



Минобрнауки России
федеральное государственное бюджетное образовательное учреждение высшего
образования
«Санкт-Петербургский государственный технологический институт
(технический университет)»

КУРСОВОЙ ПРОЕКТ

Направление подготовки	09.03.01 Информатика и вычислительная техника
Направленность	Автоматизированные системы обработки информации и управления
Факультет	Информационных технологий и управления
Кафедра	Систем автоматизированного проектирования и управления
Учебная дисциплина	Программирование
Курс 1	Группа 494
Студент	Хлебников Роман Александрович
Тема:	Программный комплекс для определения константы скорости химической реакции $2A = 2B + C$.

Дата выдачи задания:

Дата предоставления курсового проекта к защите:

Заведующая кафедрой, проф.

Т.Б. Чистякова

Лектор, доц.

И.Г. Корниенко

Консультант, ст. преп.

А.К. Федин

Выполнил

Р.А. Хлебников

Санкт-Петербург

2020

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Направление подготовки	09.03.01 Информатика и вычислительная техника
Направленность	Автоматизированные системы обработки информации и управления
Факультет	Информационных технологий и управления
Кафедра	Систем автоматизированного проектирования и управления
Учебная дисциплина	Программирование
Курс 1	Группа 494
Студент	Хлебников Роман Александрович

Тема: Программный комплекс для определения константы скорости химической реакции $2A = 2B + C$

Цель работы: Расчёт константы скорости реакции и её оценка.

Исходные данные по проекту:

- 1 Дронов, В. А. Python 3 и PyQt 5. Разработка приложений / В. А. Дронов, Н.А. Прохоренок. — 2-е изд. — СПб.: BHV, 2019. — 832 с.
- 2 Любанович, Б. Простой Python. / Б. Любанович. — СПб.: Питер, 2017 — 480 с.
- 3 Кубасов, А. А. Химическая кинетика и катализ. Часть 1 : Учебное пособие. / А. А. Кубасов. — М.: Изд-во МГУ им. М. В. Ломоносова, 2004. — 144 с.
- 4 Иванова, Г. С. Объектно-ориентированное программирование / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачёв. — Изд. 3-е, стер. — М.: Изд-во МГТУ, 2007. — 415 с.
- 5 Чистякова, Т. Б. Синтез и анализ математических моделей кинетики химических реакций : учебное пособие / Т. Б. Чистякова, Л. В. Гольцева, А. М. Островская, Ю. В. Островский. — СПб.: СПбГТИ(ТУ), 2002. — 70 с.
- 6 Гартман, Т. Н. Основы компьютерного моделирования химико-технологических процессов : учебное пособие / Т. Н. Гартман, Д. В. Клушин. — М.: Академкнига, 2008. — 70 с.

Перечень вопросов, подлежащих разработке:

- 1 Ознакомиться с теорией по синтезу и анализу математических моделей (ММ) кинетики химических реакций.
- 2 Выполнить постановку задачи по исследованию кинетики химической реакции.
- 3 Составить формализованное описание задачи. Разработать структуру входных (экспериментальных данных по изменению концентрации компонентов во времени) и выходных данных (порядок реакции, константа скорости реакции).
- 4 Уточнить методы решения математической задачи.
- 5 Разработать алгоритм (блок-схему) для определения порядка химической реакции и константы скорости химической реакции.
- 6 Спроектировать структуру программы.
- 7 Разработать пользовательский интерфейс.

- 8 Выполнить предварительную оценку кинетических констант с использованием метода регрессионного анализа.
- 9 Провести статический анализ результатов.
- 10 Составить программу, реализующую поставленную задачу.
- 11 Описать структуру данных и алгоритмов.
- 12 Описать структуру программы.
- 13 Протестировать работоспособность программного обеспечения.

Перечень графического материала:

- 1 Постановка задачи исследования кинетики химической реакции.
- 2 Формализованное описание задачи. Структура входных и выходных данных.
- 3 Метод регрессионного анализа для решения задач.
- 4 Алгоритм (блок-схема) определения порядка химической реакции и константы скорости химической реакции.
- 5 Функциональная структура программного обеспечения.
- 6 Результаты проверки адекватности разработанной модели.
- 7 Примеры тестирования работоспособности программного обеспечения.
- 8 Характеристика разработанного программного обеспечения.

Требования к программному обеспечению:

Программное обеспечение: Microsoft Windows 10, MS Office Word 2019, MS Office Power Point 2019, Python 3.7.9, Qt Designer, JetBrains PyCharm 2020.1

Требования к аппаратному обеспечению:

Шестиядерный процессор Intel Core I7 8750H с тактовой частотой 2.2 ГГц., 16 ГБ оперативной памяти, от 10 Гб до 20 Гб свободного места на жестком диске, видеокарта Nvidia GeForce GTX 1060

Дата выдачи задания:

Дата предоставления курсового проекта к защите:

Заведующая кафедрой, проф.

Лектор, доц.

Консультант, ст. преп.

Задание принял к выполнению

Т.Б. Чистякова

И.Г. Корниенко

А.К. Федин

Р.А. Хлебников

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 АНАЛИТИЧЕСКИЙ ОБЗОР	6
1.1 Обзор и анализ предметной области	6
1.2 Обзор и обоснование выбора инструментальных средств разработки программного комплекса для реализации генетического алгоритма.	7
2 ЦЕЛЬ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА	8
3 ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОГРАММНОГО КОМПЛЕКСА.....	9
4 СТРУКТУРА ДАННЫХ.....	10
5 АЛГОРИТМ ОПРЕДЕЛЕНИЯ ПОРЯДКА ХИМИЧЕСКОЙ РЕАКЦИИ И КОНСТАНТЫ СКОРОСТИ ХИМИЧЕСКОЙ РЕАКЦИИ	11
6 ОПИСАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА	13
7 ПРОВЕРКА АДЕКВАТНОСТИ МАТЕМАТИЧЕСКОЙ МОДЕЛИ КИНЕТИКИ	15
8 ПРИМЕР ТЕСТИРОВАНИЯ РАБОТОСПОСОБНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ..	15
9 ОПИСАНИЕ СТРУКТУР ДАННЫХ И АЛГОРИТМОВ	18
10 ОПИСАНИЕ СТРУКТУРЫ ПРОГРАММЫ.....	19
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	20
ПРИЛОЖЕНИЕ	20

ВВЕДЕНИЕ

В мире программирования существует множество языков программирования. С каждым годом их становится всё больше и больше. Наиболее известными и распространёнными языками являются C++, C#, Java, Python, Pascal, PHP. Этими языками пользуются, так как они удобны для программирования, и они могут работать с комплексными структурами данных.

Наибольшей популярностью пользуются языки с объектно-ориентированными возможностями. Объектно-Ориентированное Программирование (ООП) – это парадигма разработки программных систем, в которой проекты состоят из объектов и классов. Объектно-Ориентированное Программирование стало неотъемлемой частью разработки многих современных проектов. Оно способствует лучшей управляемости проектом во время разработки и его тестированию. Код таких проектов легко читается и быстро пишется.

Одним из наиболее распространенных объектно-ориентированных языков высокого уровня является Python, который был создан Гвидо ван Россумом в 1991 году. Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой набор полезных функций.

