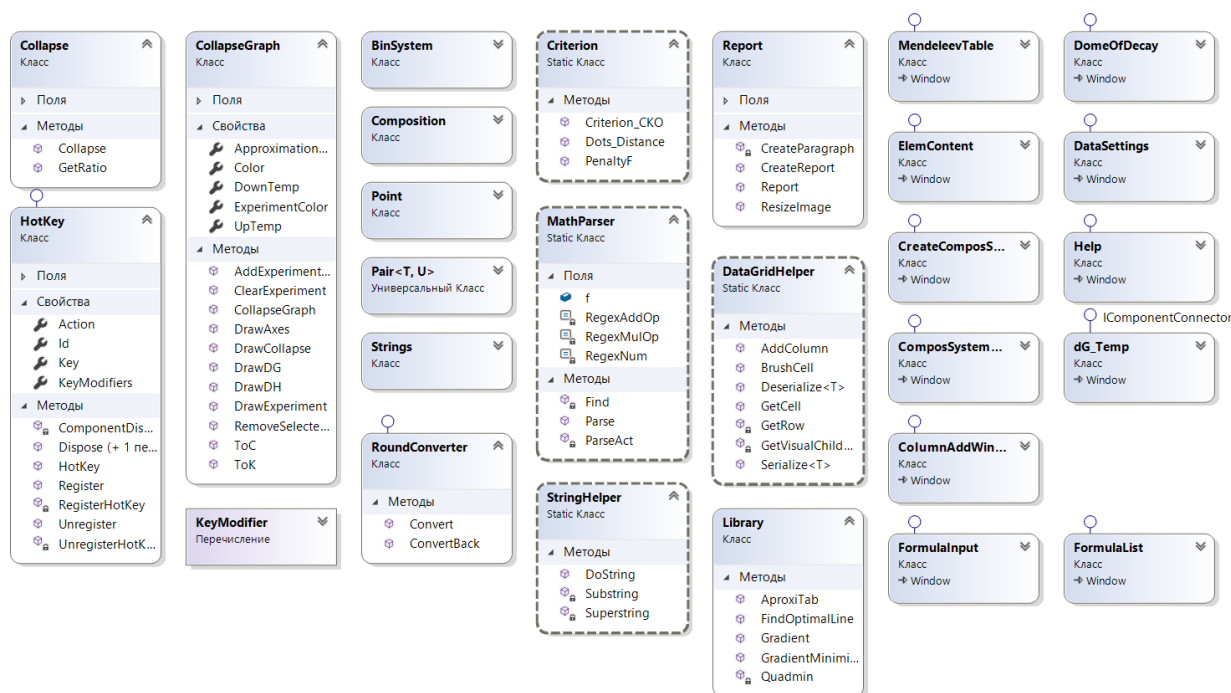


Рисунок 3 – Диаграмма классов

3.2.1 Интерактивная таблица Менделеева

3.2.1.1 Окна программы

3.2.1.1.1 ColumnAddWindow

Данный класс описывает окно добавления нового столбца в таблицу свойств (рис. 4).

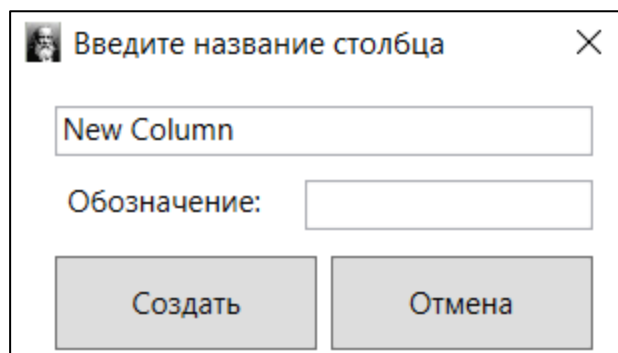


Рисунок 4 – Окно добавления нового столбца

В данном окне пользователь вводит название столбца и обозначение, с помощью которого значения ячеек, соответствующих данному столбцу, можно использовать в формулах и при построении графиков.

3.2.1.1.2 ComposSystemContent

Данный класс описывает окно свойств химического соединения или бинарной системы соединений и представляет собой редактируемую таблицу этих свойств (рис. 5).

Таблица системы NaCl-AgCl									
X, x	$R=x \cdot R(\text{Na}) + (1-x) \cdot R(\text{Ag})$	$dR/R = \text{abs}(R(\text{Na}) - R(\text{Ag})) / R$	$dS_{\text{конф}} = -1,9871 \cdot \ln(R(\text{Na}) / R(\text{Ag}))$	$dS_{\text{кол}} = x \cdot (1-x) \cdot 2 \cdot \ln(2)$	$dE = \text{abs}(E(\text{NaCl}) - E(\text{AgCl}))$	$dS = dS_{\text{конф}}(\text{NaCl}) + dS_{\text{кол}}(\text{AgCl})$	$dH = x \cdot (1-x) \cdot \Delta H_f$	$m = x \cdot 2,774 + (1-x) \cdot 2,774$	$dr/m = 0,045 / m$
1	2,8300	0,0459	0,0000	0,0000	0,0610	0,0000	0,0000	2,7740	0,0162
0,9	2,8430	0,0457	0,0006	0,0113	0,0610	0,0119	0,1359	2,7785	0,0162
0,8	2,8560	0,0455	0,0010	0,0200	0,0610	0,0210	0,2400	2,7830	0,0162
0,7	2,8690	0,0453	0,0012	0,0263	0,0610	0,0275	0,3129	2,7875	0,0161
0,6	2,8820	0,0451	0,0013	0,0300	0,0610	0,0313	0,3553	2,7920	0,0161
0,5	2,8950	0,0449	0,0014	0,0313	0,0610	0,0327	0,3676	2,7965	0,0161
0,4	2,9080	0,0447	0,0013	0,0300	0,0610	0,0313	0,3505	2,8010	0,0161
0,3	2,9210	0,0445	0,0012	0,0263	0,0610	0,0275	0,3047	2,8055	0,0160
0,2	2,9340	0,0443	0,0010	0,0200	0,0610	0,0210	0,2306	2,8100	0,0160
0,1	2,9470	0,0441	0,0006	0,0113	0,0610	0,0119	0,1288	2,8145	0,0160
0	2,9600	0,0439	0,0000	0,0000	0,0610	0,0000	0,0000	2,8190	0,0160

Редактировать таблицу

Купол распада

Рисунок 5 – Окно свойств химического соединения/системы соединений

В данном окне пользователь может просматривать и редактировать свойства химического соединения/системы соединений. Для того, чтобы получить возможность изменять таблицу необходимо нажать на кнопку «Редактировать таблицу». После этого можно изменять содержимое ячеек, добавлять столбцы и строки нажав на соответствующие кнопки. Столбцы могут представлять собой обычные свойства, имеющие название, обозначение и конкретное значение, или расчетные, имеющие обозначение и формулу расчета значений. Также, по нажатию на кнопку «Удалить столбец» можно удалить крайний справа столбец, а по нажатию кнопки «Удалить выделенные строки» - удалить все строки, выделенные мышью в таблице. Кнопка «Рассчитать и сохранить» пересчитывает все формульные свойства с точностью до 4 знаков после запятой, сохраняет значения в файл "Compositions.xml" для химических соединений и "BinarySystems.xml" для систем соединений и переводит таблицу в формат readonly. Переход к построению графиков происходит по нажатию на кнопку «Купол распада».

3.2.1.1.3 CreateComposSystem

Данный класс описывает окно выбора или создания химического соединения или системы соединений (рис. 6).

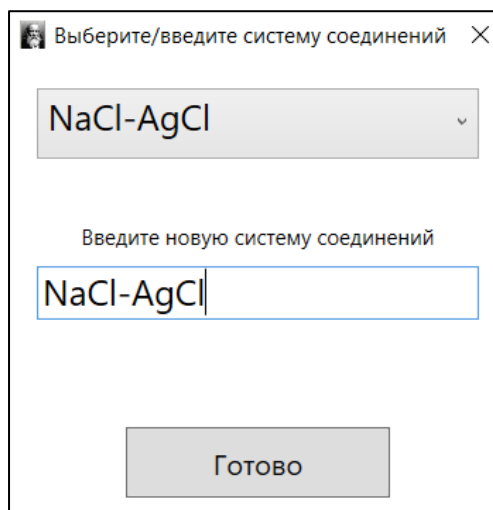
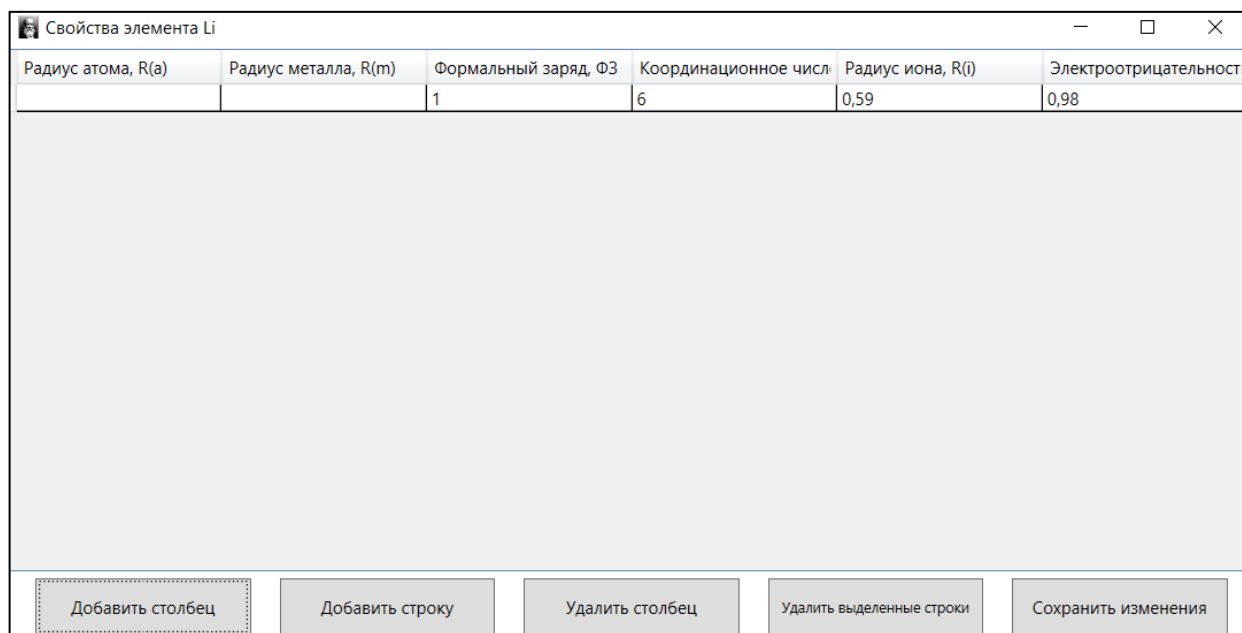


Рисунок 6 – Окно выбора химического соединения/системы соединений

В данном окне пользователь может выбрать из выпадающего списка химическое соединение/систему соединений и перейти к таблице его/её свойств или создать новое/ую, записав обозначение в текстовое поле, и перейти к пустой таблице свойств для её заполнения.

3.2.1.1.4 ElemContent

Данный класс описывает окно свойств химического элемента и представляет собой редактируемую таблицу этих свойств (рис. 7).



Радиус атома, R(a)	Радиус металла, R(m)	Формальный заряд, ФЗ	Координационное числ	Радиус иона, R(i)	Электроотрицательность
		1	6	0,59	0,98

Добавить столбец Добавить строку Удалить столбец Удалить выделенные строки Сохранить изменения

Рисунок 7 – Окно свойств химического элемента

В данном окне пользователь может просматривать и редактировать свойства химического элемента. Для того, чтобы получить возможность изменять таблицу необходимо нажать на кнопку «Редактировать таблицу». После этого можно изменять содержимое ячеек, добавлять столбцы и строки нажав на соответствующие кнопки. Столбцы представляют собой свойства, имеющие название, обозначение и

конкретное значение. Также, по нажатию на кнопку «Удалить столбец» можно удалить крайний справа столбец, а по нажатию кнопки «Удалить выделенные строки» - удалить все строки, выделенные мышью в таблице. Кнопка «Сохранить изменения» сохраняет значения в файл "Elems.xml" и переводит таблицу в формат readonly.

