**Проектирование пользовательского интерфейса десктопного приложения**

**Цель работы:** изучение принципов проектирования пользовательского интерфейса. Разработка пользовательского интерфейса десктопного приложения.

**Теоретическое описание разработки.**

**Постановка задачи. Выбор программных средств (язык и среда разработки)**

**Задача:** Найти любой математический алгоритм для программирования по теме «Рациональное природопользование» и создать обучающую программу для школьников и студентов с графическим интерфейсом (Теоретический материал по теме проекта, автоматизированный расчет на основании исходных данных, список литературы, дополнительная информация и т.д.).

Программа пишется на языке Python в IDE Pycharm. С использованием PyQT5 для графического интерфейса и Figma для разработки архитектуры.

**Анализ аналогов,**

**их достоинства и недостатки.**

Для анализа конкурентоспособности мной была выбрана Большая энциклопедия Кирилла и Мефодия.

Достоинства:

В отличие от моего проекта, Большая энциклопедия Кирилла и Мефодия содержит в себе:

Мультимедийные иллюстрации, анимированные карты, трехмерные модели, аудио- и видеофрагменты, интерактивные ленты.

Недостатки:

В отличие от моего проекта, БЭКМ не содержит в себе возможность для практического расчета теоретической части.

Так же проведя анализ веб-приложений, со схожей тематикой, сделала вывод, что в отличие от моего проекта, они не содержат в себе одновременно теоретическую и практическую части. То есть содержат либо только теорию, либо только практику без подробных объяснений.

Подводя итоги, могу сказать, что со временем можно усовершенствовать мой проект и добавить в него достоинства аналогов, тогда он на 100% будет считаться одним из конкурентоспособных приложений в области экологии.

**Пользовательские сценарии**

1)Как студент я могу просматривать список глав, для того чтобы увидеть из чего состоит обучающая программа

Задачи:

1. зайти в приложение
2. нажать на кнопку оглавление

2)Как студент я могу ознакомиться с содержанием каждой главы, для того чтобы получить информацию, которая поможет в дальнейших расчетах

Задачи:

1. зайти в приложение
2. нажать кнопку оглавление
3. нажать на кнопку необходимой главы

3)Как студент я могу просматривать отдельно теоретическую и практические части в каждой главе для того чтобы навигация по обучающей программе была более удобной

Задачи:

1. зайти в приложение
2. нажать на кнопку оглавление
3. нажать на кнопку необходимой главы
4. нажать на кнопку далее или назад

4)Как студент я могу делать расчеты формул по теме выбранной главы для того чтобы получить ответ и на практическом опыте подтвердить ранее полученную теоретическую информацию

Задачи:

1. зайти в приложение
2. нажать на кнопку оглавление
3. нажать на кнопку необходимой главы
4. нажать на кнопку далее (необходимое количество раз)
5. ввести данные для расчета
6. нажать кнопку рассчитать

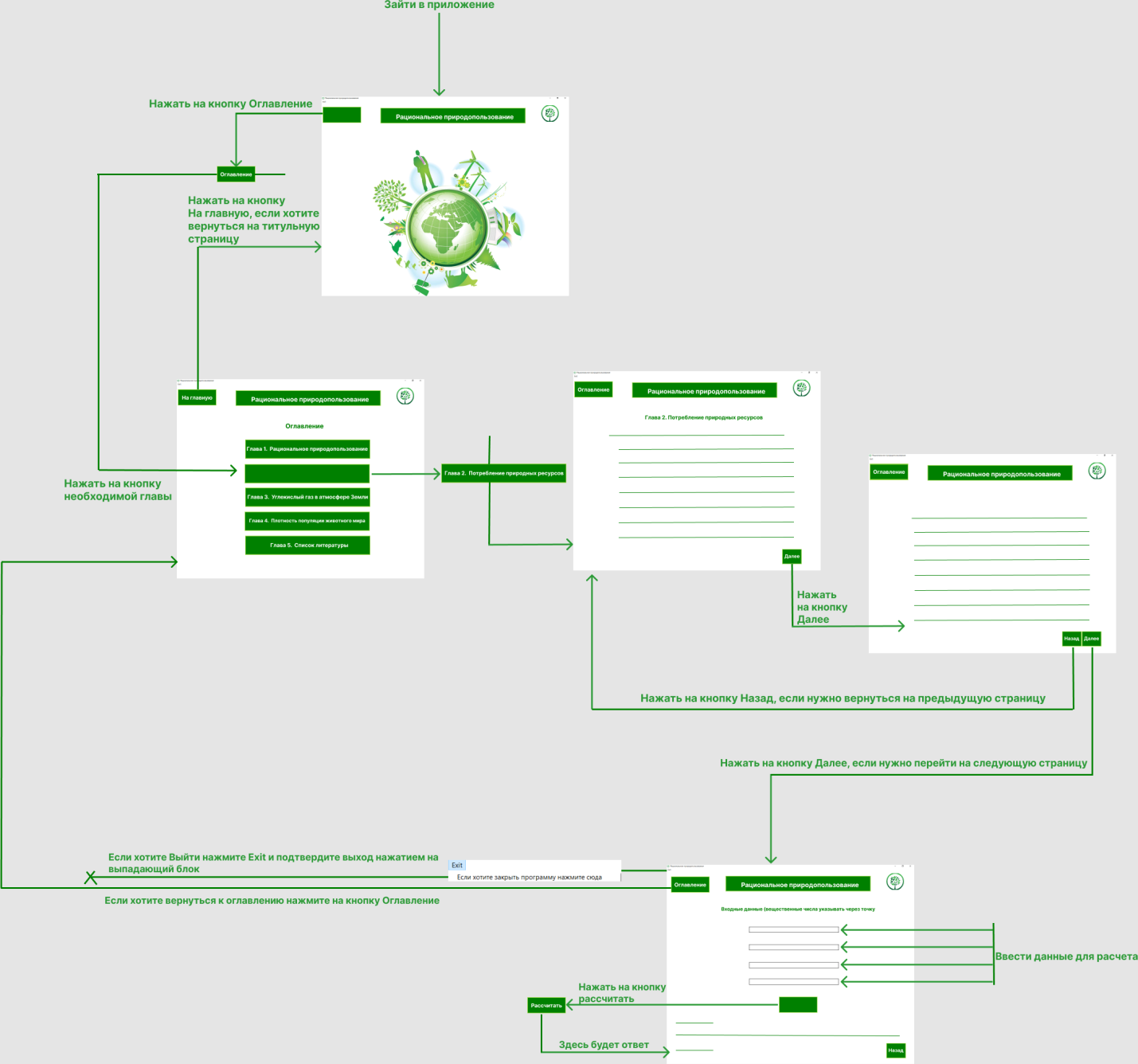
5)Как студент я могу выйти из программы для того чтобы завершить работу

Задачи:

1. зайти в приложение
2. нажать на раздел exit, который находится в меню, и подтвердить выход

**Карта экранов в части описанных**

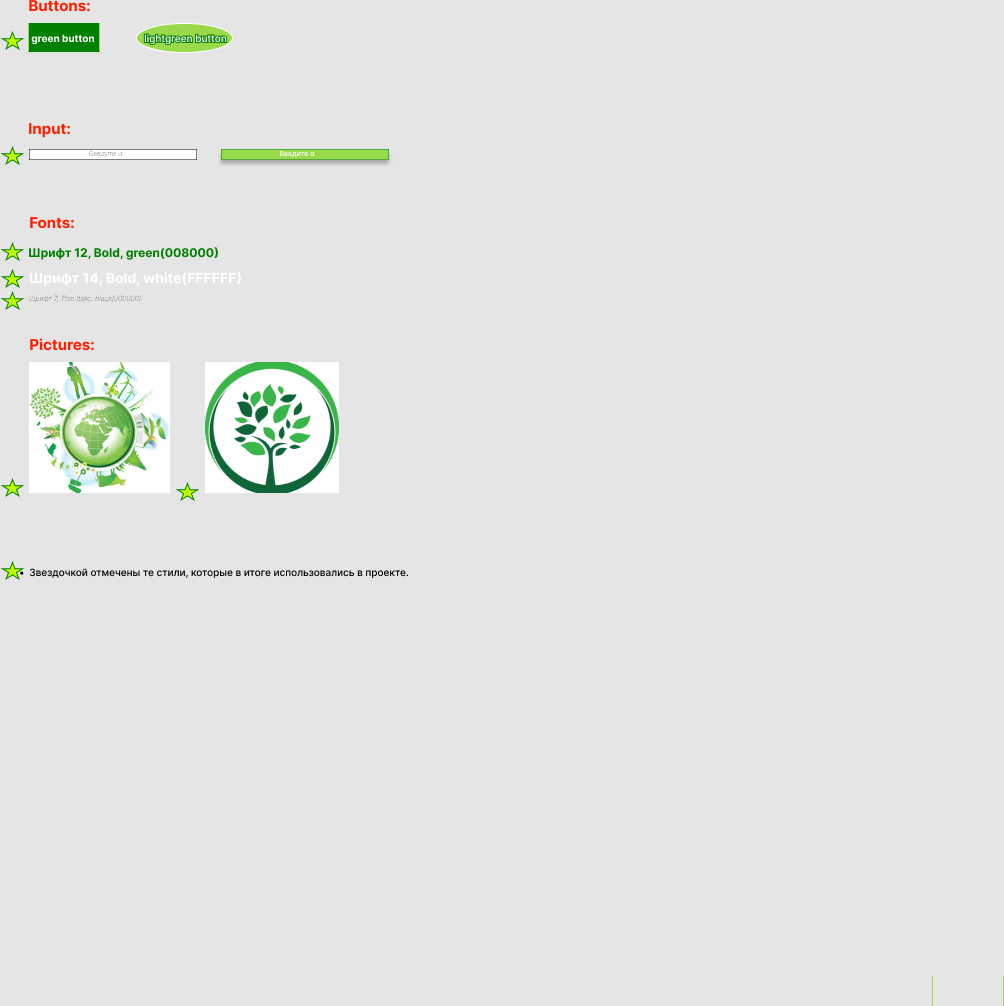
**пользовательских сценариев**

****

**Черновой прототип экранов**

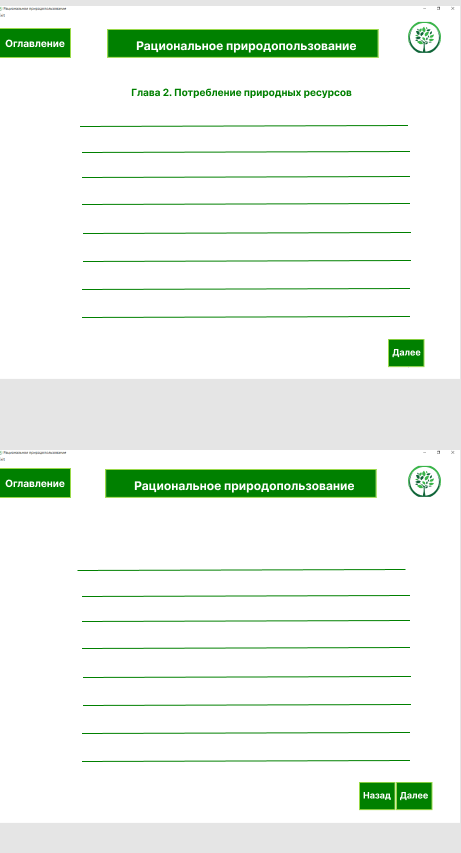
****

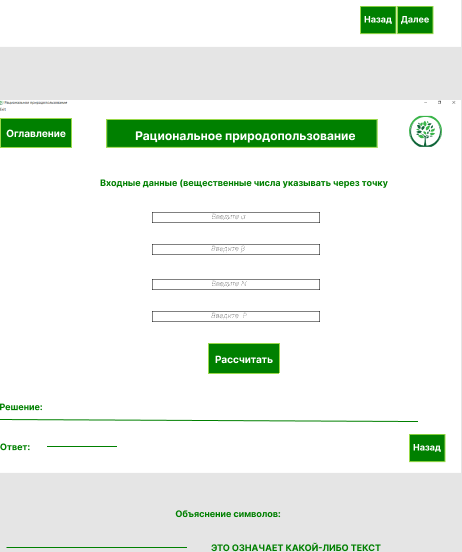
**Стилистики для приложения**

****

**Дизайн концепция приложения**

****

****

****

**Структурированное описание разработки**

**Описание функциональных возможностей.**

Код разделен на 5 частей.

**1) файл gl**

**В этот файл входят данные для заполнения контента страниц**

first = "Рациональное природопользование подразумевает эффективное использование\nприродных ресурсов, " \

"при котором человек не только не вредит природе, но бережно\nк ней относится, стремится к ее охране и " \

"восстановлению. Рациональное природопользование\nлежит в основе геоэкологии и охраны окружающей среды.\n\n\n" \

"Признаками рационального природопользования считаются следующие действия человека:\n\n \* " \

"Восстановление ресурсов природы.\n \* Сохранение гидроресурсов, земельных ресурсов, флоры и фауны.\n" \

" \* Бережное и экономное использование исчерпаемых полезных ископаемых.\n" \

" \* Применение экологичных способов получения " \

"энергии.\n \* Сохранение природного ареала обитания животных, растений и человека.\n" \

" \* Регулирование численности населения Земли.\n"

second = "Каждый год население Земли использует более 3,5 триллионов самых различных ресурсов – пресная вода, " \

"воздух, полезные ископаемые\nи энергоресурсы потребляются в огромных количествах. " \

"В среднем получается 600 тонн ресурсов в год на одного жителя планеты." \

"\n\nДо внедрения в экосистему планеты в природе поддерживался естественный баланс. " \

"Как только у человека появилось понимание\nиспользования предметов окружающей среды, " \

"он начал интенсивнее воздействовать на природный баланс,\nчто со временем становилось всё сильнее заметно. " \

"Однако до промышленной революции это было не слишком сильно заметно.\n\nС развитием промышленной и " \

"техногенной сфер, применяя новые технические и технологические разработки, в XX веке человек стал\n" \

"наносить состоянию окружающей среды урон. Этот ущерб каждый год прогрессирует и растёт, так как у " \

"населения нет понимания\nосознанного потребления, а у предпринимателей отсутствуют серьёзные рычаги давления " \

"при использовании ресурсов и утилизации\nотходов на производстве. Учитывая это, стало сложно и практически " \

"невозможно ликвидировать все последствия. Однако можно\nвовремя остановиться и не усугублять ситуацию дальше. " \

"\n\nРассмотрим конкретные измеримые данные, которые ясно показывают истощение природных ресурсов.\n\nВода: " \

"несмотря на тот факт, что 70% от состава планеты – вода, только 10% из неё является пресной. Солёную или " \

"слишком\nзагрязнённую воду пить или использовать в промышленных целях невозможно даже с учётом всех " \

"современных систем очистки и\nдистилляции. 2,5% пресной воды существует только в виде льда или снега. " \

"Поэтому на самом деле в реальном пользовании мы имеем\n7,5% от объёма пресной воды. Согласно прогнозу ФАО " \

"(Продовольственной и сельскохозяйственной Организации Объединенных Наций),\nв 2025 году около 1,8 млрд человек" \

" не будут иметь доступа к питьевой воде."

second2 = "Уголь: невозобновляемый ресурс, который используется наиболее часто " \

"в качестве топлива. Хотя пик добычи угля, по расчётам,\nпридётся на период 2025-2048 года, но всех запасов " \

"Земли по примерному прогнозу хватит ещё на 178 лет. Если спрос на уголь будет\nувеличиваться, то и срок будет" \

" приближаться быстрее.\n\nНефть: с учётом текущего прогноза всех запасов хватит лишь на ближайшие 40 лет," \

"с учётом того, что 94% всего транспорта зависит\nот нефтепродуктов.\n\nДля достижения гармоничных отношений " \

"человека и природной сферы необходимо достигнуть следующих стратегических задач\nвне зависимости от места " \