Тема моего курсового проекта «Программный комплекс для определения константы скорости химической реакции $2A = 2B + C$ ». Расчёт скорости протекания реакции и факторов, от которых зависит эта скорость, очень важная задача, так как по этим данным можно узнать много о самой химической реакции. Расчётом скорости реакции занимается раздел физической химии, называемый химической кинетикой.

Также одним из важнейших факторов являются концентрации химических веществ реакции в любой момент времени. Обычно чем выше концентрация веществ, тем выше скорость реакции.

В данной работе создаётся программный комплекс для расчёта константы скорости реакции, порядка реакции, дисперсии и корреляции. Для реализации используются формулы из химической кинетики. Результатом курсового проекта является гибкий программный комплекс, который рассчитывает константу скорости химической реакции. Наглядно показывает на графике зависимость концентрации от времени. Позволяет визуально оценить изменения концентраций всех веществ, содержащихся в данной химической реакции. Рассчитывает порядок реакции, дисперсию, корреляцию.

1 АНАЛИТИЧЕСКИЙ ОБЗОР

1.1 Обзор и анализ предметной области

Химическая кинетика – раздел физической химии, изучающий закономерности протекания химической реакции во времени, зависимости этих закономерностей от внешних условий, а также механизмы химических превращений. Предметом химической кинетики является изучение всех факторов, влияющих на скорость как всего процесса, так и всех промежуточных стадий.

Важным понятием химической кинетики является скорость химической реакции. Эта величина определяет, как изменяется концентрация компонентов реакции с течением времени. Скорость химической реакции — величина всегда положительная, поэтому, если она определяется по исходному веществу (концентрация которого убывает в процессе реакции), то полученное значение умножается на -1.

Скорость химической реакции w_r , которая определяется как число молекул, реагирующих в единицу времени, в единице объёма. Скорость реакции определяется по формуле:

$$w_r = -\frac{1}{V} \frac{dN_j}{dt}$$

Если объём постоянен, то эту формулу выражают через концентрацию:

$$w_r = -\frac{d(N_j/V)}{dt} = \frac{dC_j}{dt},$$

где N_j – число молей j-го компонента, C_j – концентрация j-го компонента.

Скорость элементарной химической реакции пропорциональна концентрациям реагирующих веществ в степенях, равных их стехиометрическим коэффициентам.

Порядок реакции по данному веществу — показатель степени при концентрации этого вещества в кинетическом уравнении реакции.

Порядок реакции определяется как сумма показателей степеней кинетического уравнения:

$$w_{ri} = k_i \prod_{j=1}^{m_i} C_{A_j}^{p_{ij}}$$
$$n_i = \sum_{j=1}^{m_i} p_{ij}.$$

При $n = 1$, $k = [1 / \text{с}]$; при $n = 2$, $k = [\text{м}^3 / \text{моль} \cdot \text{с}]$, $p_{ij} = 2$.

1.2 Обзор и обоснование выбора инструментальных средств разработки программного комплекса для реализации генетического алгоритма.

Для реализации поставленной задачи разработано программное обеспечение, включающее графический пользовательский интерфейс. Для написания кода программы был выбран язык Python. В качестве интегрированной среды разработки программного обеспечения (Integrated Development Environment, IDE) используется JetBrains PyCharm и Qt designer для создания пользовательского интерфейса.

PyCharm — интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA.

Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем путём простой компиляции программы для каждой системы без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Является полностью объектно-ориентированным, расширяемым и поддерживающим технику компонентного программирования.

Для удобной работы Python в связке Qt была использована библиотека PyQt. PyQt — набор расширений (биндингов) графического фреймворка Qt для языка программирования Python, выполненный в виде расширения Python.

2 ЦЕЛЬ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА

Целью выполнения курсового проекта является закрепление знаний и навыков в области алгоритмизации, принципов объектно-ориентированного программирования, разработки, отладки и тестирования программных продуктов на языках высокого уровня. Задачами курсового проекта являются проектирование и создание прикладного программного обеспечения, позволяющего определять порядок химической реакции, константу скорости реакции и проводить статистический анализ результатов.

- а) Ознакомиться с теорией по синтезу и анализу математических моделей кинетики химических реакций;
- б) Выполнить постановку задачи по исследованию кинетики химической реакции;
- в) Составить формализованное описание задачи. Разработать структуру входных (экспериментальных данных по изменению концентрации компонентов во времени) и выходных данных (порядок реакции, константа скорости реакции);
- г) Уточнить методы решения математической задачи;
- д) Разработать алгоритм (блок-схему) для определения порядка химической реакции и константы скорости химической реакции;
- е) Спроектировать структуру программы;
- ж) Разработать пользовательский интерфейс;
- и) Выполнить предварительную оценку кинетических констант с использованием метода регрессионного анализа;
- к) Провести статический анализ результатов;
- л) Составить программу, реализующую поставленную задачу;
- м) Протестировать работоспособность программного обеспечения.

3 ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОГРАММНОГО КОМПЛЕКСА

Самыми важными структурными компонентами программы являются модули, в которых находится программный код, и связанные с ними экранные формы. Модуль осуществляет обработку команд пользователя, поступающих с соответствующей формы. Программный код модуля состоит из взаимосвязанных подпрограмм (процедур), каждая из которых выполняет какую-либо конкретную задачу.

Формализованное описание задачи определения констант скоростей химической реакции, как объекта программирования, позволяет создать программный комплекс, обладающий следующими характеристиками:

- а) Операционная система, ОС (англ. operating system) – базовый комплекс компьютерных программ, обеспечивающий управление аппаратными средствами компьютера, работу с файлами, ввод и вывод данных, а также выполнение прикладных программ и утилит: Microsoft Windows 10;
- б) Язык программирования: Python – интерпретируемый строго типизированный язык программирования общего назначения;
- в) Два разработанных классов (class):
 - 1) MainWindow– класс главного окна программы;
 - 2) MyDelegate– класс делегата таблиц главного окна программы;
- г) Количество потоков выполнения команд (нитей, threads) - одна последовательность хода управления внутри программы: 1;
- д) Распределённость приложения: локальная;
- е) Инструментальная среда разработки: JetBrains Pycharm;
- ж) Количество исходных файлов программы: 2 (main.py, interface.ui).

4 СТРУКТУРА ДАННЫХ

Исходными данными являются:

- а) Изменение концентрации $C_A(t)$ во времени;
- б) Начальные концентрации C_B и C_C ;
- в) Количество экспериментальных точек N .

Результатом расчёта являются данные:

- а) Порядок реакции;
- б) Константа скорости химических реакций;
- в) Дисперсия;
- г) Коэффициент корреляции.

