

Школа информационных технологий и робототехники  
 Направление подготовки 09.03.04 «Программная инженерия»  
 Отделение школы (НОЦ) информационных технологий  
 Специальность «Разработка программно-информационных систем»

### БАКАЛАВРСКАЯ РАБОТА

Тема работы
<b>Разработка симулятора для долгосрочного хранения CO<sub>2</sub> с использованием методов машинного обучения</b>

УДК 004.925.84:004.415.2:004.422.8:004.85

Студент

Группа	ФИО	Подпись	Дата
8K82	Де-Гальд Владислав Александрович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Фофанов Олег Борисович	К.Т.Н		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Гасанов М.А.о.	Д.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОБД	Мезенцева И.Л.			

### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.Т.Н.		

## Результаты обучения

Код результатов	Результат обучения (выпускник должен быть готов)
<b><i>Профессиональные и общепрофессиональные компетенции</i></b>	
P1	Применять базовые и специальные естественнонаучные и математические знания для комплексной инженерной деятельности по созданию, внедрению и эксплуатации геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием геоинформационных систем и технологий, информационных систем в бизнесе, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Выполнять комплексные инженерные проекты по созданию информационных систем и технологий, а также средств их реализации (информационных, методических, математических, алгоритмических, технических и программных).
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P6	Внедрять, эксплуатировать и обслуживать современные геоинформационные системы и технологии, информационные системы и технологии в бизнесе, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
<b><i>Универсальные (общекультурные) компетенции</i></b>	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом. Владеть иностранным языком (углублённый английский язык), позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена команды, состоящей из специалистов различных направлений и квалификаций,
P10	Демонстрировать личную ответственность за результаты работы и готовность следовать профессиональной этике и нормам ведения комплексной инженерной деятельности.
P11	Демонстрировать знания правовых, социальных, экологических и культурных аспектов комплексной инженерной деятельности, а также готовность к достижению должного уровня физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности.

Школа информационных технологий и робототехники  
 Направление подготовки 09.03.04 «Программная инженерия»  
 Отделение школы (НОЦ) информационных технологий  
 Специальность «Разработка программно-информационных систем»

УТВЕРЖДАЮ:  
 Руководитель ООП  
 \_\_\_\_\_ Чердынцев Е.С.  
 (Подпись) (Дата)                      (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

бакалаврской работы
---------------------

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8K82	Де-Гальд Владислав Александрович

Тема работы:

<b>Разработка симулятора для долгосрочного хранения CO<sub>2</sub> с использованием методов машинного обучения</b>
--

Утверждена приказом и.о. директора ИШИТР	Приказ № 40-50/с от 09.02.2022 г.
--	-----------------------------------

Срок сдачи студентом выполненной работы:	
--	--

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>	Разработка веб- и десктопного приложений для интерактивного моделирования сценариев закачки и миграции, визуализации результатов долгосрочного хранения CO <sub>2</sub> , а также оптимизации расположения скважин для максимального захоронения углекислого газа в структурных ловушках и минимизации утечки газа.
---------------------------------	---

<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>	<ol style="list-style-type: none"> <li>1. Изучение модуля co2lab открытого программного обеспечения MRST, разработанного группой вычислительной науки о Земле компании SINTEF, Норвегия.</li> <li>2. Анализ мирового опыта применения машинного обучения для оптимизации расположения скважин.</li> <li>3. Исследование применения методов глубокого обучения с подкреплением для оптимизации расстановки скважин с целью максимизации захоронения углекислого газа в структурных ловушках и предотвращения утечек газа.</li> <li>4. Разработка серверной части.</li> <li>5. Проектирование пользовательского интерфейса как десктоп-, так и веб-приложений.</li> <li>6. Разработка десктопного приложения.</li> <li>7. Разработка клиентского веб-приложения.</li> <li>8. Тестирование.</li> <li>9. Документация.</li> <li>10. Релиз приложения.</li> </ol>
<b>Перечень графического материала</b>	<p>Все диаграммы Презентация в формате *.pdf</p>
<b>Консультанты по разделам выпускной квалификационной работы</b> <i>(с указанием разделов)</i>	
<b>Раздел</b>	<b>Консультант</b>
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Гасанов Магеррам Али оглы
Социальная ответственность	Мезенцева Ирина Леонидовна
<b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b>	
Заключение	

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	
---	--

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Фофанов О.Б.	к.т.н		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8К82	Де-Гальд Владислав Александрович		

Школа информационных технологий и робототехники  
 Направление подготовки 09.03.04 «Программная инженерия»  
 Отделение школы (НОЦ) информационных технологий  
 Специальность «Разработка программно-информационных систем»  
 Период выполнения весенний семестр 2021/2022 учебного года

Форма представления работы:

бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

### КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Основная часть	75
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
	Социальная ответственность	10

**СОСТАВИЛ:**

**Руководитель ВКР**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Фофанов О.Б.	к.т.н		

**СОГЛАСОВАНО:**

**Руководитель ООП**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Чердынцев Е.С.	к.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8К82	Де-Гальду Владиславу Александровичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Оклад руководителя – 30000 руб. Оклад программиста – 15000 руб.
2. Нормы и нормативы расходования ресурсов	Премимальный коэффициент руководителя 30%; Премимальный коэффициент инженера 20%; Доплаты и надбавки руководителя 30%; Доплаты и надбавки руководителя 30%; Коэффициент дополнительной заработной платы 12%; Накладные расходы 16%; Районный коэффициент 1,3.
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Коэффициент отчислений на уплату во внебюджетные фонды 30,2%

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Определение потенциального потребителя результатов исследования, SWOT-анализ разработанной стратегии
2. Планирование и формирование бюджета научных исследований	Определение структуры работы. Расчет трудоемкости выполнения работ. Подсчет бюджета исследования
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Рассчитать показатели финансовой эффективности, ресурсоэффективности и эффективности исполнения

**Перечень графического материала (с точным указанием обязательных чертежей):**

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. Альтернативы проведения НИ
4. График проведения и бюджет НИ
5. Оценка ресурсной, финансовой и экономической эффективности НИ

<b>Дата выдачи задания для раздела по линейному графику</b>	04.03.2022
---	------------

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Гасанов М.А.о.	д.э.н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8К82	Де-Гальд В.А.		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8К82	Де-Гальду Владиславу Александровичу

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

Разработка симулятора для долгосрочного хранения CO<sub>2</sub> с использованием методов машинного обучения

### Исходные данные к разделу «Социальная ответственность»:

#### Введение

- Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения.
- Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации.

*Объект исследования:* разработка веб- и десктопного приложений-симуляторов для долгосрочного хранения CO<sub>2</sub> с использованием методов машинного обучения.  
*Область применения:* ученые, нефтяники, заинтересованные вопросом подземного хранения углекислого газа.  
*Рабочая зона:* офис.  
*Размеры помещения:* 7х3х3 м<sup>3</sup>.  
*Количество и наименование оборудования рабочей зоны:* 1 персональный компьютер.  
*Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:* проектирование и разработка программного обеспечения.

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

#### 1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения

- специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;
- организационные мероприятия при компоновке рабочей зоны.

- Трудовой кодекс Российской Федерации от 30.01.2001 N 197-ФЗ.
- Рабочее место при выполнении работ сидя регулируется ГОСТом 12.2.032–78.
- СНиП 23-05-95\*. Естественное и искусственное освещение (с Изменением N 2).

#### 2. Производственная безопасность при разработке проектного решения:

- анализ выявленных вредных и опасных производственных факторов.

#### **Вредные факторы:**

- Недостаточная освещенность рабочей зоны.
- Отсутствие или недостаток естественного света;
- Отклонение параметров микроклимата.
- Умственное переутомление.

	<ul style="list-style-type: none"> <li>– Повышенный уровень электромагнитных излучений.</li> </ul> <p><b>Опасные факторы:</b></p> <ul style="list-style-type: none"> <li>– Электрический ток.</li> </ul> <p><b>Требуемые средства коллективной и индивидуальной защиты от выявленных факторов:</b></p> <ul style="list-style-type: none"> <li>– Изоляция проводов и кабелей.</li> </ul>
3. Экологическая безопасность при разработке проектного решения:	<p><b>Воздействие на литосферу:</b> неправильная утилизация комплектующих компьютеров, бумаги, иной техники.</p> <p><b>Воздействие на атмосферу:</b> Использование электричества, генерируемого тепловыми электростанциями.</p>
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения:	<p><b>Возможные ЧС:</b> пожары, аварии коммунальных систем, взрывы, наводнения, ураганы, террористические акты, эпидемии, пандемии.</p> <p><b>Наиболее типичная ЧС:</b> пожар.</p>

Дата выдачи задания для раздела по линейному графику	03.02.2022 г.
--	---------------

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ст. преподаватель ООД	Мезенцева Ирина Леонидовна			

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8К82	Де-Гальд Владислав Александрович		

## РЕФЕРАТ

Выпускная квалификационная работа содержит 125 с., 27 рис., 20 табл., 31 источников, 2 прил.

Ключевые слова: CCS, машинное обучение, оптимизация расстановки скважин, обучение с подкреплением, веб-приложение, десктопное приложение, Python, GNU Octave, MATLAB.

Объектом исследования является веб- и десктопное приложения, позволяющее выполнять симуляцию долгосрочного хранения CO<sub>2</sub>, а также оптимизировать размещение скважин.

Целью данной работы является разработка веб- и десктопного приложений, которые позволят большему числу людей получить доступ к программному обеспечению, необходимому для изучения проблемы подземного захоронения CO<sub>2</sub>, а также «среду» для применения методов машинного обучения для решения проблем нефтегазового дела.

В разделе «Теоретические аспекты моделирования подземного хранения CO<sub>2</sub>» изучен модуль MRST-co2lab открытого программного обеспечения MRST, разработанного компанией SINTEF. Произведена декомпозиция гидродинамического симулятора с целью извлечения функционала, необходимого для расчёта симуляций долгосрочного хранения CO<sub>2</sub>.

В разделе «Теоретический анализ применения машинного обучения для оптимизации расстановки скважин» изложен обзор теоретических и практических возможностей решения задачи оптимизации расстановки скважин при закачке нефти и газа (в частности, углекислого газа). Произведена оценка наиболее эффективных методов. Приведено описание методов оптимизации расстановки скважин на основе обучения с подкреплением.