"проживания:\n\nСформировать у всех поколений новое социальное и экономическое мышление, которое будет основано на" \

" моральных принципах\nрасходования ресурсов и утилитарном подходе к бытовой жизни;\nВ этих условиях нельзя " \

"по-прежнему считать, что экология и экономика — это две различные, всецело самостоятельные и никак\nне " \

"связанные между собой области знания и практики. На самом деле они тесно взаимосвязаны.\n\nЧеловечество " \

"оказалось перед проблемой выработки экономического механизма природопользования, который бы обеспечил\n" \

"выполнение условия: "

second3 = "α и β – расход природных ресурсов на 1 человека для его личных и производственных нужд;\n" \

"N – численность населения;\nP – воспроизводимый объем природных ресурсов.\n\nВ идеале эта формула должна работать" \

" так, что P оказывалась бы больше, так ресурсы тратились\nбы рационально. Однако макроэкономика в настоящее " \

"время ориентирована на максимизацию\nвалового объема благ и поэтому P может оказаться меньше."

third = "Двуокись углерода (CO2) – это бесцветный газ, который присутствует в воздухе. Хотя выбросы происходят из " \

"многих природных источников,\nпроблематичным является CO2, производимый в результате технологических процессов." \

" Например, сжигание ископаемого топлива\nи выбросы электростанций вредят окружающей среде нашей планеты и " \

"существенно влияют на изменение климата. Поэтому очень важно\nстремиться максимально уменьшить выбросы " \

"этого газа. Для этого стоит:\n\n1)Восстановить природные ресурсы. Океаны играют жизненно важную роль в поглощении" \

" углекислого газа, присутствующего в атмосфере.\nПоскольку поглощение диоксида углерода океаном является " \

"медленным процессом и может занимать сотни лет, это явление не может\nобезвредить то огромное количество газа, " \

"которое выбрасывается каждый день. Мы не можем увеличить площадь океанов на планете,\nоднако в наших силах " \

"стремиться к восстановлению и сохранению лесов для большей переработки вредного газа.\n2)Сократить количество " \

"отходов. Промышленность задействует огромные объемы энергии для производства всего, что мы используем в\n" \

"нашей повседневной жизни. Из этого следует, что, если мы сможем переработать наши отходы, потребуется меньше " \

"энергии\nдля производства новых материалов. Обязательно всегда сдавайте в переработку все, что можете, от " \

"бумаги и пластика до батарей.\n3)Уменьшить потребление энергии. При производстве электроэнергии вырабатывается " \

"в целом больше углекислого газа, чем от авто.\nМногие из наших электростанций сжигают ископаемое топливо для " \

"выработки энергии, которую мы используем. Очевидно, чем больше\nэлектричества мы потребляем, тем больше " \

"энергии нужно производить.\n\nЗаменить это можно экологически чистыми источниками энергии:\n\n1)Энергия ветра - " \

"энергия ветра, как быстроразвивающийся источник электричества, производится\nблагодаря ветротурбинам, что " \

"устанавливаются на высотах.\n2)Обезуглероженный каменный уголь - После газификации угля значительно снижается " \

"выброс оксида серы, ртути\nи оксида азота в атмосферу. Кроме того, такой комбинированный цикл газификации " \

"значительно сокращает\nвыбросы углекислого газа."

third2 = "3)Геотермальная энергия – это тепло из-под Земли. " \

"Мы можем восстановить это тепло в виде пара или горячей воды и использовать ее\nдля обогрева зданий и выработки " \

"электроэнергии.\n4)Солнечная энергия - солнечная энергия преобразовывается в электрическую и теплую. " \

"Используется для добычи электричества, для\nподогрева воды и отопления различных помещений.\nТакже о том, что " \

"нормы СО2 в помещении (ppm) действительно влияют на\nсамочувствие учащихся, проживающих и работающих, " \

"свидетельствуют многочисленные исследования, проводившиеся в странах Азии и\nЕвропы. Загрязненный воздух " \

"провоцирует ухудшение самочувствия у людей. Самые распространенные «симптомы» - сонливость, апатия,\nпотеря " \

"концентрации, тошнота, головная боль.\n\nОпределение содержания углекислого газа в атмосфере.\nРасчет " \

"содержания СО2 (в процентах) проводят по следующей формуле: "

third3 = "где V – объем воздуха открытой атмосферы(м3);\n\nV1 – объем воздуха исследуемой зоны(м3);\n\n0,04 – " \

"содержание углекислого газа в воздухе, %."

fourth = "К глобальным экологическим проблемам следует отнести и проблемы животного мира, который является неотъемлемым" \

" компонентом\nбиосферы. Животные принимают участие в биотическом круговороте энергии и веществ на планете. " \

"От устойчивости фауны зависят все\nостальные элементы экосистем. Проблема сокращения популяций животных " \

"происходит не только потому, что ухудшается экология, а и\nпотому, что люди используют их в качестве питания. " \

"\nВ природе нужны абсолютно все представители фауны: и мелкие насекомые, и травоядные, и хищники, и крупные" \

" морские животные.\nВредных видов, от которых следует избавиться, не существует. Лишь популяции клещей и " \

"грызунов-вредителей нужно контролировать.\nПричины вымирания животных. Многие виды животных исчезали до " \

"появления человека. Зарождение новых видов и вымирание старых\n— эволюционный процесс. Однако стремительное" \

" сокращение некоторых популяций напрямую связано с человеческим фактором.\n\nВлияние на окружающую среду " \

"заключается в том, что человек уничтожает естественные места обитания животных, сокращает их\nкормовую базу:\n" \

"1) Вырубка лесов;\n2) Создание ГЭС, искусственных водохранилищ;\n3) Осушение рек, озер, изменение речного русла;\n" \

"4) Распахивание земель под пашни, расширение пастбищ;\n5) Загрязнение воды, почвы и воздуха;\n6) Строительство;\n" \

"7) Добыча полезных ископаемых;\n8) Изменение экосистемы через распространение видов, нетипичных для данной " \

"территории(например, ввоз европейских кроликов\nв Австралию).\n\nИстребление животных связано с их прямым " \

"уничтожением человеком. Причины такой агрессивной деятельности\nсвязаны с активным промыслом (добыча шкур," \

" мяса, бивней, рогов), защитой сельхозугодий и скота, браконьерством\nили неосторожностью."

fourth2 = "Нанося урон экологии, человек отрицательно влияет на жизнь животных, их кормовую базу, вынуждает мигрировать" \

" или охотится\nна них, прямо сокращая численность. Деградация животного мира — значимая экологическая проблема. " \

"Для ее решения\nпринимаются меры на уровне мирового сообщества:\n1) Создание Красной книги;\n2) Отслеживание редких " \

"и исчезающих видов;\n3) Искусственное разведение в питомниках, заповедниках, зоопарках;\n4) Создание и внедрение " \

"экологических программ, направленных на защиту животных от истребления;\n5) Пропаганда экологического" \

" мировоззрения;\n6) Установление правил поведения в природе (уборка мусора, меры противопожарной безопасности, " \

"запрет на попытки одомашнить\nдетенышей диких видов);\n\nМногие виды животных истреблены человеком, они считаются" \

" вымершими. Например, Додо (маврикийский дронт): после прибытия на родину\nдодо, остров Маврикий, людей птица " \

"быстро оказалась уничтожена и в наше время стала символом уничтожения животных из-за варварского\nвторжения" \

" в среду их обитания;\nПоэтому так важно отслеживать и оценивать численность животного мира!\n\nДля того чтобы " \

"дать оценку плотности популяции определенного вида используется формула:"

fourth3 = "где, N - численность популяции;\nS1 - площадь или объем занятый популяцией;\nD - плотность."

book = "1. «Окружающая среда и человек Д. П. Никитин, Ю. В. Новиков М., Высшая Школа, 1980\n2. «Экология, " \

"природопользование, охрана окружающей среды» Т. А. Дёмина М., Аспект Пресс, 2016\n3. «Экологический словарь» " \

"Б. А. Быков Алма-Ата, Наука, 2004" \

"\n4. https://ru.wikipedia.org/ \n5. https://studref.com/523635/ekologiya/osnovnaya\_formula\_prirodopolzovaniya" \

"\n6. http://www.ievbras.ru/ecostat/Kiril/Article/A18/Vol1/Kramar1.html\n7. " \

"https://izmerkon.ru/podderzhka/publikaczii/normy-so2.html"

**2) файл project4**

**В этот файл входят страницы со списком литературы**

import project #из файла project импортируем данные и связку страниц

from content.gl import \* #из файла gl импортируем контент для страницы

import sys #импортируем системный модуль для дальнейшего вызова окна на экран

from PyQt5.QtGui import QIcon #импортируем библиотики из PyQt5

from PyQt5 import QtGui #импортируем библиотики из PyQt5

from PyQt5.QtWidgets import (QLineEdit, QLabel, QPushButton, QApplication, QMainWindow, QAction) #импортируем библиотики из PyQt5

from PyQt5.QtCore import QSize #импортируем библиотики из PyQt5

ОКНО СПИСКА ЛИТЕРАТУРЫ

class Window12(QMainWindow): #создаем окно списка литературы

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Глава 5. Список литературы")

#устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ТЕКСТ КОНТЕНТА

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(400, -30) #расположение текста

self.label\_1.setText(book) #из файла gl импортируем контент book

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_1.setFixedSize(1800, 1000) #размер области написания текста

self.label\_1.setStyleSheet("color: green;") #стиль текста

self.label\_1.setFont(font) #сохранение изменений

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

КАРТИНКА В КОНТЕНТЕ

but\_pict1 = QPushButton(self) #создаем область расположения картинки

but\_pict1.setIcon(QIcon('content/ppp.png')) #импортируем картинку

but\_pict1.setIconSize(QSize(380, 380)) #размер области картинки

but\_pict1.setFixedSize(380, 380) #размер картинки

but\_pict1.move(730, 625) #размер картинки

ЗАГОЛОВОК

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(600, 190)#расположение текста

self.label\_1.setText("Глава 5. Список литературы") #текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_1.setFixedSize(1800, 70) #размер области написания текста

self.label\_1.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_1.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

button.clicked.connect(self.window1) #после нажатия на кнопку соединение с оглавлением

self.show()#вывести на экран изменения

ФУНКЦИИ ВЫЗОВА ОКОН

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

**3) файл project3**

**В этот файл входят страницы c 4 главой и ее подглавами**

import project #из файла project импортируем данные и связку страниц

from content.gl import \* #из файла gl импортируем контент для страницы

import sys #импортируем системный модуль для дальнейшего вызова окна на экран

from PyQt5.QtGui import QIcon #импортируем библиотики из PyQt5

from PyQt5 import QtGui #импортируем библиотики из PyQt5

from PyQt5.QtWidgets import (QLineEdit, QLabel, QPushButton, QApplication, QMainWindow, QAction) #импортируем библиотики из PyQt5

from PyQt5.QtCore import QSize #импортируем библиотики из PyQt5

ОКНО ГЛАВЫ 4

class Window9(QMainWindow): #окно 4 главы 1 подглавы

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Глава 4. Плотность популяции животного мира")

#устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ТЕКСТ КОНТЕНТА

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(200, 130)#расположение текста

self.label\_1.setText(fourth)#текст из файла gl

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_1.setFixedSize(1800, 1000) #размер области написания текста

self.label\_1.setStyleSheet("color: green;) #стиль текста

self.label\_1.setFont(font) #сохранение изменений

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

ЗАГОЛОВОК

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(600, 190)#расположение текста

self.label\_1.setText("Глава 4. Плотность популяции животного мира")#текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_1.setFixedSize(1800, 70) #размер области написания текста

self.label\_1.setStyleSheet("color: green; font: bold")#стиль текста

self.label\_1.setFont(font) #сохранение изменений

КНОПКА ДАЛЕЕ ДЛЯ ПЕРЕХОДА НА СЛЕДУЮЩУЮ ПОДГЛАВУ

buttonw4 = QPushButton("Далее", self) #создаем кнопку и даем ей название

buttonw4.move(1720, 900) #расположение кнопки

buttonw4.setFixedSize(90, 70) #размер кнопки

buttonw4.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont() #импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw4.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

button.clicked.connect(self.window1) #после нажатия на кнопку соединение с оглавлением

self.show()#вывести на экран изменения

buttonw4.clicked.connect(self.window10) #после нажатия на кнопку соединение с 4 главой 2 подглавой

self.show()#вывести на экран изменения

ФУНКЦИИ ВЫЗОВА ОКОН

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

def window10(self): #функция события вызова окна с 4 главой 2 подглавой

self.w10 = Window10()#сокращенное название окна

self.w10.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

ОКНО ГЛАВЫ 4 ПОДГЛАВЫ 2

class Window10(QMainWindow): #окно 4 главы 2 подглавы

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Глава 4. Плотность популяции животного мира")

#устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ТЕКСТ КОНТЕНТА

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(200, -70)#расположение текста

self.label\_1.setText(fourth2)#текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_1.setFixedSize(1800, 1000) #размер области написания текста

self.label\_1.setStyleSheet("color: green; )#стиль текста

self.label\_1.setFont(font) #сохранение изменений

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА

self.label\_3 = QLabel("", self) #задаем область для текста

self.label\_3.move(200, 800) #расположение текста

self.label\_3.setText(fourth3) #из файла gl импортируем контент

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_3.setFixedSize(1800, 200) #размер области написания текста

self.label\_3.setStyleSheet("color: green;") #стиль текста

self.label\_3.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

КАРТИНКА В КОНТЕНТЕ

but\_pict1 = QPushButton(self) #создаем область расположения картинки

but\_pict1.setIcon(QIcon('content/animals.PNG'))#импортируем картинку

but\_pict1.setIconSize(QSize(400, 500)) #размер области картинки

but\_pict1.setFixedSize(400, 250) #размер картинки

but\_pict1.move(1000, 720) #размер картинки

КАРТИНКА В КОНТЕНТЕ(ФОРМУЛА)

but\_pict2 = QPushButton(self)

but\_pict2.setIcon(QIcon('content/4gl.PNG'))

but\_pict2.setIconSize(QSize(600, 600))

but\_pict2.setFixedSize(140, 80)

but\_pict2.move(180, 750)

КНОПКА ДАЛЕЕ ДЛЯ ПЕРЕХОДА НА СЛЕДУЮЩУЮ ПОДГЛАВУ

buttonw4 = QPushButton("Далее", self) #создаем кнопку и даем ей название

buttonw4.move(1720, 900) #расположение кнопки

buttonw4.setFixedSize(90, 70) #размер кнопки

buttonw4.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont() #импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw4.setFont(font) #сохранение изменений

КНОПКА НАЗАД ДЛЯ ВОЗВРАЩЕНИЯ НА ПРЕДЫДУЩУЮ ПОДГЛАВУ

buttonw3 = QPushButton("Назад", self) #создаем кнопку и даем ей название

buttonw3.move(1630, 900) #расположение кнопки

buttonw3.setFixedSize(90, 70) #размер кнопки

buttonw3.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont()#импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw3.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

button.clicked.connect(self.window1) #при нажатии переход на оглавление

buttonw4.clicked.connect(self.window11) #при нажатии переход вперед на 4 главу 3подглаву

buttonw3.clicked.connect(self.window9) #при нажатии переход назад на 4 главу 1 подглаву

self.show() #вывести на экран изменения

ФУНКЦИИ ВЫЗОВА ОКОН

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

def window11(self): #функция события вызова окна с 4 главой 3 подглавой

self.w8 = Window11()#сокращенное название окна

self.w8.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

def window9(self): #функция события вызова окна с 4 главой 1 подглавой

self.w9 = Window9()#сокращенное название окна

self.w9.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

ОКНО ГЛАВЫ 4 ПОДГЛАВЫ 3

class Window11(QMainWindow): #окно 4 главы 3 подглавы

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Глава 4. Плотность популяции животного мира")

#устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

ЗАГОЛОВОК

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(350, 150)#расположение текста

self.label\_1.setText("Входные данные (вещественные числа указывать через точку)")#текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_1.setFixedSize(1800, 70) #размер области написания текста

self.label\_1.setStyleSheet("color: green; font: bold")#стиль текста

self.label\_1.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА (РЕШЕНИЕ)

self.label\_4 = QLabel("", self) #задаем область для текста

self.label\_4.move(1, 650) #расположение текста

self.label\_4.setText("Решение: ") #сам текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_4.setFixedSize(500, 70) #размер области написания текста

self.label\_4.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_4.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА В РЕШЕНИИ

self.label\_5 = QLabel("", self) ) #задаем область для текста

self.label\_5.move(1, 730) #расположение текста

self.label\_5.setText("Рассчитаем формулу плотности популяции одного вида животных "

"N / S1 = D")#сам текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

self.label\_5.setFixedSize(2600, 70) #размер области написания текста

self.label\_5.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_5.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ

button.clicked.connect(self.window1) #при нажатии переход на оглавление

ВВОД ДАННЫХ ПОЛЬЗОВАТЕЛЕМ ДЛЯ ДАЛЬНЕЙШЕГО РАССЧЕТА

self.text1 = QLineEdit(self) #создание окна для ввода данных

self.text1.setPlaceholderText("Введите N") #плейсхолдер для подсказки ввода пользователю

self.text2 = QLineEdit(self) #создание окна для ввода данных

self.text2.setPlaceholderText("Введите S1") #плейсхолдер для подсказки ввода пользователю

self.setGeometry(300, 200, 600, 450) #задаем размер окна

self.text1.setGeometry(750, 250, 300, 30)#размер 1 окна ввода

self.text2.setGeometry(750, 350, 300, 30) #размер 2 окна ввода

КНОПКА РАССЧИТАТЬ

button\_count = QPushButton("Рассчитать", self) #даем название кнопке

button\_count.move(800, 500) #двигаем кнопку по экрану, задаем ей позицию

button\_count.setFixedSize(200, 80) #устанавливаем размер кнопки

button\_count.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button\_count.setFont(font) #сохраняем изменения

ТЕКСТ КОНТЕНТА (ОТВЕТ)

self.label\_4 = QLabel("", self) #задаем область для текста

self.label\_4.move(1, 900) #расположение текста

self.label\_4.setText("Решение: ") #сам текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_4.setFixedSize(500, 70) #размер области написания текста

self.label\_4.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_4.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА В ОТВЕТЕ ДО НАЖАТИЯ КНОПКИ РАССЧИТАТЬ

self.label\_answer = QLabel("N / S1 = ", self)

#задаем область для текста

self.label\_answer.move(150, 903) #расположение текста

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(10) #размер шрифта

self.label\_answer.setFixedSize(500, 70) #размер области написания текста

self.label\_answer.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_answer.setFont(font) #сохранение изменений

self.label4\_answer = QLabel("D", self)#задаем область для текста

self.label4\_answer.move(250, 903) #расположение текста

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(10) #размер шрифта

self.label4\_answer.setFixedSize(500, 70) #размер области написания текста

self.label4\_answer.setStyleSheet("color: green; font: bold") #стиль текста

self.label4\_answer.setFont(font) #сохранение изменений

self.label6\_answer = QLabel("", self)#задаем область для текста

self.label6\_answer.move(800, 903) #расположение текста

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(10) #размер шрифта

self.label6\_answer.setFixedSize(900, 70) #размер области написания текста

self.label6\_answer.setStyleSheet("color: green; font: bold") #стиль текста

self.label6\_answer.setFont(font) #сохранение изменений

КНОПКА НАЗАД ДЛЯ ВОЗВРАЩЕНИЯ НА ПРЕДЫДУЩУЮ ПОДГЛАВУ

buttonw4 = QPushButton("Назад", self) #создаем кнопку и даем ей название

buttonw4.move(1720, 900) #расположение кнопки

buttonw4.setFixedSize(90, 70) #размер кнопки

buttonw4.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont()#импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw4.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

buttonw4.clicked.connect(self.window10)#соединение с 4 главой 2 подглавой

self.show() #вывести изменения на экран

ФУНКЦИИ ДЛЯ ПРОВЕРКИ ВВОДА ДАННЫХ ПОЛЬЗОВАТЕЛЕМ И РАСЧЕТА

def checkstatus(self): #функция проверки ввода данных

if self.text1.text() == "": #если пустое значение в 1 окне ввода

print("") #то ничего не выводить

self.text1.setFocus() #возврат к вводу

if self.text2.text() == "": #если пустое значение в 2 окне ввода

print("")#то ничего не выводить

self.text2.setFocus()#возврат к вводу

else: #в другом случае

try: #отлов ошибок

self.label6\_answer.setText('| Плотность популяции данного вида животных') #вывод значения

self.label4\_answer.setText(str((float(self.text1.text()) / float(self.text2.text()))))#вывод значения

except ValueError: #отлов ошибки

self.label6\_answer.setText("Введите корректные данные!") #вывести сообщение, если пользователь ввел не число

except ZeroDivisionError: #отлов ошибки

self.label6\_answer.setText("На 0 делить нельзя!") #вывести сообщение, если пользователь делит на 0

ФУНКЦИИ ВЫЗОВА ОКОН

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

def window10(self): #функция события вызова окна 4 главой 2 подглавой

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

**4) файл project2**

**В этот файл входят страницы c 3 главой и ее подглавами**

import project #из файла project импортируем данные и связку страниц

from content.gl import \* #из файла gl импортируем контент для страницы

import sys #импортируем системный модуль для дальнейшего вызова окна на экран

from PyQt5.QtGui import QIcon #импортируем библиотики из PyQt5

from PyQt5 import QtGui #импортируем библиотики из PyQt5

from PyQt5.QtWidgets import (QLineEdit, QLabel, QPushButton, QApplication, QMainWindow, QAction) #импортируем библиотики из PyQt5

from PyQt5.QtCore import QSize #импортируем библиотики из PyQt5

ОКНО ГЛАВЫ 3 ПОДГЛАВЫ 1

class Window6(QMainWindow): #окно 3 главы 1 подглавы

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Глава 3. Углекислый газ в атмосфере Земли")

#устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ТЕКСТ КОНТЕНТА

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(200, 100)#расположение текста

self.label\_1.setText(third)#текст из файла gl

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_1.setFixedSize(1800, 1000) #размер области написания текста

self.label\_1.setStyleSheet("color: green;")#стиль текста

self.label\_1.setFont(font) #сохранение изменений

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

ЗАГОЛОВОК

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(600, 190)#расположение текста

self.label\_1.setText("Глава 3. Углекислый газ в атмосфере Земли")#текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_1.setFixedSize(1800, 70) #размер области написания текста

self.label\_1.setStyleSheet("color: green; font: bold")#стиль текста

self.label\_1.setFont(font) #сохранение изменений

КНОПКА ДАЛЕЕ ДЛЯ ПЕРЕХОДА НА СЛЕДУЮЩУЮ ПОДГЛАВУ

buttonw4 = QPushButton("Далее", self) #создаем кнопку и даем ей название

buttonw4.move(1720, 900) #расположение кнопки

buttonw4.setFixedSize(90, 70) #размер кнопки

buttonw4.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont() #импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw4.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

button.clicked.connect(self.window1)#вызов оглавления

buttonw4.clicked.connect(self.window7)#вызов 3 главы 2 подглавы

self.show()#вывести изменения на экран

ФУНКЦИИ ВЫЗОВА ОКОН

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

def window7(self): #функция события вызова окна 3 главы 2 подглавы

self.w7 = Window7()#сокращенное название окна

self.w7.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

ОКНО ГЛАВЫ 3 ПОДГЛАВЫ 2

class Window7(QMainWindow): #окно 3 главы 2 подглавы

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Глава 3. Углекислый газ в атмосфере Земли")

#устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ТЕКСТ КОНТЕНТА

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(200, -70)#расположение текста

self.label\_1.setText(third2)#текст из файла gl

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_1.setFixedSize(1800, 1000) #размер области написания текста

self.label\_1.setStyleSheet("color: green;")#стиль текста

self.label\_1.setFont(font) #сохранение изменений

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА

self.label\_3 = QLabel("", self) #задаем область для текста

self.label\_3.move(200, 750) #расположение текста

self.label\_3.setText(third3) #из файла gl импортируем контент

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_3.setFixedSize(1800, 200) #размер области написания текста

self.label\_3.setStyleSheet("color: green;") #стиль текста

self.label\_3.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

КАРТИНКА В КОНТЕНТЕ

but\_pict1 = QPushButton(self) #создаем область расположения картинки

but\_pict1.setIcon(QIcon('content/CO2.jpg'))#импортируем картинку

but\_pict1.setIconSize(QSize(400, 500)) #размер области картинки

but\_pict1.setFixedSize(370, 370) #размер картинки

but\_pict1.move(1000, 650) #размер картинки

КАРТИНКА В КОНТЕНТЕ

but\_pict2 = QPushButton(self) #создаем область расположения картинки

but\_pict2.setIcon(QIcon('content/3gl.PNG'))#импортируем картинку

but\_pict2.setIconSize(QSize(600, 600)) #размер области картинки

but\_pict2.setFixedSize(120, 50) #размер картинки

but\_pict2.move(200, 672)#нахождение картинки на экране

КНОПКА ДАЛЕЕ ДЛЯ ПЕРЕХОДА НА СЛЕДУЮЩУЮ ПОДГЛАВУ

buttonw4 = QPushButton("Далее", self) #создаем кнопку и даем ей название

buttonw4.move(1720, 900) #расположение кнопки

buttonw4.setFixedSize(90, 70) #размер кнопки

buttonw4.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont() #импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw4.setFont(font) #сохранение изменений

КНОПКА НАЗАД ДЛЯ ВОЗВРАЩЕНИЯ НА ПРЕДЫДУЩУЮ ПОДГЛАВУ

buttonw3 = QPushButton("Назад", self) #создаем кнопку и даем ей название

buttonw3.move(1630, 900) #расположение кнопки

buttonw3.setFixedSize(90, 70) #размер кнопки

buttonw3.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont()#импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw3.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

buttonw3.clicked.connect(self.window6) #вызов окна оглавления

button.clicked.connect(self.window1) #вызов окна 3 главы 1 подглавы

buttonw4.clicked.connect(self.window8) #вызов окна 3 главы 3 подглавы

self.show() #вывести изменения

ФУНКЦИИ ВЫЗОВА ОКОН

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

def window8(self): #функция события вызова окна 3 главы 1 подглавы

self.w8 = Window8()#сокращенное название окна

self.w8.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

def window6(self): #функция события вызова окна 3 главы 3 подглавы

self.w6 = Window6()#сокращенное название окна

self.w6.showMaximized()()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

ОКНО ГЛАВЫ 3 ПОДГЛАВЫ 3

class Window8(QMainWindow): #окно 3 главы 3 подглавы

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Глава 3. Углекислый газ в атмосфере Земли")

#устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

ЗАГОЛОВОК

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(350, 150)#расположение текста

self.label\_1.setText("Входные данные (вещественные числа указывать через точку)")#текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_1.setFixedSize(1800, 70) #размер области написания текста

self.label\_1.setStyleSheet("color: green; font: bold")#стиль текста

self.label\_1.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА (РЕШЕНИЕ)

self.label\_4 = QLabel("", self) #задаем область для текста

self.label\_4.move(1, 650) #расположение текста

self.label\_4.setText("Решение: ") #сам текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_4.setFixedSize(500, 70) #размер области написания текста

self.label\_4.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_4.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА В РЕШЕНИИ

self.label\_5 = QLabel("", self) ) #задаем область для текста

self.label\_5.move(1, 730) #расположение текста

self.label\_5.setText("Рассчитаем формулу содержания CO2(Углекислого газа) в атмосфере Земли" (0.04 \* V) / V1 = X и проведем анализ воздуха")#сам текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

self.label\_5.setFixedSize(2600, 70) #размер области написания текста

self.label\_5.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_5.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ

button.clicked.connect(self.window1) #при нажатии переход на оглавление

ВВОД ДАННЫХ ПОЛЬЗОВАТЕЛЕМ ДЛЯ ДАЛЬНЕЙШЕГО РАССЧЕТА

self.text1 = QLineEdit(self) #создание окна для ввода данных

self.text1.setPlaceholderText("Введите V") #плейсхолдер для подсказки ввода пользователю

self.text2 = QLineEdit(self) #создание окна для ввода данных

self.text2.setPlaceholderText("Введите V1") #плейсхолдер для подсказки ввода пользователю

self.setGeometry(300, 200, 600, 450) #задаем размер окна

self.text1.setGeometry(750, 250, 300, 30) #размер 1 окна ввода

self.text2.setGeometry(750, 350, 300, 30) #размер 2 окна ввода

КНОПКА РАССЧИТАТЬ

button\_count = QPushButton("Рассчитать", self) #даем название кнопке

button\_count.move(800, 500) #двигаем кнопку по экрану, задаем ей позицию

button\_count.setFixedSize(200, 80) #устанавливаем размер кнопки

button\_count.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button\_count.setFont(font) #сохраняем изменения

button\_count.clicked.connect(self.checkstatus)

ТЕКСТ КОНТЕНТА(ОТВЕТ)

self.label\_4 = QLabel("", self) #задаем область для текста

self.label\_4.move(1, 900) #расположение текста

self.label\_4.setText("Решение: ") #сам текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_4.setFixedSize(500, 70) #размер области написания текста

self.label\_4.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_4.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА В ОТВЕТЕ ДО НАЖАТИЯ КНОПКИ РАССЧИТАТЬ

self.label\_answer = QLabel("(0.04 \* V) / V1 = ", self)

#задаем область для текста

self.label\_answer.move(150, 903) #расположение текста

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(10) #размер шрифта

self.label\_answer.setFixedSize(500, 70) #размер области написания текста

self.label\_answer.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_answer.setFont(font) #сохранение изменений

self.label4\_answer = QLabel("X", self)#задаем область для текста

self.label4\_answer.move(350, 903) #расположение текста

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(10) #размер шрифта

self.label4\_answer.setFixedSize(500, 70) #размер области написания текста

self.label4\_answer.setStyleSheet("color: green; font: bold") #стиль текста

self.label4\_answer.setFont(font) #сохранение изменений

self.label6\_answer = QLabel("", self)#задаем область для текста

self.label6\_answer.move(800, 903) #расположение текста

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(10) #размер шрифта

self.label6\_answer.setFixedSize(900, 70) #размер области написания текста

self.label6\_answer.setStyleSheet("color: green; font: bold") #стиль текста

self.label6\_answer.setFont(font) #сохранение изменений

КНОПКА НАЗАД ДЛЯ ВОЗВРАЩЕНИЯ НА ПРЕДЫДУЩУЮ ПОДГЛАВУ

buttonw4 = QPushButton("Назад", self) #создаем кнопку и даем ей название

buttonw4.move(1720, 900) #расположение кнопки

buttonw4.setFixedSize(90, 70) #размер кнопки

buttonw4.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont()#импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw4.setFont(font) #сохранение изменений

buttonw4.clicked.connect(self.window7)#вызов 3 главы 2 подглавы

ФУНКЦИИ ДЛЯ ПРОВЕРКИ ВВОДА ДАННЫХ ПОЛЬЗОВАТЕЛЕМ И РАСЧЕТА

def checkstatus(self): #функция проверки ввода данных

if self.text1.text() == "": #если пустое значение в 1 окне ввода

print("") #то ничего не выводить

self.text1.setFocus() #возврат к вводу

if self.text2.text() == "": #если пустое значение в 2 окне ввода

print("")#то ничего не выводить

self.text2.setFocus()#возврат к вводу

else: #в другом случае

try: #отлов ошибок

if (((float(self.text1.text()) \* 0.04) / float(self.text2.text()))) <= 0.04 and \