5 АЛГОРИТМ ОПРЕДЕЛЕНИЯ ПОРЯДКА ХИМИЧЕСКОЙ РЕАКЦИИ И КОНСТАНТЫ СКОРОСТИ ХИМИЧЕСКОЙ РЕАКЦИИ

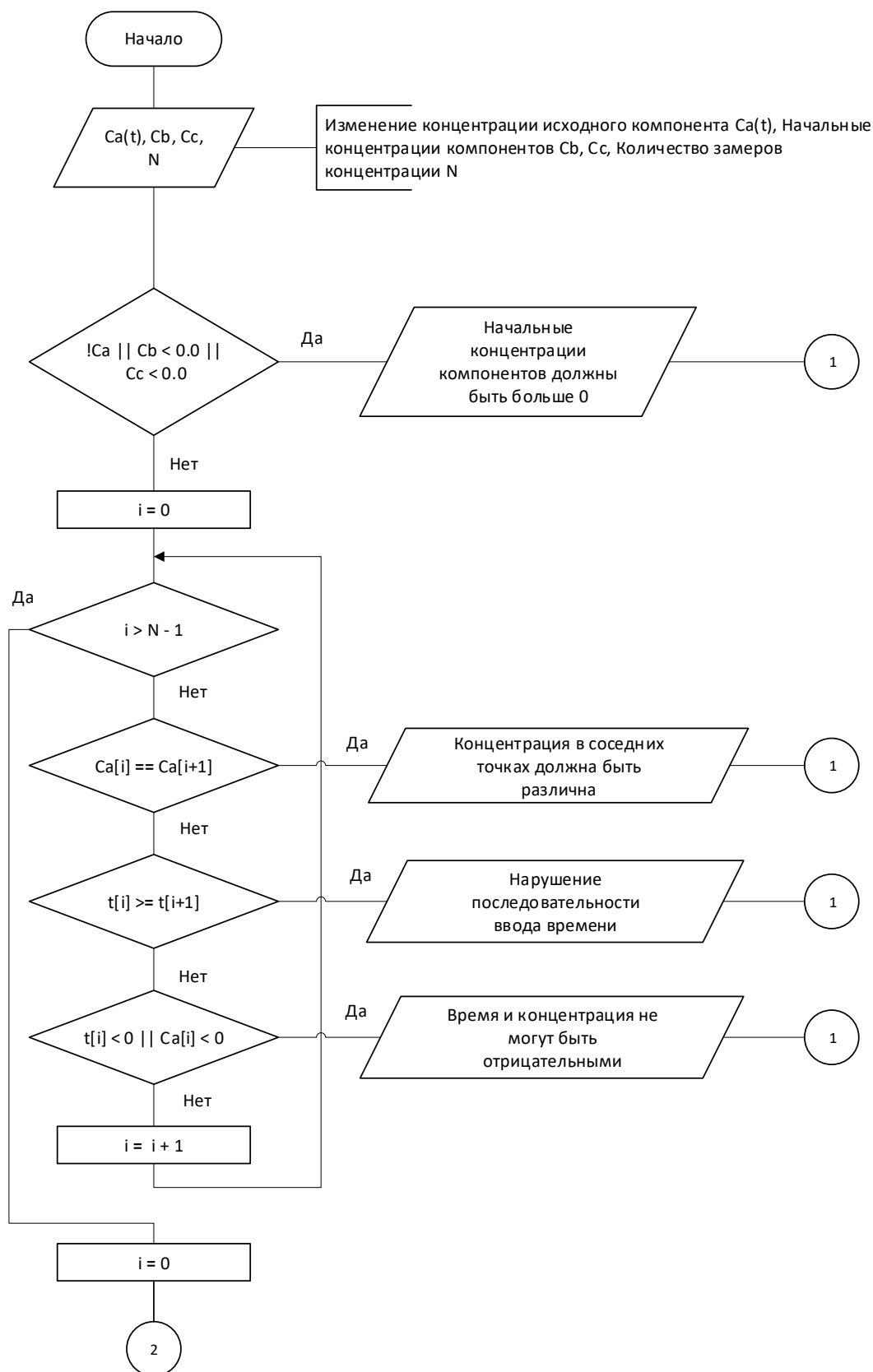


Рисунок 1, лист 1 – Блок-схема основного алгоритма программы

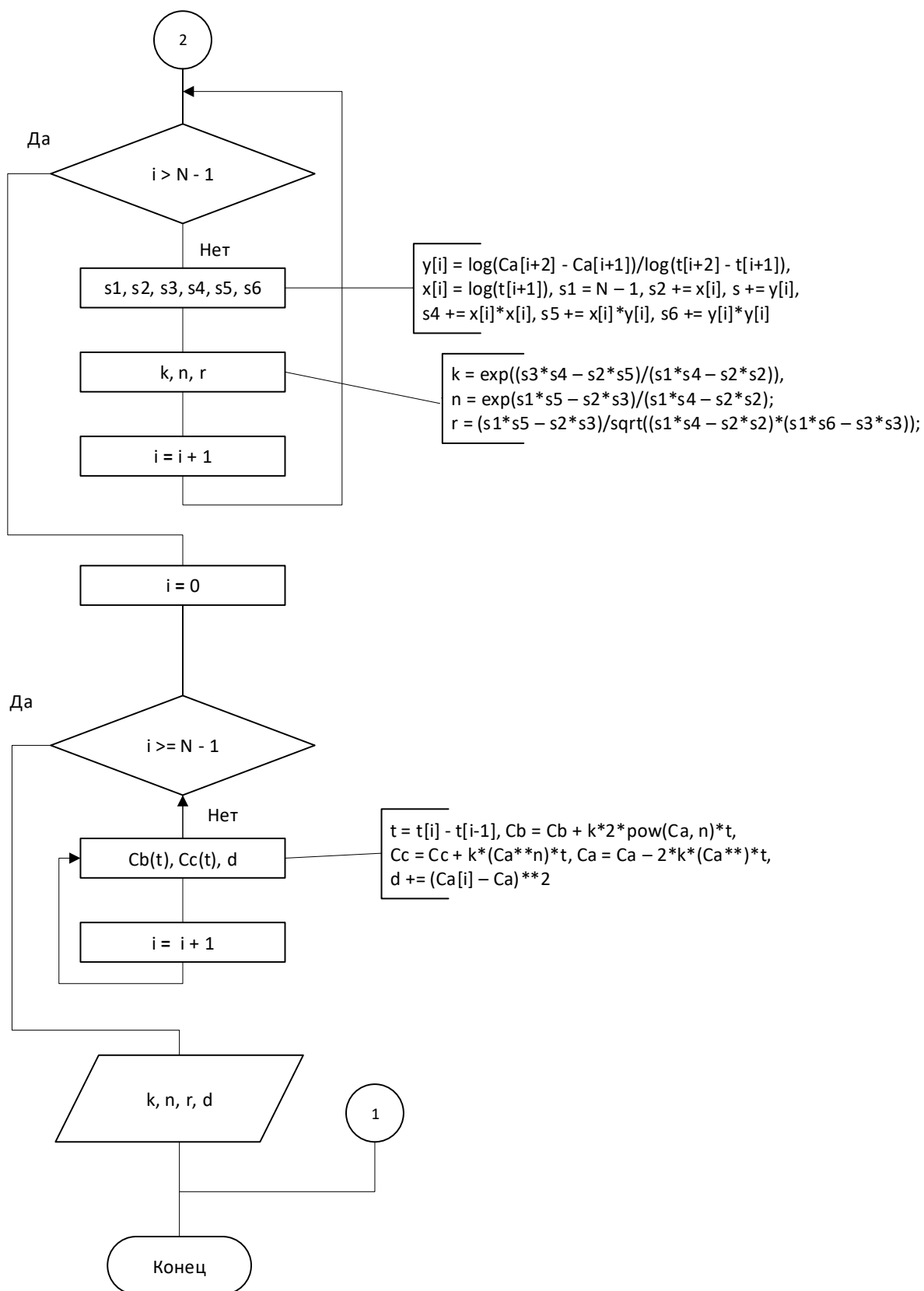


Рисунок 1, лист 2

6 ОПИСАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Программа состоит из одной, позволяющей вводить данные и отображать результат расчёта. Программа включает два исходных файла: `main.py`, `interface.ui`. Графический интерфейс включает следующие классы визуальных компонентов: формы `QmainWidget`, панели `TPanel`, группы элементов `QGroupBox`, метки `QLabel`, график `PlotWidget`, кнопки `QPushButton`, компоненты для ввода и изменения числовых значений `QDoubleSpinBox` и `QSpinBox`, таблица `QTableWidget`, диалог для сохранения данных `QFileDialog`.