В разделе «Оптимизация расстановки скважин используя обучение с подкреплением» приведены исследования по применению методов глубокого обучения с подкреплением для оптимизации расстановки скважин как для предотвращения утечек газа, так и максимального захоронения углекислого газа в структурных ловушках.

В разделе «Проектирование и разработка десктопного приложения» приведены все важные этапы создания программного обеспечения интерактивного моделирования сценариев закачки и оптимизации расстановки скважин на языке программирования Python и Octave, на котором выполняется непосредственно расчёт симуляции хранения углекислого газа.

В разделе «Проектирование и разработка веб-приложения» приведены все важные этапы проектирования и разработки веб-приложения для интерактивного моделирования сценариев закачки и миграции для долгосрочного хранения CO<sub>2</sub> и оптимизации расстановки скважин.

Раздел «Финансовый менеджмент» описывает коммерческую оценку разрабатываемого программного приложения. Оценка дает возможность определить коммерческую привлекательность программного приложения.

Раздел «Социальная ответственность» представляет собой описание экологической безопасности и производственной безопасности при разработке программного обеспечения.

В заключении приведены выводы по выполненной работе, этапы и трудности в процессе выполнения поставленной задачи, а так же пути преодоления этих трудностей. Описаны полученные опыт и навыки после выполнения работы, достигнутый результат.

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В данной работе применены следующие термины с соответствующими определениями:

**Python** – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью.

**GNU Octave** – свободная программная система для математических вычислений, использующая совместимый с MATLAB язык высокого уровня.

**Matlab** – пакет прикладных программ для решения задач технических вычислений.

**Веб-приложение** — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети.

**Десктопное приложение** — программа, которая устанавливается на компьютер пользователя и работает под управлением операционной системы.

**MRST (MATLAB Reservoir Simulation Toolbox)** – бесплатное программное обеспечение с открытым исходным кодом для моделирования пластов и симуляций, разработанное в основном группой вычислительных наук о земле на факультете математики и кибернетики в SINTEF Digital.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	16
1 ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ МОДЕЛИРОВАНИЯ ПОДЗЕМНОГО ХРАНЕНИЯ CO <sub>2</sub> .....	17
1.1 PhysicalModel .....	18
1.2 ReservoirModel .....	19
1.3 FacilityModel.....	19
1.4 CO2VEBlackOilTypeModel .....	19
1.5 SimpleWell .....	19
1.6 LinearizedProblem.....	19
1.7 LinearSolverAD.....	20
1.8 NonLinearSolver .....	20
1.9 SimulateScheduleAD.....	21
1.10 MakeReports.....	21
1.11 Free_sg.....	21
2 АНАЛИЗ МИРОВОГО ОПЫТА ПРИМЕНЕНИЯ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОПТИМИЗАЦИИ РАССТАНОВКИ СКВАЖИН.....	22
2.1 Анализ проектов с использованием обучения с подкреплением.....	22
2.2 Методология .....	23
3 ОПТИМИЗАЦИЯ РАССТАНОВКИ СКВАЖИН ИСПОЛЬЗУЯ ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ.....	24
3.1 Политика.....	25
3.2 Базовая политика .....	25
3.3 Политика нейронных сетей .....	26
3.4 Fully connected neural network .....	27
3.5 Обучение политики градиентов .....	27
3.6 Loss function .....	28
3.7 Nadam оптимизатор.....	29
3.8 Политика градиентов (REINFORCE).....	29
3.9 REINFORCE алгоритм (типичная вариация):.....	29

4 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ДЕСКТОПНОГО ПРИЛОЖЕНИЯ .....	31
4.1 Спецификация приложения.....	31
4.2 Диаграммы UML .....	33
4.2.1 Диаграмма вариантов использования.....	33
4.2.2 Диаграмма классов .....	39
4.2.3 Диаграмма последовательности.....	40
4.3 Пример использования.....	40
4.4 Руководство пользователя .....	42
4.4.1 Обзор системы .....	42
4.4.2 Уровни доступа пользователей .....	42
4.4.3 Установка симулятора .....	43
4.4.4 Использование приложения-симулятора .....	44
5 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ.....	49
5.1 Клиентская часть .....	49
5.2 Серверная часть .....	49
5.3 Логическая модель .....	50
5.4 Физическая модель.....	50
5.5 Диаграмма DFD .....	51
5.6 Диаграмма BPMN.....	52
5.7 Диаграмма EPC .....	53
5.8 Использование веб-приложения .....	53
5.9 Развертывание веб-приложения на «Яндекс.Облако».....	58
6 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ .....	61
6.1 Потенциальные потребители результатов исследования .....	61
6.2 Анализ конкурентных технических решений.....	61
6.3 SWOT-анализ .....	63
6.4 Планирование работ по научно-техническому исследованию .....	65
6.4.1 Структура работ в рамках научного исследования.....	65
6.4.2 Определение трудоёмкости выполнения работ.....	67

6.4.3	Разработка графика проведения научного исследования.....	68
6.5	Бюджет научно-технического исследования (НТИ) .....	73
6.5.1	Расчет материальных затрат НТИ.....	74
6.5.2	Расчет затрат на специальное оборудование для научных работ	74
6.5.3	Основная заработная плата исполнителя темы .....	75
6.5.4	Расчет дополнительной заработной платы .....	79
6.5.5	Отчисления во внебюджетные фонды.....	79
6.5.6	Накладные расходы.....	80
6.5.7	Формирование бюджета затрат научно-исследовательского проекта.....	81
6.6	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	81
7	СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ .....	84
	Введение .....	84
7.1	Правовые и организационные вопросы обеспечения безопасности .	85
7.1.1	Специальные правовые нормы трудового законодательства .....	85
7.1.2	Основные эргономические требования к правильному расположению и компоновке рабочей зоны .....	85
7.2	Производственная безопасность .....	87
7.2.1	Недостаточная освещенность рабочей зоны.....	88
7.2.2	Отсутствие или недостаток естественного света .....	88
7.2.3	Отклонение параметров микроклимата .....	89
7.2.4	Умственное переутомление.....	90
7.2.5	Повышенный уровень электромагнитных излучений .....	90
7.2.6	Электрический ток .....	91
7.3	Экологическая безопасность .....	92
7.4	Безопасность в чрезвычайных ситуациях .....	93
	Вывод по разделу.....	95
	ЗАКЛЮЧЕНИЕ.....	96
	СПИСОК ЛИТЕРАТУРЫ .....	97
	ПРИЛОЖЕНИЕ А. КОД DDL .....	102

ПРИЛОЖЕНИЕ Б. ИСХОДНЫЙ КОД ОСНОВНОГО МОДУЛЯ ВЕБ-ПРИЛОЖЕНИЯ .....	111
--	-----

## ВВЕДЕНИЕ

В настоящее время изменение климата наблюдается во всех регионах планеты, причем этот процесс становится все более интенсивным и связан, во многом, с ростом выбросов  $\text{CO}_2$  в атмосферу. С каждым годом тают ледники, уровень мирового океана повышается, в местах вечной мерзлоты животные сталкиваются с проблемой выживания. Одним из способов борьбы с глобальным потеплением является подземное захоронение  $\text{CO}_2$ . Многие ученые работают над решением этой проблемы, но уровень осведомленности и доступности к программному обеспечению, необходимому для моделирования сценариев хранения  $\text{CO}_2$  не высок. Если общество хочет предотвратить трагические последствия выбросов  $\text{CO}_2$  в атмосферу, необходим ресурс, который позволит большему числу людей быть вовлеченными в решение проблем, связанных с подземным захоронением углекислого газа.

Целью данной работы является разработка веб- и десктопного приложений, которые позволят большему числу людей получить доступ к программному обеспечению, необходимому для изучения проблемы подземного захоронения  $\text{CO}_2$ , а также «среду» для применения методов машинного обучения для решения проблем нефтегазового дела.

*Симулятор для долгосрочного хранения  $\text{CO}_2$  с использованием методов машинного обучения* – это приложение, которое охватывает весь цикл создания и настройки сценариев закачки, миграции, запуска моделирования и визуализации результатов, а также оптимизации расстановки скважин. Приложение имеет библиотеку с готовыми геологическими моделями, подробная документация снижает порог вхождения для начала работы на платформе. Приложение является абсолютно бесплатным, возможность добровольных пожертвований, специальные грантовые программы и конкурсы помогут проекту развиваться в нужной степени.

## 1 ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ МОДЕЛИРОВАНИЯ ПОДЗЕМНОГО ХРАНЕНИЯ CO<sub>2</sub>

Существуют различные программные продукты, обладающие функционалом для моделирования долгосрочного хранения CO<sub>2</sub>: tNavigator, UTCHEM, Petrel, Eclipse, CMG-GEM, tNavigator, UTCHEM. Из их числа можно выделить Eclipse, а именно ECLIPSE Compositional simulator – симулятор, разработанный компанией Schlumberger для изучения долгосрочного хранения CO<sub>2</sub>. Данный симулятор включает в себя богатый функционал динамического моделирования для хранения CO<sub>2</sub> [12]:

1. Eclipse CO2STORE – предназначен для моделирования геологического хранения CO<sub>2</sub> в водоносных горизонтах.

2. GASWAT – предназначен для закачки CO<sub>2</sub> в истощённое месторождение газа с целью моделирования методов увеличения нефтеотдачи или хранения CO<sub>2</sub>.

3. CO2SOL – предназначен для закачки CO<sub>2</sub> в истощённое месторождение нефти с целью моделирования методов увеличения нефтеотдачи или хранения CO<sub>2</sub>.

Перечисленные выше гидродинамические симуляторы доступны только специалистам-нефтяникам или учёным, работающим в крупных компаниях или научно-исследовательских центрах, поэтому научный прогресс проблемы подземного хранения CO<sub>2</sub> идет медленнее, чем необходимо.

Также следует выделить MRST-co2lab, который реализован как дополнительный модуль к MRST. Он состоит из семейства вычислительных инструментов, которые были разработаны специально для изучения долгосрочного хранения CO<sub>2</sub> в крупномасштабных системах водоносных горизонтов. Несмотря на открытую лицензию данного программного обеспечения, для работы с MRST-co2lab необходимо иметь лицензию

MATLAB, что в настоящее время является проблемой для компаний и граждан Российской Федерации, а также знание языка программирования MATLAB является дополнительным препятствием для специалистов-нефтяников.

