**КОМПЬЮТЕРНАЯ АКАДЕМИЯ «ШАГ»**

Секция: Разработка программного обеспечения

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

К ДИПЛОМНОМУ ПРОЕКТУ ПО ТЕМЕ:

Разработка Android-приложения для создания структурных формул органических соединений

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Development of an Android application for drawing up structural formulas of organic compounds

СЛУШАТЕЛЬ 24ППВ12 Головизнина Екатерина Владимировна

Goloviznina Kateryna

РУКОВОДИТЕЛЬ ПРОЕКТА Задерей Юрий Николаевич

ДИРЕКТОР ХФ КА «ШАГ» Канд. техн. наук, доцент Макушенко Т.В.

Харьков 2017 г.

**КОМПЬЮТЕРНАЯ АКАДЕМИЯ «ШАГ»**

Секция: Разработка программного обеспечения

УТВЕРЖДАЮ

Зав. учебной частью

Остапова Т.В.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2017 г.

**ЗАДАНИЕ**

НА ДИПЛОМНЫЙ ПРОЕКТ СЛУШАТЕЛЮ

Головизниной Екатерине Владимировне

1. Тема проекта:

2. Срок сдачи слушателем законченного проекта:

3. Исходные данные к проекту:

4. Содержание пояснительной записки (перечень вопросов, подлежащих  разработке):

5. Дата выдачи задания:

Руководитель

Задание принял к выполнению

СОДЕРЖАНИЕ

АННОТАЦИЯ 4

ВВЕДЕНИЕ 5

ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ 6

1. Выбор технологии реализации проекта 7

2. Разработка структуры системы 8

3. Разработка алгоритмов функционирования системы 14

4. Разработка базы данных для системы 22

5. Разработка интерфейса системы 24

6. Руководство пользователя 29

ВЫВОДЫ 32

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 33

ПРИЛОЖЕНИЯ 34

АННОТАЦИЯ

Целью данной работы являлась разработка мобильного приложения – редактора структурных формул органических соединений для операционной системы (ОС) Android.

Актуальность данного вопроса заключается практически в полном отсутствии правильно функционирующих приложений подобной направленности под ОС Android.

Данное приложение ориентированно на широкую аудиторию: школьники 9-11 классов, студенты I – VI курсов химических, химико-технологических, биологических, медицинских, экологических, геолого-географических факультетов вузов I-IV уровней аккредитации, аспирантов и преподавателей, сотрудников научно-исследовательских институтов химии, физики и биологии и всех, кто интересуется химией.

ВВЕДЕНИЕ

В современном мире химия является наукой, развитие которой не останавливается ни на минуту. Разработка новых лекарственных препаратов, открытие новых соединений, химические сенсоры и т.д. Каждый год публикуются миллионы статей в научных журналах различного профиля. И порядка 90% публикаций тем или иным образом связаны с органической химией. Это влечет за собой необходимость создания структурных формул органических соединений в качестве иллюстративного материала к научным статьям.

В настоящее время существует широкий спектр программных пакетов для Windows, предоставляющих такие возможности. Среди наиболее известных следует отметить ACD/ChemSketch, [ChemDraw,](https://www.google.com.ua/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&sqi=2&ved=0ahUKEwiK6aqU6KHSAhWGjCwKHZsICFcQFggtMAI&url=https%3A%2F%2Fru.wikipedia.org%2Fwiki%2FChemDraw&usg=AFQjCNEKYi2yydCLr9oW_g4wGJf8cI4LXw&sig2=izuPOr7GDH8ZaEVyqs6rLA&bvm=bv.147448319,d.bGg) MarvinSketch, ChemDoodle. В то же время технологии не стоят на месте и молодое поколение уже не может обходиться без смартфонов. Будущие великие ученые в настоящее время являются простыми студентами и аспирантами, которые заняты написанием дипломных работ и кандидатских диссертаций. С этим и связана необходимость делать упор на мобильные технологии.

Анализ существующих приложений-редакторов химических формул для ОС Android показывает, что качественные приложения просто отсутствуют. Это и подчеркивает актуальность проблемы создания редактора химических формул.

ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Главной целью данной работы являлось создание мобильного приложения «ChemPaint» для ОС Android, основной функцией которого являлось работа со структурными формулами органических соединений, а именно:

* создание и редактирование структурных формул с учетом всех ограничений на валентность атомов и групп, порядок и длину связи, валентные углы;
* наличие Периодической системы Д. И. Менделеева и возможностью работы с ней;
* наличие шаблонов для углеродных скелетов (алифатические, циклические и ароматические соединения);
* наличие шаблонов для функциональных групп (гидрокси-, метокси-, карбокси-, кето-, нитро-, нитрозо- и др.)
* создания пользовательских молекул с возможностью хранения в базе данных;
* экспорт полученных структурных формул в виде графических файлов.

Для выполнения поставленных задач и целей будет создано мобильное приложение «ChemPaint» средствами языка программирования C# с использованием фреймворка Xamarin, технологии ADO.NET, СУБД SQLite.

1. Выбор технологии реализации проекта

Данное приложение разрабатывалось для операционной системы Android. В настоящее время создание приложений под такую ОС возможно, в основном, на языках программирования Java, Objective-C и C#. Использование последнего предполагает работу с фреймворком Xamarin.

Xamarin – это фреймворк для кроссплатформенной разработки мобильных приложений (iOS, Android, Windows Phone) с использованием языка C#. Xamarin основан на open-source реализации платформы .NET – Mono. Эта реализация включает в себя собственный компилятор C#, среду выполнения, а также основные .NET библиотеки [1-3].

Данный фреймворк позволяет использовать привычные для пользователя C# лямбда-выражения, LINQ и др. технологии. В то же время присутствует полный доступ ко всем возможностям SDK платформы и родному механизму создания UI. Для асинхронной разработки Xamarin предоставляет возможность использовать как классы из пространства имен System.Threading.Thread и System.Threading.ThreadPool, так и полный спектр возможностей, предоставляемых Task Parallel Library [2-5].

Таким образом, его выбор является более предпочтительным, нежели разработка с использованием Java, и более интересным с точки зрения освоения новых технологий.

В качестве базы данных для разработки [6-8] была выбрана база данных SQLite [9], которая является стандартной и нет необходимости ее подключать или запрашивать разрешение на работу с ней. SQLite является легковесным фреймвoрком, который, с одной стороны, дает по максимуму использовать возможности SQL, с другой – бережно относится к ресурсам устройства. Не смотря на некоторые минусы SQLite, например, ограниченность размера базы данных или отсутствие индексирования, использование ее более предпочтительно, так как разрабатываемое приложение не требует большого размера хранилища, а многоуровневые и сложные запросы к базе данных отсутствуют.

Доступ к базе данных осуществляется при помощи технологии ADO.NET. Очевидно, что более правильным было бы использование Entity Framework, однако использование данной технологии вместе с Xamarin сейчас доступно только стадии бета-тестирования [1]. В то же время, его использование под iOS в настоящее время невозможно в принципе. Так как данное приложение разрабатывается для дальнейшей реализацией не только под Android, но и iOS, то наличие этого фреймворка может затруднить переход от одной операционной системе к другой.

2. Разработка структуры системы

Для реализации данного приложения была предложена следующая структура.

Для взаимодействия с пользователем используются три Layout (Main, TableLayout, TemplateLayout) и соответствующие им Activity (MainActivity, TableActivity, TemplateActivity).