Форма `MainWindow` – это основное окно, с которым будет работать пользователь. Здесь происходит ввод исходных данных, расчёт всех параметров и вывод конечных результатов на экран. В этом окне также присутствуют функции сохранения результата в файл, вывод окна со справкой о работе и авторе программы. Интерфейс программы представляет собой несколько областей, а его реализация является достаточно простой и интуитивно ясной для пользователей. Основная часть программы разделена на две области – «Ввод исходных данных» и график зависимости концентрации от времени. В области «Ввод исходных данных» вводятся концентрации реагентов и количество точек. Если исходные данные введены неверно, то появляется сообщение о некорректном вводе. С помощью стрелок на счётчике можно увеличить или уменьшить количество точек на графике. Также изменится количество строк в таблице с результатом и в таблице с концентрациями C_b и C_c от времени. После нажатия кнопки «Провести расчёты» на экране появятся графики. Графики содержат четыре линии, которые отражают изменение концентраций C_a , C_b и C_c от времени, также содержит точки аппроксимации, выделенные красным цветом. Под графиками создается таблица с изменением рассчитанных значений концентраций C_a , C_b и C_c .

В результате работы данного программного комплекса на экран выводятся следующие данные: график; константа скорости химических реакций; дисперсия; порядок реакции; коэффициент корреляции. Для построения графиков в программу был заложен метод наименьших квадратов.

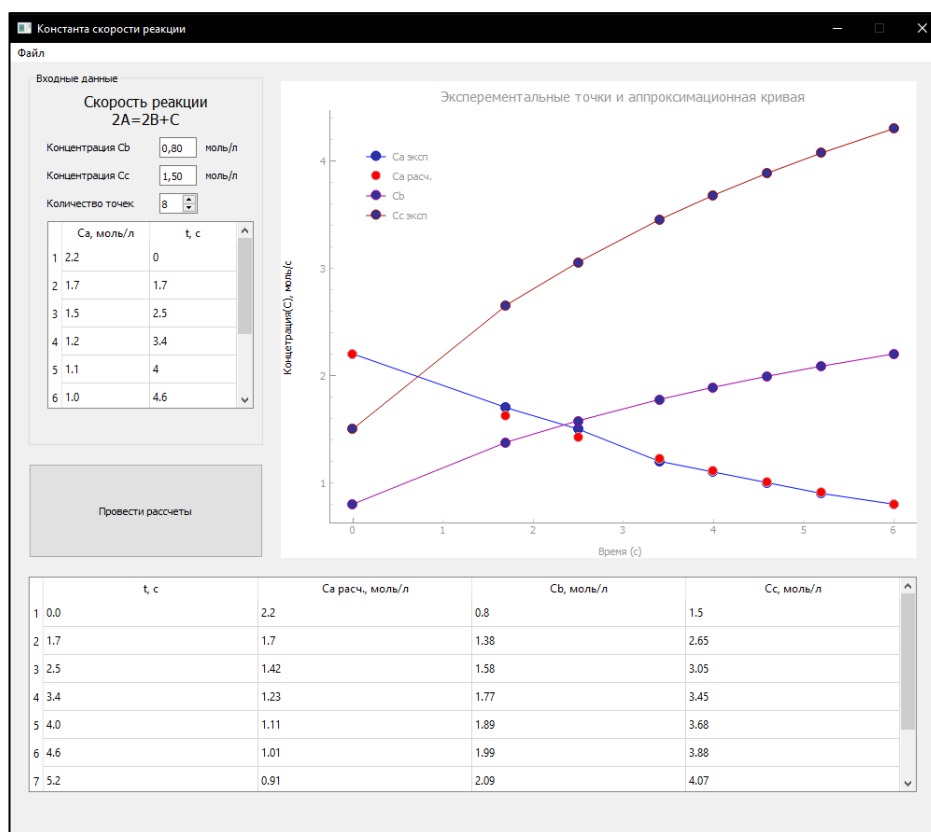


Рисунок 2 – Главное окно приложения

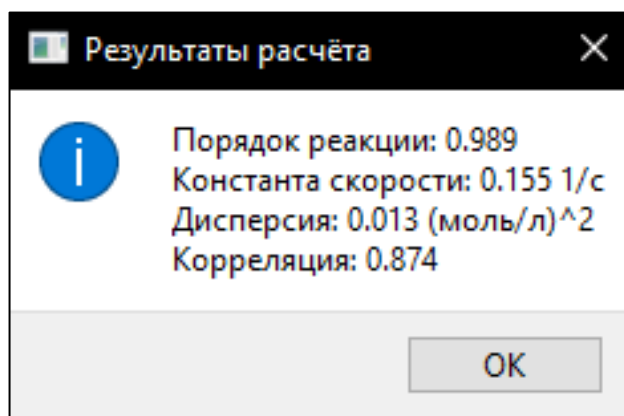


Рисунок 3 – Окно с результатами расчёта

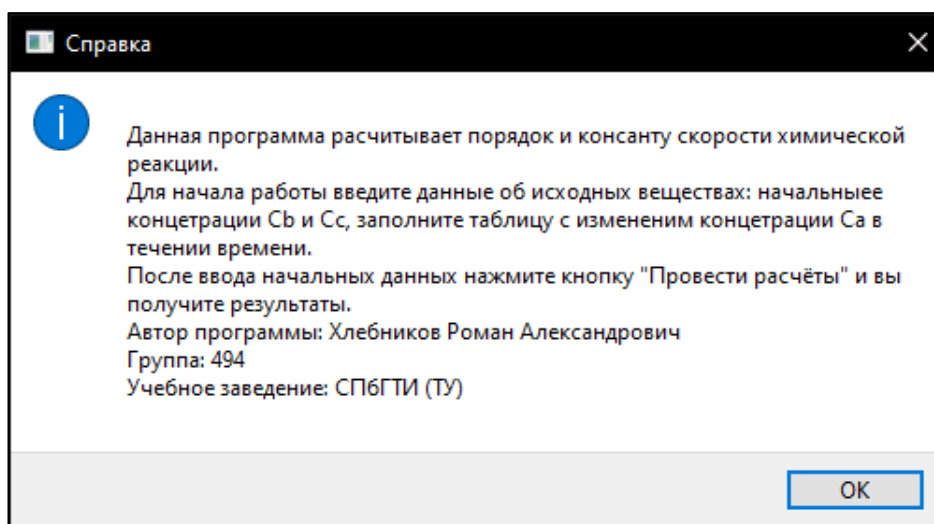


Рисунок 4 – Окно со справкой о программе

7 ПРОВЕРКА АДЕКВАТНОСТИ МАТЕМАТИЧЕСКОЙ МОДЕЛИ КИНЕТИКИ

Полученное значение коэффициента корреляции $r = 0,874$ говорит о том, что связь между экспериментальными данными и расчётами, выполненными в математической модели существенная.