В данной главе приведено описание объектно-ориентированного фреймворка решателей, разработанного компанией SINTEF в рамках проекта MRST (рис. 1), основанного на автоматическом дифференцировании. Данный модуль был декомпозирован с целью извлечения «ядра», необходимого для вычисления симуляций долгосрочного хранения CO<sub>2</sub>.

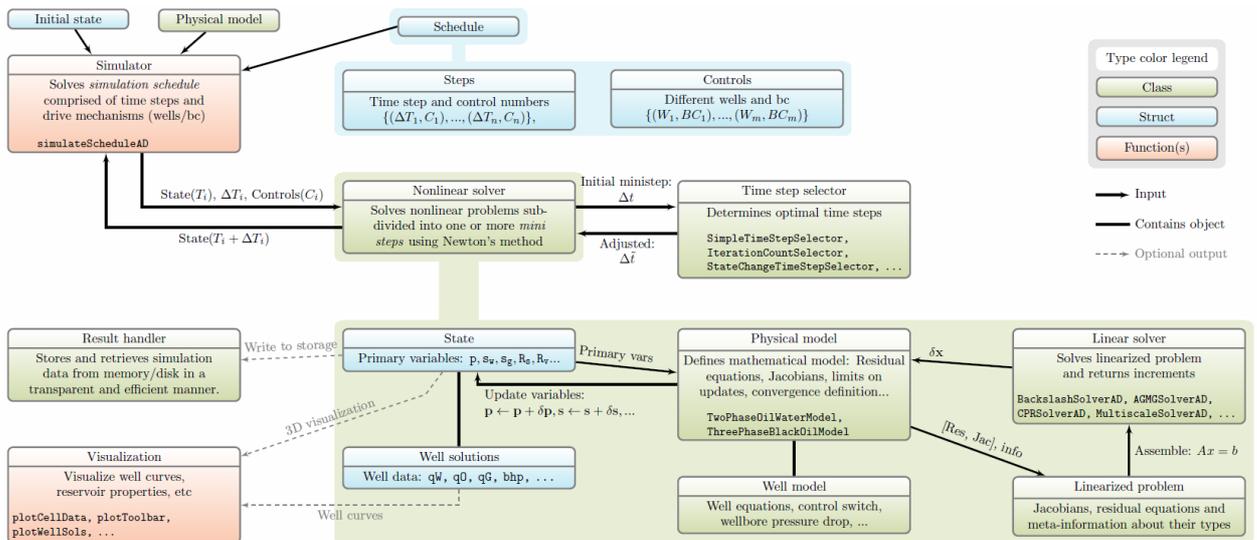


Рисунок 1 – Общая структура решателей [3]

## 1.1 PhysicalModel

Базовый класс для реализации физических моделей для использования с автоматической дифференциацией. Данный класс является абстрактным. Физическая модель состоит из набора дискретных операторов, которые могут быть использованы для определения уравнений модели, и нелинейного параметра, который определяет, насколько значения должны быть близки к нулю, прежде чем уравнения могут считаться решёнными. В большинстве случаев операторы определяются по сетке модели [1].

## **1.2 ReservoirModel**

Наследник класса «PhysicalModel» для задания свойств фаз и горной породы, а также иных необходимых параметров [1].

## **1.3 FacilityModel**

«FacilityModel» – это слой между «ReservoirModel» и скважинами. Скважины реализованы в собственных наследованных классах. Скважины имеют разные управляющие уравнения, исходные данные и критерии сходимости [1].

## **1.4 CO2VEBlackOilTypeModel**

«CO2VEBlackOilTypeModel» представляет собой модель типа нелетучей нефти для вертикально интегрированного потока газа или воды [1].

Класс, представляющий собой модель с вертикально интегрированным двухфазным потоком (CO<sub>2</sub> и солёная вода), основанным на s-составе, где увеличенное насыщение является первичной переменной, а растворимость газа в водной фазе является дополнительной опцией [1].

## **1.5 SimpleWell**

Базовый класс, реализующий единую, мгновенную равновесную модель скважины. Для этой модели скважины допущено, что поток скважины является быстрым по сравнению с временными шагами симулятора, что делает мгновенное равновесие и смешивание в скважине разумным предположением [1].

## **1.6 LinearizedProblem**

Класс, реализующий хранение экземпляра линеаризованной задачи, дискретизированной с помощью автоматической дифференциации. Этот класс содержит остаточные уравнения, вычисляемые для одного состояния, а также

метаинформацию об уравнениях и первичных переменных, по отношению к которым они дифференцированы [1].

Линеаризированная задача может быть преобразована в линейную систему и решена с помощью подклассов, производных от «LinearSolverAD», учитывая, что число уравнений соответствует числу переменных [1].

В частности, класс содержит методы для [1]:

1. сборки линейной системы из матриц-якобиан, хранящихся для каждого уравнения;
2. добавления дополнительных уравнений;
3. использования метода Гаусса для исключения отдельных переменных или всех переменных, не имеющих заданного типа;
4. восстановления приращений, соответствующих переменным, которые ранее были исключены;
5. вычисления нормы каждого остаточного уравнения.

### **1.7 LinearSolverAD**

Базовый класс для линейных решателей. Реализация методов решения линеаризованных задач и смежных задач. Также поддерживает функции установки/очистки до/после использования для инициализации однотипного использования [1].

### **1.8 NonLinearSolver**

Класс «NonLinearSolver» является общим нелинейным решателем, основанным на методе Ньютона. Он способен к выбору и уменьшению временных шагов на основе скоростей сходимости и может быть расширен с помощью подклассов или модульных линейных решателей и классов временного шага [1].

Сходимость осуществляется классом «PhysicalModel». «NonLinearSolver» отвечает за то, что модель сообщает с точки зрения сходимости, чтобы обеспечить некоторый уровень инкапсуляции [1].

### **1.9 SimulateScheduleAD**

Запуск симуляции для нелинейной физической модели с использованием автоматической дифференциации [1].

«SimulateScheduleAD» — это внешняя функция учёта, используемая для выполнения симуляций с нетривиальными изменениями управления. Она опирается на модель и (не)линейные классы решателей для выполнения сложных вычислений [1].

### **1.10 MakeReports**

Эта функция выполняет промежуточную обработку данных моделирования для создания графиков инвентаризации захоронения углекислого газа.

### **1.11 Free\_sg**

Предполагая чёткую границу раздела, эта функция определяет величину настоящего насыщения, которое находится в области «газовой шапки» (в отличие от области ниже подвижной фазы, где  $\text{CO}_2$  удерживается капиллярными силами). Таким образом, «подвижная часть настоящего насыщения» включает  $\text{CO}_2$  в подвижной фазе, которая впоследствии будет удерживаться капиллярными силами, но не  $\text{CO}_2$ , который уже является капиллярно-защемлённым газом [1].

## **2 АНАЛИЗ МИРОВОГО ОПЫТА ПРИМЕНЕНИЯ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОПТИМИЗАЦИИ РАССТАНОВКИ СКВАЖИН**

### **2.1 Анализ проектов с использованием обучения с подкреплением**

Оптимизация размещения скважин является одним из основных комплексных факторов разработки коллекторов в нефтегазовой отрасли. Целевая функция оптимизации размещения скважин рассматривается как многомерная, прерывистая и многомодельная. За последнее десятилетие для решения этой проблемы были реализованы как градиентные, так и неградиентные методы оптимизации. В этой области использовались эвристические алгоритмы оптимизации без градиентов, такие как оптимизация роя частиц, генетический алгоритм, стратегия эволюции адаптации ковариационной матрицы и дифференциальная эволюция [8].

Последовательные искусственные нейронные сети являются одним из методов получения рационального местоположения скважины. В методе последовательной искусственной нейронной сети (ANN) пространство поиска уменьшается по мере последовательного построения ANN. ANN переобучаются на каждой итерации с использованием обучающего набора данных, который постепенно добавляется с заданным количеством точек данных в ограниченном пространстве поиска, что повышает производительность оценки сети [9].

Проблема оптимизации размещения скважин может быть решена с помощью глубокого обучения с подкреплением. Глубокое обучение с подкреплением является перспективным подходом для решения комбинаторных задач и позволяет адаптировать предметную область и напрямую оптимизировать недифференцированные целевые функции.

Изучение политики обучения с подкреплением является очень сложной задачей, отчасти из-за краткости обновлений градиента [7].

## **2.2 Методология**

Рассмотренная литература по смежным темам показывает, что методы обучения с подкреплением являются одними из наиболее подходящих для решения проблемы эффективного размещения скважин. В данном проекте рассматривается проблема оптимизации расстановки скважин с целью максимального захоронения CO<sub>2</sub> в структурных ловушках и минимизации утечки газа. Этот подход более подробно рассматривается в следующей главе.

### 3 ОПТИМИЗАЦИЯ РАССТАНОВКИ СКВАЖИН ИСПОЛЬЗУЯ ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ

В данной главе приведены результаты исследований автора данной работы по применению методов обучения с подкреплением к проблеме эффективной расстановки скважин с целью максимального захоронения углекислого газа в структурных ловушках и минимизации утечек CO<sub>2</sub>.

Обучение с подкреплением (RL) — это область машинного обучения, связанная с тем, как интеллектуальные агенты должны предпринимать действия в среде, чтобы максимизировать понятие кумулятивного вознаграждения. Обучение с подкреплением является одной из трех основных парадигм машинного обучения, наряду с обучением с учителем и без учителя [11].

Обучение с подкреплением отличается от обучения с учителем тем, что не нуждается в представлении размеченных пар ввода-вывода и в том, что не нуждается в явном исправлении неоптимальных действий. Вместо этого основное внимание уделяется поиску баланса между разведкой неизведанной территории и эксплуатацией текущих знаний.

В обучении с подкреплением программный агент делает наблюдения и предпринимает действия в окружающей среде, а взамен получает вознаграждение. Его цель состоит в том, чтобы научиться действовать таким образом, чтобы со временем максимизировать ожидаемые награды. То есть, агент действует в окружающей среде и учится методом проб и ошибок максимизировать свое ‘удовольствие’ и минимизировать ‘боль’ [6].

Примеры применения обучения с подкреплением:

1. Робототехника.
2. Персонализированные рекомендации.
3. Игры.

4. Торги и реклама.
5. Конфигурация веб-систем.