Main и MainActivity отвечают за главное окно приложения. Именно в нем осуществляется процесс рисование молекул пользователем, сохранение и редактирование данной молекулы в базе данных. В данном окне представлены самые широко распространенные в органической химии элементы, основные типы химической связи и кнопки перехода в другие окна.

TableLayout и TableActivity отвечают за окно с Периодической системой химических элементов. В ней представлены все возможные элементы, с которыми может работать пользователь.

TemplateLayout и TemplateActivity отвечают за окно шаблонов, в котором представлены как шаблоны углеводородных скелетов (в том числе цепей, циклических и ароматических структур), так и функциональных групп.

Для обмена данными между activity имеется отдельный статический класс Data (приложение А), в котором хранятся экземпляр молекулы, с которой осуществляется работа, экземпляр базы данных, поля с указанием id объектов (элементов, шаблонов), выбранных во вспомогательных activity (параметры рисования (начальные координаты и масштаб), флаги и др).

Все возможные операции взаимодействия с базой данных представлены в классе Database, а именно создание базы данных, заполнения таблиц с шаблонами, получение молекул, атомов, связей, углов из базы данных, сохранение и открытие пользовательских молекул из базы данных.

Молекулы, с которыми работает пользователь, реализованы через несколько классов: Element, Atom, Group, Bond, Angle, Molecule (данные классы являются сериализируемыми).

Для более полного понимания химической структуры рассмотрим следующие определения. Как известно, молекулой называют электрически нейтральную частицу, образованную из двух или более связанных ковалентными связями атомов. Атомы представляют собой частицы вещества микроскопических размеров и массы, которые являются наименьшими частями химического элемента, проявляющие его свойства. В пространстве атомы соединены химическими связями, что в случае ковалентных связей представлено одной или нескольким общими электронными парами. Максимальное количество связей, которое может образовать отдельный атом характеризуется его валентностью. Расстояние между ядрами атомов в молекуле называется длиной связью, а угол, образованный двумя химическими связями, называется валентным углом. В качестве иллюстрации на Рис 2.1 представлено строение молекулы воды.

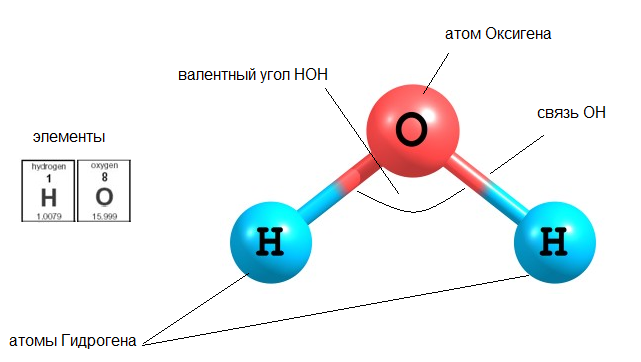


Рис 2.1. Молекула воды

Класс Element (листинг 2.1) содержит информацию об элементе Периодической системы – порядковый номер элемента, название, символ, относительную атомную массу либо нуклонное число наиболее стабильного изотопа для радиоактивных элементов и валентность.

public class Element

{

public int Id { get; set; }

public string Name { get; set; }

public double Mas { get; set; }

public string Symbol { get; set; }

public int Valence { get; set; }

}

Листинг 2.1. Класс Element

В настоящий момент известно 118 химическим элементов (Рис 2.2), существование которых официально подтверждено International Union of Pure and Applied Chemistry. Информация о них в данном приложении хранится в базе данных и в статическом поле Dictionary<int, Element> elements класса Data, где ключ словаря соответствует порядковому номеру элемента.

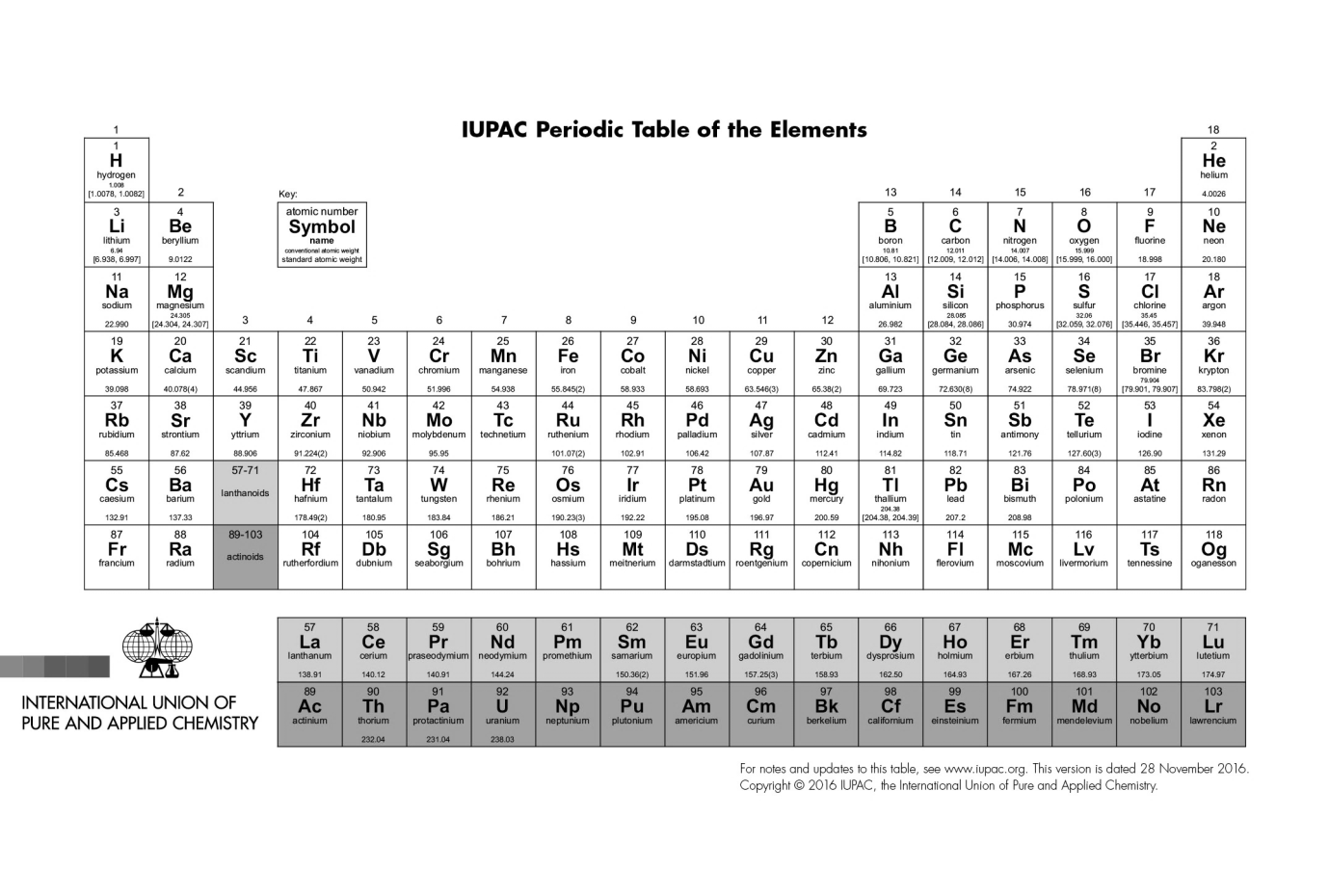


Рис 2.2. Периодическая система химических элементов (IUPAC, 28.11.2016)

Класс Atom (листинг 2.2) хранит информацию об Id атома в базе данных, его порядковом номере в молекуле, химическом элементе, которым он представлен, его заряде и координатах. Дополнительно имеются поля, указывающие, необходимо ли прорисовывать символ химического элемента и является ли он группой атомов.

public class Atom

{

public int Id { get; set; }

public int Mumber { get; set; }

public Element El { get; set; }

public int Visible { get; set; }

public int Charge { get; set; }

public float X { get; set; }

public float Y { get; set; }

public float Z { get; set; }

public bool IsGroup { get; set; }

}

Листинг 2.2. Класс Atom

От класса Atom наследуется класс Group. Атом будет рассматриваться как группа атомов в случае, если флаг IsGroup равен true. Класс Group имеет одно единственное поле типа GroupType, описание которого будет представлено ниже.