Полученное значение дисперсии $d = 0,013 \text{ (моль/л)}^2$ – мало, что также свидетельствует о малом разбросе данных, полученных при использовании математической модели, относительно данных, полученных в результате эксперимента.

Из этих данных можно сделать вывод, что математическая модель может быть использована для определения константы скорости химической реакции.

8 ПРИМЕР ТЕСТИРОВАНИЯ РАБОТОСПОСОБНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Тестирование разработанного в среде PyCharm программного обеспечения проводилось при входных данных:

- а) Концентрация $C_b = 0,8 \text{ моль/л}$.
- б) Концентрация $C_c = 1,5 \text{ моль/л}$.
- в) Количество точек 8.

Таблица 1 – Входные данные концентрации и времени

Концентрация C_a , моль/л	Время t , с
2,2	0
1,7	1,7
1,5	2,5
1,2	3,4
1,1	4
1	4,6
0,9	5,2
0,8	6

Таблица 2 – Зависимость концентраций реагентов от времени

Время t , с	Концентрация C_a расч., моль/л	Концентрация C_b расч., моль/л	Концентрация C_c расч., моль/л	Концентрация C_d расч., моль/л
0	2,2	0,8	1,5	2
1,7	1,6	1,4	2,7	2,1
2,5	1,4	1,6	3,1	2,2
3,4	1,2	1,8	3,4	2,2
4	1,1	1,9	3,7	2,3
4,6	1	2	3,9	2,3
5,2	0,91	2,1	4,1	2,3
6	0,8	2,2	4,3	2,3

Входные данные

Скорость реакции
 $2A=2B+C$

Концентрация C_b моль/л

Концентрация C_c моль/л

Количество точек

	C_a , моль/л	t , с
1	2.2	0
2	1.7	1.7
3	1.5	2.5
4	1.2	3.4
5	1.1	4
6	1.0	4.6

Рисунок 5 – Форма ввода исходных данных

Результаты расчёта

Порядок реакции: 0.989
 Константа скорости: 0.155 1/с
 Дисперсия: 0.013 (моль/л)²
 Корреляция: 0.874

OK

Рисунок 6 – Форма отображения результатов расчёта

Результаты расчёта:

Порядок реакции $n = 1$ (0,989)

Константа скорости реакции $k = 0,155$ 1/с

Дисперсия $d = 0,013$ (моль/л)²

Коэффициент корреляции $r = 0,874$

9 ОПИСАНИЕ СТРУКТУР ДАННЫХ И АЛГОРИТМОВ

Таблица 3 – Переменные, используемые в программе

Имя	Тип	Описание
i	int	Счетчик
reg_up	int	Проверка допустимых значений
reg_down	int	Проверка допустимых значений
s1	float	Общее число опытов
s2	float	Сумма всех измеренных величин x
s3	float	Сумма всех измеренных величин y
s4	float	Сумма квадрата, измеренных величин x
s5	float	Сумма произведения измеренных величин x и y
s6	float	Сумма квадрата, измеренных величин y
ca	float	Концентрация вещества А
cb	float	Концентрация вещества В
cc	float	Концентрация вещества С
d	float	Дисперсия
k	float	Константа скорости реакции
n	float	Порядок реакции
r	float	Коэффициент корреляции
t	float	Время
*x	float	Временная координата абсцисс
*y	float	Временная координата ординат

10 ОПИСАНИЕ СТРУКТУРЫ ПРОГРАММЫ

Программа разбита на 2 модуля: main.py с основной программой и interface.ui с разметкой интерфейса.

Таблица 4 – Функции, составляющие main.py (основная программа)

Имя	Описание
<code>__init__</code>	Инициализирует форму
<code>save</code>	Сохраняет данные (кнопка «Сохранить данные»)
<code>faq</code>	Выводит справку (кнопка «Справка»)
<code>changeTableSize</code>	Изменяет размер таблицы при изменении <code>spinbox_2</code>
<code>dataIsBad</code>	Очищает таблицу с данными, удаляет и делает неактивным график
<code>plot</code>	Запускает вычисления (кнопка «Провести расчеты»)
<code>showErrorMessage</code>	Выводит сообщений об ошибке
<code>getCell</code>	Получает данные из таблицы

Таблица 6 – Классы, используемые в программе

Имя	Описание
<code>MyDelegate</code>	Класс делегата таблиц главного окна программы
<code>MainWindow</code>	Класс главного окна программы;

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта было создано прикладное программное обеспечение, позволяющее определять порядок химической реакции, константу скорости реакции и проводить статистический анализ результатов.

Был разработан алгоритм, позволяющий на основании исходных данных определить константу скорости реакции, порядок реакции, коэффициент корреляции и

дисперсии. Исходными данными являются: изменение концентрации исходного компонента $C_a(t)$ во времени, начальные концентрации компонентов C_b и C_c , количество экспериментальных точек N .

Для реализации алгоритма решения обратной задачи кинетики было разработано программное обеспечение, включающее графический пользовательский интерфейс. В качестве интегрированной среды разработки программного обеспечения (Integrated Development Environment, IDE) используется JetBrains PyCharm.

Программа состоит из одной формы, позволяющей вводить начальные данные и отображать результаты расчёта. Программа включает два исходных файла: `main.py` и `interface.ui`.

Было проведено тестирование работоспособности программного продукта и проведена оценка адекватности математической модели по полученным значениям дисперсии и корреляции. Также было определено, что математическая модель может быть использована для определения константы скорости химической реакции.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1 Дронов, В. А. Python 3 и PyQt 5. Разработка приложений / В. А. Дронов, Н.А. Прохоренок. — 2-е изд. — СПб.: BHV, 2019. — 832 с.

2 Любанович, Б. Простой Python. / Б. Любанович. — СПб.: Питер, 2017 — 480 с.

3 Кубасов, А. А. Химическая кинетика и катализ. Часть 1 : Учебное пособие. / А. А. Кубасов. — М.: Изд-во МГУ им. М. В. Ломоносова, 2004. — 144 с.

4 Иванова, Г. С. Объектно-ориентированное программирование / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачёв. — Изд. 3-е, стер. — М.: Изд-во МГТУ, 2007. — 415 с.

5 Чистякова, Т. Б. Синтез и анализ математических моделей кинетики химических реакций : учебное пособие / Т. Б. Чистякова, Л. В. Гольцева, А. М. Островская, Ю. В. Островский. — СПб.: СПбГТИ(ТУ), 2002. — 70 с.

6 Гартман, Т. Н. Основы компьютерного моделирования химико-технологических процессов : учебное пособие / Т. Н. Гартман, Д. В. Клушин. — М.: Академкнига, 2008. — 70 с.