### 3.1 Политика

Алгоритм, используемый программным агентом для определения своих действий, называется его политикой. Для этого проекта в качестве политик используются нейронная сеть, принимающая наблюдения в качестве входных данных и возвращая действие, которое нужно предпринять, и базовая политика.

### 3.2 Базовая политика

Внедрение метода обучения с подкреплением, который использует «базовую» политику для поиска оптимального расположения скважин.

«Базовая» политика включает в себя следующие шаги:

1.  $N$  локаций расположения скважин выбираются случайным образом.
2. Для каждого случайного местоположения выполняются  $k$  шагов с целью нахождения оптимального расположения скважин в данной области модели.
3. Каждый шаг требует 5 симуляций, работающих одновременно: текущее местоположение скважины, левая, правая, верхняя и нижняя ячейки относительно выбранной скважины.
4. Местоположение следующей скважины выбирается исходя из максимального значения разницы между наградами смещенных скважин и вознаграждением текущей скважины.

Вознаграждение рассчитывается как разница между массой  $\text{CO}_2$ , удерживаемого структурными ловушками, и количеством  $\text{CO}_2$ , вышедшим за пределы геологической модели.

### 3.3 Политика нейронных сетей

Нейронная сеть принимает наблюдение в качестве входных данных и возвращает действие, которое будет выполнено. Она оценивает вероятность каждого действия, а затем действие будет выбрано случайным образом, согласно оценочным вероятностям. В этом проекте есть четыре возможных действия (влево, вправо, вверх и вниз), поэтому последний слой имеет четыре нейрона (рис. 2).

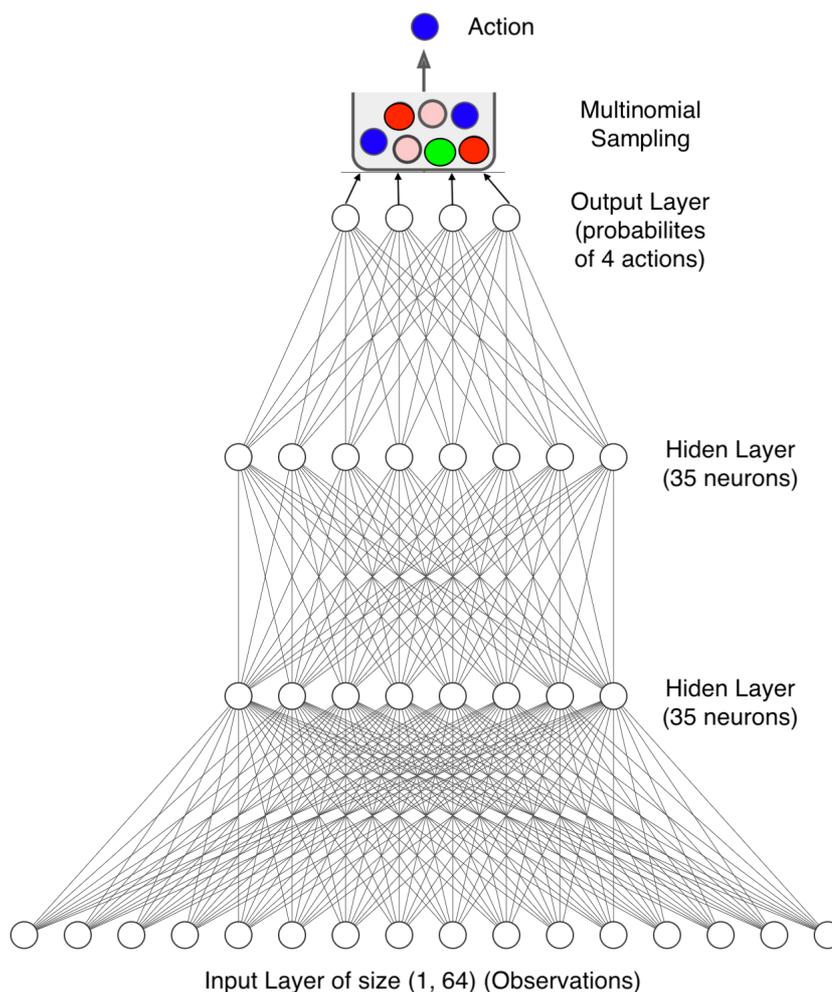


Рисунок 2 – Политика нейронной сети

### 3.4 Fully connected neural network

Полностью связанные нейронные сети (FCNN) представляют собой тип искусственной нейронной сети, где архитектура такова, что все узлы или нейроны в одном слое связаны с нейронами в следующем слое.

Нейронная сеть представляет собой набор нейронов, организованных в слои. Каждый нейрон представляет собой математическую операцию, которая принимает его входные данные, умножает их на свои веса (1), а затем передает сумму через функцию активации (2) другим нейронам (рис. 3).

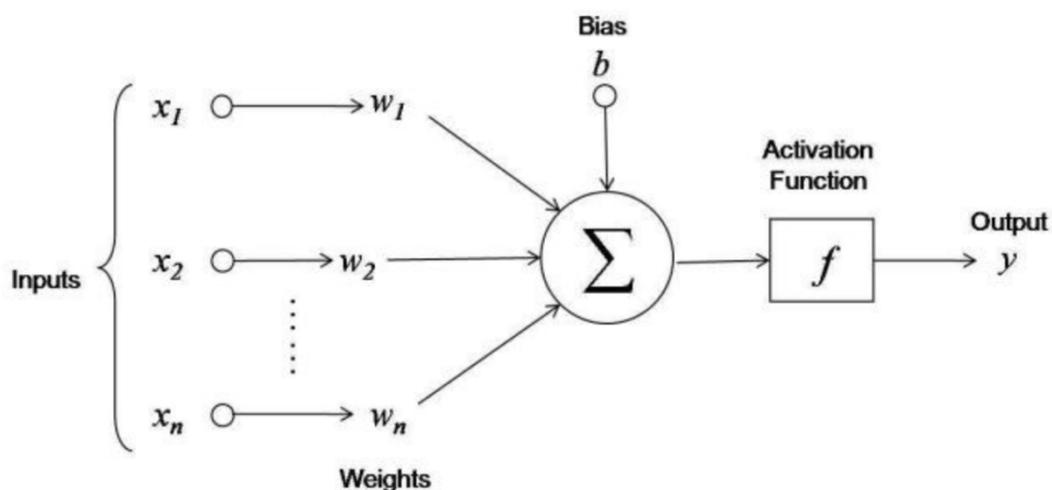


Рисунок 3 – Операции нейрона [2]

$$z = w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_ix_i + b = x^T w + b \quad (1)$$

$$y = f(z) \quad (2)$$

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \quad (3)$$

где  $f(z)$  – тангенциальная функция активации.

### 3.5 Обучение политики градиентов

Согласно [6], политика градиентов тренируется в течение многих эпох. В каждой эпохе есть две фазы. На первом этапе она собирает обучающие

данные из окружающей среды для одного эпизода. На втором этапе она использует сохраненные данные для обучения нейронной сети.

Действие, которое предсказывает политика нейронной сети, считается верным, таким образом могут быть рассчитаны функция потерь и ее градиенты. Эти градиенты не используются для тренировки на данном этапе и будут модифицироваться позже в зависимости от того, насколько «хорошим» оказалось действие.

После выполнения нескольких эпизодов и шагов формируется вложенный список наград (один список наград на эпизод, содержащий одну награду за шаг), а также вложенный список градиентов (один список градиентов на эпизод, каждый из которых содержит один кортеж градиентов на шаг и каждый кортеж, содержащий один тензор градиента на обучаемую переменную).

Выполнив вышеупомянутые шаги несколько раз, алгоритм вернется назад и посмотрит на все вознаграждения, дисконтирует их и нормализует.

Затем вычисляется средневзвешенное значение градиентов для каждой обучаемой переменной по всем эпизодам и всем шагам, взвешенное по финальной награде.

Наконец, эти средние градиенты применяются с помощью оптимизатора: обучаемые переменные модели будут подстроены для улучшения качества модели.

### 3.6 Loss function

Функция потерь в данном проекте рассчитывается с помощью разреженной категориальной кросс-энтропийной функции:

$$Loss = - \sum_{i=1}^{output\ size} y_i \cdot \log \hat{y}_i, \quad (4)$$

где  $y_i, \hat{y}_i$  являются целевым значением и выходными данными модели соответственно.

### 3.7 Nadam оптимизатор

Оптимизаторы — это алгоритмы или методы, используемые для изменения атрибутов нейронной сети, таких как веса и скорость обучения, для снижения потерь.

Nadam — аббревиатура от оптимизатора Нестерова и Адама:

$$\omega_{t+1} = \omega_t - \frac{\alpha}{\sqrt{\hat{u}_t + \epsilon}} \left( \beta_1 \hat{m}_t + \frac{1 - \beta_1}{1 - \beta_1^t} \cdot \frac{\partial L}{\partial \omega_t} \right) \quad (5)$$

### 3.8 Политика градиентов (REINFORCE)

Политика градиентов:

$$\nabla J = E \nabla \log \pi_{\theta}(a|s) \cdot Q(s, a) \quad (6) [10]$$

Приблизительная политика градиентов с выборкой:

$$J \approx \frac{1}{N} \sum_{i=0}^N \sum_{s, a \in z_i} \nabla \log \pi_{\theta}(a|s) \cdot Q(s, a) \quad (7) [10]$$

### 3.9 REINFORCE алгоритм (типичная вариация):

Алгоритм REINFORCE [6]:

1. Политика нейронной сети выполняет симуляцию несколько раз, и на каждом шаге вычисляет градиенты, которые сделают выбранное действие еще более вероятным, но пока не применяет эти градиенты.

2. После получения нескольких эпизодов вычисляется преимущество каждого действия.

3. Если преимущество действия положительное, это означает, что действие, вероятно, было хорошим, и необходимо применить градиенты, рассчитанные ранее, чтобы сделать действие еще более вероятным для выбора в будущем. Однако, если преимущество действия отрицательное, это означает, что действие, вероятно, было плохим, и необходимо применить противоположные градиенты, чтобы сделать это действие немного менее вероятным в будущем. Решение состоит в том, чтобы умножить каждый вектор градиента на преимущество соответствующего действия.

4. Наконец, вычисляется среднее из всех результирующих векторов градиента, и затем оно используется для выполнения шага градиентного спуска.