3.2.1.1.5 FormulaInput

Данный класс описывает окно добавления формулы в таблицу свойств (рис. 8).

Рисунок 8 – Окно добавления формулы

В данном окне пользователь вводит формулу и обозначение, с помощью которого значения ячеек, соответствующих данному столбцу, можно использовать в других формулах и при построении графиков. Формулы могут состоять из:

- a. Вещественных чисел, в виде десятичных дробей;
- b. Конструкций, по которым ищется значение в таблицах:
 - $\langle \text{обозначение искомого свойства} \rangle \{ \langle \text{элемент/соединение/система} \rangle \}$ – получает значение искомого свойства элемента/соединения/системы соединений находящегося в первой строке таблицы свойств;
 - $\langle \text{обозначение искомого свойства} \rangle \{ \langle \text{вспомогательное свойство, по которому ищем искомое} \rangle \{ \langle \text{значение вспомогательного свойства} \rangle \} \langle \text{элемент/соединение/система} \rangle \}$ - получает значение искомого свойства элемента/соединения/системы соединений соответствующего значению вспомогательного свойства из таблицы свойств);
 - $\langle \text{элемент/соединение/система} \rangle \{ \langle \text{номер строки} \rangle; \langle \text{номер столбца} \rangle \}$ – получает значение ячейки таблицы свойств элемента/соединения/системы соединений на пересечении заданной строки и столбца(нумерация строк и столбцов начинается с нуля);
 - $\langle \text{элемент/соединение/система} \rangle \{ \langle \text{номер строки} \rangle; \langle \text{обозначение искомого свойства} \rangle \}$ – получает значение искомого свойства элемента/соединения/системы соединений на заданной строке(нумерация

строк начинается с нуля);

- с. Символов математических операций("+", "-", "*", "/", "%" - остаток от деления);
- d. Скобок, обозначающих приоритет математических действий("(" и ")");
- e. Функций, описанные ниже(если аргументов больше одного - следует их разделять символом ";". Пример: min(32; sin(4)).

Список функций, которые поддерживаются программой:

- sin(a) - вычисляет синус a;
- cos(a) - вычисляет косинус a;
- tan(a) - вычисляет тангенс a;
- ctan(a) - вычисляет котангенс a;
- abs(a) - вычисляет модуль a;
- ln(a) - вычисляет натуральный логарифм a;
- exp(a) - вычисляет экспоненту степени a;
- min(a; b) - находит минимум из a и b;
- max(a; b) - находит максимум из a и b;
- pow(a; b) - возводит a в степень b.

Также, при нажатии на кнопку «Список формул» откроется окно с таблицей формул и обозначений, которые уже присутствуют в различных таблицах программы.

3.2.1.1.6 FormulaList

Данный класс описывает окно, которое представляет собой таблицу формул и обозначений, используемых в таблицах свойств (рис. 9).

Обозначение	Название/формула
R(a)	Радиус атома
R(m)	Радиус металла
ФЗ	Формальный заряд
KЧ	Координационное число
R(i)	Радиус иона
x	Электроотрицательность
e	$1 - \Phi_3(\text{Na}) / KЧ(\text{Na}) * \exp(-0.25 * \text{pow}(x(\text{Na}) - x(\text{Cl}); 2))$
e	$1 - \Phi_3(\text{Ag}) / KЧ(\text{Ag}) * \exp(-0.25 * \text{pow}(x(\text{Ag}) - x(\text{Cl}); 2))$
e	$1 - \Phi_3(\text{Si}) / KЧ(\text{Si}) * \exp(-0.25 * \text{pow}(x(\text{Si}) - x(\text{O}); 2))$
e	$1 - \Phi_3(\text{Ti}) / KЧ(\text{Ti}) * \exp(-0.25 * \text{pow}(x(\text{Ti}) - x(\text{O}); 2))$
e	$1 - \Phi_3(\text{Sn}) / KЧ(\text{Sn}) * \exp(-0.25 * \text{pow}(x(\text{Sn}) - x(\text{O}); 2))$
e	$1 - \Phi_3(\text{Zr}) / KЧ(\text{Zr}) * \exp(-0.25 * \text{pow}(x(\text{Zr}) - x(\text{O}); 2))$
x	X
R	$x * Ti(1;R(i)) + (1 - x) * R(i)(Zr) + R(i)(O)$
dR/R	$\text{abs}(Ti(1;R(i)) - R(i)(Zr)) / R(TiO_2-ZrO_2)$
dСконф	$-1.98716 * 0.001 * (x * \ln(x) + (1-x) * \ln(1-x))$
dСкол	$x * (1-x)^{2.725} * (\text{abs}(Ti(1;R(i)) - R(i)(Zr)) / (\min(Ti(1;R(i)); R(i)(Zr)) + R(i)(O)$
de	$\text{abs}(TiO_2(1;e) - e(ZrO_2))$
dH	$x * (1 - x) * (332 * 4.8 * \text{pow}(\text{de}(TiO_2-ZrO_2); 2) / R(TiO_2-ZrO_2) + 25 * 3$
dS	$d\text{Сконф}(TiO_2-ZrO_2) + d\text{Скол}(TiO_2-ZrO_2)$

Рисунок 9 – Окно «Список формул»

В данном окне пользователь может посмотреть и скопировать обозначения

различных свойств, их названия, а также формулы, которые были использованы в различных таблицах свойств программы.

3.2.1.1.7 MendeleevTable

Данный класс описывает стартовое окно программы и представляет собой специализированную таблицу Менделеева (рис. 10).

Таблица Менделеева

Работа с соединением

Работа с системой соединений

Справка

Ia	IIa	IIIa	IVa	Va	VIa	VIIa	VIIIa	I6	II6	III6	IV6	V6	VI6	VII6	VIII6		
														H	He		
Li	Be									B	C	N	O	F	Ne		
Na	Mg									Al	Si	P	S	Cl	Ar		
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba	*La	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra	**Ac															I газ
T-Металлы										B1-Металлы				B2-Металлы			
*Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu	R-Лантаноиды			
**	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	-	TR-Актиноиды			

Рисунок 10 – Окно «Таблица Менделеева»

В данном окне пользователь может перейти к таблице свойств любого представленного в таблице химического элемента по двойному нажатию на него левой кнопкой мыши. Нажав на пункт меню «Работа с соединениями» или «Работа с системой соединений» можно перейти в окно выбора соединения или системы соответственно. Также можно перейти в окно справки нажав на соответствующий пункт меню или клавишу F1.

3.2.1.2 Composition

Данный класс создан для хранения информации о химическом элементе или соединении (табл. 1).

Таблица 1 – Класс Composition

Имя	Модификатор доступа	Тип	Описание
Name	Public get Public set	String	Название соединения(элемента)
Properties	Public get Public set	System.Collections.Generic.List< Pair< Pair<String, String>, System.Collections.Generic.List<String> >>	Лист свойств и соответствующих им значений
DataTable	Public get Public set	System.Data.DataTable	Таблица свойств элементов

3.2.1.3 DataGridViewHelper

Данный класс является вспомогательным для работы с DataGridView.

Он включает в себя методы поиска строки и ячейки по ее координатам в таблице:

```

    /// <summary>
    /// Получает строку из DataGridView
    /// </summary>
    /// <param name="index">номер строки</param>
    /// <param name="dg">DataGridView</param>
    /// <returns>Строку DataGridView</returns>
    private static DataGridView GetRow(int index, DataGridView dg)
    {
        try
        {
            DataGridView row =
(DataGridView) dg.ItemContainerGenerator.ContainerFromIndex(index);
            if (row == null)
            {
                dg.UpdateLayout();
                dg.ScrollIntoView(dg.Items[index]);
                row =
(DataGridView) dg.ItemContainerGenerator.ContainerFromIndex(index);
            }
            return row;
        }
        catch (Exception ex)
        {
            MessageBox.Show("Ошибка получения данных из таблицы функция
(GetRow)!\n\n" + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return null;
        }
    }

    /// <summary>
    /// Получает ячейку из DataGridView
    /// </summary>
    /// <param name="row">номер строки</param>
    /// <param name="column">номер столбца</param>
    /// <param name="dg">DataGridView</param>
    /// <returns>Ячейку DataGridView</returns>
    public static DataGridViewCell GetCell(int row, int column, DataGridView dg)
    {
        try
        {
            DataGridView rowContainer = GetRow(row, dg);

```

```
        if (rowContainer != null)
        {
            System.Windows.Controls.Primitives.DataGridCellsPresenter
presenter =
GetVisualChild<System.Windows.Controls.Primitives.DataGridCellsPresenter>(rowContainer
);

            DataGridCell cell =
(DataGridCell)presenter.ItemContainerGenerator.ContainerFromIndex(column);
            if (cell == null)
            {
                dg.ScrollIntoView(rowContainer, dg.Columns[column]);
                cell =
(DataGridCell)presenter.ItemContainerGenerator.ContainerFromIndex(column);
            }
            return cell;
        }
        return null;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Ошибка получения данных из таблицы функция
(GetCell)!\n\n" + ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
        return null;
    }
}
```

Также, данный класс включает в себя некоторые методы манипуляций с данными DataGrid. Одним из таких методов является BrushCell(int row, int column, Brush color, DataGrid dg), который перекрашивает ячейку таблицы *dg* с заданными координатами *row* и *column* в цвет *color*. Этот метод используется при начальном заполнении таблицы Менделеева (рис. 10) Другая функция позволяет добавлять новые столбцы в таблицы свойств:

```
/// <summary>
/// Добавляет текстовый столбец в таблицу
/// </summary>
/// <param name="dg">DataGrid</param>
/// <param name="dat">таблица данных</param>
/// <param name="f">флаг: true - соединение(элемент), false - система</param>
/// <returns>Можно ли удалить столбцы?</returns>
public static bool AddColumn(ref DataGrid dg, ref DataTable dat, bool f =
true)
{
    ColumnAddWindow form = new ColumnAddWindow();
    form.ShowDialog();
    try
    {
        if (form.name != "")
        {
            if (form.symbol != "" && form.symbol != " ")
                foreach (DataColumn v in dat.Columns)
                    if (v.Caption == form.symbol)
                        throw new DuplicateNameException();

            DataColumn col = new DataColumn(form.name) { Caption = form.symbol
};

            dat.Columns.Add(col);
            dg.Columns.Add(new DataGridTextColumn()
            {
                Header = (form.symbol == "" || form.symbol == " ") ? form.name
: form.name + ", " + form.symbol,
                Binding = new Binding "[" + dg.Columns.Count + "]"
            });
        }
    }
    catch (DuplicateNameException)
    {
        29
    }
}
```

```
{
    MessageBox.Show("Столбец с данным именем(обозначением) уже принадлежит
данной таблице!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}

if (f && dat.Columns.Count > 0 || !f && dat.Columns.Count > 1)
    return true;
return false;
}
```

Данный класс также включает в себя методы сериализации и десериализации листов в xml файл, для сохранения данных о химических элементах, соединениях и системах соединений.

3.2.1.4 MathParser

Данный класс создан для вычисления значения формул, заданных пользователем в таблицах свойств химического соединений и систем соединений.