ПРИЛОЖЕНИЕ

1. main.py

```
import os
from math import log, fabs, sqrt, exp

import pyqtgraph as pg
from PyQt5.QtWidgets import QTableWidgetItem, QMessageBox, QApplication
from pyqtgraph.Qt import QtCore, QtGui
from tabulate import tabulate

pg.mkQApp()

# сглаживание для графиков
pg.setConfigOptions(antialias=True)

# Подключение файла с интерфейсом
path = os.path.dirname(os.path.abspath(__file__))
uiFile = os.path.join(path, 'interface.ui')
WindowTemplate, TemplateBaseClass = pg.Qt.loadUiType(uiFile)

# Функция для более удобного доступа к ячейке
def getCell(table, n, m):
    return (float(table.item(n, m).text()))

# Функция для удобного показа сообщений
def showErrorMessage(text, title):
    msg = QMessageBox()
    msg.setIcon(QMessageBox.Critical)
    msg.setText(text)
    msg.setWindowTitle(title)
    msg.exec_()

# Делегат таблицы(ячейки таблицы после объявления делегата становятся объектами
QDoubleSpinBox)

class MyDelegate(QtGui.QItemDelegate):

    def createEditor(self, parent, option, index):
        return QtGui.QDoubleSpinBox(parent, maximum=1000, decimals=3,
buttonSymbols=QtGui.QSpinBox.NoButtons)
```

```

# Класс главного окна
class MainWindow(TemplateBaseClass):
    # Определение инициализации формы
    def __init__(self):
        TemplateBaseClass.__init__(self)
        self.setWindowTitle('Расчет константы скорости химической реакции 2A= 2B +
C')

    # Create the main window
    self.ui = WindowTemplate()
    self.ui.setupUi(self)
    delegate = MyDelegate()
    self.ui.tableWidget.setItemDelegate(delegate)
    self.ui.plot.setBackground('w')
    self.ui.plot.addLegend()
    self.ui.plot.setLabel('bottom', 'Время (с)')
    self.ui.plot.setLabel('left', 'Концентрация(C), моль/с', labelTextSize='50pt')
    self.ui.plot.setTitle("Экспериментальные точки и аппроксимационная кривая")
    self.ui.plot.setEnabled(False)
    # Установка начальных значений
    self.ui.tableWidget.setColumnCount(2)
    self.ui.tableWidget.setRowCount(8)
    self.ui.tableWidget.setHorizontalHeaderLabels(["Ca, моль/л", "t, с"])

    self.ui.tableWidget.setItem(0, 0, QTableWidgetItem("2.2"))
    self.ui.tableWidget.setItem(1, 0, QTableWidgetItem("1.7"))
    self.ui.tableWidget.setItem(2, 0, QTableWidgetItem("1.5"))
    self.ui.tableWidget.setItem(3, 0, QTableWidgetItem("1.2"))
    self.ui.tableWidget.setItem(4, 0, QTableWidgetItem("1.1"))
    self.ui.tableWidget.setItem(5, 0, QTableWidgetItem("1.0"))
    self.ui.tableWidget.setItem(6, 0, QTableWidgetItem("0.9"))
    self.ui.tableWidget.setItem(7, 0, QTableWidgetItem("0.8"))

    self.ui.tableWidget.setItem(0, 1, QTableWidgetItem("0"))
    self.ui.tableWidget.setItem(1, 1, QTableWidgetItem("1.7"))
    self.ui.tableWidget.setItem(2, 1, QTableWidgetItem("2.5"))
    self.ui.tableWidget.setItem(3, 1, QTableWidgetItem("3.4"))
    self.ui.tableWidget.setItem(4, 1, QTableWidgetItem("4"))
    self.ui.tableWidget.setItem(5, 1, QTableWidgetItem("4.6"))
    self.ui.tableWidget.setItem(6, 1, QTableWidgetItem("5.2"))
    self.ui.tableWidget.setItem(7, 1, QTableWidgetItem("6"))

    self.ui.tableWidget_2.setColumnCount(4)

```

```

        self.ui.tableWidget_2.setHorizontalHeaderLabels(["t, c", "Ca расч., моль/л",
"Cb, моль/л", "Cc, моль/л"])

        for i in range(self.ui.tableWidget_2.columnCount()):
            self.ui.tableWidget_2.horizontalHeader().setResizeMode(i,
QtGui.QHeaderView.Stretch)

        for i in range(self.ui.tableWidget.columnCount()):
            self.ui.tableWidget.horizontalHeader().setResizeMode(i,
QtGui.QHeaderView.Stretch)

        # Подключение функций к элементам интерфейса
        self.ui.pushButton.clicked.connect(self.plot)
        self.ui.spinBox_2.valueChanged.connect(self.changeTableSize)
        self.ui.action.triggered.connect(self.save)
        self.ui.action_2.triggered.connect(qApp.quit)
        self.ui.action_3.triggered.connect(self.faq)

        self.show()

        self.n = 0
        self.k = 0
        self.d = 0
        self.r = 0
        self.isDataNormal = False

        # функция сохранения результата
        def save(self):
            if self.isDataNormal:
                text = "Порядок реакции: " + str(round(self.n, 3)) + "\nКонстанта скорости:
" + str(
                    round(self.k, 3)) + " 1/c\nДисперсия: " + str(round(self.d, 3)) + "
(моль/л)^2\nКорреляция: " + str(
                    round(self.r, 3)) + "\n\nТаблица вычисленных значений\n"
                values = [["t, c", "Ca расч., моль/л", "Cb, моль/л", "Cc, моль/л"]]
                for i in range(self.ui.tableWidget_2.rowCount()):
                    values.append([])
                    for j in range(self.ui.tableWidget_2.columnCount()):
                        values[i + 1].append(str(getCell(self.ui.tableWidget_2, i, j)))
                text += tabulate(values)

                fileName = QtGui.QFileDialog.getSaveFileName(self, "Сохранить результаты",
"Результаты вычислений",
                                                                    "Текстовый документ
(*.txt)")[0]
                print(fileName)

```

```

        with open(fileName, 'w') as fp:
            fp.write(text)
    else:
        showErrorMessage("Сначала должны быть произведены расчеты.", "Ошибка
сохранения файла")

```

```

# функция вызова справки
def faq(self):
    msg = QMessageBox()
    msg.setIcon(QMessageBox.Information)
    msg.setText("""

```

Данная программа рассчитывает порядок и константу скорости химической реакции.

Для начала работы введите данные об исходных веществах: начальные концентрации C_b и C_c , заполните таблицу с изменением концентрации C_a в течении времени.

После ввода начальных данных нажмите кнопку "Провести расчёты" и вы получите результаты.