(((float(self.text1.text()) \* 0.04) / float(self.text2.text()))) >= 0.03:

self.label6\_answer.setText('| Уличный воздух') #вывести сообщение

if (((float(self.text1.text()) \* 0.04) / float(self.text2.text()))) < 0.03:

self.label6\_answer.setText('| Крайне низкое содержание CO2 для атмосферы Земли! ') #вывести сообщение

if (((float(self.text1.text()) \* 0.04) /

float(self.text2.text()))) > 0.04 and (((float(self.text1.text()) \* 0.04) /

float(self.text2.text()))) < 0.5:

self.label6\_answer.setText('| Рекомендованная концентрация CO2 в помещении') #вывести сообщение

if (((float(self.text1.text()) \* 0.04) /

float(self.text2.text()))) >= 0.5 and (((float(self.text1.text()) \* 0.04) /

float(self.text2.text()))) < 1.5:

self.label6\_answer.setText(

'| Предельная концентрация в течении 8 часов. Найдите способ снизить СО2') #вывести сообщение

if (((float(self.text1.text()) \* 0.04) /

float(self.text2.text()))) >= 1.5:

self.label6\_answer.setText('| Смертельно опасная и высокая концентрация CO2!!!') #вывести сообщение

else: #в другом случае

print("") #ничего не выводить

self.label4\_answer.setText((str((float(self.text1.text()) \* 0.04) /

float(self.text2.text()))))#вывести сообщение

except ValueError: #отлов ошибок

self.label6\_answer.setText("Введите корректные данные!") #вывести сообщение, если пользователь ввел не число

except ZeroDivisionError: #отлов ошибок

self.label6\_answer.setText("На 0 делить нельзя!") #вывести сообщение, если пользователь делит на 0

ФУНКЦИИ ВЫЗОВА ОКОН

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

def window7(self): #функция события вызова окна с 3 главой 2 подглавой

self.w7 = Window7()#сокращенное название окна

self.w7.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

**5) файл project**

**В этот файл входят страницы с титульным листом, оглавлением, 1 и 2 главой и их подглавы.**

import project2 #импортируем в главный файл (c содержанием) данные из файла project2

import project3 #импортируем в главный файл (c содержанием) данные из файла project3

import project4 #импортируем в главный файл (c содержанием) данные из файла project4

from content.gl import \* #из файла gl импортируем контент для страницы оглавления, титульной, 1 и 2 главы и их подглав)

import sys #импортируем системный модуль для дальнейшего вызова окна на экран

from PyQt5.QtGui import QIcon #импортируем библиотики из PyQt5

from PyQt5 import QtGui #импортируем библиотики из PyQt5

from PyQt5.QtWidgets import (QLineEdit, QLabel, QPushButton, QApplication, QMainWindow, QAction) #импортируем библиотики из PyQt5

from PyQt5.QtCore import QSize #импортируем библиотики из PyQt5

ОКНО 2 ПОДГЛАВЫ 2 ГЛАВЫ

class Window5(QMainWindow): #создаем 2 подглаву главы 2

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Глава 2. Потребление природных ресурсов") #устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ТЕКСТ КОНТЕНТА

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(200, -30) #расположение текста

self.label\_1.setText(second2) #из файла gl импортируем контент second2

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_1.setFixedSize(1800, 1000) #размер области написания текста

self.label\_1.setStyleSheet("color: green;") #стиль текста

self.label\_1.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА

self.label\_3 = QLabel("", self) #задаем область для текста

self.label\_3.move(500, 730) #расположение текста

self.label\_3.setText(second3) #из файла gl импортируем контент second3

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_3.setFixedSize(1800, 200) #размер области написания текста

self.label\_3.setStyleSheet("color: green;") #стиль текста

self.label\_3.setFont(font) #сохранение изменений

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

КАРТИНКА ДЛЯ КОНТЕНТА СТРАНИЦЫ (формула)

but\_form2g = QPushButton(self) #создаем область расположения картинки

but\_form2g.setIcon(QIcon('content/2gl.png')) #импортируем картинку

but\_form2g.setIconSize(QSize(3000, 2000)) #размер области картинки

but\_form2g.setFixedSize(220, 100) #размер картинки

but\_form2g.move(150, 800) #расположение картинки

КНОПКА ДАЛЕЕ ДЛЯ ПЕРЕХОДА НА СЛЕДУЮЩУЮ ПОДГЛАВУ

buttonw2 = QPushButton("Далее", self) #создаем кнопку и даем ей название

buttonw2.move(1720, 900) #расположение кнопки

buttonw2.setFixedSize(90, 70) #размер кнопки

buttonw2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont() #импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw2.setFont(font) #сохранение изменений

КНОПКА НАЗАД ДЛЯ ВОЗВРАЩЕНИЯ НА ПРЕДЫДУЩУЮ ПОДГЛАВУ

buttonw3 = QPushButton("Назад", self) #создаем кнопку и даем ей название

buttonw3.move(1630, 900) #расположение кнопки

buttonw3.setFixedSize(90, 70) #размер кнопки

buttonw3.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont()#импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw3.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

button.clicked.connect(self.window1) #при нажатии переход на оглавление

buttonw2.clicked.connect(self.window2) #при нажатии переход вперед на подглаву

buttonw3.clicked.connect(self.window4) #при нажатии переход назад на подглаву

self.show() #вывести на экран изменения

ФУНКЦИИ ВЫЗОВА ОКОН

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

def window2(self): #функция события вызова подглавы

self.w2 = Window2() #сокращенное название окна

self.w2.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

def window4(self): #функция события вызова подглавы

self.w4 = Window4()#сокращенное название окна

self.w4.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

ОКНО 1 ПОДГЛАВЫ 2 ГЛАВЫ

class Window4(QMainWindow): #создаем 1 подглаву главы 2

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Глава 2. Потребление природных ресурсов") #устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ТЕКСТ КОНТЕНТА

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(200, 100) #расположение текста

self.label\_1.setText(second2) #из файла gl импортируем контент second2

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_1.setFixedSize(1800, 1000) #размер области написания текста

self.label\_1.setStyleSheet("color: green;") #стиль текста

self.label\_1.setFont(font) #сохранение изменений

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

ЗАГОЛОВОК

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(500, 190) #расположение текста

self.label\_1.setText("Глава 2. Потребление природных ресурсов") #текст для заголовка

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_1. setFixedSize (1800, 70) #размер области написания текста

self.label\_1.setStyleSheet("color: green; font: bold") #стиль шрифта

self.label\_1.setFont(font) #сохранение изменений

КНОПКА ДАЛЕЕ ДЛЯ ПЕРЕХОДА НА СЛЕДУЮЩУЮ ПОДГЛАВУ

buttonw4 = QPushButton("Далее", self) #создаем кнопку и даем ей название

buttonw4.move(1720, 900) #расположение кнопки

buttonw4.setFixedSize(90, 70) #размер кнопки

buttonw4.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont() #импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw4.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

button.clicked.connect(self.window1) #при нажатии переход на оглавление

buttonw4.clicked.connect(self.window5) #при нажатии переход на 2 подглаву 2 главы

self.show()#вывести на экран изменения

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

def window5(self): #функция события вызова окна с 2 подглавой 2 главы

self.w5 = Window5()#сокращенное название окна

self.w5.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

ОКНО 3 ПОДГЛАВЫ 2 ГЛАВЫ

class Window2(QMainWindow): #создаем 3 подглаву главы 2

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle('Глава 2. Потребление природных ресурсов') #устанавливаем название, которое будет выводиться сверху окна

self.setWindowIcon(QIcon('content/planet.jpg')) #устанавливаем лого окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

ЗАГОЛОВОК

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(350, 150) #расположение текста

self.label\_1.setText("Входные данные (вещественные числа указывать через точку)") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_1.setFixedSize(1800, 70) #размер области написания текста

self.label\_1.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_1.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА (РЕШЕНИЕ)

self.label\_4 = QLabel("", self) #задаем область для текста

self.label\_4.move(1, 750) #расположение текста

self.label\_4.setText("Решение: ") #сам текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_4.setFixedSize(500, 70) #размер области написания текста

self.label\_4.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_4.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА В РЕШЕНИИ

self.label\_5 = QLabel("", self) ) #задаем область для текста

self.label\_5.move(1, 830) #расположение текста

self.label\_5.setText("Рассчитаем формулу природопользования (α + β) \* N <(или >) P"

" и выясним больше или меньше темп возобновления "

"ресурсов P в сравнении с изъятием ресурсов (α + β) \* N") #сам текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

self.label\_5.setFixedSize(2600, 70) #размер области написания текста

self.label\_5.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_5.setFont(font) #сохранение изменений

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ

button.clicked.connect(self.window1) #при нажатии переход на оглавление

ВВОД ДАННЫХ ПОЛЬЗОВАТЕЛЕМ ДЛЯ ДАЛЬНЕЙШЕГО РАССЧЕТА

self.text1 = QLineEdit(self) #создание окна для ввода данных

self.text1.setPlaceholderText("Введите α") #плейсхолдер для подсказки ввода пользователю

self.text2 = QLineEdit(self) #создание окна для ввода данных

self.text2.setPlaceholderText("Введите β") #плейсхолдер для подсказки ввода пользователю

self.text3 = QLineEdit(self) #создание окна для ввода данных

self.text3.setPlaceholderText("Введите N") #плейсхолдер для подсказки ввода пользователю

self.text4 = QLineEdit(self) #создание окна для ввода данных

self.text4.setPlaceholderText("Введите P") #плейсхолдер для подсказки ввода пользователю

self.setGeometry(300, 200, 600, 450) #задаем размер блока с окнами ввода данных

self.text1.setGeometry(750, 250, 300, 30) #задаем размер 1 окна ввода данных

self.text2.setGeometry(750, 350, 300, 30) #задаем размер 2 окна ввода данных

self.text3.setGeometry(750, 450, 300, 30) #задаем размер 3 окна ввода данных

self.text4.setGeometry(750, 550, 300, 30) #задаем размер 4 окна ввода данных

КНОПКА РАССЧИТАТЬ

button\_count = QPushButton("Рассчитать", self) #даем название кнопке

button\_count.move(800, 650) #двигаем кнопку по экрану, задаем ей позицию

button\_count.setFixedSize(200, 80) #устанавливаем размер кнопки

button\_count.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button\_count.setFont(font) #сохраняем изменения

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ

button\_count.clicked.connect(self.checkstatus) #соединение кнопки и функции рассчета

КНОПКА НАЗАД ДЛЯ ВОЗВРАЩЕНИЯ НА ПРЕДЫДУЩУЮ ПОДГЛАВУ

buttonw4 = QPushButton("Назад", self) #создаем кнопку и даем ей название

buttonw4.move(1720, 900) #расположение кнопки

buttonw4.setFixedSize(90, 70) #размер кнопки

buttonw4.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

font = QtGui.QFont()#импортируем шрифт

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(12) #размер шрифта

buttonw4.setFont(font) #сохранение изменений

buttonw4. clicked.connect (self.window5) #соединение кнопки и окна

ТЕКСТ КОНТЕНТА (ОТВЕТ)

self.label\_4 = QLabel("", self) #задаем область для текста

self.label\_4.move(1,900) #расположение текста

self.label\_4.setText("Решение: ") #сам текст

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_4.setFixedSize(500, 70) #размер области написания текста

self.label\_4.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_4.setFont(font) #сохранение изменений

ТЕКСТ КОНТЕНТА В ОТВЕТЕ ДО НАЖАТИЯ КНОПКИ РАССЧИТАТЬ

self.label\_answer = QLabel("(α + β) \* N", self) #задаем область для текста

self.label\_answer.move(150, 903) #расположение текста

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(10) #размер шрифта

self.label\_answer.setFixedSize(500, 70) #размер области написания текста

self.label\_answer.setStyleSheet("color: green; font: bold") #стиль текста

self.label\_answer.setFont(font) #сохранение изменений

self.label3\_answer = QLabel("< или (>)", self)#задаем область для текста

self.label3\_answer.move(250, 903) #расположение текста

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(10) #размер шрифта

self.label3\_answer.setFixedSize(500, 70) #размер области написания текста

self.label3\_answer.setStyleSheet("color: green; font: bold") #стиль текста

self.label3\_answer.setFont(font) #сохранение изменений

self.label4\_answer = QLabel("P", self)#задаем область для текста

self.label4\_answer.move(340, 903) #расположение текста

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(10) #размер шрифта

self.label4\_answer.setFixedSize(500, 70) #размер области написания текста

self.label4\_answer.setStyleSheet("color: green; font: bold") #стиль текста

self.label4\_answer.setFont(font) #сохранение изменений

self.label6\_answer = QLabel("P", self)#задаем область для текста

self.label6\_answer.move(500, 903) #расположение текста

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(10) #размер шрифта

self.label6\_answer.setFixedSize(900, 70) #размер области написания текста

self.label6\_answer.setStyleSheet("color: green; font: bold") #стиль текста

self.label6\_answer.setFont(font) #сохранение изменений

ФУНКЦИИ ДЛЯ ПРОВЕРКИ ВВОДА ДАННЫХ ПОЛЬЗОВАТЕЛЕМ И РАСЧЕТА

def checkstatus(self): #функция проверки ввода данных

if self.text1.text() == "": #если пустое значение в 1 окне ввода

print("") #то ничего не выводить

self.text1.setFocus() #возврат к вводу

if self.text2.text() == "": #если пустое значение в 2 окне ввода

print("")#то ничего не выводить

self.text2.setFocus()#возврат к вводу

if self.text3.text() == "": #если пустое значение в 3 окне ввода

print("")#то ничего не выводить

self.text3.setFocus()#возврат к вводу

if self.text4.text() == "": #если пустое значение в 4 окне ввода

print("")#то ничего не выводить

self.text4.setFocus()#возврат к вводу

else: #в другом случае

try: #отлов ошибок

self.label\_answer.setText(str((float(self.text1.text()) + float(self.text2.text())) \*

float(self.text3.text())))#вывод значения

if (((float(self.text1.text()) + float(self.text2.text())) \*

float(self.text3.text()))) > (((float(self.text4.text())))):

self.label3\_answer.setText('>')

self.label6\_answer.setText('| Значит изъятие ресурсов больше темпа их возобновления') #если значение больше, то вывести этот текст

else: #в другом случае

self.label3\_answer.setText('<')

self.label6\_answer.setText('| Значит изъятие ресурсов меньше темпа их возобновления')

self.label4\_answer.setText(str((float(self.text4.text()))))#если значение меньше, то вывести этот текст

except ValueError: #исключения

self.label6\_answer.setText("Введите корректные данные!") #если пользователь вводит что-то кроме чисел, то вывод этого сообщения на экран

ФУНКЦИИ ВЫЗОВА ОКОН

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

def window5(self): #функция события вызова окна 2 подглавы 2 главы

self.w5 = Window5()#сокращенное название окна

self.w5.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

ОКНО 1 ГЛАВЫ

class Window3(QMainWindow): #создаем 1 главу

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Глава 1. Рациональное природопользование") #устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ТЕКСТ КОНТЕНТА