Класс Bond (листинг 2.3) хранит информацию о химической связи - ее Id в базе данных, атомах, которые образуют данную ковалентную связь, длину связи и ее тип.

public class Bond

{

public int Id { get; set; }

public Atom A1 { get; set; }

public Atom A2 { get; set; }

public double Length { get; set; }

public int Type { get; set; }

}

Листинг 2.3. Класс Bond

Класс Angle (листинг 2.4) содержит информацию об валентных углах в молекуле. Каждый валентный угол задается двумя химическими связями, представленными тремя атомами. Дополнительно указывается значение валентного угла и Id угла в базе данных.

public class Angle

{

public int Id { get; set; }

public Atom A1 { get; set; }

public Atom A2 { get; set; }

public Atom A3 { get; set; }

public double Angle { get; set; }

}

Листинг 2.4. Класс Angle

Класс Molecule (листинг 2.5) является основным и связывает классы Atom, Group, Bond и Angle в единое целое. Он содержит информацию об Id молекулы, ее названии и типе, а также словари для хранения атомов, связей и углов. В нем также имеется два конструктора и ряд функций для работы с данной молекулой.

public class Molecule

{

public int Id { get; private set; }

public string Name { get; set; }

public int Type { get; set; }

public Dictionary<int, Atom> Atoms { get; set; }

public Dictionary<int, Bond> Bonds { get; set; }

public Dictionary<int, Angle> Angles { get; set; }

…

}

Листинг 2.5. Свойства класса Molecule

Данный подход к представлению молекул в виде набора атомов, связей и углов является широко распространенным в программных пакетах, реализующих задачи как квантовой химии (Gamess, Gaussian и др.), так и молекулярно-динамического моделирования (Gromacs, Travis и др.). На листинге 2.6 представлен входной файл для молекулы воды в программном пакете Gromacs.

[ moleculetype ]

; molname nrexcl

SOL 2

[ atoms ]

; id at type res nr res name at name cg nr charge mass

1 OW\_spc 1 SOL OW 1 -0.82 15.99940

2 HW\_spc 1 SOL HW1 1 0.41 1.00800

3 HW\_spc 1 SOL HW2 1 0.41 1.00800

[ bonds ]

; i j funct length force.c.

1 2 1 0.1 345000 0.1 345000

1 3 1 0.1 345000 0.1 345000

[ angles ]

; i j k funct angle force.c.

2 1 3 1 109.47 383 109.47 383

Листинг 2.6. Пример входного файла для программного пакета Gromacs

В разделе [atoms] представлены id молекулы, а в разделах [bonds] и [angles] при помощи индексов i, j, k реализована связь, аналогичная связи «первичный ключ – внешний ключ» в базе данных. Также приведены значения длин связей (length) и валентных углов (angle).

Особенностью органических соединений является наличие функциональных групп, которые определяют принадлежность соединения к различному классу. Функциональной группой называют структурный фрагмент органической молекулы (некоторая группа атомов), определяющий её химические свойства.

В качестве примера рассмотрим молекулу ванилина. Она имеет ароматическое ядро (бензойное кольцо) и три функциональные группы – альдегидную, гидроксильную и эфирную. Данное соединение будет одновременно проявлять свойства ароматического альдегида, фенола и простого эфира (Рис 2.3).

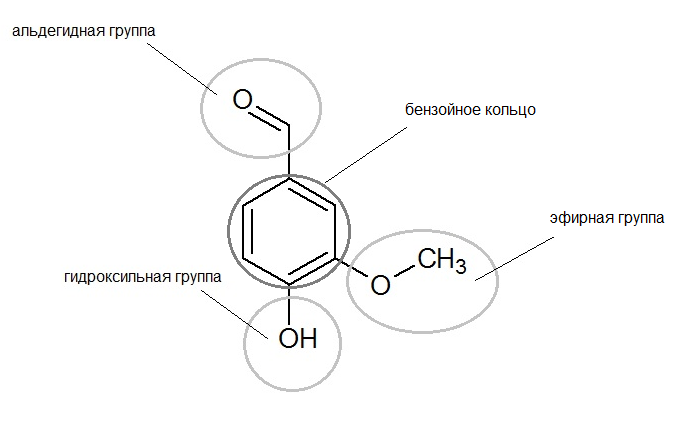


Рис 2.3. Молекула ванилина

Для описания функциональных групп имеется класс GroupType (листинг 2.7), который хранит информацию о возможных функциональных группах в молекуле: Id в базе данных, название, символ и максимально допустимую валентность.

class GroupType

{

public int Id { get; set; }

public string Name { get; set; }

public string Symbol { get; set; }

public int Valence { get; set; }

}

Листинг 2.7. Класс Angle

Информация о них в данном приложении хранится в базе данных и в статическом поле Dictionary<int, GroupType> group\_types класса Data, где ключ словаря соответствует порядковому номеру функциональной группы.

Для реализации процесса прорисовки молекулы присутствует отдельный класс DrawMolecule, который наследуется от View. Основным его методом, который широко используется в данной работе – это перегруженный метод OnDraw. Он позволяет осуществлять прорисовку углеродного скелета, кратных связей, «видимых» атомов в молекуле и функциональных групп. Данный класс представлен в виде элемента в layout Main.

3. Разработка алгоритмов функционирования системы

Прежде чем описывать алгоритмы функционирования самой системы, включающие работу с базой данных, создание и редактирование молекулы, прорисовка молекул, следует рассмотреть доступные в приложении шаблоны, функциональные группы и способы их выражения.

В первую очередь рассмотрим шаблоны. В данном приложении присутствует ряд шаблонов углеродных скелетов, а именно – цепи (этан, этен, этин, пропан, бутан, изобутан, пентан, неопентан), циклы (циклопропан, циклобутан, циклопентан, циклогексан, циклогептан, циклооктан), би- и трициклические структуры (пергидрофенантрен, пергидроантрацен, бицикло[4,3,0]нонан и декалин), ароматические структуры (бензол, нафталин, антрацен и фенантрен). Структурные формулы представлены на Рис 3.1.



Рис 3.1. Структурные формулы шаблонов углеродных скелетов

Координаты атомов для каждого шаблона хранятся в базе данных в такой системе координат, чтобы начало отсчета соответствовало точке (0;0), а длины всех связей были равны 1.

Рассмотрим процедуру расчета координат атомов на примере молекулы циклогептана. Циклогептан является правильным семиугольником. Значение внутреннего угла для многоугольника равно:

, (3.1)

где n – число углов (в случае семиугольника значение внутреннего угла составляет около 128.57º).

На Рис 3.2 представлен семиугольник с указанием основных переменных, необходимых для расчета, и полученных на их основании координат атомов. Синим цветов обозначены номера атомов, красным – отрезки, зеленым – углы.

Аналогичным образом при помощи ряда тригонометрических выражений задаются координаты всех атомов в шаблонах молекул.