## 4 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ДЕСКТОПНОГО ПРИЛОЖЕНИЯ

### 4.1 Спецификация приложения

*Предметная область:*

Симулятор долгосрочного хранения CO<sub>2</sub> с использованием методов машинного обучения является интерактивным инструментом для настройки и моделирования долгосрочных сценариев хранения CO<sub>2</sub> в масштабах всего водоносного горизонта. Приложение имеет библиотеку с готовыми геологическими моделями для хранения CO<sub>2</sub>.

*Функциональные требования:*

1. Симулятор должен позволять пользователям указывать местоположение скважины.
2. Симулятор должен позволять пользователям изменять уровень закачки от 1 до 10 мегатонн в год.
3. Симулятор должен позволять пользователям изменять время закачки и миграции от 1 до 1000 лет.
4. Симулятор должен позволять пользователям изменять временные шаги закачки и миграции от 1 до 100.
5. Симулятор должен позволять пользователям выбирать геологические модели из предоставленного списка.
6. Симулятор должен выводить график инвентаризации захоронения углекислого газа с 6-ю категориями: 'Structural residual', 'Residual', 'Residual in plume', 'Structural plume', 'Free plume', 'Exited'.
7. Симулятор должен иметь возможность интеллектуального определения оптимальных местоположений скважин для минимизации утечки

газа и максимального хранения CO<sub>2</sub> в структурных ловушках с использованием алгоритмов обучения с подкреплением.

8. Симулятор должен иметь возможность интуитивного определения оптимальных местоположений скважин для минимизации утечки газа и максимального хранения CO<sub>2</sub> в структурных ловушках с использованием базового алгоритма.

9. Симулятор должен иметь возможность принудительно останавливать поиск оптимальных местоположений скважин.

*Нефункциональные требования:*

1. Каждая симуляция должна быть выполнена в течение 1-й минуты.

2. Местоположение скважин должно быть установлено в виде координат x, y.

3. Симулятор должен иметь несколько примеров геологических моделей для моделирования хранения CO<sub>2</sub>.

4. Симулятор должен иметь интуитивно понятный графический пользовательский интерфейс для настройки сценариев симуляции или выбора оптимизации местоположений скважин.

*Требования к интерфейсу:*

1. Сетка геологической модели должна быть представлена интерактивным 2D графиком.

2. Графический интерфейс пользователя должен позволять пользователям выбирать местоположение скважины, нажимая на геологическую модель.

3. Графический интерфейс должен позволять пользователям указывать следующие параметры: уровень закачки, период закачки, временные шаги закачки, период миграции, временные шаги миграции.

4. Графический интерфейс должен позволять пользователям запускать моделирование с заданными начальными параметрами.

5. Графический интерфейс должен позволять пользователям запускать интеллектуальный поиск местоположения скважин с помощью алгоритма обучения с подкреплением, используя политику нейронной сети или «базовую» политику.

## 4.2 Диаграммы UML

### 4.2.1 Диаграмма вариантов использования

Диаграмма вариантов использования симулятора для долгосрочного хранения CO<sub>2</sub> представлена на рисунке 4.

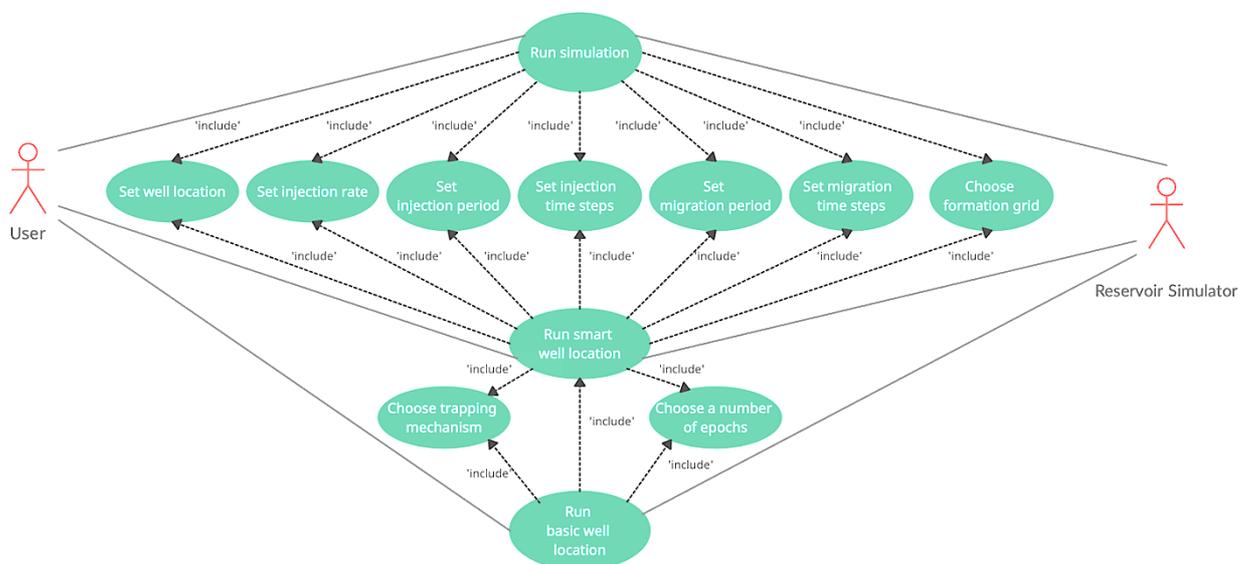


Рисунок 4 – Диаграмма вариантов использования

Варианты использования:

1. Run simulation (Запустить симуляцию):

1. Короткое описание:

1. «Run simulation» позволяет пользователю начать симуляцию в соответствии с заданным сценарием.

2. Актёры:

1. Пользователь.
2. Симулятор.
3. Предварительные условия:
  1. «Run simulation» можно запустить только после выполнения следующих условий:
    1. Необходимо выбрать геологическую модель.
    2. Необходимо выбрать местоположение скважины.
    3. Необходимо установить уровень закачки.
    4. Необходимо установить период закачки.
    5. Необходимо установить временные шаги закачки.
    6. Необходимо установить период миграции.
    7. Необходимо задать временные шаги миграции.
4. Поток главного события:
  1. Сценарий использования начинается, когда клиент нажимает кнопку «Run simulation».
  2. Симулятор запускает моделирование с заданными параметрами.
  3. Симулятор возвращает клиенту график инвентаризации захоронения CO<sub>2</sub>.
  4. Конец варианта использования.
5. Альтернативный поток событий 1. Некоторые из значений предварительных условий неверны:
  - 4a1. Сообщение об ошибке предлагает пользователю проверить введенные значения и изменить их на корректные.
  - 4a2. Клиент нажимает кнопку «Run simulation» еще раз.

4а3. Симулятор начинает моделирование с заданными параметрами.

4а4. Симулятор возвращает клиенту график инвентаризации захоронения CO<sub>2</sub>.

4а5. Конец варианта использования.

2. Run smart well location (Запустить интеллектуальное расположение скважин):

1. Короткое описание:

1. «Run smart well location» позволяет пользователю найти оптимальные местоположения скважин для минимизации утечки газа и максимизации захоронения CO<sub>2</sub> в структурных ловушках с помощью алгоритма обучения с подкреплением с политикой нейронной сети.

2. Актёры:

1. Пользователь.

2. Симулятор.

3. Предварительные условия:

1. «Run smart well location» можно запустить только после выполнения следующих условий.

1. Необходимо выбрать геологическую модель.

2. Необходимо выбрать местоположение скважины.

3. Необходимо установить уровень закачки.

4. Необходимо установить период закачки.

5. Необходимо установить временные шаги закачки.

6. Необходимо установить период миграции.

7. Необходимо задать временные шаги миграции.

4. Поток главного события:

1. Сценарий использования начинается, когда клиент нажимает кнопку «Run smart well location».

2. Симулятор начинает поиск оптимального расположения скважин.

3. Оптимальное расположение скважин динамически отображаются на геологической модели.

4. Скважина с самым высоким вознаграждением в исследуемом регионе аннотируется наградой.

5. Конец варианта использования.

5. Альтернативный вентиляционный поток 1. Некоторые из значений предварительных условий неверны:

4a1. Сообщение об ошибке предлагает пользователю проверить введенные значения и изменить их на корректные.

4a2. Пользователь еще раз нажимает кнопку «Run smart well location».

4a3. Симулятор начинает поиск оптимального расположения скважин.

4a4. Оптимальное расположение скважин динамически отображается на геологической модели.

4a5. Скважина с самым высоким вознаграждением в исследуемом регионе аннотируется значением вознаграждения.

4a6. Конец варианта использования.

3. Run basic well location (Запустить интуитивное расположение скважин):

1. Короткое описание:

1. «Run basic well location» позволяет пользователю найти оптимальное местоположение скважин для минимизации утечки газа и максимизации захоронения CO<sub>2</sub> в структурных ловушках с помощью интуитивного алгоритма поиска местоположения скважин.

2. Актёры:

1. Пользователь.

2. Симулятор.

3. Предварительные условия:

1. «Run basic well location» можно только после выполнения следующих условий.

1. Необходимо выбрать геологическую модель.

2. Необходимо установить уровень закачки.

3. Необходимо установить период закачки.

4. Необходимо установить временные шаги закачки.

5. Необходимо установить период миграции.

6. Необходимо задать временные шаги миграции.

4. Поток главного события:

1. Сценарий использования начинается, когда клиент нажимает кнопку «Run basic well location».

2. Симулятор начинает поиск оптимальных местоположений расположения скважин.

3. Оптимальное расположение скважин динамически отображается на геологической модели.
  4. Скважина с самым высоким «вознаграждением» в исследуемой области аннотируется данным «вознаграждением» на геологической модели.
  5. Конец варианта использования.
5. Альтернативный поток событий 1. Некоторые из значений предпосылок неправильны:
- 4a1. Сообщение об ошибке предлагает пользователю проверить введенные значения и изменить их на соответствующие.
  - 4a2. Клиент еще раз нажимает кнопку «Run basic well location».
  - 4a3. Симулятор начинает поиск оптимальных мест расположения скважин.
  - 4a4. Оптимальное расположение скважин динамически отображается на геологической модели.
  - 4a5. Скважина с самой высокой наградой в исследуемой области аннотируется наградой.
  - 4a6. Конец варианта использования.

## 4.2.2 Диаграмма классов

Диаграмма классов для симулятора долгосрочного хранения CO<sub>2</sub> представлена на рисунке 5.