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(400, 1) #расположение текста

self.label\_1.setText(first) #из файла gl импортируем контент first

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(15) #размер шрифта

self.label\_1.setFixedSize(1300, 1000) #размер области написания текста

self.label\_1.setStyleSheet("color: green;") #стиль текста

self.label\_1.setFont(font) #сохранение изменений

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

ЗАГОЛОВОК

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(480, 190) #расположение текста

self.label\_1.setText("Глава 1. Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(20) #размер шрифта

self.label\_1.setFixedSize(1800, 70) #размер области написания текста

self.label\_1.setStyleSheet("color: green; font: bold;") #стиль текста

self.label\_1.setFont(font) #сохранение изменений

КАРТИНКА CХЕМЫ В КОНТЕНТЕ

sxema = QPushButton(self) #создаем область расположения картинки

sxema.setIcon(QIcon('content/sxema.jpg')) #импортируем картинку

sxema.setIconSize(QSize(500, 400)) #размер области картинки

sxema.setFixedSize(500, 290) #размер картинки

sxema.move(650, 700) #расположение картинки

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

button.clicked.connect(self.window1) #при нажатии переход на оглавление

self.show()#вывести на экран изменения

ФУНКЦИЯ ВЫЗОВА ОКНА С ОГЛАВЛЕНИЕМ

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопку

ОКНО ОГЛАВЛЕНИЯ

class MenuGlav(QMainWindow): #создаем оглавление

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Рациональное природопользование") #устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА НА ГЛАВНУЮ

button = QPushButton("На главную", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;")#стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ КОНТЕНТА СТРАНИЦЫ

but = QPushButton(self) #создаем область расположения картинки

but.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but.setIconSize(QSize(80, 80)) #размер области картинки

but.setFixedSize(80, 80) #размер картинки

but.move(1700, 30) #расположение картинки

ОКНО

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowTitle('Рациональное природопользование') #задаем название окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

КНОПКА 1 ГЛАВЫ

button1 = QPushButton("Глава 1. Рациональное природопользование", self) #обозначаем и называем кнопку

button1.move(600, 400) #расположение кнопки

button1.hide() #не переходить до нажатия

self.button1 = button1 #перейти если произошло нажатие

button1.setFixedSize(600, 80) #размер кнопки

font.setFamily('CeraPro-Bold') # сам шрифт

font.setPointSize(14) # размер шрифта

button1.setFont(font) #сохранение изменений

button1.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

КНОПКА 2 ГЛАВЫ

button2 = QPushButton("Глава 2. Потребление природных ресурсов", self) #обозначаем и называем кнопку

button2.move(600, 500) #расположение кнопки

button2.hide() #не переходить до нажатия

self.button2 = button2 #перейти если произошло нажатие

button2.setFixedSize(600, 80) #размер кнопки

font.setFamily('CeraPro-Bold') # сам шрифт

font.setPointSize(14) # размер шрифта

button2.setFont(font) #сохранение изменений

button2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

КНОПКА 3 ГЛАВЫ

button3 = QPushButton("Глава 3. Углекислый газ в атмосфере Земли", self) #обозначаем и называем кнопку

button3.move(600, 600) #расположение кнопки

button3.hide() #не переходить до нажатия

self.button3 = button3 #перейти если произошло нажатие

button3.setFixedSize(600, 80) #размер кнопки

font.setFamily('CeraPro-Bold') # сам шрифт

font.setPointSize(14) # размер шрифта

button3.setFont(font) #сохранение изменений

button3.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

КНОПКА 4 ГЛАВЫ

button4 = QPushButton("Глава 4. Плотность популяции животного мира", self) #обозначаем и называем кнопку

button4.move(600, 700) #расположение кнопки

button4.hide() #не переходить до нажатия

self.button4 = button4 #перейти если произошло нажатие

button4.setFixedSize(600, 80) #размер кнопки

font.setFamily('CeraPro-Bold') # сам шрифт

font.setPointSize(14) # размер шрифта

button4.setFont(font) #сохранение изменений

button4.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

КНОПКА 5 ГЛАВЫ

button5 = QPushButton("Глава 5. Список литературы", self)#обозначаем и называем кнопку

button5.move(600, 800) #расположение кнопки

button5.hide() #не переходить до нажатия

self.button5 = button5 #перейти если произошло нажатие

button5.setFixedSize(600, 80) #размер кнопки

font.setFamily('CeraPro-Bold') # сам шрифт

font.setPointSize(14) # размер шрифта

button5.setFont(font) #сохранение изменений

button5.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль кнопки

ЗАГОЛОВОК

self.label\_1 = QLabel("", self) #задаем область для текста

self.label\_1.move(720, 250)#расположение текста

self.label\_1.setText("Оглавление" ") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_1.setFixedSize(320, 70) #размер области написания текста

self.label\_1.setStyleSheet("color: green; font: bold;") #стиль текста

self.label\_1.setFont(font) #сохранение изменений

ВЫВОД КНОПОК ГЛАВ НА ЭКРАН

self.button1.show() #вывод кнопки 1 главы

self.button2.show()#вывод кнопки 2 главы

self.button3.show()#вывод кнопки 3 главы

self.button4.show()#вывод кнопки 4 главы

self.button5.show()#вывод кнопки 5 главы

self.label\_1.show()#вывод заголовка

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

self.button1.clicked.connect(self.window3) #при нажатии переход на 1 главу

self.button2.clicked.connect(self.window4) при нажатии переход на 2 главу

self.button3.clicked.connect(self.window6) при нажатии переход на 3 главу

self.button4.clicked.connect(self.window9) при нажатии переход на 4 главу

self.button5.clicked.connect(self.window12) при нажатии переход на 5 главу

ФУНКЦИИ ВЫЗОВА ОКОН

def window3(self): #функция события вызова 1 главы

self.w3 = Window3()#сокращенное название окна

self.w3.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

def window4(self): #функция события вызова 2 главы

self.w4 = Window4()#сокращенное название окна

self.w4.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

def window6(self): #функция события вызова 3 главы

self.w6 = Window6()#сокращенное название окна

self.w6.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

def window9(self): #функция события вызова 4 главы

self.w9 = Window9()#сокращенное название окна

self.w9.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

def window12(self): #функция события вызова 5 главы

self.w12 = Window12()#сокращенное название окна

self.w12.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

def title(self): #функция события вызова титульного листа

self.ttl = Titul()#сокращенное название окна

self.ttl.showMaximized()#при открытие программы размер окна на весь экран

self.hide()#скрыть окно до нажатия на кнопку

ОКНО ТИТУЛЬНОГО ЛИСТА

class Titul(QMainWindow): #создаем титульный лист

def \_\_init\_\_(self): #вызываем функцию, которая отвечает за отрисовку окна

super().\_\_init\_\_() #связываем это окно с окном оглавления

self.setWindowTitle("Рациональное природопользование") #устанавливаем название, которое будет выводиться сверху окна

self.setGeometry(500, 500, 500, 500) #задаем размер окна размер окна

self.setWindowIcon(QIcon('content/planet.jpg')) #добавляем логотип окна

self.setStyleSheet("background-color: white;") #указываем, что фон нашего окна будет белый

self.initUI() #связываем этот класс с функцией в нем, которая отвечает за контент и логику окна

КНОПКА ВЫХОДА

def initUI(self): #функция контента и логики окна

exitAction = QAction('Если хотите закрыть программу нажмите сюда', self) #запрос на подтверждение выхода из программы

exitAction.triggered.connect(self.close) #при нажатии на exitAction будет произведен выход из программы

МЕНЮ

menubar = self.menuBar() #отрисовывем меню в окне

fileMenu = menubar.addMenu('Exit') #прописываем название раздела

fileMenu.addAction(exitAction) #добавляем раздел Выхода в меню

КНОПКА ОГЛАВЛЕНИЯ

button = QPushButton("Оглавление", self) #даем название кнопке

button.move(1, 30) #двигаем кнопку по экрану, задаем ей позицию

button.setFixedSize(200, 80) #устанавливаем размер кнопки

button.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #задаем дизайн кнопке

font = QtGui.QFont() #вызываем модуль, который отвечает за шрифт

font.setFamily('CeraPro-Bold') #обозначаем название шрифта

font.setPointSize(14) #размер шрифта

button.setFont(font) #сохраняем изменения

ЗАГОЛОВОК

self.label\_2 = QLabel("", self) #задаем область для текста

self.label\_2.move(400, 30) #расположение текста

self.label\_2.setText(" Рациональное природопользование") #сам текст с заголовком

font.setFamily('CeraPro-Bold') #шрифт

font.setPointSize(30) #размер шрифта

self.label\_2.setFixedSize(1050, 80) #размер области написания текста

self.label\_2.setStyleSheet("background-color : green; color: white; font: bold; border: 3px solid lightgreen;") #стиль текста

self.label\_2.setFont(font) #сохранение изменений

КАРТИНКА ДЛЯ КРАСОТЫ В ПРАВОМ ВЕРХНЕМ УГЛУ ЭКРАНА

but\_pict = QPushButton(self) #создаем область расположения картинки

but\_pict.setIcon(QIcon('content/planet.jpg')) #импортируем картинку

but\_pict.setIconSize(QSize(80, 80)) #размер области картинки

but\_pict.setFixedSize(80, 80) #размер картинки

but\_pict.move(1700, 30) #расположение картинки

КАРТИНКА В КОНТЕНТЕ

sxema = QPushButton(self) #создаем область расположения картинки

sxema.setIcon(QIcon('content/sxema.jpg')) #импортируем картинку

sxema.setIconSize(QSize(800, 800)) #размер области картинки

sxema.setFixedSize(790, 770) #размер картинки

sxema.move(520, 230) #расположение картинки

СОЗДАЕМ СОЕДИНЕНИЕ КНОПОК И СОБЫТИЙ НАЖАТИЯ НА ОКНА

button.clicked.connect(self.window1) #при нажатии переход на оглавлени

self.show()#вывести на экран изменения

ФУНКЦИИ ВЫЗОВА ОКОН

def window1(self): #функция события вызова окна с оглавлением

self.w = MenuGlav() #сокращенное название окна

self.w.showMaximized() #при открытие программы размер окна на весь экран

self.hide() #скрыть окно до нажатия на кнопки

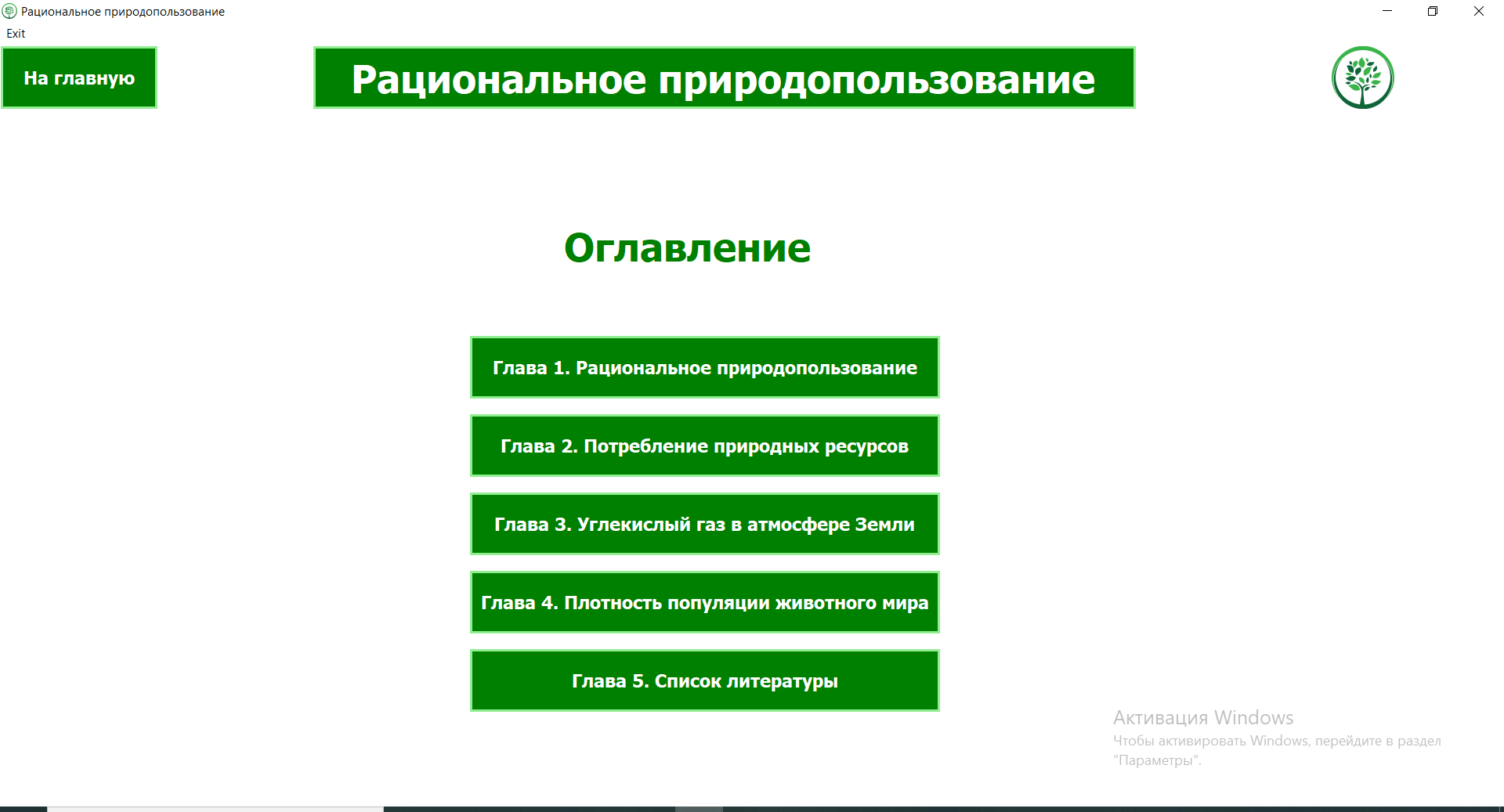
if \_\_name\_\_ == '\_\_main\_\_': #структура сбора окон между собой

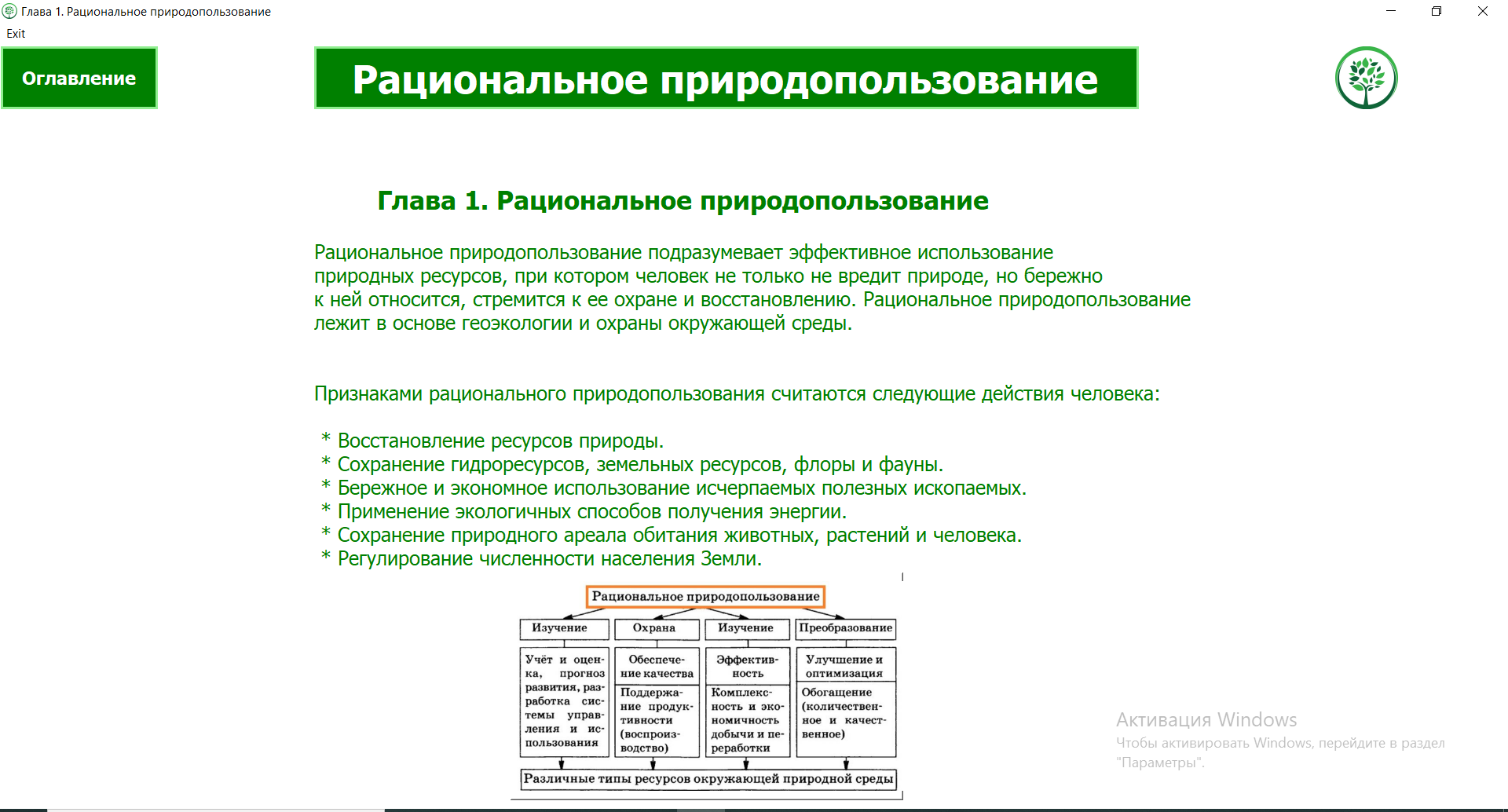
app = QApplication(sys.argv) #вывод окон