Основной метод данного класса Parse(string *str*, ref DataTable *dat*, int *u*) выполняет парсинг строки *str*, находящейся в ячейке таблицы *dat* строки под номером *u*. Парсинг выполняется последовательно в несколько этапов:

1. Парсинг математических функций

```
string[] func = { "sin", "cos", "tan", "ctan", "abs", "exp", "ln", "min",
"max", "pow" };
for (i = 0; i < func.Length; i++)
{
    k = str.IndexOf(func[i]);
    if (k >= 0)
    {
        left = str.Substring(0, k);
        k += func[i].Length;
        j = 0;
        bool f = false;
        right = "";
        inner = "";
        while (k < str.Length)
        {
            if (!f && str[k] == '(' && k < str.Length)
            {
                j++;
                if (j == 1)
                {
                    left += right;
                    right = "";
                    k++;
                }
            }

            right += str[k];

            if (!f && k < str.Length - 1 && str[k + 1] == ')')
            {
                j--;
                if (j == 0)
                {
                    inner += right;
                    right = "";
                    f = true;
                    k++;
                }
            }
        }
    }
}
```

```
        k++;
    }

    switch (i)
    {
        case 0:
            return Parse(left + Math.Sin(Parse(inner, ref dat, u)) +
right, ref dat, u);

        case 1:
            return Parse(left + Math.Cos(Parse(inner, ref dat, u)) +
right, ref dat, u);

        case 2:
            return Parse(left + Math.Tan(Parse(inner, ref dat, u)) +
right, ref dat, u);

        case 3:
            return Parse(left + 1.0 / Math.Tan(Parse(inner, ref dat,
u)) + right, ref dat, u);

        case 4:
            return Parse(left + Math.Abs(Parse(inner, ref dat, u)) +
right, ref dat, u);

        case 5:
            return Parse(left + Math.Exp(Parse(inner, ref dat, u)) +
right, ref dat, u);

        case 6:
            return Parse(left + Math.Log(Parse(inner, ref dat, u)) +
right, ref dat, u);

        case 7:
            int a = 0, b = 0, v = 0, g = 0;
            string inleft = "";
            while (inner[g] != ';' || a != 0 || b != 0 || v != 0)
            {
                inleft += inner[g];
                if (inner[g] == '(')
                    a++;
                else if (inner[g] == ')')
                    a--;
                else if (inner[g] == '[')
                    b++;
                else if (inner[g] == ']')
                    b--;
                else if (inner[g] == '{')
                    v++;
                else if (inner[g] == '}')
                    v--;
                g++;
            }

            string inright = inner.Substring(g + 1);
            return Parse(left + Math.Min(Parse(inleft, ref dat, u),
Parse(inright, ref dat, u)) + right, ref dat, u);

        case 8:
            a = 0; b = 0; v = 0; g = 0;
            inleft = "";
            while (inner[g] != ';' || a != 0 || b != 0 || v != 0)
            {
                inleft += inner[g];
                if (inner[g] == '(')
                    a++;
                else if (inner[g] == ')')
                    a--;
                else if (inner[g] == '[')
                    b++;
                else if (inner[g] == ']')
                    b--;
```

```

        b--;
        else if (inner[g] == '{')
            v++;
        else if (inner[g] == '}')
            v--;
        g++;
    }

    inright = inner.Substring(g + 1);
    return Parse(left + Math.Max(Parse(inleft, ref dat, u),
Parse(inright, ref dat, u)) + right, ref dat, u);

case 9:
    a = 0; b = 0; v = 0; g = 0;
    inleft = "";
    while (inner[g] != ';' || a != 0 || b != 0 || v != 0)
    {
        inleft += inner[g];
        if (inner[g] == '(')
            a++;
        else if (inner[g] == ')')
            a--;
        else if (inner[g] == '[')
            b++;
        else if (inner[g] == ']')
            b--;
        else if (inner[g] == '{')
            v++;
        else if (inner[g] == '}')
            v--;
        g++;
    }

    inright = inner.Substring(g + 1);
    return Parse(left + Math.Pow(Parse(inleft, ref dat, u),
Parse(inright, ref dat, u)) + right, ref dat, u);
    }
}
}

```

2. Парсинг символа «X»

```

Match matchFuncx = Regex.Match(str, @"(x)");
if (matchFuncx.Groups.Count > 1)
{
    left = str.Substring(0, matchFuncx.Index);
    right = str.Substring(matchFuncx.Index + matchFuncx.Length);
    return Parse(left + double.Parse(dat.Rows[u]["X"].ToString()) + right,
ref dat, u);
}

```

3. Парсинг бесконечности

```

Match matchinf = Regex.Match(str, @"(∞)");
if (matchinf.Groups.Count > 1)
{
    left = str.Substring(0, matchinf.Index);
    right = str.Substring(matchinf.Index + matchinf.Length);
    return Parse(left + "1000000000000000000" + right, ref dat, u);
}

```

4. Парсинг конструкций для получения значений из ячеек таблиц свойств

```

string str1 = "";
int r = -1, c = -1;
try
{
    Match matchElem = Regex.Match(str, @"([\w\[\]\.\+\-\
\\*\%^\(\)\0123456789]*){([\w\[\]\.\+\-\\\*\%^\(\)\0123456789 ]*){([\[\]\d\.\+\-\
\\*\%^\(\) ]*)}{([\w\[\]\.\+\-\\\*\%^\(\)\0123456789 ]*)}|([\w\[\]\.\+\-\
\\*\%^\(\)\0123456789]*){([\w\[\]\.\+\-\\\*\%^\(\)\0123456789 ]*)}");
    if (matchElem.Groups.Count > 1)
    {

```



```

        inner = StringHelper.DoString(str.Substring(matchElem.Index,
matchElem.Length)).Trim(' ');
        left = str.Substring(0, matchElem.Index);
        right = str.Substring(matchElem.Index + matchElem.Length);
        LinkedList<string> strs = new LinkedList<string>();
        string symbol = "";
        k = 0;
        while (k < inner.Length)
        {
            if (inner[k] == '{')
            {
                k++;
                while (k < inner.Length && inner[k] != '}' && inner[k] !=
'{' )
                {
                    str1 += inner[k];
                    k++;
                }
                strs.AddLast(str1);
                str1 = "";
            }
            else if (inner[k] == '}')
            {
                k++;
                while (k < inner.Length && inner[k] != '}' && inner[k] !=
'{' )
                {
                    str1 += inner[k];
                    k++;
                }
                strs.AddFirst(str1);
                str1 = "";
            }
            else
            {
                symbol += inner[k];
                k++;
            }
        }
        strs.AddLast(symbol);
        strs.RemoveFirst();
        return Parse(left + Find(strs, u).ToString() + right, ref dat, u);
    }

    matchElem = Regex.Match(str, @"([\w\[\]\.\+\-
\*\%\/\^\_\(\)\0123456789]*){([\d ]*;[\d ]*)}");
    if (matchElem.Groups.Count > 1)
    {
        inner = StringHelper.DoString(str.Substring(matchElem.Index,
matchElem.Length)).Trim(' ');
        left = str.Substring(0, matchElem.Index);
        right = str.Substring(matchElem.Index + matchElem.Length);
        str1 = "";
        k = 0;
        Composition e = null;
        DataTable d = null;

        while (k < inner.Length)
        {
            if (inner[k] == '{')
            {
                k++;
                e = MendeleevTable.Elems.Find(x => x.Name == str1);
                if (e == null)
                    e = MendeleevTable.Compos.Find(x => x.Name == str1);

                if (e == null)
                    d = MendeleevTable.BinarySystem.Find(x => x.TableName
== str1);

                if (d == null & e == null)

```

```
        throw new Exception("Отсутствует соединение, элемент  
или система" + str1 + ".", new Exception("MyException"));  
        str1 = "";  
    }  
    else if (inner[k] == ';')  
    {  
        k++;  
        int.TryParse(str1, out r);  
        str1 = "";  
    }  
    else if (inner[k] == '}')  
    {  
        k++;  
        int.TryParse(str1, out c);  
        str1 = "";  
    }  
    else  
    {  
        str1 += inner[k];  
        k++;  
    }  
    }  
    if (e != null)  
        return Parse(left + e.Properties[c].Second[r] + right, ref  
dat, u);  
    return Parse(left + d.Rows[r][c] + right, ref dat, u);  
}  
  
matchElem = Regex.Match(str, @"([\w\[\]\.\+\-  
\\*\%^\^_\(\)0123456789]*){([\d ]*[\w\[\]\.\+\-\\*\%^\^_\(\)0123456789 ]*)}");  
if (matchElem.Groups.Count > 1)  
{  
    inner = StringHelper.DoString(str.Substring(matchElem.Index,  
matchElem.Length)).Trim(' ');  
    left = str.Substring(0, matchElem.Index);  
    right = str.Substring(matchElem.Index + matchElem.Length);  
    str1 = "";  
    k = 0;  
    Composition e = null;  
    DataTable d = null;  
  
    while (k < inner.Length)  
    {  
        if (inner[k] == '{')  
        {  
            k++;  
            e = MendeleevTable.Elems.Find(x => x.Name == str1);  
            if (e == null)  
                e = MendeleevTable.Compos.Find(x => x.Name == str1);  
  
            if (e == null)  
                d = MendeleevTable.BinarySystem.Find(x => x.TableName  
== str1);  
  
            if (d == null & e == null)  
                throw new Exception("Отсутствует соединение, элемент  
или система" + str1 + ".", new Exception("MyException"));  
            str1 = "";  
        }  
        else if (inner[k] == ';')  
        {  
            k++;  
            int.TryParse(str1, out r);  
            str1 = "";  
        }  
        else if (inner[k] == '}')  
        {  
            k++;  
            if (e != null)  
            {  
                for (c = 0; c < e.Properties.Count; c++)
```

```
                if (e.Properties[c].First.Second == str1)
                    break;
            }
            else
                for (c = 0; c < d.Columns.Count; c++)
                    if (d.Columns[c].Caption == str1)
                        break;
            str1 = "";
        }
        else
        {
            str1 += inner[k];
            k++;
        }
    }
    if (e != null)
        return Parse(left + e.Properties[c].Second[r] + right, ref
dat, u);
        return Parse(left + d.Rows[r][c] + right, ref dat, u);
    }
}
catch (FormatException)
{
    throw new FormatException(string.Format("Неверная входная строка
'{0}'", str), new Exception("MyException"));
}
catch (ArgumentOutOfRangeException)
{
    throw new FormatException(string.Format("Отсутствует строка {0} или
столбец {1} у данного элемента/соединения.", r, c), new Exception("MyException"));
}
catch (NullReferenceException)
{
    throw new FormatException("Отсутствует содержание ячейки, к которой вы
обратились.", new Exception("MyException"));
}
}
```

5. Парсинг скобок

```
Match brackets = Regex.Match(str, @"\[([^\w\d\[\\].\+|-\\*\\/%^\_\\(\|
]*)\\)");
if (brackets.Groups.Count > 1)
{
    i = 0;
    j = 0;
    left = "";
    right = "";
    inner = "";
    while (i < str.Length)
    {
        if (str[i] == '(' && i < str.Length)
        {
            j++;
            if (j == 1)
            {
                left += right;
                right = "";
                i++;
            }
        }

        if (i < str.Length && str[i] == ')')
        {
            j--;
            if (j == 0)
            {
                inner += right;
                right = str.Substring(i + 1);
                return Parse(left + Parse(inner, ref dat, u) + right, ref
dat, u);
            }
        }
    }
}
```

```
        right += str[i];
        i++;
    }
    if (inner != "")
        return Parse(left + Parse(inner, ref dat, u) + right, ref dat, u);
}
```

6. Парсинг математических операций

```
Match matchMulOp = Regex.Match(str,
string.Format(@"({0})\s?({1})\s?({0})\s?", RegexOptions, RegexOptions));
Match matchAddOp = Regex.Match(str,
string.Format(@"({0})\s?({1})\s?({0})\s?", RegexOptions, RegexOptions));
var match = (matchMulOp.Groups.Count > 1) ? matchMulOp :
(matchAddOp.Groups.Count > 1) ? matchAddOp : null;
if (match != null)
{
    left = str.Substring(0, match.Index);
    right = str.Substring(match.Index + match.Length);
    return Parse(left +
ParseAct(match).ToString(CultureInfo.InvariantCulture) + right, ref dat, u);
}
```

7. Парсинг числа

```
str1 = "";
for (i = 0; i < str.Length; i++)
    if (str[i] != ' ')
        if (str[i] == '.')
            str1 += ',';
        else
            str1 += str[i];
if (double.TryParse(str1, out double num))
    return num;
else
    throw new FormatException(string.Format("Неверная входная строка
'{0}'", str), new Exception("MyException"));
```

3.2.1.5 RoundConverter

Данный класс создан для округления значений в таблицах свойств.