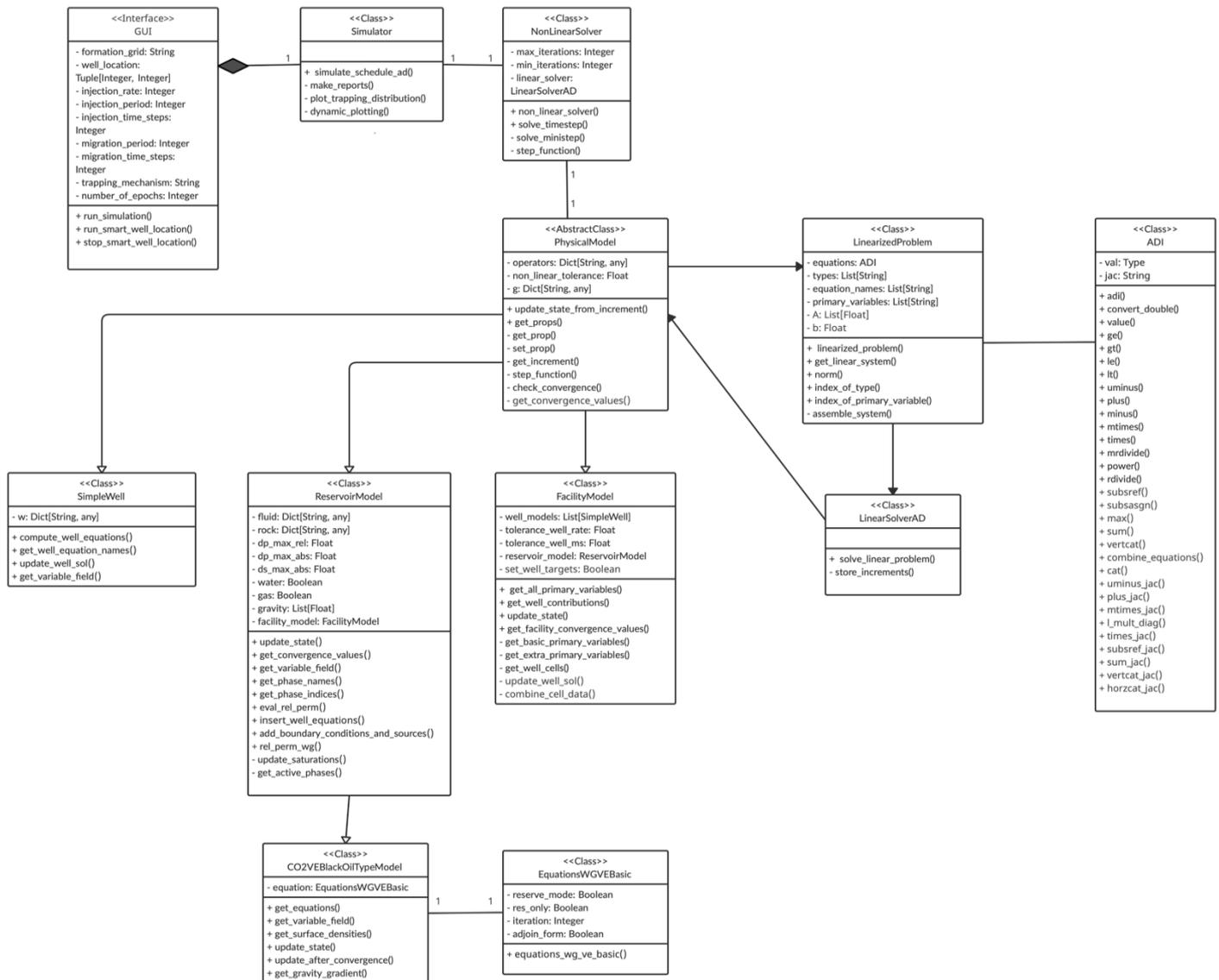


Рисунок 5 – Диаграмма классов

Классы:

1. GUI.
2. Simulator.
3. NonLinearSolver.
4. PhysicalModel.

5. FacilityModel.
6. ReservoirModel.
7. SimpleWell.
8. CO2BlackOilTypeModel.
9. EquationsWGVEBasic.
10. LinearizedProblem.
11. LinearSolverAD.
12. ADI.

### 4.2.3 Диаграмма последовательности

Диаграмма последовательности симулятора для долгосрочного хранения CO<sub>2</sub> представлена на рисунке 6.

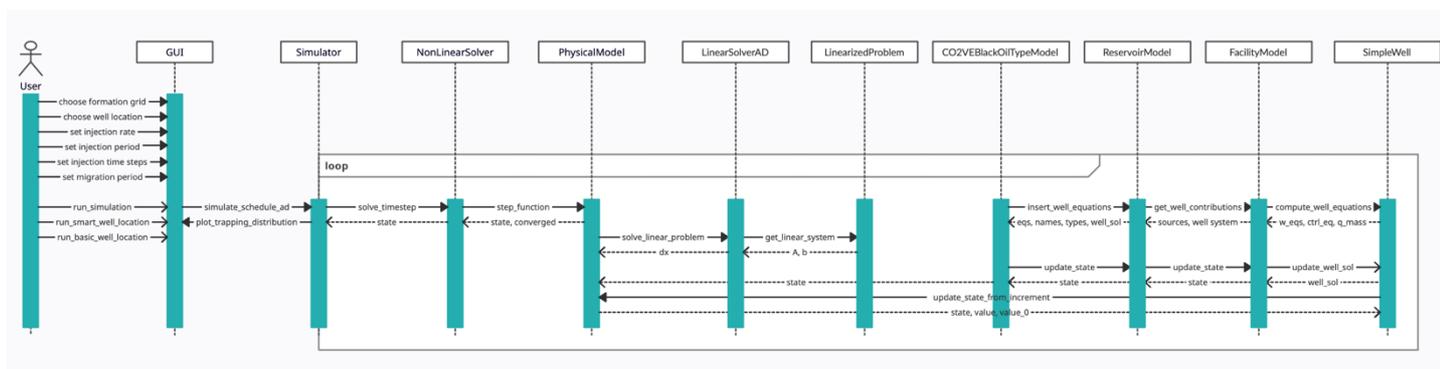


Рисунок 6 – Диаграмма последовательности

### 4.3 Пример использования

1. Пользователь задает начальные параметры моделирования: геологическая модель, местоположение скважины, уровень закачки, период закачки, временные шаги закачки, период миграции, временные шаги миграции.
2. Пользователь выбирает запуск моделирования («Run simulation») или поиск оптимальных мест скважин для заданных исходных параметров с помощью алгоритмов обучения с подкреплением («Run smart well location») или «Run basic well location»).

3. Если выбрано «Run simulation»:
  1. Симулятор запускает сценарий моделирования по заданным параметрам.
  2. После завершения моделирования симулятор возвращает график инвентаризации захоронения CO<sub>2</sub> с 6-ю категориями: ‘Structural Residual’, ‘Residual’, ‘Residual in Plume’, ‘Structural Plume’, ‘Free Plume’, ‘Exited’ (рис. 7).

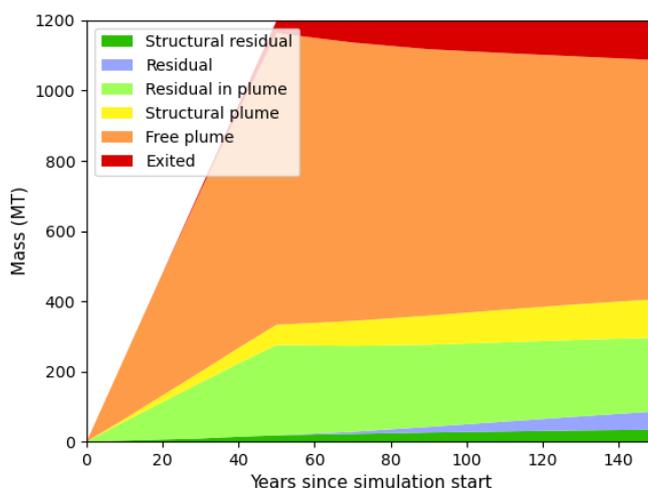


Рисунок 7 – График инвентаризации захоронения CO<sub>2</sub>

4. Если выбрано «Run smart well location» или «Run basic well location»:
  1. Симулятор выполняет множество симуляций и находит оптимальные местоположения скважин на основе алгоритма обучения с подкреплением с политикой нейронной сети или интуитивной политикой.
  2. После каждой эпохи симулятор возвращает график с оптимальным расположением скважин в регионе, изученном в эту эпоху (рис. 8). Каждый исследуемый регион имеет одно числовое значение, которое соответствует местоположению скважины с

максимальным вознаграждением (разница между массой  $\text{CO}_2$ , хранимой в структурной ловушке и массой  $\text{CO}_2$ , вышедшей за границы моделируемой области).

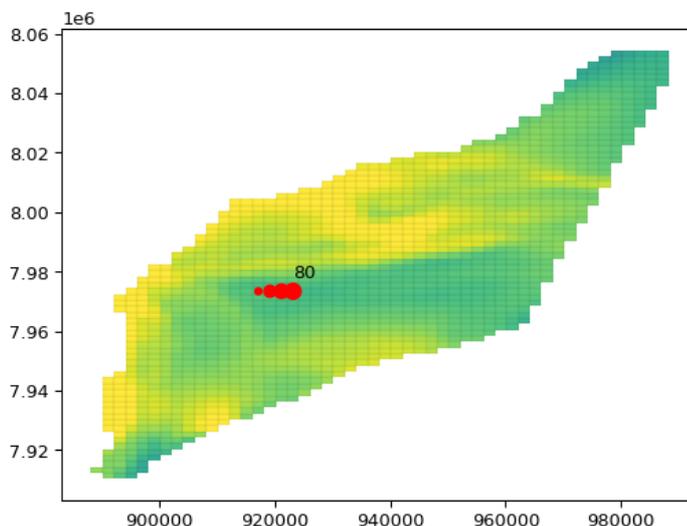


Рисунок 8 – Оптимальные местоположения скважин

3. Пользователь может остановить поиск оптимального местоположения скважин.