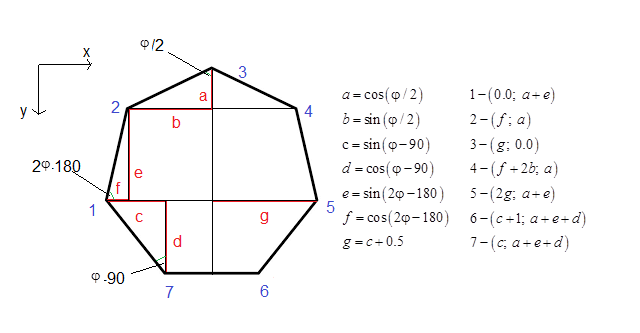


Рис 3.2. Расчет координат атомов Карбона в циклогептане

В данном приложении существует возможность добавления ряда функциональных групп в молекулу. На Рис 3.3 представлены полные формулы функциональных групп, которые доступны в приложении. Для удобства пользователя также создан отдельный интерфейс для добавления кратких формул функциональных групп в текущую молекулу.



Рис 3.3. Функциональные группы

Как было сказано ранее, функциональные группы реализованы отдельным классом Group, который является наследником класса Atom. Это позволяет их хранить в виде словаря вместе со всеми атомами в молекуле и приводить к типу группы при необходимости, путем проверки значения флага IsGroup.

При старте приложения создается экземпляр класса Database, обеспечивающего работу с базой данных, ссылка на который сохраняется в статическом классе Data. При первом запуске приложения происходит создание базы данных на устройстве, заполнение необходимых таблиц с шаблонами и функциональными, а также ряда служебных таблиц.

После создание экземпляра класса Database производится заполнение словарей elements, содержащего список всех химических элементов, group\_types, содержащего список всех возможных функциональных групп, и листа user\_molecules, содержащего названия молекул.

Необходимость хранения этих данных в отдельных коллекциях связано с высокой частотой обращения к этим данным, что влекло бы за собой снижение производительности, если бы для получения этих данных каждый раз приходилось бы обращаться с запросом в базу данных. В то же время, шаблоны молекул не хранятся в коллекциях. Обращение к ним происходит один раз за жизненный цикл молекулы и хранение целого набора молекул со всеми атомами, связями и углами было бы нецелесообразным.

Далее взаимодействие с базой осуществляется непосредственно после прямого указания пользователя. Например, создание молекулы или ее сохранение в базу данных.

Для того чтобы пользователь начал работу с приложением, ему необходимо создать молекулу и открыть существующую. Создание пользователем молекулы подразумевает выбор углеводородного скелета (шаблона) в TemplateLayout и сохранения его Id в поле cycle\_id статического класса Data. После того, как пользователь размещает молекулу во View (определяет начальные координаты молекулы путем прикосновения к экрану) происходит создание экземпляра класса Molecule и заполнение его полей (название молекулы, ее тип, словари атомов, связей и углов) в конструкторе с использованием методов класса Database.

Последующее редактирование молекулы может включать: добавление новых химических связей, изменение типа существующих и их удаление; замена атомов одних элементов на другие в молекуле; добавление новых функциональных групп.

Добавление новых химических связей происходит при прикосновении пользователя к какому-либо уже существующему атому. В зависимости от точки прикосновения пользователя к экрану смартфона в событии OnTouch для View происходит вычисление координат и осуществляется проверка, попадает ли данная координата в круг радиусом 1/3 от длины химический связи с центром в точке местонахождения атома. Если данное прикосновение относится к какому-либо атому, то вызывается метод AddNewBond у текущего экземпляра молекулы. После проверки на допустимую валентность происходит вычисление координаты нового атома с учетом поправок на тип химической связи (например, наличие тройной связи у текущего атома говорит о необходимости создание связи под углом 180º, в то время как для двойных связей данный угол составляет 120º) и возможностью не зацикливать молекулу. Далее происходит создание нового атома (или выбор существующего при неизбежном создании циклической структуры), новой химической связи и нового угла с добавлением в соответствующие словари atoms, bonds и angles экземпляра класса Molecule.

Изменение типа существующих химических связей подразумевает изменение кратности связи – связь может быть ординарной, двойной и тройной. Пользователь выбирает тип химической связи, на которую он хочет изменить, и саму связь. Тип химической связи записывается в поле bond\_type\_id класса Data. В зависимости от точки прикосновения пользователя к экрану происходит вычисление координат и соотнесение их с существующими химическими связями. Связь будет считаться выбранной только в том случае, если сумма векторов (вектора из первого атома в точку касания и вектора из второго атома в точку касания) не будет превышать 1.5 длины связи. После осуществления проверки на допустимую валентность атома происходит изменение порядка выбранной связи.

Удаление атомов, химических связей и углов возможно лишь в случае концевых атомов (атомов, имеющих только одну связь с атомом Карбона). Данное ограничение связано с тем, что при удалении неконцевых атомов будет осуществляться разбиение молекулы на несколько независимых частей, что является недопустимым. При удалении атома пользователь должен нажать соответствующую кнопку на панели управления и указать атом, подлежащий удалению. После прохождения всех проверок происходит удаление атома, связи и угла из соответствующих словарей.

Замена атомов одних элементов на другие в молекуле является возможностью создания пользовательских функциональных групп в полной их записи, в то время как шаблоны функциональных групп предоставляют возможность лишь сокращенной записи. Пользователь может выбрать атом как из панели управления в главном окне, так и атом из таблицы Менделеева в TableLayout. В обоих случаях происходит запись порядкового номера выбранного атома в поле atom\_id класса Data. Так как каждый атом может иметь различную валентность, то производится сравнение текущего количества связей у заменяемого атома и максимально возможное число связей у нового атома. После успешного прохождения данного проверки производится замена свойства типа Element в экземпляре класса Atom.

Добавление новых функциональных групп осуществляется путем выбора соответствующей функциональной группы в TemplateLayout с занесением ее Id в поле group\_id класса Data, и выбора атома, к которому будет присоединена соответствующая функциональная группа. Процесс добавление функциональной группы аналогичен созданию новой связи, однако при создании атома создается экземпляр класса Group с указанием GroupType и IsGroup = true.

В процессе рисования также доступны опции отмены и возврата. Их действие основано на создании глубокой копии экземпляра класса Molecule и его хранении в классе Data. Глубокая копия создается путем сериализации и десериализации в методе Clone (листинг 3.1) в классе Molecule, который вызывается в событии OnTouch при любом изменении молекулы.

public static object Clone(object obj)

{

using (MemoryStream buffer = new MemoryStream())

{

BinaryFormatter formatter = new BinaryFormatter();

formatter.Serialize(buffer, obj);

buffer.Position = 0;

object temp = formatter.Deserialize(buffer);

return temp;

}

}

Листинг 3.1. Метод Clone класса Molecule

Процесс прорисовки молекулы полностью реализован в методе OnDraw класса DrawMolecule, который состоит из следующих этапов:

* проверка существования молекулы для рисования;
* прорисовка ординарных С-С связей при помощи метода DrawLine, вызываемого у экземпляра класса Canvas; рисование каждой связи осуществляется на основании координат атомов, образующих данную связь в соответствии с данными в словарях atoms и bonds экземпляра класса Molecule;
* прорисовка двойных или тройных связей; если такая необходимость существует, то в зависимости от месторасположения атомов, образующих данную химическую связь, происходит рисование дополнительных параллельных линий к основной ординарной связи;
* прорисовка «видимых атомов» (под видимыми атомами понимаются все атомы, кроме атомов Карбона и Гидрогена); при необходимости происходит дорисовка атомов Гидрогена возле «видимых атомов» для правильного воспроизведения валентности атомов; данный этап осуществляется при помощи методов DrawRect и DrawText, вызываемых у экземпляра класса Canvas.