Он реализует метод `Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)` интерфейса `System.Windows.Data.IValueConverter`, округляющий число до четвертого знака после запятой.

3.2.1.6 StringHelper

Данный статический класс является вспомогательным для работы со строковыми значениями в программе.

Он включает в себя методы преобразования цифр и некоторых специальных символов в надстрочные (`Superstring(char c)`) и подстрочные (`Substring(char c)`). Также реализована функция `DoString(string str)`, которая преобразует числовые символы после буквенных и подстрочных символов в подстрочные:

```
/// <summary>
/// Преобразует числовые символы после буквенных и подстрочных символов в
подстрочные
/// </summary>
/// <param name="str">исходная строка</param>
/// <returns>преобразованная строка</returns>
```

```

public static string DoString(string str)
{
    int i = 0;
    string str1 = "";
    while (i < str.Length)
    {
        if (i > 0 && "0123456789".Contains(str[i].ToString()) &&
"0123456789) QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm".Contains(str1[i -
1].ToString()))
            str1 += Substring(str[i]);
        else
            if (i > 1 && '.' == str[i - 1] &&
"0123456789) QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm".Contains(str1[i -
2].ToString()))
                str1 += Substring(str[i]);
            else
                str1 += str[i];
            i++;
    }
    return str1;
}

```

3.2.1.7 Strings

Данный класс создан для заполнения таблицы Менделеева (рис. X) химическими элементами (табл. 2).

Таблица 2 – Класс Strings

Имя	Модификатор доступа	Тип	Описание
Ia, IIa, IIIa, IVa, Va, VIa, VIIa, VIIIa_1, VIIIa_2, VIIIa_3, Ib, IIb, IIIb, IVb, Vb, VIb, VIIb, VIIIb	Public	String	Элемент определенной группы и подгруппы

3.2.2 Построение графиков

3.2.2.1 Окна программы

3.2.2.1.1 DataSettings

Данный класс описывает окно настроек входных параметров для графиков (рис. 11).

Настройка данных

	Символ химического элемента	Номер строки
Первый химический элемент:	Na	0
Второй химический элемент:	Ag	0
Общий химический элемент:	Cl	0
Обозначение радиуса иона элемента в таблице данных:	R(i)	
Обозначение электроотрицательности элемента в таблице данных:	x	
Обозначение формального заряда элемента в таблице данных:	ФЗ	
A =	1,745	m = 2
n =	6	z = 1

Сохранить изменения и закрыть окно

Рисунок 11 – Окно настроек данных

В данном окне пользователь может задать некоторые значения входных параметров, а также указать обозначения свойств, получаемых из таблиц, если они отличаются от значений по умолчанию. Оно изначально открывается при нажатии кнопки «Купол распада» в окне свойств химического соединения/системы соединений (рис. 5).

3.2.2.1.2 dG_Temp

Данный класс описывает окно настройки температурных пределов для графика свободной энергии Гиббса (рис. 12).

Диапазон температур

Нижний предел температуры (°C): 0

Верхний предел температуры (°C): 1

Шаг температуры: 100

Готово

Рисунок 12 – Окно настройки температурных пределов

В данном окне пользователь может задать начальную и конечную температуру, а также шаг. Графики строятся для каждой температуры, соответствующей данному диапазону, с указанным интервалом.

3.2.2.1.3 DomeOfDecay

Данный класс описывает окно построения графиков (рис. 13).

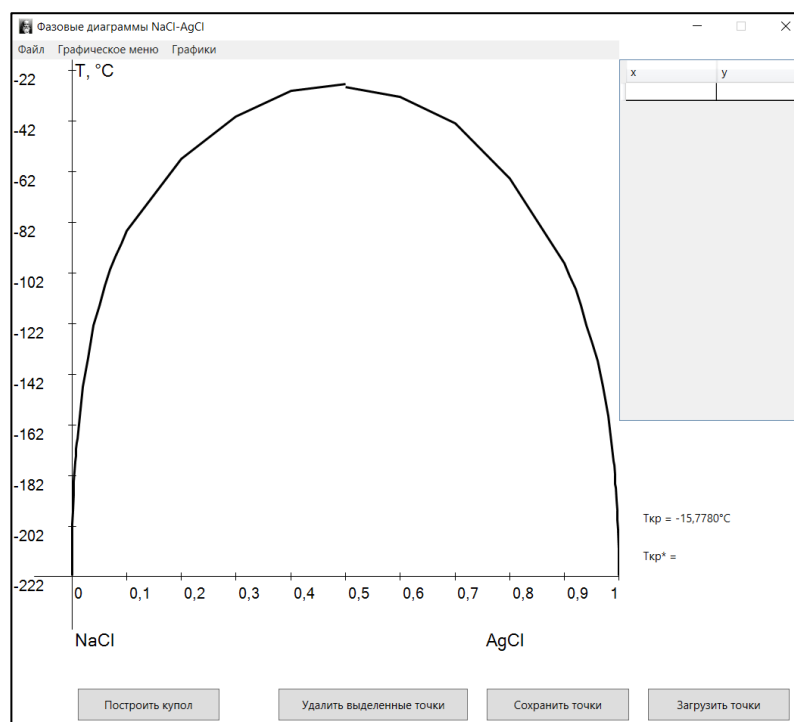


Рисунок 13 – Окно построения графиков (купол распада)

В данном окне пользователь может построить три различных графика. Первым из них является купол распада. Он строится по умолчанию при переходе в данное окно и по нажатию кнопки «Построить купол». Второй – это термодинамическая функция смешения. Её можно построить, выбрав пункт меню «Графики/Функция H_{sm} ». Также имеется возможность добавлять экспериментальные точки. Их можно ввести вручную в таблицу справа или загрузить из файла формата txt. Третий график – это свободная энергия Гиббса. Начать построение диаграммы можно по нажатию на пункт меню «Графики/Функция G_{sm} ». Также выбрав «Графики/Оценка чувствительности» можно оценить влияние изменяемых параметров на теплоту смешения посредством их изменения через слайдеры (рис. 14).

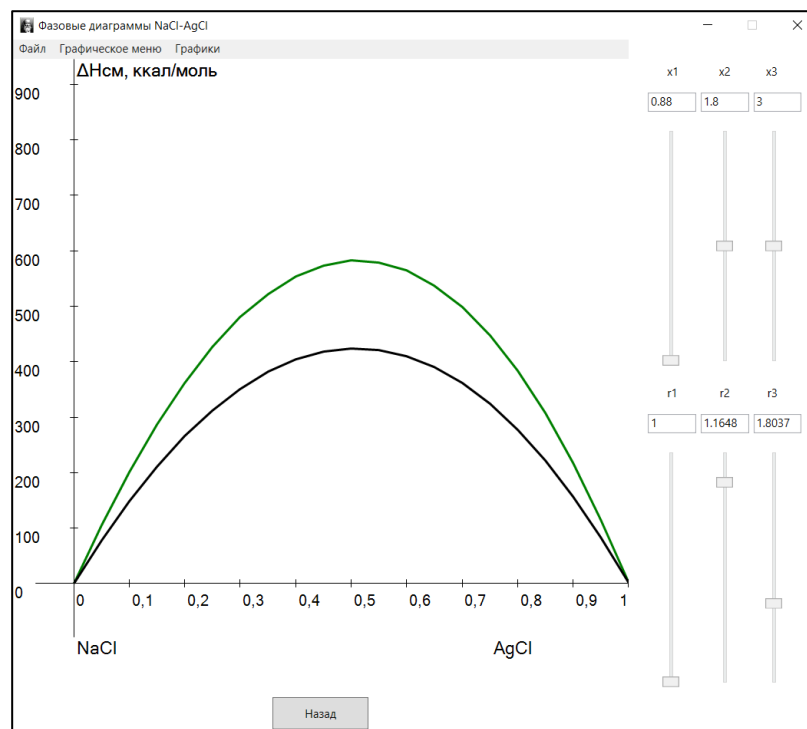


Рисунок 14 – Окно построения графиков (оценка чувствительности)

3.2.2.2 BinSystem

Данный класс создан для хранения информации о бинарной системы соединений (табл. 3).

Таблица 3 – Класс BinSystem

Имя	Модификатор доступа	Тип	Описание
kN	Private	const Double	Универсальная газовая постоянная
A	Public get Private set	Double	Константа Моделунга
zX	Public get Private set	Double	Формальный заряд общего химического элемента
sourceString	Private	readonly String	Обозначение системы соединений
elemA, elemB, elemX	Private	Composition	Химический элемент
c	Public get	Double	Эмпирический параметр
m	Public get Private set	Double	Число структурных единиц.
n	Public get Private set	Double	Координационное число
z	Private	Double	Формальный заряд
r_1, r_2, r_3	Public	Double	Ионные радиусы
x_1, x_2, x_3	Public	Double	Электроотрицательность.
ElementA, ElementB, ElementX	Public get	String	Обозначения химических элементов
delR	Public get	Double	Разность радиусов

Продолжение таблицы 3

delEps	Public get	Double	Разность степеней ионности	
Tmax	Public get	Double	Критическая температура	
Методы (если не указано иное, типы параметров методов класса принадлежат пространству имён System).				
Имя	Модификатор доступа	Тип	Параметры	Описание
Ssm	Public	Double	Double x	Энтропия смешения
R	Public	Double	Double x	Среднее межатомное расстояние
Eps	Public	Double	Int32 i	Степень ионности
Hsm	Public	Double	Double x	Теплота смешения
Gsm	Public	Double	Double x, Double T	Свободная энергия Гиббса
ToString	Public	override String		Возвращает обозначение системы
Clone	Public	BinSystem		Создает копию системы

3.2.2.3 Collapse

Данный класс создан для хранения набор точек фазовой диаграммы и содержит методы их получения:

```

/// <summary>
/// Получает точки для фазовой диаграммы
/// </summary>
/// <param name="system">система соединений</param>
public Collapse(BinSystem system)
{
    string r = GetRatio(system.delR / Math.Min(system.R(1), system.R(0)));

    System.Windows.Resources.StreamResourceInfo ri =
System.Windows.Application.GetResourceStream(new Uri("DrawingClasses/Collapse.xml",
UriKind.Relative));
    System.IO.Stream data = ri.Stream;

    XDocument doc = XDocument.Load(data);
    string[] x1values = doc.Root.Elements().First(p =>
p.Attribute("ratio").Value == r).Element("x1").Value.Split(';');
    string[] x2values = doc.Root.Elements().First(p =>
p.Attribute("ratio").Value == r).Element("x2").Value.Split(';');
    string[] y1values = doc.Root.Elements().First(p =>
p.Attribute("ratio").Value == r).Element("y1").Value.Split(';');
    string[] y2values = doc.Root.Elements().First(p =>
p.Attribute("ratio").Value == r).Element("y2").Value.Split(';');

    if (system.R(1) >= system.R(0))
    {
        right = new Point[x1values.Length];
        left = new Point[x2values.Length];

        for (int i = 0; i < x1values.Length; i++)
            right[i] = new Point(double.Parse(x1values[i]),
double.Parse(y1values[i]));

        for (int i = 0; i < x2values.Length; i++)
            left[i] = new Point(double.Parse(x2values[i]),
double.Parse(y2values[i]));
    }
}