#### 4.4 Руководство пользователя

##### 4.4.1 Обзор системы

Симулятор долгосрочного хранения  $\text{CO}_2$  с использованием методов машинного обучения – это приложение с интуитивно понятным пользовательским интерфейсом, который позволяет пользователю вводить необходимую информацию для запуска моделирования сценариев хранения  $\text{CO}_2$  и оптимизации расстановки скважин.

##### 4.4.2 Уровни доступа пользователей

Существует несколько типов пользователей. С одной стороны, симулятор хранения  $\text{CO}_2$  может использоваться в качестве программного обеспечения типа «graybox», которое обеспечивает надежное моделирование

различных сценариев. С другой стороны, симулятор доступен с полным доступом к исходному коду с открытой лицензией GPL-3.0 License.

#### 4.4.3 Установка симулятора

##### *MacOS*

1. Открыть Terminal.

2. Загрузить и установить Miniconda для Python 3:

```
curl https://repo.anaconda.com/miniconda/Miniconda3-py38_4.10.3-MacOSX-x86_64.sh' > Miniconda.sh
```

3. Установить Git.

4. Загрузить симулятор хранения CO<sub>2</sub>:

```
git clone https://github.com//De-Gald/CO2_simulator.git
```

5. Перейти в папку с симулятором:

```
cd CO2_simulator
```

6. Создайте среду с именем *CO2\_simulator*, установите Python 3.8:

```
conda create --name CO2_simulator python=3.8
```

7. Установить все необходимые зависимости:

```
pip install -r requirements.txt
```

8. Установить GNU Octave:

```
brew install octave
```

9. Запустите тесты, чтобы убедиться, что все работает правильно:

```
python -m unittest
```

10. Открыть GUI симулятора:

```
python gui.py
```

11. Установить начальные параметры.

12. Запустить симулятор в одном из 3 режимов «Simulation», «Smart well location» и «Basic well location».

## Windows

1. Открыть командную строку Windows.

2. Загрузить и установить Miniconda для Python 3:

*[https://repo.anaconda.com/miniconda/Miniconda3-py38\\_4.10.3-Windows-x86\\_64.exe](https://repo.anaconda.com/miniconda/Miniconda3-py38_4.10.3-Windows-x86_64.exe)*

3. Установить Git.

4. Скачать симулятор хранения CO<sub>2</sub>:

*`git clone https://github.com/De-Gald/CO2_simulator.git`*

5. Перейти в папку с симулятором:

*`cd CO2_simulator`*

6. Создайте среду с именем *CO2\_simulator*, установить Python 3.8:

*`conda create --name CO2_simulator python=3.8`*

7. Установите все необходимые зависимости:

*`pip install -r requirements.txt`*

8. Скачать GNU Octave.

*<https://ftpmirror.gnu.org/octave/windows/octave-6.4.0-w64-installer.exe>*

9. Установить GNU Octave.

10. Запустить тесты, чтобы убедиться, что все работает правильно:

*`python -m unittest`*

13. Открыть GUI симулятора:

*`python gui.py`*

14. Установить начальные параметры.

15. Запустить симулятор в одном из 3 режимов «Simulation», «Smart well location» и «Basic well location».

### 4.4.4 Использование приложения-симулятора

При открытии графического пользовательского интерфейса загружается форма настроек (рис. 9), позволяющая пользователям вводить начальные параметры симуляции. Настройки должны вводиться при каждом

запуске графического интерфейса, но моделирование можно запускать столько раз, сколько необходимо.

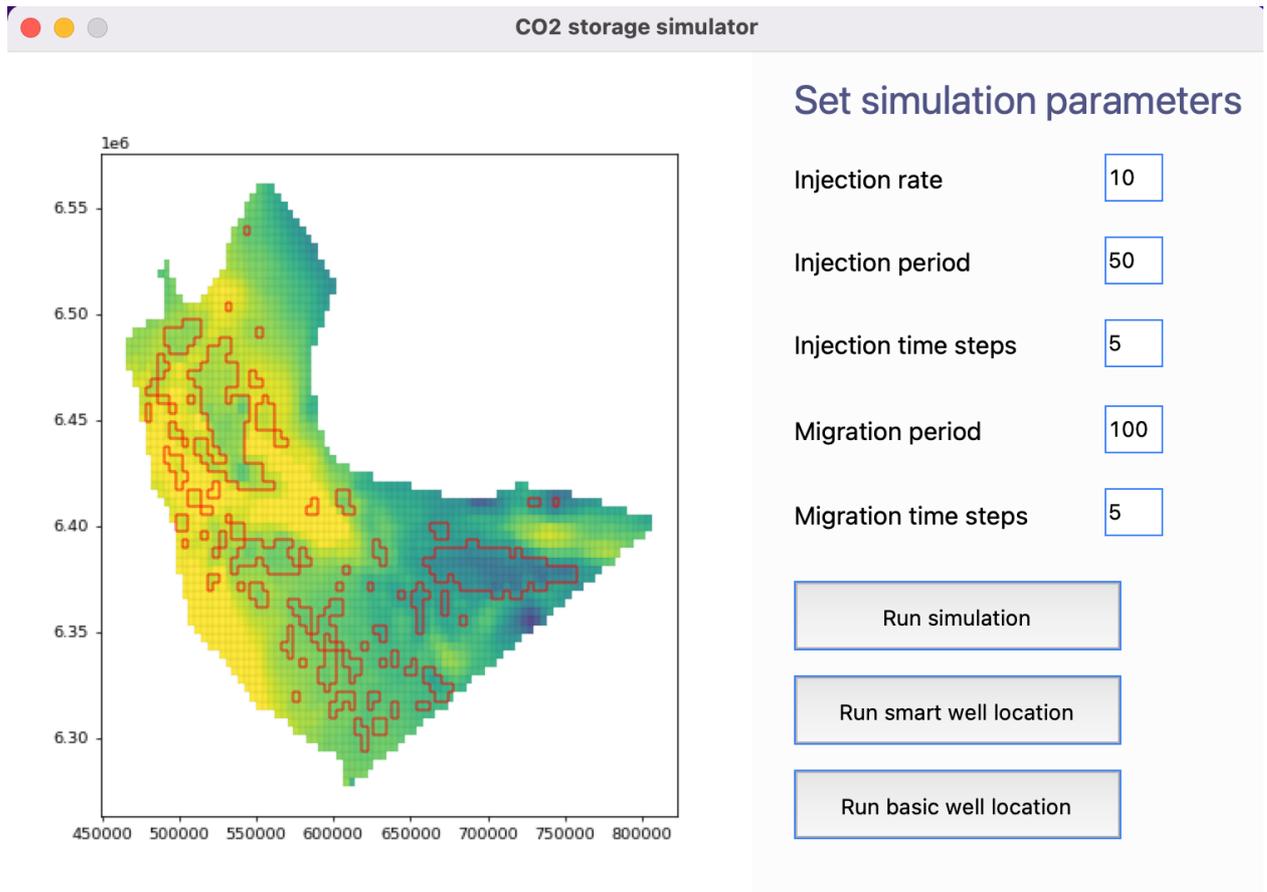


Рисунок 9 – Графический пользовательский интерфейс приложения

- *Injection rate:*

Уровень закачки CO<sub>2</sub> в скважину в мегатоннах/год.

- *Injection period:*

Продолжительность периода закачки.

- *Injection time steps:*

Количество временных шагов моделирования, используемых для представления периода закачки.

- *Migration period:*

Продолжительность периода миграции, следующего за периодом закачки.

- *Migration time steps:*

Количество временных шагов моделирования, используемых для представления периода миграции.

- *Run simulation:*

После того, как все начальные параметры заполнены в окне приложения («Injection rate», «Injection period», «Injection time steps», «Migration period», «Migration time steps»), сценарии долгосрочного хранения CO<sub>2</sub> в масштабах всего водоносного горизонта запускаются нажатием кнопки «Run simulation». Результатом выполнения симуляции является график на рисунке 10.

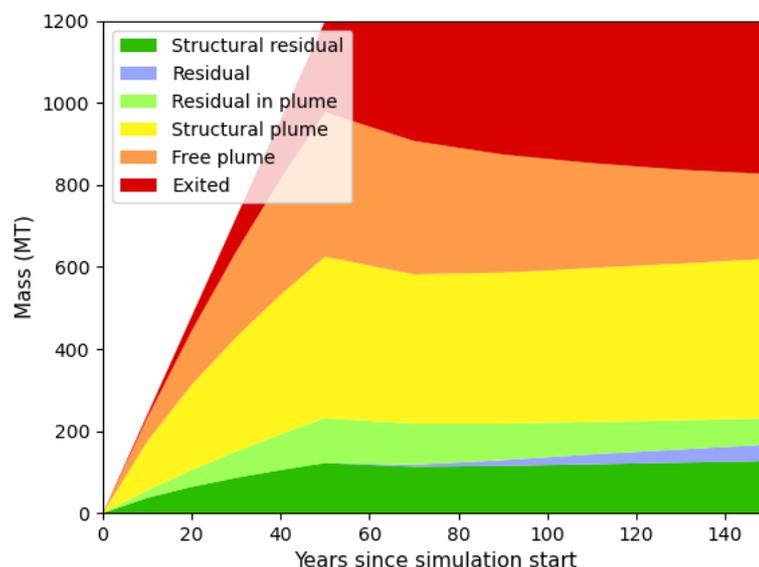


Рисунок 10 – График инвентаризации захоронения CO<sub>2</sub>

- *Run smart well location:*

Для максимального хранения CO<sub>2</sub> в структурной ловушке и минимизации утечки газа за пределы рассматриваемой геологической модели используется алгоритм обучения с подкреплением с политикой нейронной сети. Оптимизируя размещение точек закачки, оптимальные местоположения скважин обнаруживаются алгоритмом и отображаются при нажатии кнопки «Run smart well location». Для каждой исследуемой области местоположение скважины с максимальной «наградой» аннотируется соответствующим

значением, равным разнице между массой  $\text{CO}_2$ , хранимой структурной ловушкой, и массой  $\text{CO}_2$ , вышедшей за границы геологической модели (рис. 11).