Взаимное расположение некоторых функциональных групп и «видимых атомов» имеет свои особенности, которые следует проиллюстрировать. На Рис 3.4 представлены некоторые первичные и вторичные амины. В обоих случаях можно увидеть различное расположение атомов Гидрогена, в зависимости от того, куда направлены связи C-N.

Для реализации данного правильного расположения атомов в функциональных группах в методе OnDraw отдельно проводится расчет взаимного расположения связей, примыкающей к данной функциональной группе.



Рис 3.4. Расположение атомов Гидрогена в первичной и вторичной аминогруппе

Как было сказано ранее, при работе с приложением пользователь может, как создавать новые молекулы, так и работать с уже существующими. Для этого ему необходимо выбрать соответствующий пункт меню и выбрать по названию молекулы соответствующий RadioButton в RadioGroup в появившемся AlertDialog. Заполнение RadioGroup кнопками (RadioButton) со значениями основано на названиях пользовательских молекул, хранящихся в списке user\_molecules класса Data. Если в текущий момент времени у пользователя была открыта какая-либо молекула, то будет предложена возможность ее сохранить перед тем, как открывать другую молекулу.

Пользователь может сохранять текущие молекулы, выбрав соответствующий пункт меню и записав имя молекулы в соответствующее поле. При этом если пользователь работал с уже сохраненной молекулой, то он может сохранить ее под тем же именем.

Хранение пользовательских молекул производится в отдельной таблице в базе данных. После каждого добавления сохраненной молекулы также производится обновление списка user\_molecules класса Data.

Экспорт молекулы в виде графического файла осуществляется при выборе соответствующего пункта меню. При этом необходимо задать имя молекулы, если оно не было задано (листинг 3.2).

public void SaveAsImage()

{

GetPermission();

Bitmap bitmap = v.GetDrawingCache(true);

using (System.IO.Stream stream = File.Create(Android.Provider.MediaStore.Images.Media.ExternalContentUri.Path +"/"+Data.molecule\_name+".jpg"))

{

bitmap.Compress(Bitmap.CompressFormat.Jpeg, 30, stream);

}

}

Листинг 3.2. Метод SaveAsImage класса MainActivity

4. Разработка базы данных для системы

База данных приложения реализована с использованием SQLite и состоит из 8 таблиц: Elements, Molecules, Atoms, Bonds, Angles, BondTypes, GroupTypes, UsersMolecules.

Первые 7 таблиц являются не редактируемыми. Их создание и заполнение происходит при первом старте приложения. Последняя таблица может редактироваться в процессе работы с приложением. На листинге 4.1 представлен метод создания таблиц в базе данных.

public async void CreateTables()

{

using (var conn = new SqliteConnection((connectionString)))

{

await conn.OpenAsync();

using (var command = conn.CreateCommand())

{

command.CommandText = "CREATE TABLE Elements (ElementId INTEGER PRIMARY KEY AUTOINCREMENT, AtomName NTEXT, Symbol NTEXT, Ar REAL, Valence INTEGER); ";

command.CommandText += "CREATE TABLE BondTypes (BondTypeId INTEGER PRIMARY KEY AUTOINCREMENT, TypeName NTEXT); ";

command.CommandText += "CREATE TABLE Molecules (MoleculeId INTEGER PRIMARY KEY AUTOINCREMENT, MolName NTEXT, UserType INTEGER DEFAULT 0); ";

command.CommandText += "CREATE TABLE Atoms (AtomId INTEGER PRIMARY KEY AUTOINCREMENT, ElId INTEGER NOT NULL, N INTEGER, MolId INTEGER NOT NULL, Charge INTEGER DEFAULT 0, Visible INTEGER DEFAULT 0, X REAL, Y REAL, Z REAL, FOREIGN KEY(MolId) REFERENCES Molecules(MoleculeId), FOREIGN KEY(ElId) REFERENCES Elements(ElementId)); ";

command.CommandText += "CREATE TABLE Bonds (BondId INTEGER PRIMARY KEY AUTOINCREMENT, MolId INTEGER NOT NULL, N1 INTEGER NOT NULL, N2 INTEGER NOT NULL, Length REAL, BtId INTEGER DEFAULT 0, FOREIGN KEY(MolId) REFERENCES Molecules(MoleculeId), FOREIGN KEY(BtId) REFERENCES BondTypes(BondTypeId)); ";

command.CommandText += "CREATE TABLE Angles (AngleId INTEGER PRIMARY KEY AUTOINCREMENT, MolId INTEGER NOT NULL, N1 INTEGER NOT NULL, N2 INTEGER NOT NULL, N3 INTEGER NOT NULL, Angle REAL, FOREIGN KEY(MolId) REFERENCES Molecules(MoleculeId)); ";

command.CommandText += "CREATE TABLE GroupTypes(GroupId INTEGER PRIMARY KEY AUTOINCREMENT, GroupName NTEXT, GroupSymbol NTEXT, Valence INTEGER DEFAULT 1); ";

command.CommandText += "CREATE TABLE UsersMolecules(MoleculeId INTEGER PRIMARY KEY AUTOINCREMENT, MoleculeName NTEXT NOT NULL, Molecule BLOB NOT NULL, Size INTEGER NOT NULL); ";

command.CommandType = CommandType.Text;

await command.ExecuteNonQueryAsync();

}

}

}

Листинг 4.1. Запросы создания таблиц базы данных

Таблица Elements хранит данные об элементах Периодической системы имени Д. И. Менделеева, а именно порядковый номер элемента, который совпадает с первичным ключом, название и символ, относительную атомную массу и валентность. В данной таблице имеется 118 строк, которые соответствуют 118 элементам.

Таблицы Molecules, Atoms, Bonds, Angles необходимы для хранения данных о молекулах. Взаимосвязи между данными таблицами происходят через внешний ключ «FOREIGN KEY(MolId) REFERENCES Molecules(MoleculeId)». В таблице Molecules хранится информация о молекуле и ее типе. Таблица Atoms состоит из атомов, каждый из которых представлен порядковым номером элемента, связанным с таблицей Elements по типу «внешний ключ – первичный ключ», порядковым номером данного атома в молекуле, его зарядом и координатами в пространстве. В таблице Bonds химические связи реализованы парами атомов с указанием их порядкового номера в молекуле и длиной связи, тип связи является строго заданным с использованием таблицы BondTypes. Таблица Angles характеризует валентные углы между парами химических связей, которые заданы по трем атомам, значение валентного угла также приведено в таблице.

Таблицы BondTypes и GroupTypes является служебными. В первой представлены возможные типы химической связи (одинарная, двойная, тройная и др.), а во второй содержатся возможные типы функциональных групп с указанием названия, сокращенной записи и максимально допустимой валентности.

Как можно видеть, поля данных таблиц практически полностью совпадают с полями аналогичных классов приложения.

Для хранения пользовательских молекул используется таблица UsersMolecules. Молекулы хранятся в бинарном виде в поле blob-типа, также присутствует название молекулы и объем хранимых бинарных данных. Внесение новых данных в таблицу и редактирование уже существующих осуществляет непосредственно пользователь приложения.

5. Разработка интерфейса системы

Приложение состоит из трех Layout (Main, TableLayout, TemplateLayout).