```

```
    }
    else
    {
        left = new Point[x1values.Length];
        right = new Point[x2values.Length];

        for (int i = 0; i < x1values.Length; i++)
            left[i] = new Point(double.Parse(x1values[i]),
double.Parse(y1values[i]));

        for (int i = 0; i < x2values.Length; i++)
            right[i] = new Point(double.Parse(x2values[i]),
double.Parse(y2values[i]));
    }
}

/// <summary>
/// Получает соотношение радиусов
/// </summary>
/// <param name="ratio">delR/Rmin</param>
/// <returns>соотношение радиусов</returns>
public static string GetRatio(double ratio)
{
    ratio = Math.Round(ratio, 3);
    if ((ratio >= 0) && (ratio < 0.025)) return "0,00";
    else if (ratio < 0.075) return "0,05";
    else if (ratio < 0.125) return "0,10";
    else if (ratio < 0.175) return "0,15";
    else if (ratio < 0.225) return "0,20";
    else if (ratio < 0.275) return "0,25";
    else if (ratio < 0.325) return "0,30";
    else throw new Exception("Недопустимое отношение радиусов!", new
Exception("MyException"));
}
```

3.2.2.4 CollapseGraph

Данный класс создан для отрисовки различных элементов диаграммы. Он включает в себя методы отрисовки:

- Осей графика

```
/// <summary>
/// Рисует оси координат
/// </summary>
public void DrawAxes(bool f = false)
{
    g.DrawString(system.ElementA + system.ElementX, new Font("X", 14),
Brushes.Black, new Point(80, width + 30));
    g.DrawString(system.ElementB + system.ElementX, new Font("X", 14),
Brushes.Black, new Point(width - 100, width + 30));

    g.DrawLine(Pens.Black, 80, 0, 80, width + 30);
    g.DrawLine(Pens.Black, 30, width - 40, width + 80, width - 40);

    for (double x = 0; x <= 1; x += 0.1)
    {
        g.DrawString(x.ToString(), new Font("X", 12), Brushes.Black, 80 +
(float)(width * x), width - 30);
        g.DrawLine(Pens.Black, 80 + (int)(width * x), width - 45, 80 +
(int)(width * x), width - 35);
    }
    g.DrawString("1", new Font("X", 12), Brushes.Black, width + 65, width -
30);

    double c = Math.Round((UpTemp - DownTemp) / 100.0);
    if (c == 0)
        c = 1;
    if (c < (UpTemp - DownTemp) / 100)
        c = (c + 1) * 10;
```

```
        else
            c *= 10;

        if (f)
            for (double x = DownTemp; x <= UpTemp; x += c)
            {
                g.DrawString(ToC(x).ToString(), new Font("X", 12), Brushes.Black,
0, width - 40 - (int)((x - DownTemp) / (UpTemp - DownTemp) * width));
                g.DrawLine(Pens.Black, 75, width - 40 - (int)((x - DownTemp) /
(UpTemp - DownTemp) * width), 85, width - 40 - (int)((x - DownTemp) / (UpTemp -
DownTemp) * width));
            }
        else
            for (double x = DownTemp; x <= UpTemp; x += c)
            {
                g.DrawString(x.ToString(), new Font("X", 12), Brushes.Black, 0,
width - 40 - (int)((x - DownTemp) / (UpTemp - DownTemp) * width));
                g.DrawLine(Pens.Black, 75, width - 40 - (int)((x - DownTemp) /
(UpTemp - DownTemp) * width), 85, width - 40 - (int)((x - DownTemp) / (UpTemp -
DownTemp) * width));
            }
    }
}
```

- Эксперимента

```
/// <summary>
/// Рисует эксперимент
/// </summary>
/// <param name="f">флаг: true - купол распада, false - теплота
смещения</param>
public void DrawExperiment(bool f = true)
{
    List<Point> left = new List<Point>();
    List<Point> right = new List<Point>();

    foreach (var item in experiment)
    {
        int x = 80 + (int)(item.X * width);
        int y;
        if (f)
            y = width - 40 - (int)((ToK(item.Y) - DownTemp) / (UpTemp -
DownTemp) * width);
        else
            y = width - 40 - (int)((item.Y - DownTemp) / (UpTemp - DownTemp) *
width);

        y = y > width ? width : y;

        if (item.X <= 0.5)
            left.Add(new Point(x, y));
        else
            right.Add(new Point(x, y));
    }

    foreach (var item in left)
    {
        g.FillEllipse(penExp.Brush, item.X - 8, item.Y - 8, 16, 16);
    }
    foreach (var item in right)
    {
        g.FillEllipse(penExp.Brush, item.X - 8, item.Y - 8, 16, 16);
    }
}
```

- Купола распада

```
/// <summary>
/// Рисует купол распада
/// </summary>
/// <param name="f">флаг: true - аппроксимация, false - теория</param>
public void DrawCollapse(bool f = false)
{
}
```

```

Collapse collapse = new Collapse(system);
UpTemp = UpTemp == -1 ? (int)system.Tmax + 10 : UpTemp;
DownTemp = DownTemp == -1 ? (int)(0.20 * system.Tmax) : DownTemp;

right = new Point[collapse.right.Length];
for (int i = 0; i < right.Length; i++)
{
    int x = 80 + (int)(width * (1 - collapse.right[i].X));
    int y = width - 40 - (int)(width * ((collapse.right[i].Y * system.Tmax
- DownTemp) / (UpTemp - DownTemp)));
    y = y > width ? width : y;

    right[i] = new Point(x, y);
}

left = new Point[collapse.left.Length];
for (int i = 0; i < left.Length; i++)
{
    int x = 80 + (int)(width * collapse.left[i].X);
    int y = width - 40 - (int)(width * ((collapse.left[i].Y * system.Tmax
- DownTemp) / (UpTemp - DownTemp)));
    y = y > width ? width : y;

    left[i] = new Point(x, y);
}

g.DrawString("T, °C", new Font("X", 14), Brushes.Black, new Point(80, 0));
if (f)
{
    g.DrawLines(penApp, right);
    g.DrawLines(penApp, left);
}
else
{
    g.DrawLines(pen, right);
    g.DrawLines(pen, left);
}
}

```

- Термодинамической функции смещения

```

/// <summary>
/// Рисует термодинамическую функцию смещения
/// </summary>
/// <param name="f">флаг: true - аппроксимация, false - теория</param>
public void DrawDH(bool f = true)
{
    DownTemp = DownTemp == -1 ? 0 : DownTemp;
    UpTemp = UpTemp == -1 ? (int)(system.Hsm(0.5) * 1000) + 100 : UpTemp;

    Point[] dh = new Point[21];
    for (double i = 0; i < 1; i += 0.05)
        dh[(int)Math.Round(i * 20)] = new Point(80 + (int)(i * width), width -
40 - (int)((system.Hsm(i) * 1000 - DownTemp) / (UpTemp - DownTemp) * width));
    dh[20] = new Point(80 + width, width - 40 - (int)((system.Hsm(0.999999) *
1000 - DownTemp) / (UpTemp - DownTemp) * width));
    g.DrawString("ΔHсм, ккал/моль", new Font("X", 14), Brushes.Black, new
Point(80, 0));

    if (f)
        g.DrawLines(penApp, dh);
    else
        g.DrawLines(pen, dh);
}

```

- Графика свободной энергии Гиббса

```

/// <summary>
/// Рисует график свободной энергии Гиббса
/// </summary>
/// <param name="tempD">нижняя граница температуры</param>

```

```
/// <param name="tempU">верхняя граница температуры</param>
/// <param name="tempInt">шаг температуры</param>
public void DrawDG(int tempD, int tempU, int tempInt)
{
    DownTemp = DownTemp == -1 ? 0 : DownTemp;
    UpTemp = UpTemp == -1 ? (int)(system.Hsm(0.5) * 1000) + 100 : UpTemp;

    Point[] dh = new Point[21];
    for (double i = 0; i < 1; i += 0.05)
        dh[(int)Math.Round(i * 20)] = new Point(80 + (int)(i * width), width -
40 - (int)((system.Hsm(i) * 1000 - DownTemp) / (UpTemp - DownTemp) * width));
    dh[20] = new Point(80 + width, width - 40 - (int)((system.Hsm(0.99999) *
1000 - DownTemp) / (UpTemp - DownTemp) * width));
    g.DrawLines(penApp, dh);

    Point[] dg = new Point[21];
    Point[] ds = new Point[21];

    for (int t = tempD; t < tempU; t += tempInt)
    {
        for (double i = 0.00001; i < 1; i += 0.05)
        {
            ds[(int)Math.Round(i * 20)] = new Point(80 + (int)(i * width),
width - 40 - (int)((-t * system.Ssm(i) * 1000 - DownTemp) / (UpTemp - DownTemp) *
width));
            dg[(int)Math.Round(i * 20)] = new Point(80 + (int)(i * width),
width - 40 - (int)((system.Gsm(i, t) * 1000 - DownTemp) / (UpTemp - DownTemp) *
width));
        }
        dg[20] = new Point(80 + width, width - 40 - (int)((system.Gsm(0.99999,
t) * 1000 - DownTemp) / (UpTemp - DownTemp) * width));
        ds[20] = new Point(80 + width, width - 40 - (int)((-t *
system.Ssm(0.99999) * 1000 - DownTemp) / (UpTemp - DownTemp) * width));
        g.DrawLines(pen, dg);
        g.DrawLines(penExp, ds);
    }

    g.DrawString("ΔGсм, ккал/моль", new Font("X", 14), Brushes.Black, new
Point(80, 0));
}
```

Также, в данном классе содержатся вспомогательные методы для отрисовки, такие как: добавление и удаление точек из эксперимента или перевод градусов из Цельсия в Кельвины и наоборот.

3.2.2.5 Report

Данный класс создан для генерации отчета по конкретной системе в формате PDF при помощи библиотеки iTextSharp. Запустить создание отчета можно выбрав пункт меню «Файл/Сформировать отчет» в окне построения графиков (рис. 13). Отчет состоит из 1–4 страниц, в зависимости от количества графиков, сохраненных при помощи «Файл/Сохранить график в отчет». Первая страница содержит название системы, таблицу её свойств и изначальные значения параметров. На второй расположен график купола распада и новые значения параметров, если таковые имеются. На третьей странице изображен график термодинамической функции смешения и новые значения параметров, если таковые имеются. На четвертой странице находится график свободной энергии Гиббса, а также температурный интервал и шаг этого графика.

3.2.3 Аппроксимация

3.2.3.1 Criterion

Данный класс создан для хранения методов, оценивающих отклонение функции при оптимизации.

Он включает в себя среднеквадратичное отклонение:

$$\sqrt{\frac{\sum f(x_i) - y_i^2}{n}}, \quad (9)$$

где n – это количество экспериментальных точек, а f – это функция, параметры которой мы оптимизируем, и штрафную функцию, ограничивающую оптимизацию параметров по заданным жестким пределам. Также данный класс содержит метод для определения расстояния между двумя наборами точек, который используется для аппроксимации эксперимента купола распада:

$$\sum \min(\sqrt{(x_{1i} - x_{2j})^2 + (y_{1i} - y_{2j})^2}) \quad (10)$$

3.2.3.2 Library

Данный класс создан для аппроксимации набора экспериментальных точек функцией.