Автор программы: Хлебников Роман Александрович

Группа: 494

Учебное заведение: СПбГТИ (ТУ)

```

        """)
    msg.setWindowTitle("Справка")
    msg.exec_()

```

```

# функция изменяющая размер таблицы при изменении значения количества точек
def changeTableSize(self):
    s = self.ui.spinBox_2.value()
    self.ui.tableWidget.setRowCount(s)

```

```

# функция очищающая таблицы, график
def dataIsBad(self):
    while self.ui.tableWidget_2.rowCount() > 0:
        self.ui.tableWidget_2.removeRow(0)
    self.ui.plot.plot(clear=True)
    self.isDataNormal = False
    self.ui.plot.setEnabled(False) # выключение графика

```

```

# функция вычисления
def plot(self):
    try:
        self.d = 0
        spinboxvalue = self.ui.spinBox_2.value()
        self.ui.tableWidget_2.setRowCount(spinboxvalue)
        s1 = spinboxvalue - 1
        reg_down = reg_up = 0
        s2 = s3 = s4 = s5 = s6 = 0
        x = []

```



```

y = []

ca = getCell(self.ui.tableWidget, 0, 0)
cb = (self.ui.doubleSpinBox.value())
cc = (self.ui.doubleSpinBox_2.value())

if ca < 0 or cb < 0 or cc < 0:
    self.dataIsBad()
    showErrorMessage("Начальные концентрации компонентов должны быть
больше 0.", "Введены неверные данные")
    return
for i in range(spinboxvalue - 1):
    if getCell(self.ui.tableWidget, i, 0) > getCell(self.ui.tableWidget,
i + 1, 0):
        reg_down += 1
    else:
        reg_up += 1
    if getCell(self.ui.tableWidget, i, 0) == getCell(self.ui.tableWidget,
i + 1, 0):
        self.dataIsBad()
        showErrorMessage("Концетрации в соседних точках должна быть
различна.", "Введены неверные данные")
        return
    if getCell(self.ui.tableWidget, i, 1) >= getCell(self.ui.tableWidget,
i + 1, 1):
        self.dataIsBad()
        showErrorMessage("Нарушение последовательности ввода времени.",
"Введены неверные данные")
        return
    if getCell(self.ui.tableWidget, i, 1) < 0 or
getCell(self.ui.tableWidget, i, 0) < 0:
        self.dataIsBad()
        showErrorMessage("Время и концентрация не могут быть
отрицательными.", "Введены неверные данные")
        return

for i in range(spinboxvalue - 1):
    y.append(log((fabs(getCell(self.ui.tableWidget, i + 1, 0) -
(getCell(self.ui.tableWidget, i, 0)))) / (
        getCell(self.ui.tableWidget, i + 1, 1) -
getCell(self.ui.tableWidget, i, 1))))
    x.append(log(getCell(self.ui.tableWidget, i, 0)))
    s2 += x[i]
    s3 += y[i]
    s4 += x[i] * x[i]
    s5 += x[i] * y[i]

```

```

        s6 += y[i] * y[i]
self.k = exp((s3 * s4 - s2 * s5) / (s1 * s4 - s2 * s2))
self.n = (s1 * s5 - s2 * s3) / (s1 * s4 - s2 * s2)
self.r = (s1 * s5 - s2 * s3) / sqrt((s1 * s4 - s2 * s2) * (s1 * s6 - s3 *
s3))

series1 = [], []
series2 = [], []
series3 = [], []
series4 = [], []

for i in range(spinboxvalue):
    series1[0].append(getCell(self.ui.tableWidget, i, 1))
    series1[1].append(getCell(self.ui.tableWidget, i, 0))
    self.ui.tableWidget_2.setItem(1, 1,
QTableWidgetItem(str(getCell(self.ui.tableWidget, 1, 0))))
    self.ui.tableWidget_2.setItem(i, 0,
QTableWidgetItem(str(getCell(self.ui.tableWidget, i, 1))))

    if ca < 0:
        self.dataIsBad()
        showMessage("Модель невозможно описать линейной регрессией",
"Ошибка")

        return
    if i == 0:
        series2[0].append(getCell(self.ui.tableWidget, i, 1))
        series2[1].append(ca)
        self.ui.tableWidget_2.setItem(i, 1,
QTableWidgetItem(str(round(ca, 2))))
        series3[0].append(getCell(self.ui.tableWidget, i, 1))
        series3[1].append(cb)
        self.ui.tableWidget_2.setItem(i, 2,
QTableWidgetItem(str(round(cb, 2))))
        series4[0].append(getCell(self.ui.tableWidget, i, 1))
        series4[1].append(cc)
        self.ui.tableWidget_2.setItem(i, 3,
QTableWidgetItem(str(round(cc, 2))))
    else:
        t = getCell(self.ui.tableWidget, i, 1) -
getCell(self.ui.tableWidget, i - 1, 1)
        cb = cb + self.k * (ca ** self.n) * t
        cc = cc + self.k * (ca ** self.n) * t * 2

        series3[0].append(getCell(self.ui.tableWidget, i, 1))
        series3[1].append(cb)
        self.ui.tableWidget_2.setItem(i, 2,
QTableWidgetItem(str(round(cb, 2))))

```

```

        series4[0].append(getCell(self.ui.tableWidget, i, 1))
        series4[1].append(cc)
        self.ui.tableWidget_2.setItem(i,
3,
QTableWidgetItem(str(round(cc, 2))))
        if reg_down > reg_up:
            ca = ca - self.k * (ca ** self.n) * t
        else:
            ca = ca + self.k * (ca ** self.n) * t
        series2[0].append(getCell(self.ui.tableWidget, i, 1))
        series2[1].append(ca)
        self.ui.tableWidget_2.setItem(i,
1,
QTableWidgetItem(str(round(ca, 2))))
        self.d += (getCell(self.ui.tableWidget, i, 0) - ca) ** 2

allCValues = []
for i in range(spinboxvalue):
    for j in range(1, 4):
        allCValues.append(getCell(self.ui.tableWidget_2, i, j))
xmax = getCell(self.ui.tableWidget_2, spinboxvalue - 1, 0)
ymax = max(allCValues)

# установка ограничений для графика
self.ui.plot.setLimits(xMin=-1, xMax=xmax * 1.2,
                        minXRange=0.1, maxXRange=xmax * 1.2,
                        yMin=0, yMax=ymax * 1.2,
                        minYRange=0.1, maxYRange=ymax * 1.2)

# отрисовка линий
self.ui.plot.plot(series1[0], series1[1], clear=True, pen='#000DFF',
symbolPen='#000DFF', name='Ca эксп')
self.ui.plot.plot(series2[0], series2[1], pen=(0,0,0,0), symbolBrush="r",
name='Ca расч.')
self.ui.plot.plot(series3[0], series3[1], pen='#A600A0',
symbolPen='#A600A0', name='Cb')
self.ui.plot.plot(series4[0], series4[1], pen='#A60600',
symbolPen='#A60600', name='Cc эксп')
# вывод сообщения
msg = QMessageBox()
msg.setIcon(QMessageBox.Information)
msg.setText("Порядок реакции: " + str(round(self.n, 3)) + "\nКонстанта
скорости: " + str(
        round(self.k, 3)) + " 1/с\nДисперсия: " + str(round(self.d, 3)) + "
(моль/л)^2\nКорреляция: " + str(
        round(self.r, 3)))
msg.setWindowTitle("Результаты расчёта")
msg.exec_()

```

```

        self.isDataNormal = True
        self.ui.plot.setEnabled(True)
    except:
        showErrorMessage("Произошла непредвиденная ошибка, попробуйте изменить
данные", "Ошибка")

win = MainWindow()

if __name__ == '__main__':
    import sys

    if (sys.flags.interactive != 1) or not hasattr(QtCore, 'PYQT_VERSION'):
        QtGui.QApplication.instance().exec_()

```

2. interface.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>MainWindow</class>
<widget class="QMainWindow" name="MainWindow">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>1000</width>
      <height>850</height>
    </rect>
  </property>
  <property name="minimumSize">
    <size>
      <width>1000</width>
      <height>850</height>
    </size>
  </property>
  <property name="maximumSize">
    <size>
      <width>1000</width>
      <height>850</height>
    </size>
  </property>
  <property name="baseSize">
    <size>
      <width>2</width>
      <height>0</height>
    </size>
  </property>

```