Главное окно (Рис 5.1) состоит из панели управления, области для рисования и меню.



Рис 5.1. Layout Main

Панель управления состоит из разделов Atoms и Bonds and groups. Раздел Atoms включает в себя кнопки выбора наиболее распространенных атомов (Карбон, Гидроген, Оксиген, Нитроген, Сульфур, Силиций, Хлор, Бром) и кнопку перехода в окно, содержащую таблицу Менделеева. Раздел Bonds and groups состоит из кнопок выбора типа химической связи (ординарная, двойная, тройная), кнопки перехода в окно шаблонов, кнопок отмены текущего действия, отмены и возврата, и удаления атома и связи. Удерживание любой из кнопок приводит к появлению всплывающего сообщения с информацией о функции данной кнопки.

Область для рисования представлена View и занимает большую часть экрана. Позиционирование молекулы определяется начальными координатами x\_begin и y\_begin, которые являются статическими полями класса Data. Пример нарисованной молекулы представлен на Рис 5.2.

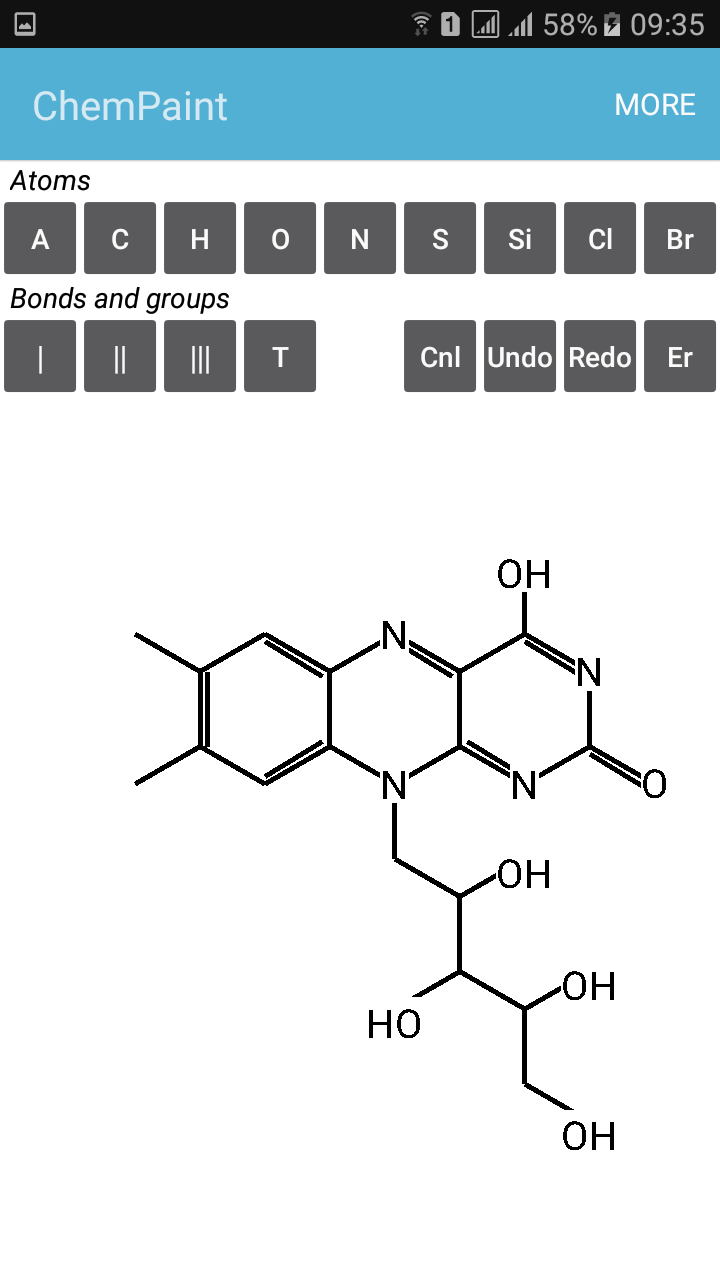


Рис 5.2. Структурная формула витамина В2

Меню (Рис 5.3) представлено элементами New (создание новой молекулы без возможности возвращения к предыдущей), Open (открыть сохраненную молекулу), Save (обновление молекулы, которая ранее была сохранения), Save as (сохранение молекулы под новым названием), Save as image (экспорт молекулы в виде графического файла в память устройства), Clear (очищения поля для рисования с возможностью восстановления предыдущей молекулы), Help (информационная справка) и Exit (выход из приложения).

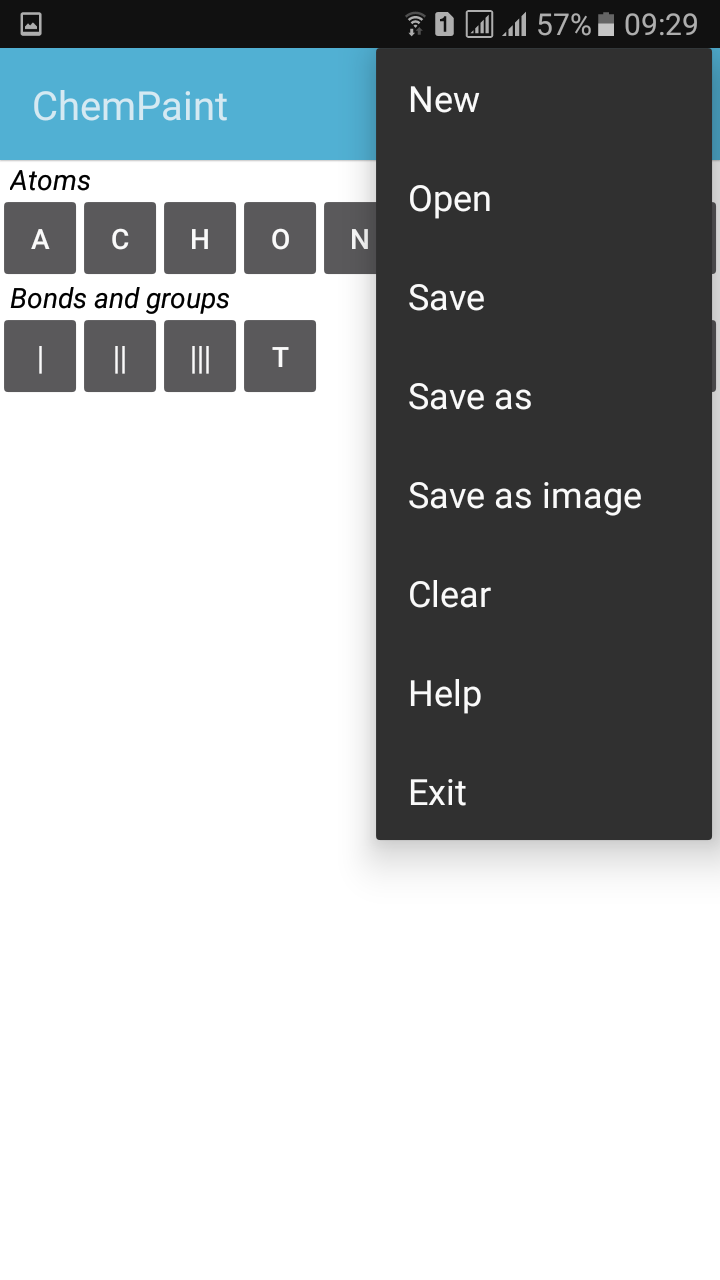


Рис 5.3. Меню

В окне TableLayout содержит в себе таблицу Менделеева, которая состоит из семи периодов. Первые три периода являются малыми, остальные – большими. В зависимости от строения электронной оболочки элементы обозначены различными цветами: s-элементы (красный), p-элементы (желтый), d-элементы (голубой), f-элементы (светло-зеленый и зеленый). Так как в настоящий момент элементорганическая химия переходных металлов, актиноидов и лантаноидов не является широко изученной, то данные группы не доступны для выбора пользователю (Рис 5.4).