Он включает в себя метод градиентной минимизации:

```
/// <summary>
/// Метод наискорейшего спуска (метод градиентной минимизации)
/// </summary>
/// <param name="funN">исследуемая функция</param>
/// <param name="X0">начальный вектор параметров</param>
/// <param name="Delta">допустимое отклонение для ширины интервала</param>
/// <param name="Epsilon">допустимое отклонение для |f(b) - f(a)|</param>
/// <param name="Max">максимальное число итераций</param>
/// <returns>оптимальный вектор параметров</returns>
public static double[] GradientMinimization(Func<double[], double> funN,
double[] X0, double Delta = 1E-8, double Epsilon = 1E-11, int Max = 100)
{
    int NC = X0.Length;
    double[] Q1 = new double[NC];
    double[] Q2 = new double[NC];
    double F1, F2;
    int iter = 0;
    double deltaX = 0;
    X0.CopyTo(Q2, 0);

    do
    {
        iter++;
        Q2.CopyTo(Q1, 0);
        F1 = funN(Q1);
        Q2 = Quadmin(funN, Q1);
        F2 = funN(Q2);
        deltaX = 0;
        for (int k = 0; k < NC; k++)
        {
            double ZN = Math.Abs(Q2[k]);
            if (ZN > 0.0)
            {
```

```

        double DR = Math.Abs((Q2[k] - Q1[k]) / ZN);
        deltaX = DR > deltaX ? DR : deltaX;
    }
}
while (iter < Max & Math.Abs(F1 - F2) > Epsilon & deltaX > Delta);

return Q2;
}

/// <summary>
/// Вычисление градиента и направляющих вектора перемещения
/// </summary>
/// <param name="funN">исследуемая функция</param>
/// <param name="X0">вектор параметров - исследуемая точка</param>
/// <param name="del">относительная вариация каждого параметра</param>
/// <returns>вектор перемещения вдоль градиента</returns>
public static double[] Gradient(Func<double[], double> funN, double[] X0,
double del = 0.001)
{
    int NC = X0.Length;
    double[] G = new double[NC];
    double[] dx = new double[NC];
    for (int j = 0; j < NC; j++)
        if (X0[j] == 0)
            dx[j] = del;
        else
            dx[j] = Math.Abs(X0[j] * del);

    double[] V = new double[NC];
    X0.CopyTo(V, 0);
    double Fma, Fmi, dFi;
    for (int j = 0; j < NC; j++)
    {
        V[j] = X0[j] + dx[j];
        Fma = funN(V);
        V[j] = X0[j] - dx[j];
        Fmi = funN(V);
        dFi = Fma - Fmi;
        G[j] = dFi / (2 * dx[j]);
        V[j] = X0[j];
    }

    double[] S = new double[NC];
    double len = 0;
    for (int j = 0; j < NC; j++)
        len += Math.Pow(G[j], 2);

    len = Math.Sqrt(len);
    for (int j = 0; j < NC; j++)
        S[j] = -G[j] / len;

    return S;
}

/// <summary>
/// Вычисление минимума вдоль градиента
/// </summary>
/// <param name="funN">исследуемая функция</param>
/// <param name="X0">начальный вектор параметров</param>
/// <param name="Delta">допустимое отклонение для ширины интервала</param>
/// <param name="Epsilon">допустимое отклонение для |f(b) - f(a)|</param>
/// <returns>параметры (точка) минимума вдоль градиента</returns>
private static double[] Quadmin(Func<double[], double> funN, double[] X0,
double Delta = 1E-5, double Epsilon = 1E-7)
{
    int NC = X0.Length;
    double Y0 = funN(X0);
    double[] P0 = new double[NC];
    X0.CopyTo(P0, 0);
    double[] P1 = new double[NC];

```

```
double[] P2 = new double[NC];
double[] S = new double[NC];
double H = 1.0;
double Err = 1.0;
int Jmax = 20;
double H0, H1, H2, Hmin, E0, E1, E2, Y1, Y2, D, Ymin;
int i;
int Cond = 0;
int J = 0;

S = Gradient(funN, X0);
for (i = 0; i < NC; i++)
{
    P1[i] = P0[i] + H * S[i];
    P2[i] = P0[i] + 2 * H * S[i];
}

Y1 = funN(P1);
Y2 = funN(P2);
double[] Pmin = new double[NC];
while ((J < Jmax) & (Cond == 0))
{
    if (Y0 < Y1)
    {
        Y2 = Y1;
        H = H / 2.0;
        for (i = 0; i < NC; i++)
        {
            P2[i] = P1[i];
            P1[i] = P0[i] + H * S[i];
        }
        Y1 = funN(P1);
    }
    else
    {
        if (Y2 < Y1)
        {
            Y1 = Y2;
            H = 2.0 * H;
            for (i = 0; i < NC; i++)
            {
                P1[i] = P2[i];
                P2[i] = P0[i] + 2.0 * H * S[i];
            }
            Y2 = funN(P2);
        }
        else Cond = -1;
    }
}

if (H < Delta) Cond = 1;
D = 4.0 * Y1 - 2.0 * Y0 - 2.0 * Y2;
if (D < 0)
    Hmin = H * (4.0 * Y1 - 3.0 * Y0 - Y2) / D;
else
{
    Cond = 4;
    Hmin = H / 3.0;
}

for (i = 0; i < NC; i++)
    Pmin[i] = P0[i] + Hmin * S[i];

Ymin = funN(Pmin);
H0 = Math.Abs(Hmin);
H1 = Math.Abs(Hmin - H);
H2 = Math.Abs(Hmin - 2.0 * H);
if (H0 < H) H = H0;
if (H1 < H) H = H1;
if (H2 < H) H = H2;
if (H < Delta) Cond = 1;
```



```
E0 = Math.Abs(Y0 - Ymin);
E1 = Math.Abs(Y1 - Ymin);
E2 = Math.Abs(Y2 - Ymin);
if (E0 < Err)
    Err = E0;
else if (E1 < Err)
    Err = E1;
else if (E2 < Err)
    Err = E2;
else if ((E0 == 0) && (E1 == 0) && (E2 == 0))
    Err = 0;

if (Err < Epsilon)
    Cond = 2;
if ((Cond == 2) && (H < Delta))
    Cond = 3;

J++;
return Pmin;
}
```

Также данный класс содержит метод, предназначенный для аппроксимации набора точек аналитическими выражениями, описывающими купол распада, у которого нет единой формулы для построения.

```
/// <summary>
/// Метод аппроксимации набора точек аналитическими выражениями, описывающими
купол распада
/// </summary>
/// <param name="sys">данные системы</param>
/// <param name="dat">набор точек</param>
/// <param name="t">критическая температура</param>
/// <returns>оптимальный набор параметров</returns>
public static double[] DomeApprox(List<List<double>> dat, int t,
HelperClasses.BinSystem sys)
{
    System.Windows.Resources.StreamResourceInfo ri =
Application.GetResourceStream(new Uri("DrawingClasses/Collapse.xml",
UriKind.Relative));
    Stream data = ri.Stream;
    System.Xml.Linq.XDocument doc = System.Xml.Linq.XDocument.Load(data);

    List<Point> ExpDots = new List<Point>();
    List<Point> Dots = new List<Point>();
    foreach (List<double> point in dat)
        ExpDots.Add(new Point(point[0], point[1]));

    double r1 = sys.r_1, r2 = sys.r_2, r3 = sys.r_3, x1 = sys.x_1, x3 =
sys.x_3;
    string r = "", r_old = "1";
    double min = 1000000, min_t = 1000000;
    double[] par_min = new double[] { sys.r_1, sys.r_2, sys.r_3, sys.x_1,
sys.x_3 };
    for (r1 = sys.r_1 - 0.02; r1 <= sys.r_1 + 0.02; r1 += 0.002)
        if (r1 > 0)
            for (r2 = sys.r_2 - 0.02; r2 <= sys.r_2 + 0.02; r2 += 0.002)
                if (r2 > 0)
                    for (r3 = sys.r_3 - 0.02; r3 <= sys.r_3 + 0.02; r3 += 0.002)
                        if (r3 > 0)
                            for (x1 = sys.x_1 - 0.02; x1 <= sys.x_1 + 0.02; x1 += 0.002)
                                if (x1 > 0)
                                    for (x3 = sys.x_3 - 0.02; x3 <= sys.x_3 + 0.02; x3 += 0.002)
                                        if (x3 > 0)
                                            {
                                                if (r != r_old)
                                                {
                                                    r = Collapse.GetRatio(Math.Abs(r1 - r2) / Math.Min(r1 + r3, r2 + r3));
                                                    string[] xlvalues = doc.Root.Elements().First(p =>
p.Attribute("ratio").Value == r).Element("x1").Value.Split(';');

```

```
        string[] x2values = doc.Root.Elements().First(p =>
p.Attribute("ratio").Value == r).Element("x2").Value.Split(';');
        string[] y1values = doc.Root.Elements().First(p =>
p.Attribute("ratio").Value == r).Element("y1").Value.Split(';');
        string[] y2values = doc.Root.Elements().First(p =>
p.Attribute("ratio").Value == r).Element("y2").Value.Split(';');

        Dots.Clear();
        for (int i = 0; i < x1values.Length; i++)
            Dots.Add(new Point(double.Parse(x1values[i]), double.Parse(y1values[i]) * t));

        for (int i = 0; i < x2values.Length; i++)
            Dots.Add(new Point(double.Parse(x2values[i]), double.Parse(y2values[i]) * t));

        r_old = r;
    }
    double _t = (33.33 * (1 - (sys.z / sys.n) * Math.Exp((x1 - x3) * (x1 - x3) * -0.25)) +
8.83) * sys.m * sys.n * sys.z * sys.zX * Math.Pow(Math.Abs(r1 - r2) / Math.Min(r1 +
r3, r2 + r3), 2) / (1.9844 * 0.002);
    if (Math.Abs(CollapseGraph.ToK(t) - _t) < min_t)
    {
        min_t = Math.Abs(CollapseGraph.ToK(t) - _t);
        if (min == -1 || Criterion.Dots_Distance(ExpDots, Dots) <= min)
        {
            min = Criterion.Dots_Distance(ExpDots, Dots);
            par_min[0] = r1;
            par_min[1] = r2;
            par_min[2] = r3;
            par_min[3] = x1;
            par_min[4] = x3;
        }
    }
}
return par_min;
}
```

3.2.4 Help

Данный класс описывает окно справки. В нем пользователь может посмотреть краткие инструкции о том, как работать с различными окнами программы. Это окно можно открыть, нажав на пункт меню «Справка» стартового окна (рис. 10) или клавишу F1 в любом окне программы.

3.2.5 HotKey

Данный класс создан для привязки горячих клавиш к действиям. Основными методами данного класса являются Register(), который регистрирует сочетание клавиш, переданное при создании экземпляра, и Unregister(), который утилизирует управляемые ресурсы.

```
/// <summary>
/// Регистрирует сочетание клавиш
/// </summary>
/// <returns>Зарегистрирован или нет?</returns>
public bool Register()
{
    int virtualKeyCode = KeyInterop.VirtualKeyFromKey(Key);
    Id = virtualKeyCode + ((int)KeyModifiers * 0x10000);
    bool result = RegisterHotKey(IntPtr.Zero, Id, (uint)KeyModifiers,
(uint)virtualKeyCode);

    if (_dictHotKeyToCallbackProc == null)
    {
        _dictHotKeyToCallbackProc = new Dictionary<int, HotKey>();
    }
}
```

```

        ComponentDispatcher.ThreadFilterMessage += new
        ThreadMessageEventHandler (ComponentDispatcher.ThreadFilterMessage);
    }

    if (!_dictHotKeyToCallBackProc.ContainsKey(Id))
        _dictHotKeyToCallBackProc.Add(Id, this);

    Debug.Print(result.ToString() + ", " + Id + ", " + virtualKeyCode);
    return result;
}

/// <summary>
/// Утилизирует управляемые ресурсы
/// </summary>
public void Unregister()
{
    HotKey hotKey;
    if (_dictHotKeyToCallBackProc.TryGetValue(Id, out hotKey))
        UnregisterHotKey(IntPtr.Zero, Id);
}

```

3.2.6 Pair<T, U>

Данный класс создан для хранения двух разнородных объектов как единое целое (табл. 4).

Таблица 4 – Класс Pair<T, U>

Имя	Модификатор доступа	Тип	Описание
First	Public get Public set	T	Первый аргумент
Second	Public get Public set	U	Второй аргумент

3.2.7 Point

Данный класс создан для хранения информации о точку с двумя численными координатами (табл. 5).

Таблица 5 – Класс Point

Имя	Модификатор доступа	Тип	Описание
X	Public get Public set	Double	Координата X
Y	Public get Public set	Double	Координата Y

3.3 Тестирование

Расчет деталей фазовых диаграмм выполнен с помощью уравнений (3) - (5) и (7), (8) для расчета энтальпии смещения - $\Delta H_{см}$ и критической температуры распада твердых растворов – $T_{кр}$, соответственно.