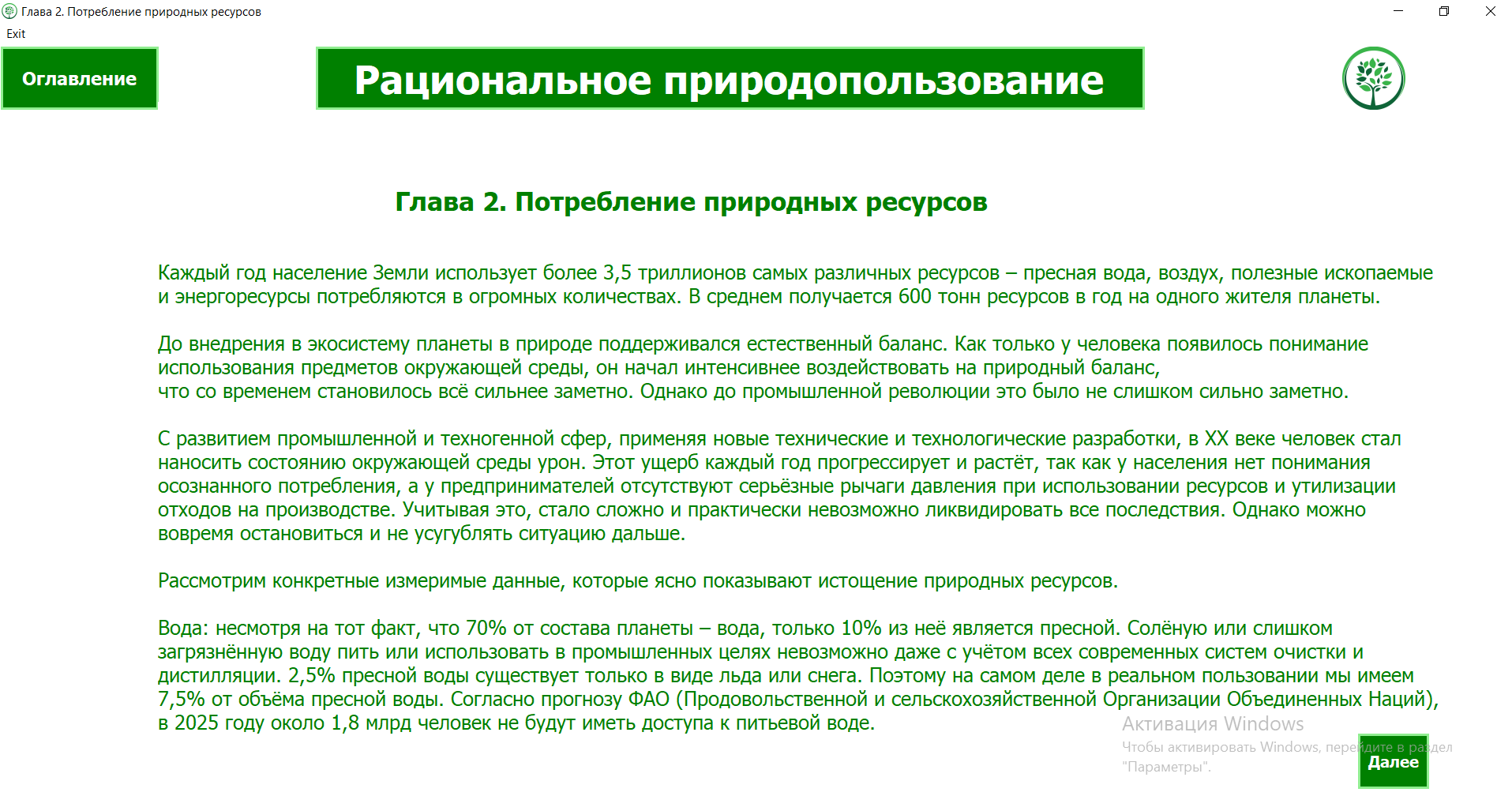
mg = Titul() #первое окно, которое показывается при запуске программы

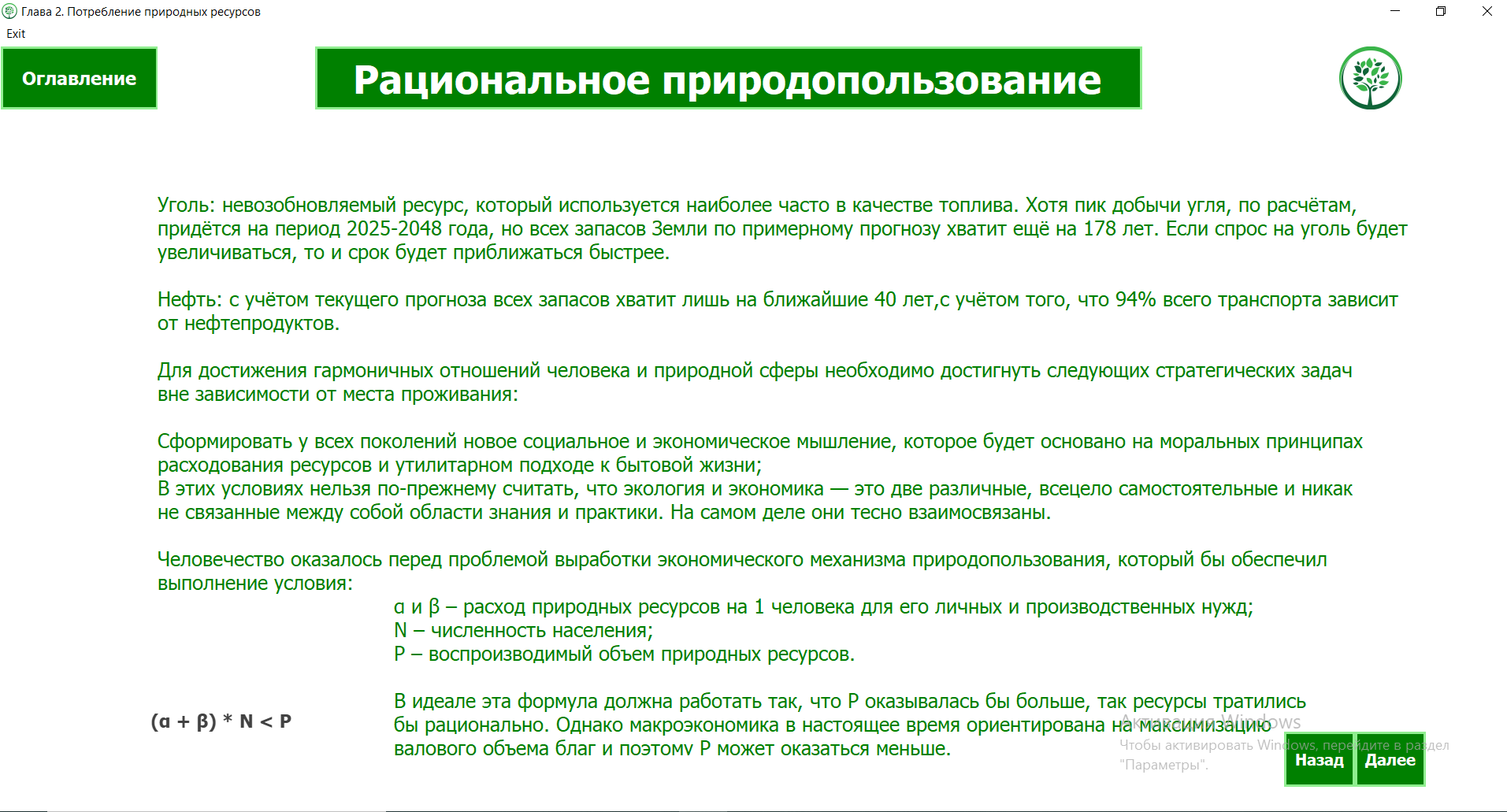
mg.showMaximized() #размер окна на весь экран