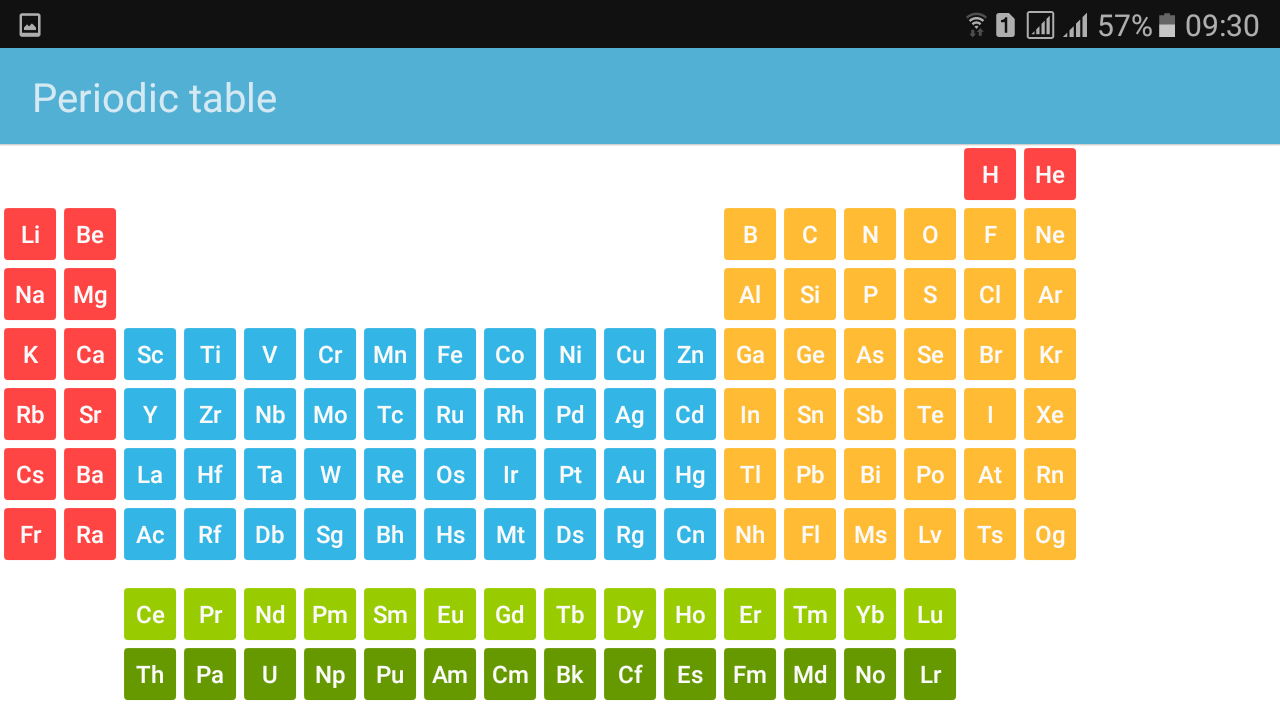


Рис 5.4. TableLayout

В данной таблице существует возможность выбора химических элементов. После выбора химического элемента пользователя возвращают в главное окно приложения, а название выбранного элемента показывается в виде всплывающего сообщения.

Окно TemplateLayout является окном шаблонов. В верхней его части расположена панель управления, нижняя часть представлена областью для рисования (Рис 5.5).

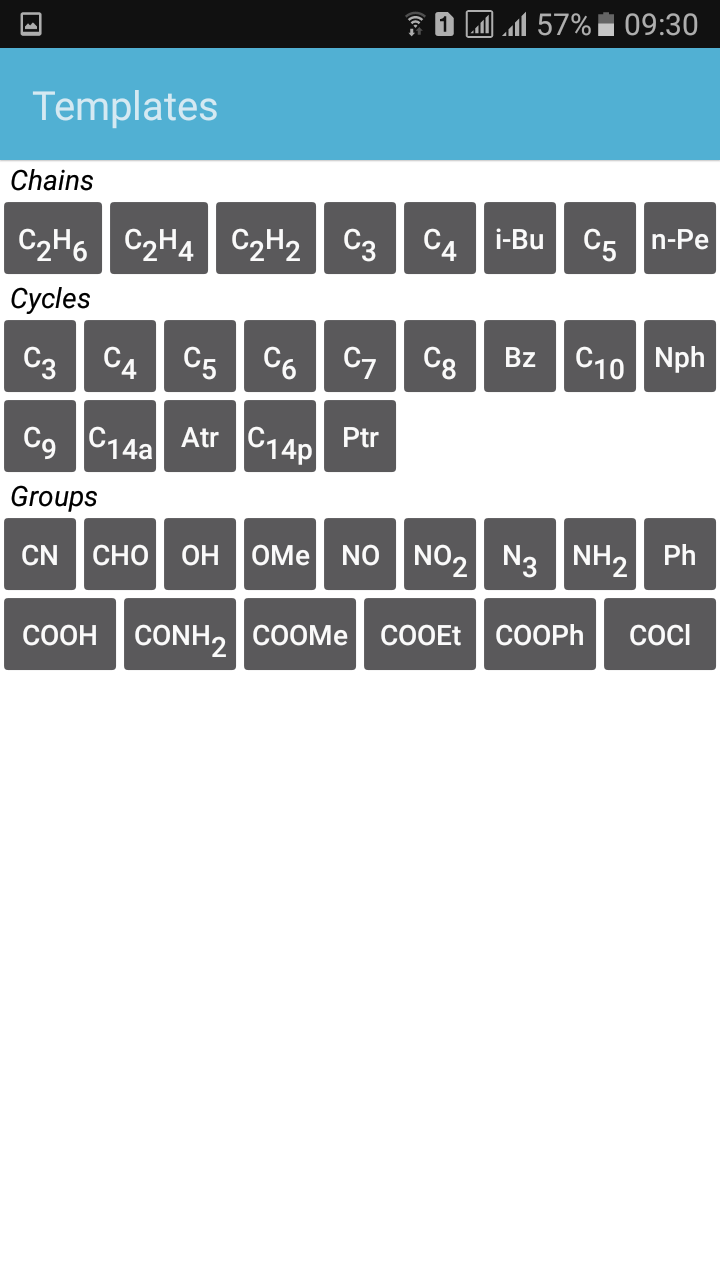


Рис 5.5. Панель управления TemplateLayout

Панель управления состоит из трех разделов, которые можно объединить в две группы по своему функциональному назначению. Первые два раздела (Chains и Cycles) содержат шаблоны для рисования структурных формул. Раздел Chains включает в себя молекулы этана, этена, этина, пропана, бутана, изобутана, пентана, неопентена; раздел Cycles - циклопропана, циклобутана, циклопентана, циклогексана, циклогептана, циклооктана, бензола, декалина, нафталина, бицикло[4,3,0]нонана, пергидроантрацена, антрацена, пергидрофенантрена и фенантрена. Структурные формулы данных соединений представлены на Рис 3.1. Удерживание любой из кнопок шаблона углеводородного скелета приводит к появлению preview данной молекулы в области для рисования, которое исчезает, когда пользователь отпускает палец.