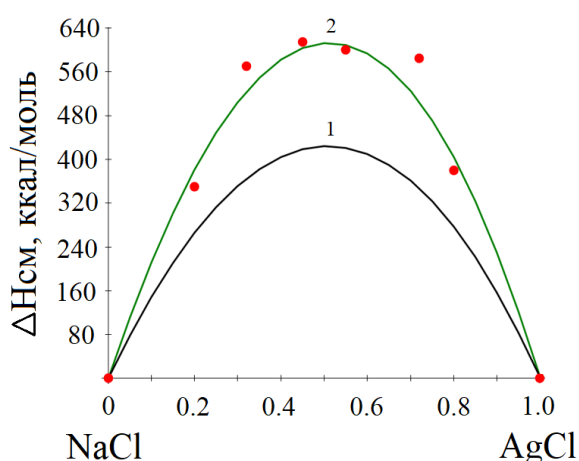
В табл. 6-9 представлены исходные данные для вычисления $\Delta H_{см}$ во всем интервале концентраций в системах NaCl-AgCl, SnO₂-TiO₂, TiO₂-ZrO₂ и SiO₂-TiO₂.

Система NaCl-AgCl. Расчетные данные для системы NaCl-AgCl (рис. 15а, табл. 6) показали, что экспериментальные точки [17] и рассчитанная кривая 1 не

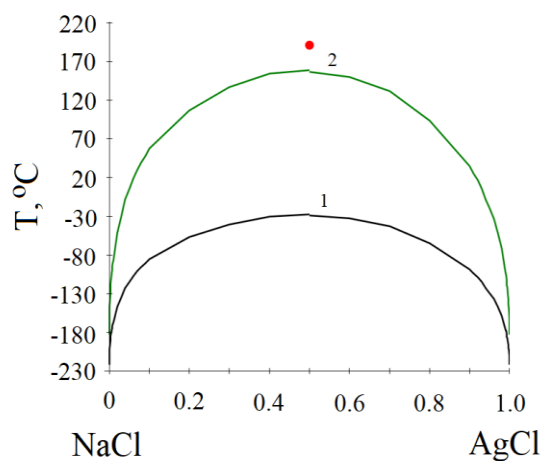
совпадают.

Таблица 6 – Данные для расчета термодинамической функции смешения и построения купола распада системы NaCl-AgCl

Размерный фактор				Структурный фактор		Фактор химической связи							
R_1 , Å	R_2 , Å	$R_{обм}$, Å	$\Delta R/R_{мин}$, Å	$n_1 = n_2$	A	χ_1 , отн. ед	χ_2 , отн. ед	χ_0 , отн. ед	c , ккал/моль	ε_1	ε_2	$\Delta\varepsilon$	$T_{кр}$, °C
Na	Ag	Cl				Na	Ag	Cl		NaCl	AgCl		
1.02/ 1.00*	1.15/ 1.17*	1.81/ 1.79*	0.046/ 0.061*	6	1.745	0.9/ 0.88*	1.8	3.0/ 2.98*	40.33	0.945	0.88	0.065	-16/ 180*
-0.02	+0.02	-0.02	+0.015			-0.02		-0.02	0	0	0	0	+196



а)



б)

Рисунок 15 – Кривые $\Delta H_{см}$, ккал/моль (а) и T , °C (б): рассчитанные (кривая 1) по исходным данным (табл. 6) и по оптимизированным значениям (в табл. 6 отмечены *) (кривая 2).
● – экспериментальные величины [17].

Варьируемыми параметрами при расчете $\Delta H_{см}$ (уравнение (3)) являются $\Delta\varepsilon$ с изменяемыми величинами χ_i (по разным системам разные значения) и c , связанные друг с другом (фактор химической связи), и $\Delta R/R$ (размерный фактор). В табл. 6 звездочкой отмечены оптимизированные значения параметров, по которым рассчитана кривая 2, хорошо описывающая экспериментальные значения (рис. 15а).

Для приближения теоретических значений $\Delta H_{см}$, ккал/моль к экспериментальным проведена оценка изменения величины $\Delta H_{см}$ вариацией аргументов уравнения (3) (значения $\Delta\varepsilon$, c и $\Delta R/R$ изменяются при оптимизации) с использованием метода градиентной минимизации, где критерием оптимальности послужило среднеквадратичное отклонение значений функции от точек эксперимента. В результате выполненных расчетов найдено, что вариация величинами c и $\Delta R/R$ в реальных пределах оказывают наименьшее влияние на функцию $\Delta H_{см}$, а лучшая сходимость рассчитанных (рис. 15а, кривая 2) и

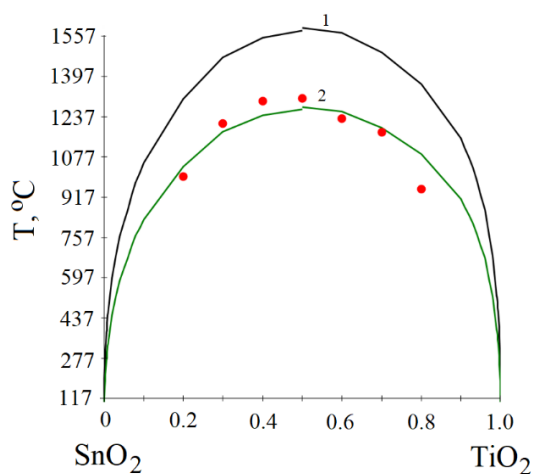
экспериментальных данных наблюдается при значениях параметров, отмеченные звездочкой в табл. 6. Большее влияние на величину $\Delta H_{см}$, ккал/моль оказывает размерный фактор.

Нереальные значения $T_{кр.}$ (рис. 15б линия 1) получены с использованием исходных параметров (табл. 6). В уравнения (7) и (8) для расчета $T_{кр.}$ входят изменяемая величина c и связанные с ней χ_i и ε как и для расчета $\Delta H_{см}$, и новая величина $\Delta R/R_{min}$ (уравнение (4)). Оптимизация этих параметров позволила вычислить новое значение $T_{кр.}$ (рис. 15б линия 2), которое на 196 °С отличалось от литературного [17]. Важно отметить, что при оптимизации величин $\Delta H_{см}$ и $T_{кр.}$ больше всего изменялись значения радиусов каждого элемента системы и электроотрицательность Na и Cl. А после вычислений стало ясно, что наибольшее значение оказывает размерный фактор (табл. 6 значения со *).

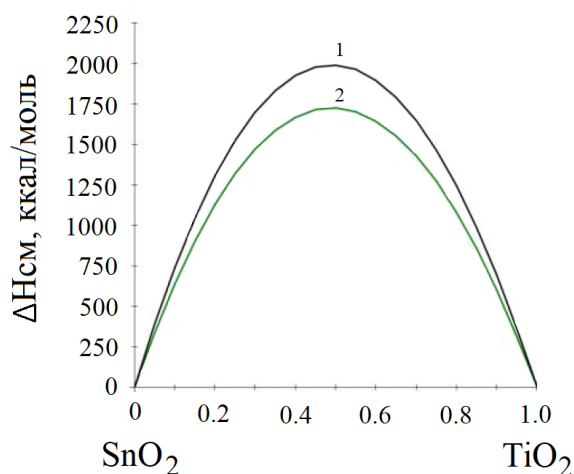
Система SnO_2 - TiO_2 . По исходным данным (табл. 7) выполнен расчет $T_{кр.}$ (уравнения (8)) в системе SnO_2 - TiO_2 (рис. 16а, кривая 1) значения которой известны из литературы (красные точки на рис. 16а) [18].

Таблица 7 – Данные для расчета термодинамической функции смешения и построения купола распада системы SnO_2 - TiO_2

Размерный фактор				Структурный фактор		Фактор химической связи								T _{кр} , °C
R ₁ , Å	R ₂ , Å	R _{общ} , Å	ΔR/R _{min} , Å	n ₁ =n ₂	A	γ ₁ , отн. ед	γ ₂ , отн. ед	γ _O , отн. ед	c, ккал/моль	ε ₁	ε ₂	Δε		
Ti	Sn	O				Ti	Sn	O		TiO ₂	SnO ₂			
0.61/ 0.597*	0.69/0. 67*	1.35/ 1.356*	0.041/ 0.037*	6	4.80	1.6	1.7/ 1.714*	3.5/ 3.48*	32.28/ 31.97*	0.73/ 0.725*	0.70/ 0.694*	0.03/ 0.031		
-0.013	-0.02	+0.006	-0.04			0	+0.014	-0.02	-0.31	-0.005	0.006	0.001		



а)



б)

Рисунок 16 – Кривые купола распада твердых растворов (а) и теплоты смешения ($\Delta H_{см}$, ккал/моль) (б) в системе SnO_2 - TiO_2 : 1 - расчет по исходным данным (табл. 7), 2 - расчет по оптимизированным данным, ● - экспериментальные значения [18].

Как видно из рис. 16а, рассчитанный купол распада твердых растворов в системе SnO_2 - TiO_2 не совпадает с экспериментальными данными [18], а согласие (рис. 16а, кривая 2) наблюдается при оптимизации значений $\Delta R/R_{min}$, \bar{A} и c , ккал/моль (отмечено звездочками в табл. 7), причем больше всего изменялись значения радиуса олова и электроотрицательность кислорода. Оптимизация величин, необходимых для оценки $\Delta H_{см}$ (уравнения (3), отмечено * в табл. 7) дала возможность построить концентрационную зависимость $\Delta H_{см}$ (рис. 16б, кривая 2).

Система TiO_2 - ZrO_2 . Аналогично по исходным данным (табл. 8) был выполнен расчет $T_{кр}$ (уравнения (8)) в системе TiO_2 - ZrO_2 (рис. 17а, кривая 1), значения которой известны из литературы (красные точки на рис. 17а) [19].

Таблица 8 – Данные для расчета термодинамической функции смешения и построения купола распада системы TiO_2 - ZrO_2

Размерный фактор				Структурный фактор		Фактор химической связи								
R ₁ , Å	R ₂ , Å	R _{обш} , Å	ΔR/R _{мин} , Å	n ₁ =n ₂	A	χ ₁ , отн. ед	χ ₂ , отн. ед	χ _O , отн. ед	c, ккал/моль	ε ₁	ε ₂	Δε	T _{кр} , °C	
Ti	Zr	O				Ti	Zr	O		TiO ₂	ZrO ₂			
0.61/ 0.605*	0.72/ 0.70*	1.35/ 1.364*	0.056/ 0.048*	6	4.80	1.6/ 1.617*	1.4	3.5/ 3.48*	33.15/ 32.83*	0.73/ 0.72*	0.78/ 0.774*	0.05/ 0.054*	3515/ 2500*	
-0.005	-0.02	+0.014	-0.008			+0.017	0	-0.02	-0.32	-0.01	-0.006	+0.004	-1015	

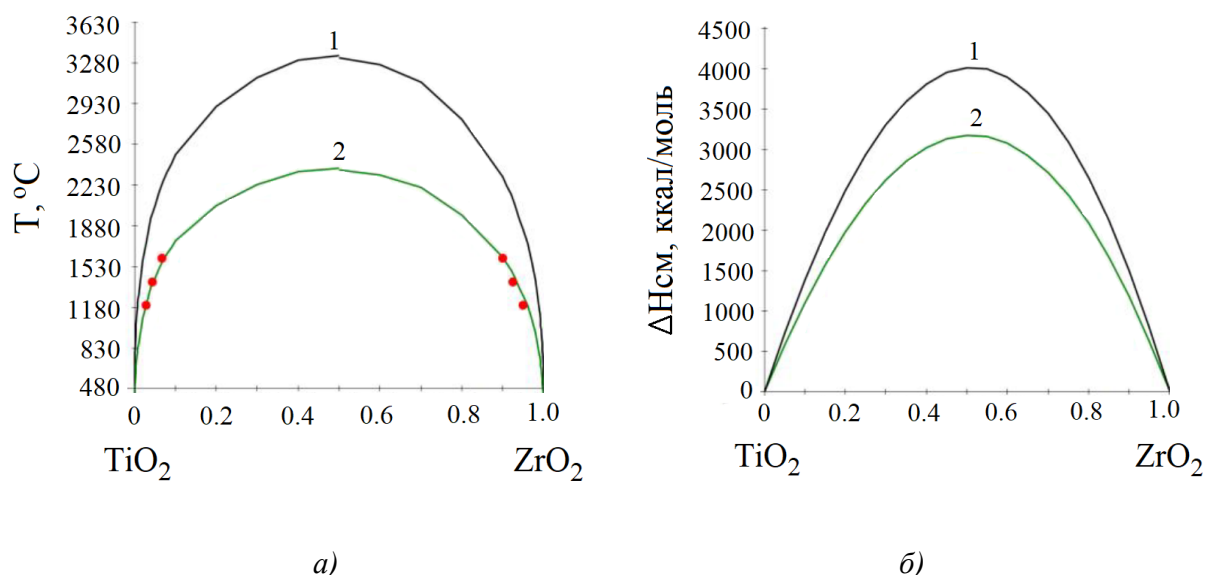


Рисунок 17 – Кривые купола распада твердых растворов (а) и теплоты смешения ($\Delta H_{см}$, ккал/моль) (б) в системе TiO_2 - ZrO_2 : 1 - расчет по исходным данным (табл. 8), 2 - расчет по оптимизированным данным, ● - экспериментальные значения [19].