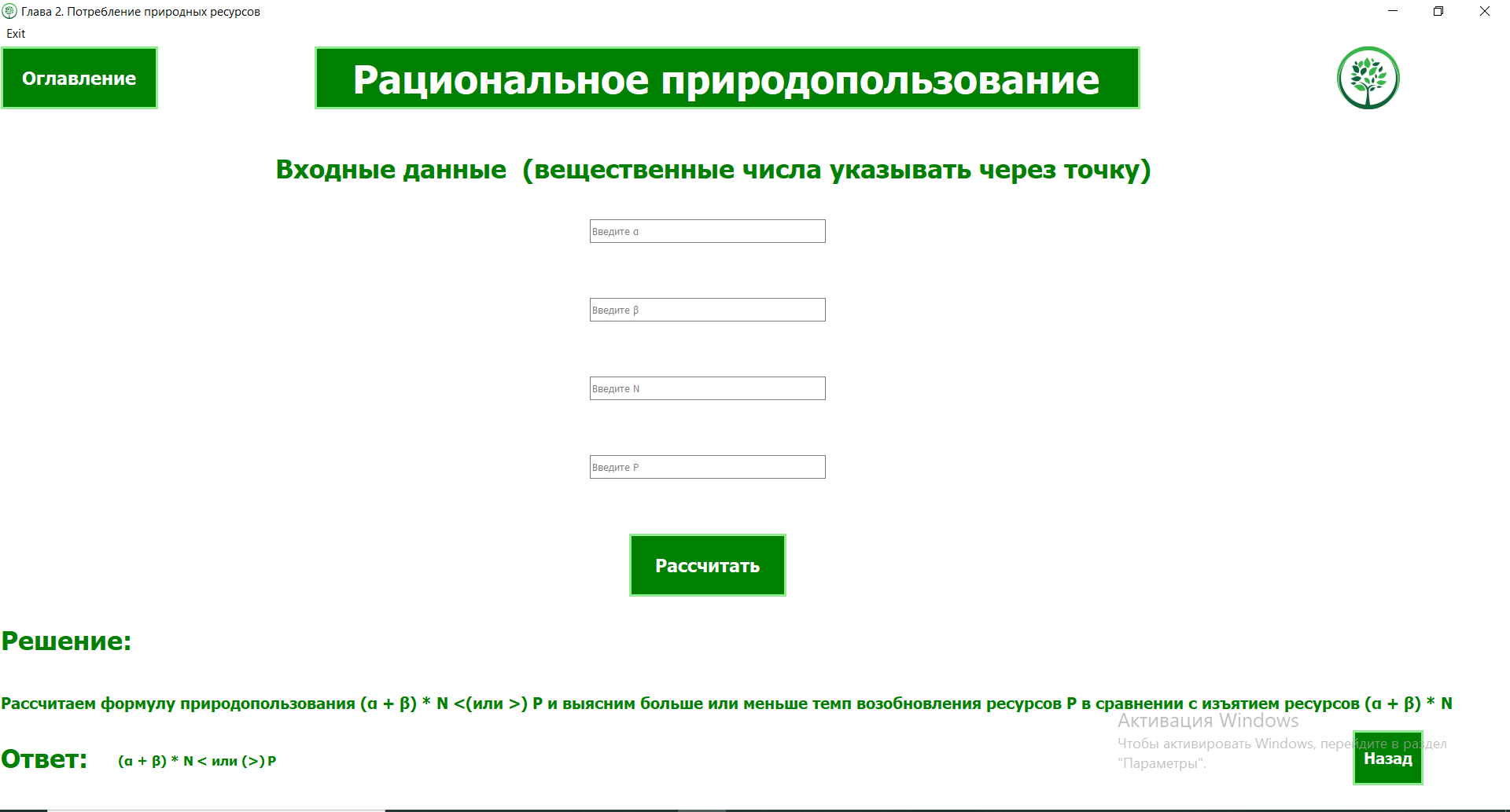
sys.exit(app.exec\_()) #системный вызов

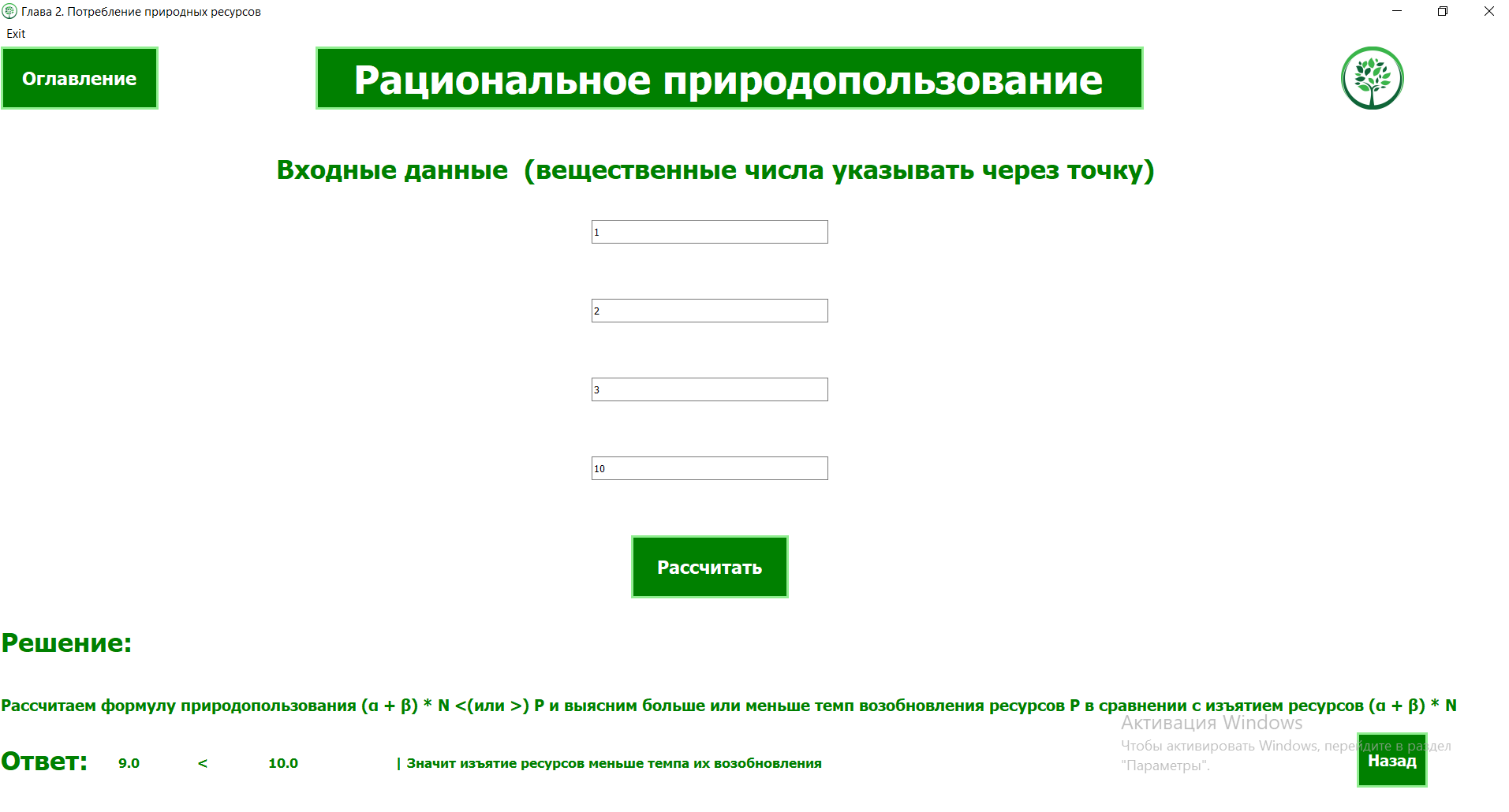
 

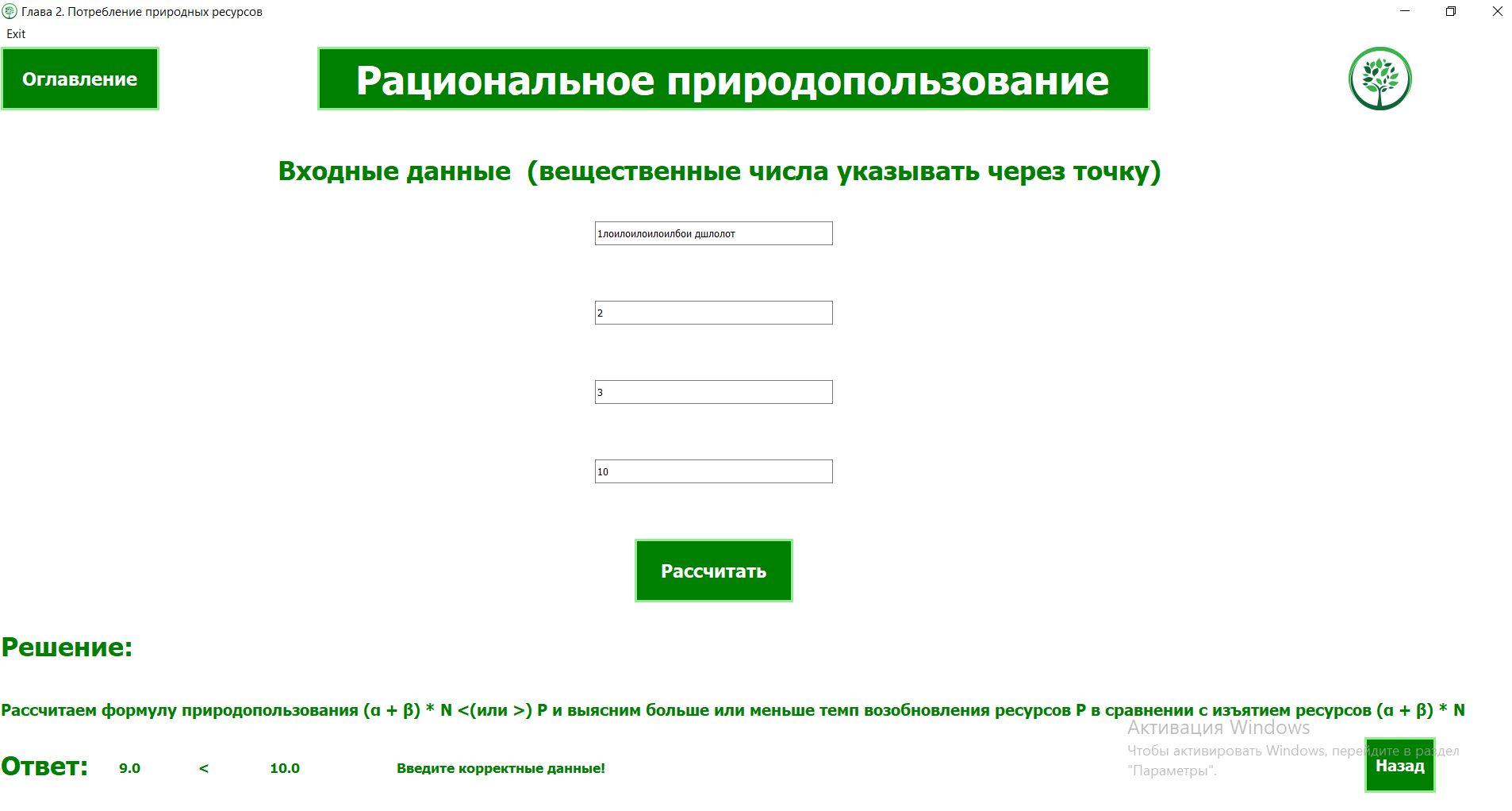


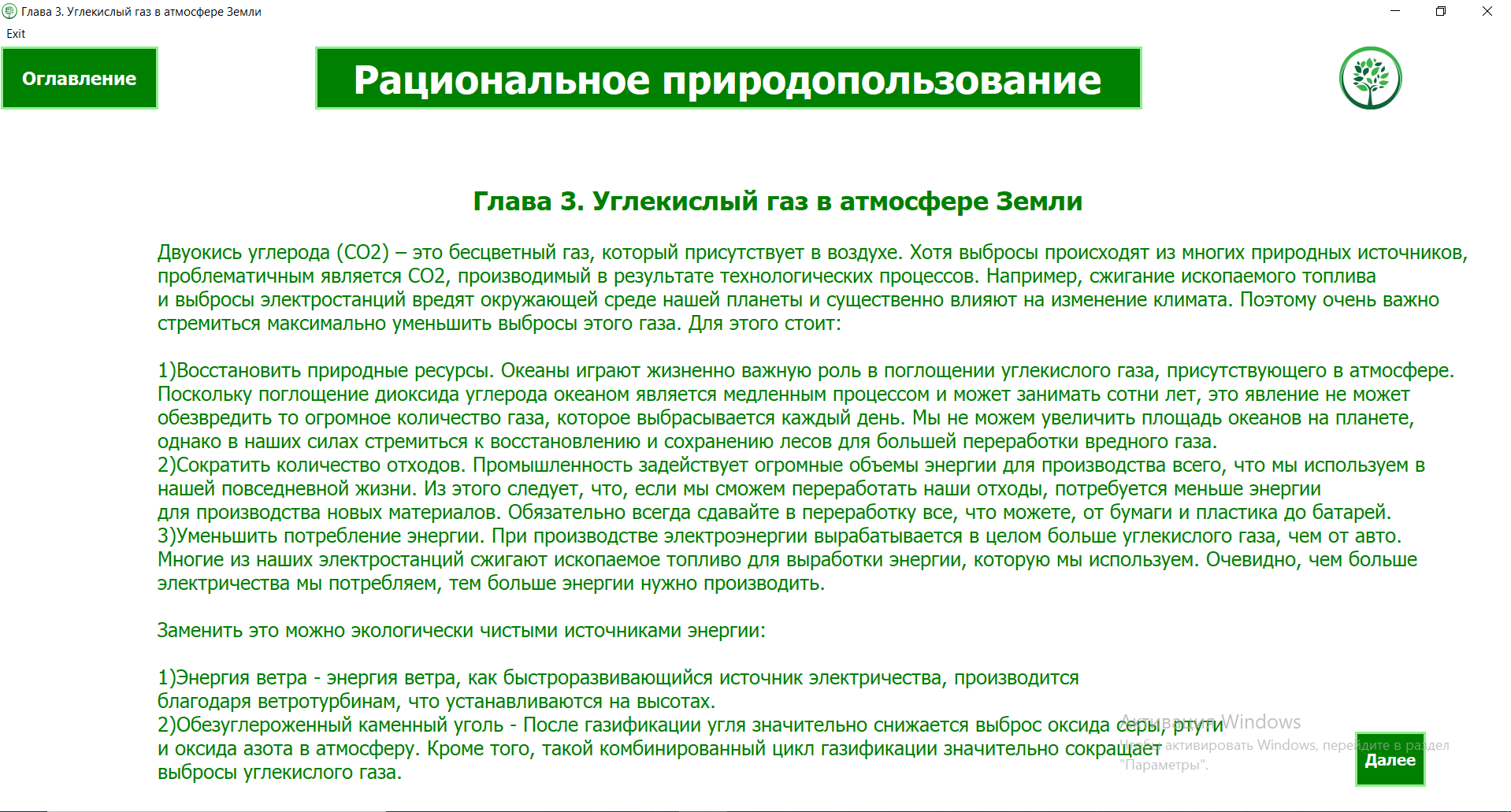




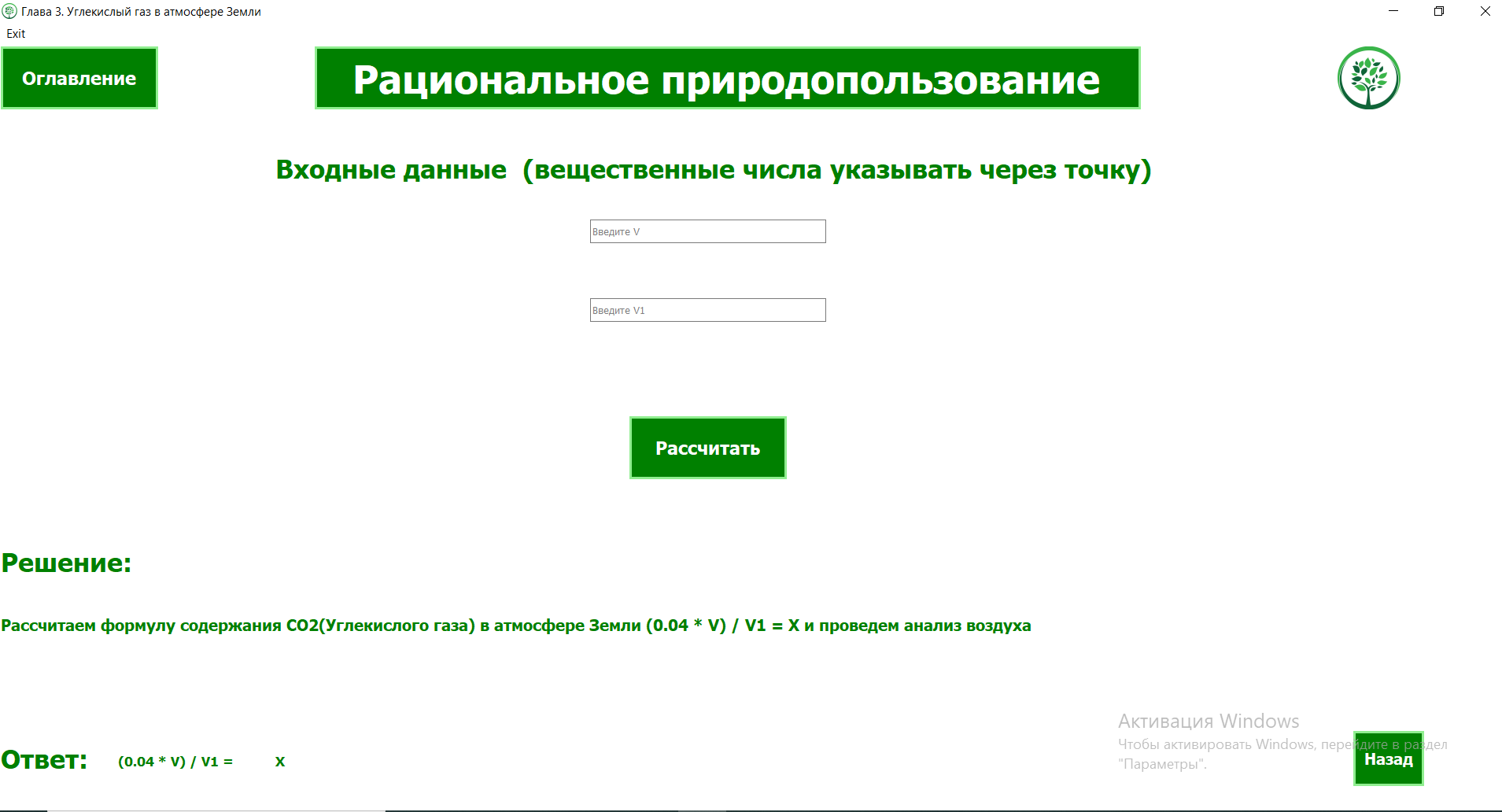