Раздел Groups содержит функциональные группы, которые пользователь может добавить к своей молекуле, а именно циано-, формил-, гидрокси-, метокси-, нитрозо-, нитро-, азидо-, амино-, фенил-, карбокси-, амидо-, сложноэфирных групп с различным спиртовым остатком (метил-, этил-, фенил-) и хлорангидридной групп. Структурные формулы представлены на Рис 3.2.

После выбора шаблона или функциональной группы пользователь автоматически возвращается в основное окно, а данные о выбранной им структуре или фрагменте заносятся в класс Data.

6. Руководство пользователя

Данная программа предназначена для создания структурных формул органических соединений. Первая запуск программы может занять некоторое непродолжительно время, что связано с необходимостью создания хранилища шаблонов и функциональных групп на Вашем устройстве.

Для того, чтобы начать работу с приложением, зайдите в окно шаблонов и выберите интересующий Вас шаблон. Для этого нажмите кнопку «Т» в разделе Bonds and groups. Шаблоны расположены в разделах Chains и Cycles. При удерживании кнопки выбранного шаблона доступен предварительный просмотр. После выбора Вы будете перенаправлены в основное окно, где Вам необходимо будет выбрать любую точку на поле для рисования, задав начало координат. После этого выбранная молекула будет автоматически нарисована.

Для того чтобы добавить новые химические связи, прикоснитесь к атому, для которого Вы хотите создать химическую связь. В случае успешного создания химической связи она будет автоматически нарисована. Если Вы выберете атом, текущая валентность которого равна максимальной (например, максимальная валентность атома Карбона равна четырем), то во всплывающем окне Вы увидите сообщение «Inappropriate valence».

Для того, чтобы изменить тип существующей химической связи, Вам необходимо выбрать новый тип связи в разделе Bonds and groups и указать, какую химическую связь Вы хотите изменить. В случае успешной операции она будет автоматически отображена, в противном случае Вы увидите всплывающее сообщение о превышении допустимой валентности атома.

Для того, чтобы изменить тип атома, Вам необходимо выбрать, на какой элемент Вы хотите произвести замену. Для этого необходимо нажать на нужный элемент в разделе Atoms или перейти в окно с таблицей Менделеева. Для перехода в окно с таблицей Менделеева необходимо выбрать кнопку «А» в разделе Atoms. После выбора элемента будет произведен автоматический переход в главное окно. Название выбранного элемента отобразиться в виде всплывающего сообщения. После этого необходимо выбрать атом в молекуле для замены, прикоснувшись к нему. В случае успешной замены автоматически отобразиться атом, в ином случае – сообщение ошибке.

Для того, чтобы добавить функциональную группу в молекулу, Вам необходимо перейти в окно шаблонов. В окне шаблонов функциональные группы расположены в разделе Groups. После выбора функциональной группы выберите атом, к которому вы ходите ее присоединить. При этом будет создана новая химическая связь от выбранного атома к функциональной группе.

Для того, чтобы удалить атом и связь между ним и оставшейся частью молекулы, нажмите кнопку «Er». После этого выберите атом для удаления. Удаление атома возможно лишь в том случае, когда данный атом является концевым, т.е. он содержит только одну связь любой кратности. В ином случае Вы увидите всплывающее сообщение об ошибке.

На панели управления также расположены три кнопки для отмены и возврата действий. Кнопка «Cln» отменяет незаконченное действие. Например, она может отменить действие изменение атома после того, как Вы выбрали новый элемент, но до того, как Вы выбрали атом в молекуле для изменения. Кнопки «Undo» и «Redo» позволяют, соответственно, отменить и вернуть полное действие. Например, отменить действия добавления новой химической связи или изменение ее типа уже после того, как это действие было завершено.

Меню находится в верхней части главного окна приложения. Оно состоит из следующий пунктов:

* New – очистка области для рисования без возможности отмены данного действия;
* Open – открыть сохраненную пользовательскую молекулу; при этом, если в области для рисования находилась какая-либо несохраненная молекула, то программа предложит Вам сохранить ее перед открытие новой молекулы;
* Save – сохранение текущей молекулы при условии, что она уже находится в базе данных сохраненных молекул; в противном случае Вам будет предложено внести ее в базу данных с указание ее названия;
* Save as – сохранение новой молекулы в базу данных с указание ее названия;
* Save as image – сохранение молекулы в качестве графического файла; Вам также будет предложено ввести имя файла;
* Help – справочная информация по работе с программой;
* Exit – выход из приложения.

ВЫВОДЫ

В ходе проделанной работы было разработано приложение ChemPaint, которое является редактором химических формул для ОС Android. В качестве технологий реализации были выбран язык C#, фреймворк Xamarin, база данных SQLite и технология ADO.Net для взаимодействия с ней.

Данное приложение реализует весь заявленный функционал, а именно позволяет создавать и редактировать структурные формулы органических соединений, имеет ряд шаблонов углеводородных скелетов (цепей и циклов), функциональных групп, имеет встроенную периодическую систему химических элементов, позволяет сохранять и редактировать пользовательские молекулы, а также экспортировать графические файлы для дальнейшей работы с ними.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. <https://www.xamarin.com/>
2. Petzold Ch. Creating Mobile Apps with Xamarin.Forms: Cross-platform C# programming for iOS, Android, and Windows [Электронный ресурс] / Ch. Petzold. – 2016. – 310 p.
3. Peppers J. Xamarin Cross-platform Application Development / J. Peppers. – NY: Packt Publishing, 2014 . – 262 p.
4. Hermes D. Xamarin Mobile Application Development: Cross-Platform C# and Xamarin.Forms Fundamentals / D. Hermes. – L: Apress, 2015. – 510 p.
5. Miller Ch. Cross-platform Localization for Native Mobile Apps with Xamarin / Ch. Miller. – L: Apress, 2015. – 298 p.
6. Диго С.М. Базы данных: проектирование и использование [Текст]: учеб. / С.М.Диго. – М.:Финансы и статистика, 2005. – 518 с.
7. Гарсия-Молина Г. Системы баз данных. Полный курс. [Текст]: Пер. с англ./ Г.Гарсия-Молина, Дж.Ульман, Дж.Уидом. – М.:Издательский дом Вильямс, 2003. – 1088 с.
8. Кренке Д. Теория и практика построения баз данных. [Текст] / Д.Кренке - 8-е изд. – СПб.: Питер, 2003. – 800 с.
9. Serpico A. SQLite Database Programming for Xamarin: Cross-platform C# database development for iOS and Android using SQLite.XM [Электронный ресурс] / A. Serpico. – 2015. – 384 p.

ПРИЛОЖЕНИЯ

Приложение А

Класс Data

static class Data

{

static public Molecule m = null;

static public Database d = null;

static public Molecule temp\_m = null;

static public Dictionary<int, Element> elements = new Dictionary<int, Element>();

static public Dictionary<int, GroupType> group\_types = new Dictionary<int, GroupType>();

static public List<string> user\_molecules = new List<string>();

static public float X;

static public float Y;

static public float x\_begin;

static public float y\_begin;

static public float coeff;

static public int atom\_id = 0;

static public int bond\_type\_id=0;

static public int cycle\_id = 0;

static public int group\_id = 0;

static public bool erase = false;

static public bool isMolecule = false;

static public bool isSaved = false;

static public bool toSave = false;

static public bool toOpen = false;

static public bool toSaveAsImage = false;

static public string molecule\_name = "";

static public string error = "";

}