```

    </size>
</property>
<property name="windowTitle">
    <string>PµPSPSCfC,P°PSC,P° C fPePsCßPSCfC,Pë CßPuP°PeC†PëPë</string>
</property>
<widget class="QWidget" name="centralwidget">
    <widget class="QGroupBox" name="groupBox">
        <property name="geometry">
            <rect>
                <x>20</x>
                <y>10</y>
                <width>251</width>
                <height>401</height>
            </rect>
        </property>
        <property name="title">
            <string>P'C...PsPrPSC< Pµ PrP°PSPSC< Pµ</string>
        </property>
        <widget class="QTableWidget" name="tableWidget">
            <property name="geometry">
                <rect>
                    <x>20</x>
                    <y>160</y>
                    <width>221</width>
                    <height>201</height>
                </rect>
            </property>
            <property name="baseSize">
                <size>
                    <width>0</width>
                    <height>0</height>
                </size>
            </property>
            <property name="gridStyle">
                <enum>Qt::SolidLine</enum>
            </property>
        </widget>
        <widget class="QDoubleSpinBox" name="doubleSpinBox">
            <property name="geometry">
                <rect>
                    <x>140</x>
                    <y>70</y>
                    <width>40</width>
                    <height>22</height>
                </rect>
            </property>
            <property name="buttonSymbols">
                <enum>QAbstractSpinBox::NoButtons</enum>
            </property>
            <property name="specialValueText">
                <string/>
            </property>
            <property name="correctionMode">
                <enum>QAbstractSpinBox::CorrectToPreviousValue</enum>
            </property>
            <property name="prefix">

```

```

    <string/>
  </property>
  <property name="suffix">
    <string/>
  </property>
  <property name="value">
    <double>0.8000000000000000</double>
  </property>
</widget>
<widget class="QDoubleSpinBox" name="doubleSpinBox_2">
  <property name="geometry">
    <rect>
      <x>140</x>
      <y>100</y>
      <width>40</width>
      <height>22</height>
    </rect>
  </property>
  <property name="buttonSymbols">
    <enum>QAbstractSpinBox::NoButtons</enum>
  </property>
  <property name="value">
    <double>1.5000000000000000</double>
  </property>
</widget>
<widget class="QLabel" name="label">
  <property name="geometry">
    <rect>
      <x>20</x>
      <y>70</y>
      <width>103</width>
      <height>21</height>
    </rect>
  </property>
  <property name="text">
    <string>РѡPsPSC†PuPSC, CБP°C†PёCЦ Cb</string>
  </property>
</widget>
<widget class="QLabel" name="label_2">
  <property name="geometry">
    <rect>
      <x>20</x>
      <y>100</y>
      <width>103</width>
      <height>21</height>
    </rect>
  </property>
  <property name="text">
    <string>РѡPsPSC†PuPSC, CБP°C†PёCЦ Cc    </string>
  </property>
</widget>
<widget class="QLabel" name="label_5">
  <property name="geometry">
    <rect>
      <x>190</x>
      <y>70</y>

```

```

        <width>47</width>
        <height>21</height>
    </rect>
</property>
<property name="text">
    <string>PjPsP»CH/P»</string>
</property>
</widget>
<widget class="QLabel" name="label_6">
    <property name="geometry">
        <rect>
            <x>190</x>
            <y>100</y>
            <width>47</width>
            <height>21</height>
        </rect>
    </property>
    <property name="text">
        <string>PjPsP»CH/P»</string>
    </property>
</widget>
<widget class="QSpinBox" name="spinBox_2">
    <property name="geometry">
        <rect>
            <x>140</x>
            <y>130</y>
            <width>41</width>
            <height>22</height>
        </rect>
    </property>
    <property name="minimum">
        <number>1</number>
    </property>
    <property name="value">
        <number>8</number>
    </property>
</widget>
<widget class="QLabel" name="label_10">
    <property name="geometry">
        <rect>
            <x>20</x>
            <y>130</y>
            <width>101</width>
            <height>21</height>
        </rect>
    </property>
    <property name="text">
        <string>PЉPsP»PĖC†PµCŦC,PIPs C,PsC†PµPe</string>
    </property>
</widget>
<widget class="QLabel" name="label_3">
    <property name="geometry">
        <rect>
            <x>20</x>
            <y>20</y>
            <width>211</width>

```

```

        <height>41</height>
    </rect>
</property>
<property name="font">
    <font>
        <pointsize>12</pointsize>
    </font>
</property>
<property name="text">
    <string>PŸPePsCᄁPsCᄁC, Cᄁ CᄁPᄁP°PeC†PᄁPᄁ
2A=2B+C</string>
</property>
<property name="textFormat">
    <enum>Qt::AutoText</enum>
</property>
<property name="alignment">
    <set>Qt::AlignCenter</set>
</property>
</widget>
</widget>
<widget class="QTableWidget" name="tableWidget_2">
    <property name="geometry">
        <rect>
            <x>20</x>
            <y>551</y>
            <width>951</width>
            <height>231</height>
        </rect>
    </property>
</widget>
<widget class="QPushButton" name="pushButton">
    <property name="geometry">
        <rect>
            <x>20</x>
            <y>430</y>
            <width>251</width>
            <height>101</height>
        </rect>
    </property>
    <property name="text">
        <string>PᄁCᄁPsPIPᄁCᄁC, Pᄁ CᄁP°CᄁCᄁC†PᄁC, C<</string>
    </property>
</widget>
<widget class="PlotWidget" name="plot">
    <property name="geometry">
        <rect>
            <x>290</x>
            <y>20</y>
            <width>681</width>
            <height>511</height>
        </rect>
    </property>
    <property name="interactive">
        <bool>true</bool>
    </property>
</widget>

```



```

</widget>
<widget class="QMenuBar" name="menubar">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>1000</width>
      <height>22</height>
    </rect>
  </property>
  <widget class="QMenu" name="menu">
    <property name="title">
      <string>P&P°PN°P</string>
    </property>
    <addaction name="action"/>
    <addaction name="action_3"/>
    <addaction name="action_2"/>
  </widget>
  <addaction name="menu"/>
</widget>
<widget class="QStatusBar" name="statusbar"/>
<action name="action">
  <property name="text">
    <string>PŸPsC...CŦP°PSPëC, CŦ PrP°PSPSC< Pµ</string>
  </property>
  <property name="shortcutContext">
    <enum>Qt::WindowShortcut</enum>
  </property>
</action>
<action name="action_2">
  <property name="text">
    <string>P' C< C...PsPr</string>
  </property>
</action>
<action name="action_3">
  <property name="text">
    <string>PŸPiCŦP°PIPeP°</string>
  </property>
</action>
</widget>
<customwidgets>
  <customwidget>
    <class>PlotWidget</class>
    <extends>QGraphicsView</extends>
    <header>pyqtgraph</header>
  </customwidget>
</customwidgets>
<resources/>
<connections/>
<designerdata>
  <property name="gridDeltaX">
    <number>10</number>
  </property>
  <property name="gridDeltaY">
    <number>10</number>

```

```
</property>
<property name="gridSnapX">
  <bool>true</bool>
</property>
<property name="gridSnapY">
  <bool>true</bool>
</property>
<property name="gridVisible">
  <bool>true</bool>
</property>
</designerdata>
</ui>
```