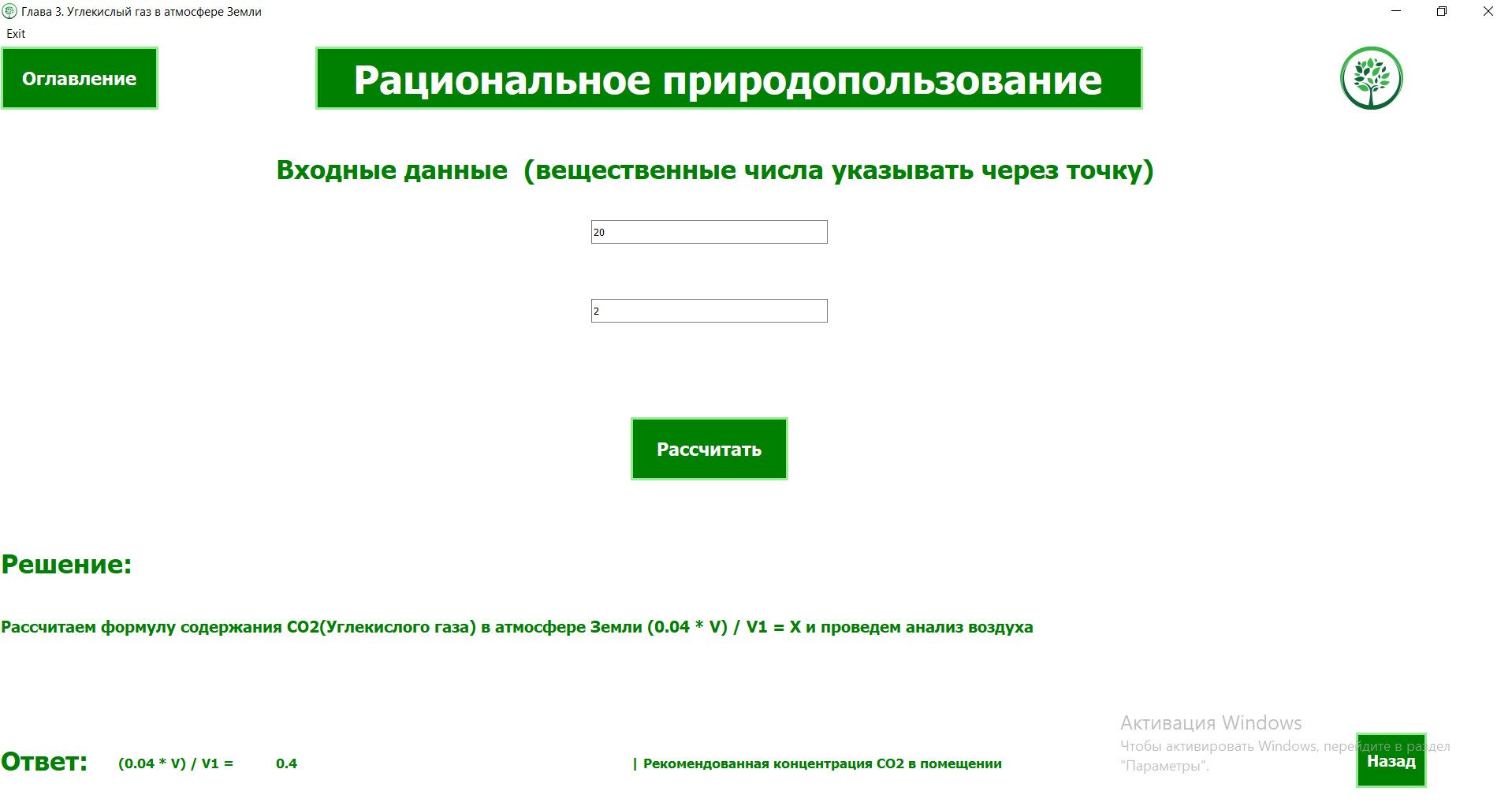


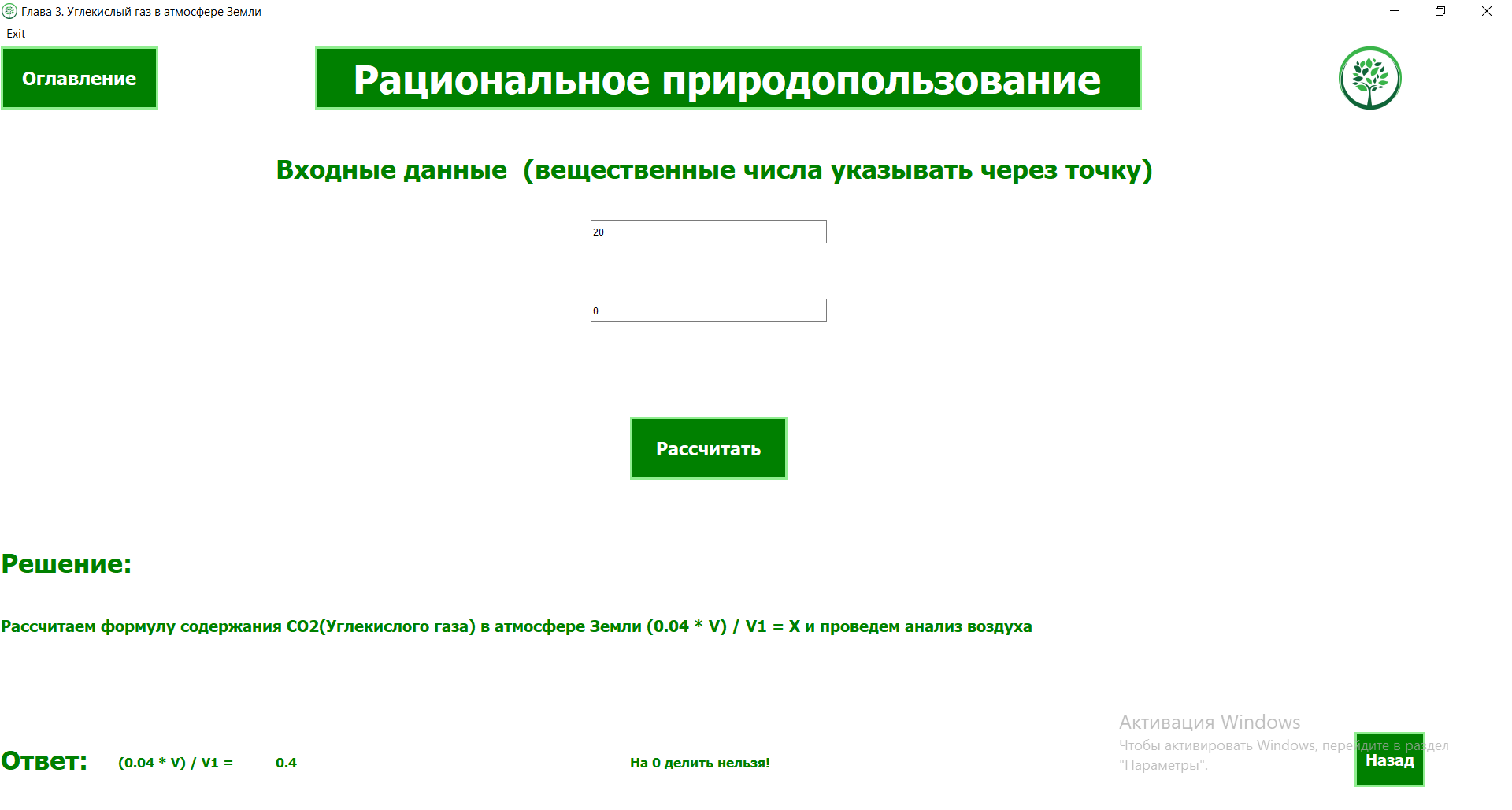


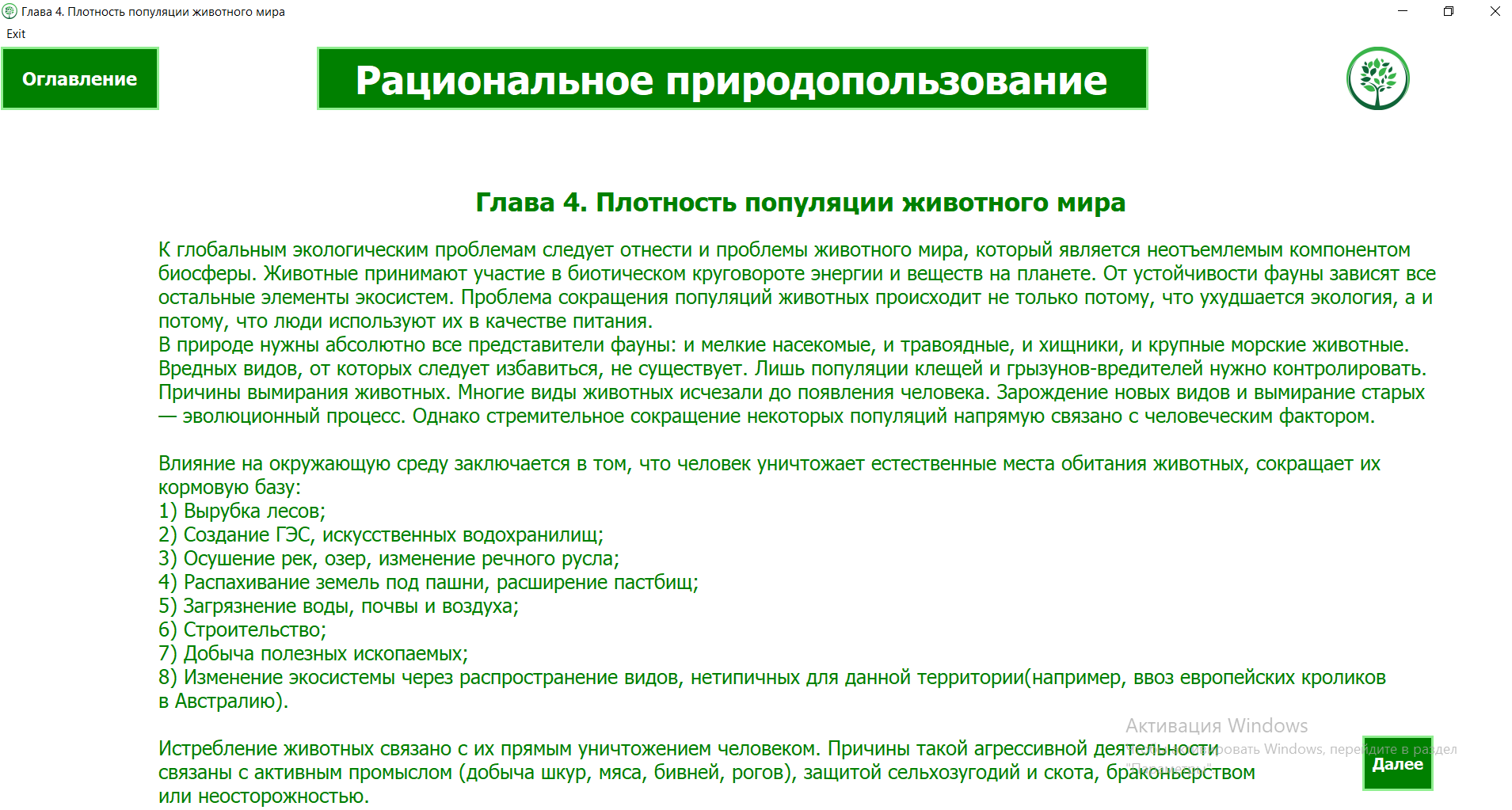


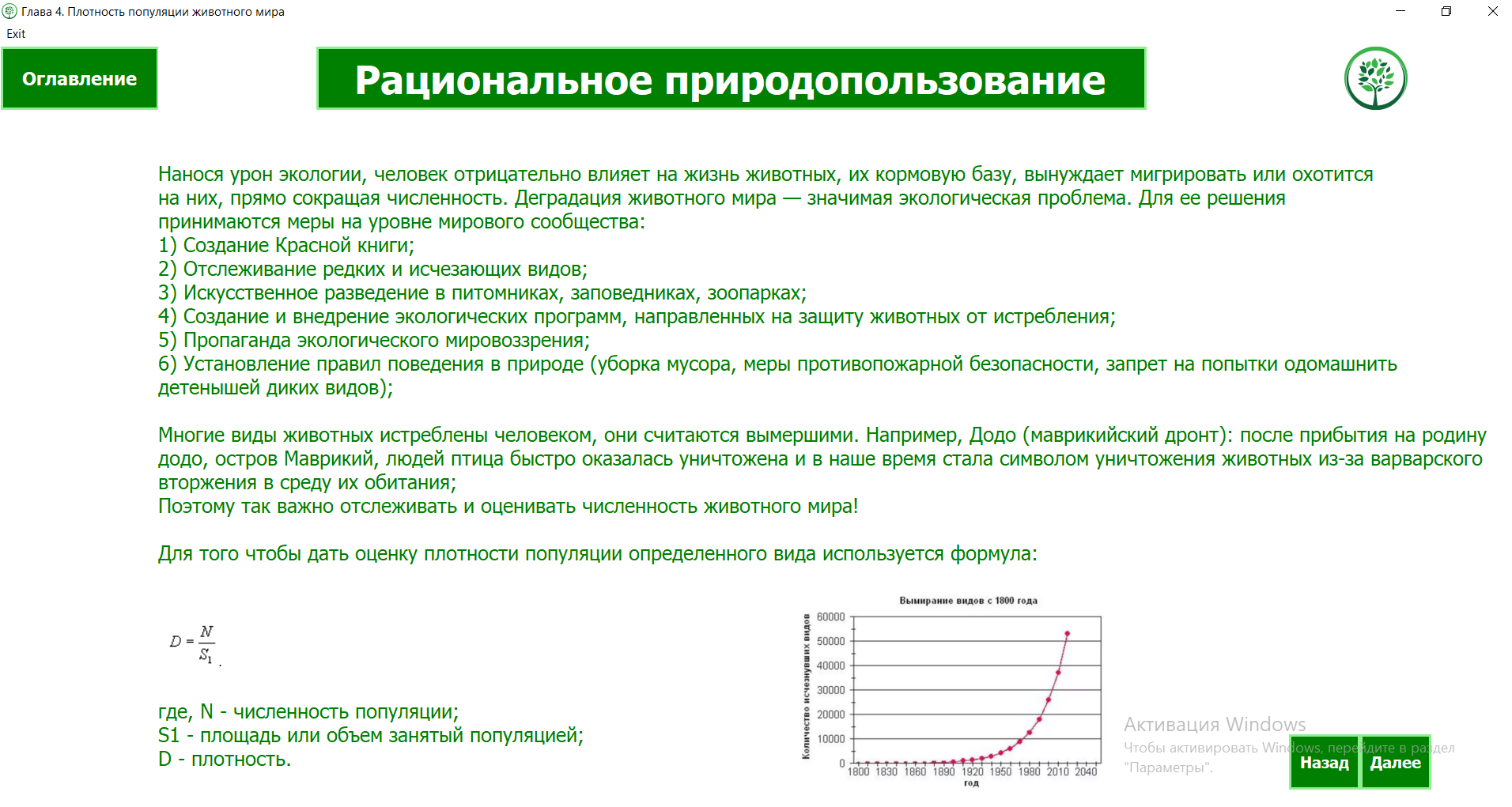


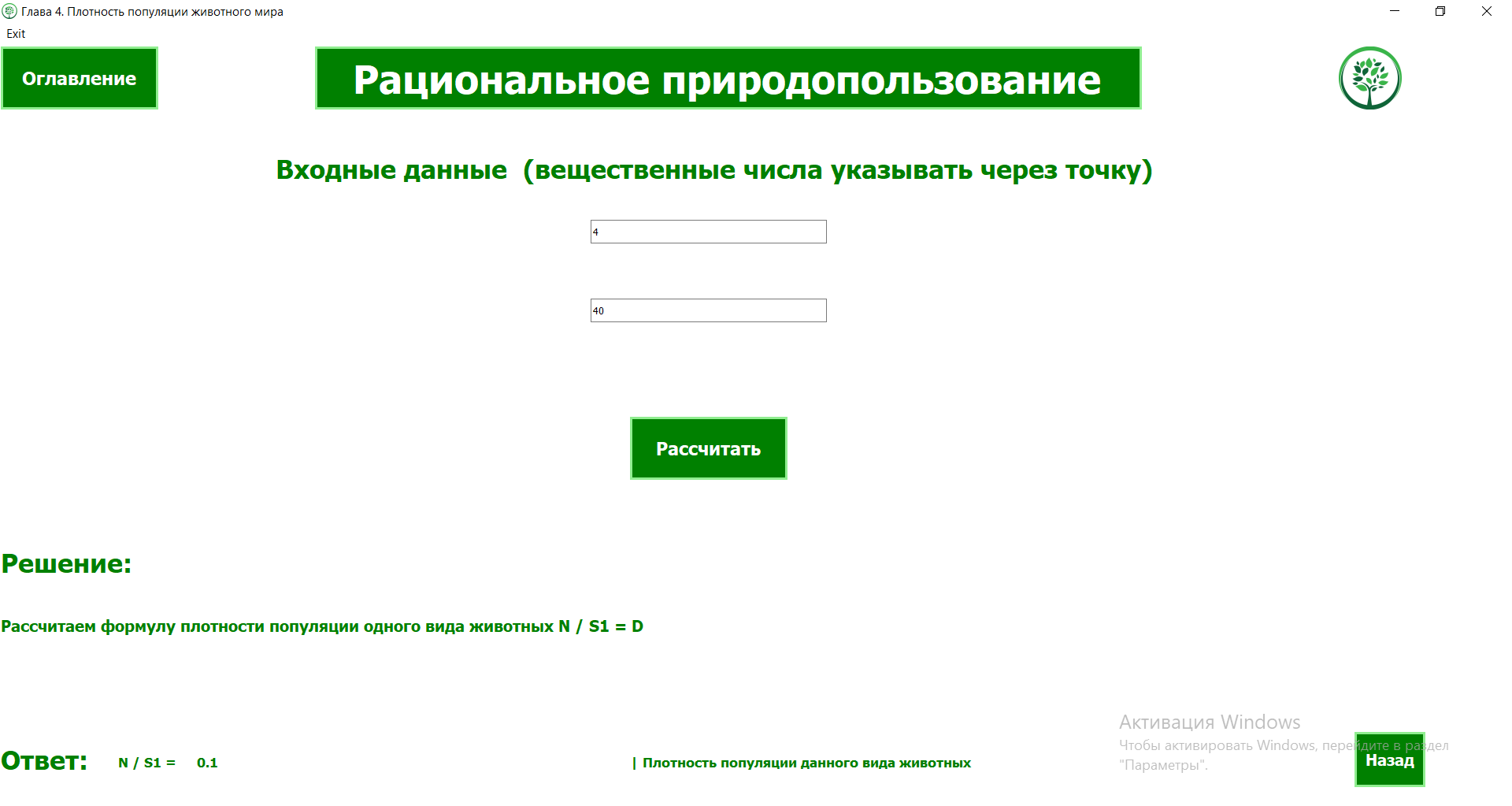














****