Согласно Padurov et al.[19], величина $T_{кр} \sim 1330 ^\circ C$ (данные представлены с 900

°C), что хорошо согласуется с рассчитанным значением $T_{кр} = 1350$ °C (рис. 17, кривая 2). При оптимизации данных $T_{кр}$ больше всего изменялись значения радиуса циркония и электроотрицательность кислорода. Установлено, что больше влияние на величину $\Delta H_{см}$ оказывает размерный фактор.

Система SiO₂-TiO₂. Согласно исходным данным для системы SiO₂-TiO₂ (табл. 9), взятым из работы С. А. Кирилловой и др. [20] был выполнен пересчет величины $T_{кр}$ в два этапа (табл. 9): со стороны TiO₂ и со стороны SiO₂.

Таблица 9 – Данные для расчета термодинамической функции смешения и построения купола распада системы SiO₂-TiO₂

Размерный фактор				Структурный фактор		Фактор химической связи								
R ₁ , Å	R ₂ , Å	R _{общ} , Å	ΔR/R _{min} , Å	n ₁ =n ₂	A	χ ₁ , отн. ед	χ ₂ , отн.ед	χ ₀ , отн. ед	c, ккал/м оль	ε ₁	ε ₂	Δε	T _{кр} , °C	
РАСЧЕТ СО СТОРОНЫ SiO2														
Si	Ti	O				Si	Ti	O		TiO ₂	SiO ₂			
0.26/ 0.28*	0.41/ 0.39*	1.35/ 1.37*	0.093/ 0.066*	4	4.439	1.8/ 1.82*	1.6	3.5/ 3.48*	26/ 25.4*	0.59/ 0.59*	0.51/ 0.5*	0.08/ 0.09*	5181/ 2460*	
+0.02	-0.02	+0.02	-0.027			+0.02	0	-0.02	-0.6	0	-0.01	+0.01	-2721	
РАСЧЕТ СО СТОРОНЫ TiO2														
Si	Ti	O				Si	Ti	O		TiO ₂	SiO ₂			
0.40	0.61	1.35	0.12	6	4.800	1.8	1.6	3.5	31	0.73	0.68	0.05	16117	

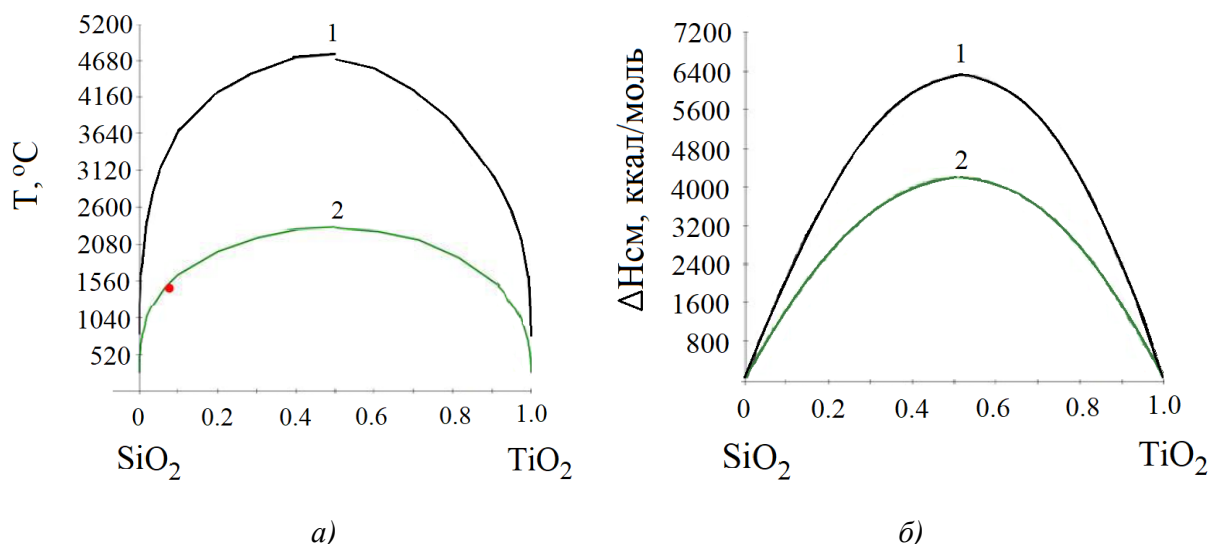


Рисунок 18 – Кривые купола распада твердых растворов (а) и теплоты смешения ($\Delta H_{\text{см}}, \text{ккал/моль}$) (б) в системе $\text{SiO}_2\text{-TiO}_2$: 1 - расчет по исходным данным (табл. 8), 2 - расчет по оптимизированным данным, • - экспериментальные значения [20].

Согласно С. А. Кирилловой и др. [20], величина $T_{\text{кр}} \sim 5181 ^\circ\text{C}$ (со стороны SiO_2) при оптимизации изменилась на $2721 ^\circ\text{C}$ (табл. 9, рис. 18а), и на ее пересчет больше всего влияло изменение значений радиусов каждого элемента системы, а также электроотрицательность кремния и кислорода. По оптимизированным данным был произведен пересчет теплоты смешения и построена концентрационная зависимость (рис. 18б). Пересчет со стороны TiO_2 оказался не корректным: пересчитанная $T_{\text{кр}} \sim 16000 ^\circ\text{C}$.

3.4 Выводы по главе

В третьей главе были рассмотрены основные детали реализации программы, выбранные алгоритмы и методы тестирования.

Заключение

В рамках данной ВКР было разработано приложение, позволяющее определить границы растворимости твердых растворов замещения с изовалентными компонентами.

В процессе выполнения ВКР были выполнены следующие задачи:

- Реализована возможность получения информации об атомах, химических соединениях и бинарных системах соединений;
- Реализована возможность изменения (добавления) данных об атомах, ионах и соединениях, а также добавления новых химических соединений/систем соединений;
- Решена задача аппроксимации табличной зависимости (полученной при экспериментальной оценке границы фаз бинарной системы) функциональной зависимостью, теоретически определяющей термодинамическую функцию смешения;
- Решена задача аппроксимации табличной зависимости (полученной при экспериментальной оценке границы фаз бинарной системы) зависимостью, теоретически определяющей купол распада;
- Реализована возможность проведения оптимизации параметров функции купола распада по заданным экспериментальным точкам и критической температуре;
- Реализована возможность проведения оценки чувствительности (влияния) параметров функции смешения на конечный результат;
- Реализована возможность, на основании полученных данных путем оптимального выбора параметров функций, построения графика свободной энергии Гиббса в заданном температурном диапазоне;
- Программа протестирована на 4-х химических системах;
- Подготовлена техническая документация.

Код приложения находится в свободном доступе и открыт для скачивания на https://github.com/kkozshakin/Activision_Mendeleyev_table.

Направления дальнейшей разработки включают в себя построение графиков свободной энергии Гиббса для каждой их проверенных систем и визуализация фазовой диаграммы. Также к направлениям дальнейшей разработки можно отнести дополнительное расширение или редактирование функционала по требованию заказчика.

Список использованных источников

- [1] Урусов В. С. Теория изоморфной смесимости. – Изд-во Наука, Москва, 1977. – 252 с.
- [2] Урусов В. С. Энергетическая кристаллохимия. – Изд-во Наука, Москва, 1975. – 335 с.
- [3] R.D. Shannon Revised effective ionic radii and systematic studies of interatomic distances in halides and chalcogenides. – Acta Crystallogr. A 32, 1976 – pp. 751-767.
- [4] Официальный сайт Microsoft Visual Studio [Электронный ресурс]. URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 02.12.2020).
- [5] Руководство по WPF [Электронный ресурс]. URL: <https://metanit.com/sharp/wpf/> (дата обращения: 02.12.2020).
- [6] Петрова Т. Г. Компьютерное моделирование локальной структуры и свойств бинарных оксидных твердых растворов со структурой типа хлористого натрия. – Москва, 2004. – 25 с.
- [7] Талис Р. А. Компьютерное атомистическое моделирование локальной структуры, свойств смешения и стабильности твердых растворов корунд-эсколаит, корунд-гематит, гематит-эсколаит. – Москва, 2007. – 70 с.
- [8] Изоморфизм. Твердые растворы. Петьков В. И., Грудзинская Е. Ю. Электронное учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2010. – 144 с.
- [9] Урусов В. С. Локальная структура твердых растворов по результатам компьютерного моделирования и экспериментальным данным / В. С. Урусов, Н. Н. Еремин // Журнал структурной химии. – 2015. – Том 56, №4. – 786-800 с.
- [10] Буданов А. Б. Теоретическое исследование механических свойств твёрдых растворов щелочных галогенидов с изоморфными замещениями в катионных и анионных подрешётках. – Москва, 2015. – 59 с.
- [11] Замятин Д. А. Кристаллохимия и спектроскопия циркона в решении вопросов его микрозондового химического U-Th-Pb-Датирования. – Екатеринбург, 2017. – 22 с.
- [12] Кузьмичева Г. М. Основные кристаллохимические категории: учебное пособие / М.: МИТХТ, 2001. – 72 с.
- [13] Проблемы кристаллологии. Выпуск 7: сборник научных статей / Н. Н. Еремин (отв. ред.) – М.: «КДУ», «Университетская книга», 2019. – 338 с.
- [14] Специфика объекта и предмета материаловедения [Электронный ресурс]. URL: https://bstudy.net/621169/tehnika/spetsifika_obekta_predmeta_materialovedeniya/ (дата обращения: 23.04.2021).
- [15] Кожакин К. Г., Подбельский В. В. «Программа активации таблицы Д. И. Менделеева».
- [16] Кожакин К. Г., Подбельский В. В. «Программа для расчета и визуализации бинарных фазовых диаграмм в системе изоструктурных компонентов».
- [17] Зенин Г.С., Привалова Т.А., Пенкина Н.В. Физическая химия: Ч.3. Фазовые равновесия и учение о растворах: Текст лекций. — СПб.: СЗПИ, 2003. — 113с.
- [18] N. N. Radurow, Die Naturwissenschaften. 1956. B.43. S.395.
- [19] Торопов Н. А., Барзаковский В.П., Курцева Н.Н., Лапин В.В. Диаграммы состояния силикатных систем. Вып. 1. Двойные системы. М., Наука. 1965. – 822 с.
- [20] С. А. Кириллова, В. И. Альмяшев, В. В. Гусаров, Спинодальный распад в системе SiO₂–TiO₂ и формирование иерархически организованных наноструктур, Наносистемы: физика, химия, математика, 2012, том 3, выпуск 2, 100–115.
- [21] Миллер В. А., Подбельский В. В. «Программа для расчета областей смесимости неорганических соединений в бинарной системе».
- [22] Википедия [Электронный ресурс]. URL: <https://ru.wikipedia.org> (дата обращения: 23.04.2021).