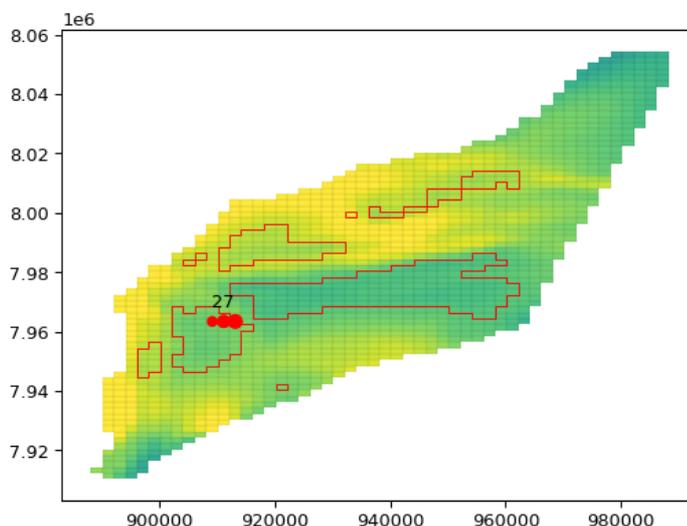


Рисунок 11 – Оптимальные области закачки  $\text{CO}_2$

- *Run basic well location:*

Разница между «Run smart well location» и «Run basic well location» заключается в политике, используемой алгоритмом обучения с подкреплением. При нажатии кнопки «Run basic well location» интуитивная «простая» политика используется для оптимизации размещения местоположений скважин. Результаты оптимизации местоположения скважин динамически отображаются на геологической модели (рис. 12).

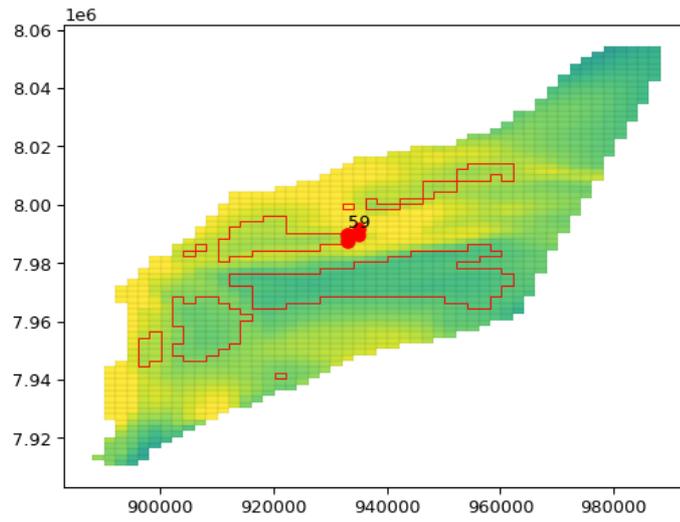


Рисунок 12 – Оптимальные области закачки CO<sub>2</sub>

## **5 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ**

### **5.1 Клиентская часть**

Для разработки клиентской части веб-приложения был использован фреймворк Dash, который является инструментом для интерактивной визуализации, которую можно интегрировать в приложение [13]. Он написан с использованием Flask, React.js, Plotly.js и позволяет строить веб-приложения на языке программирования Python.

Компоненты Dash — это классы Python, которые кодируют свойства и значения конкретного компонента React и сериализуются как JSON. Состояние приложения Dash хранится в веб-браузере. Это позволяет использовать приложения Dash одновременно несколькими пользователями, создавая независимые сеансы при одновременном взаимодействии [15].

### **5.2 Серверная часть**

Для разработки серверной части приложения использовался Python и GNU Octave.

GNU Octave обладает мощным математически ориентированным синтаксисом со встроенными инструментами построения графиков и визуализации в 2D/3D, совместим с большинством программ, написанных на MATLAB, а также является бесплатным программным обеспечением [14].

Python предлагает простые решения без лишних деталей, экономит время и обеспечивает высокий уровень безопасности, в то же время является мощным языком для веб-разработки посредством использования различных фреймворков, а также один из главных языков программирования для работы с машинным обучением.

### 5.3 Логическая модель

Логическая модель предметной области «Симуляция долгосрочного хранения CO<sub>2</sub>» представлена на рисунке 13. Для управления данными предметной области используется система управления базами данных MongoDB, так как данные представлены в полуструктурированной форме.

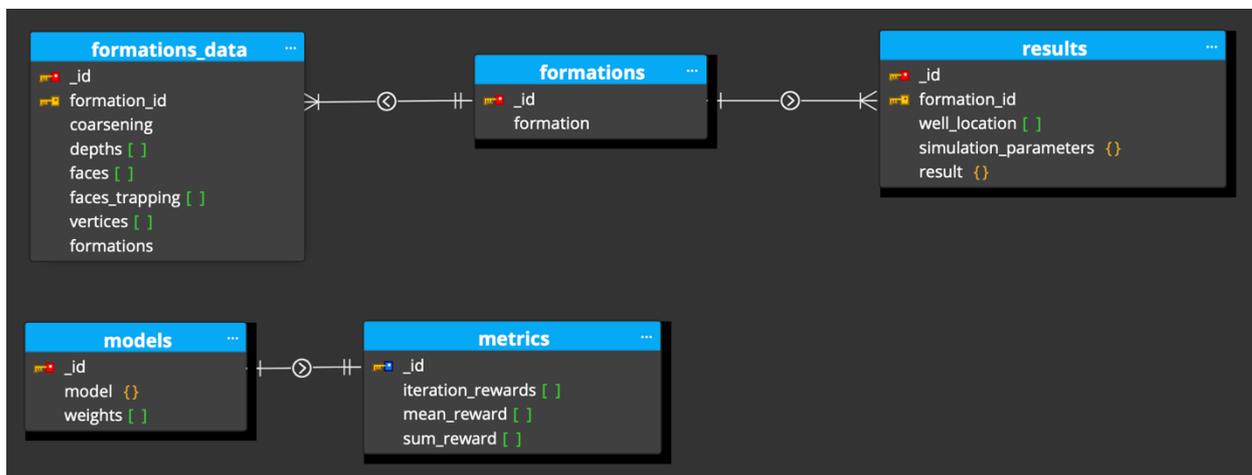


Рисунок 13 – Логическая модель

### 5.4 Физическая модель

Физическая модель предметной области «Симуляция долгосрочного хранения CO<sub>2</sub>» представлена на рисунке 14.

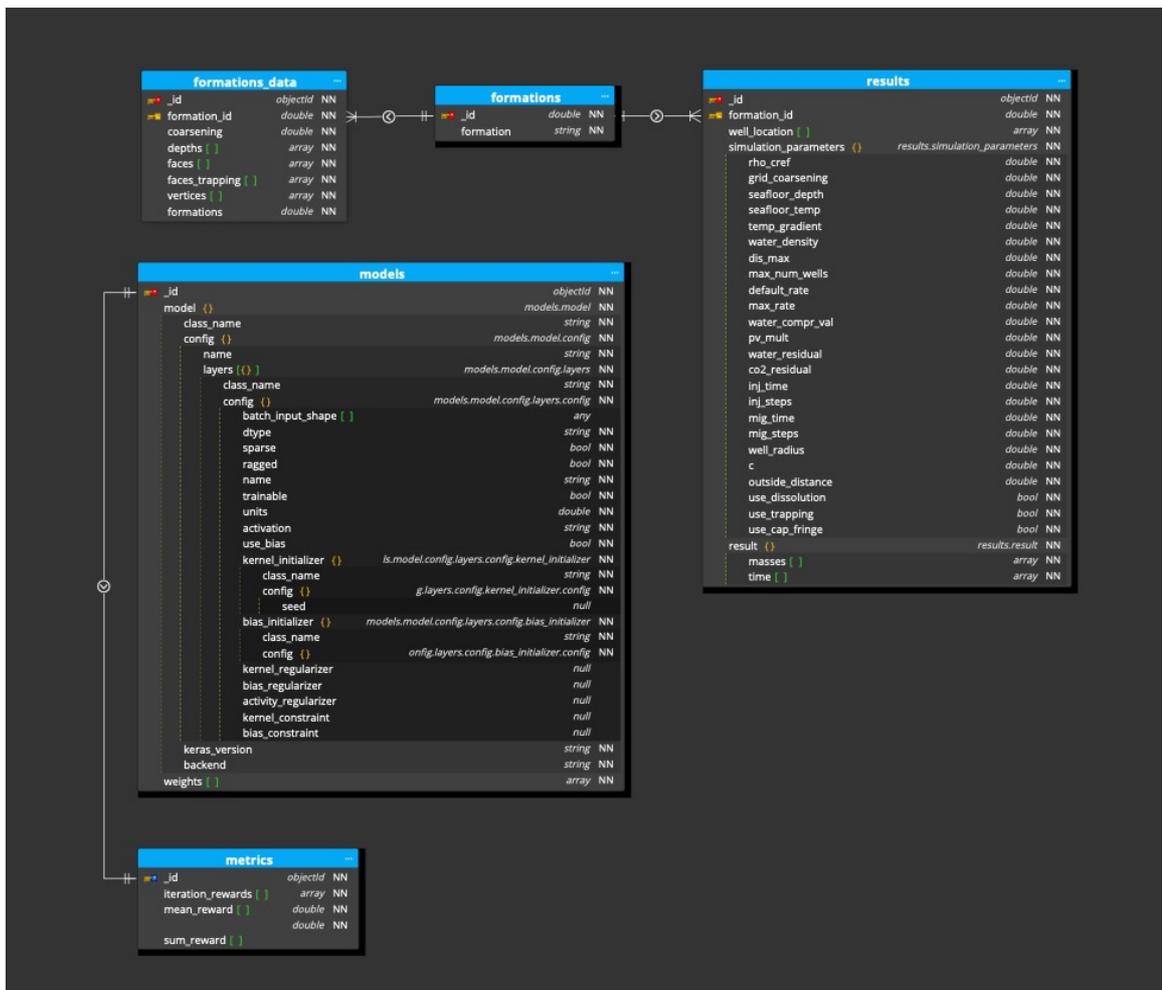


Рисунок 14 – Физическая модель

Коллекции предметной области:

1. Formations – список геологических моделей.
2. Formations\_data – данные геологических моделей.
3. Results – результаты симуляций долгосрочного хранения CO<sub>2</sub>.
4. Models – список нейронных сетей.
5. Metrics – метрики качества соответствующих нейронных сетей.

### 5.5 Диаграмма DFD

Диаграмма DFD функции «Оптимизация расположения скважин, используя политику нейронных сетей» представлена на рисунке 15. Внешние сущности представлены прямоугольниками, накопители данных 2-я параллельными отрезками, а процессы окружностями.

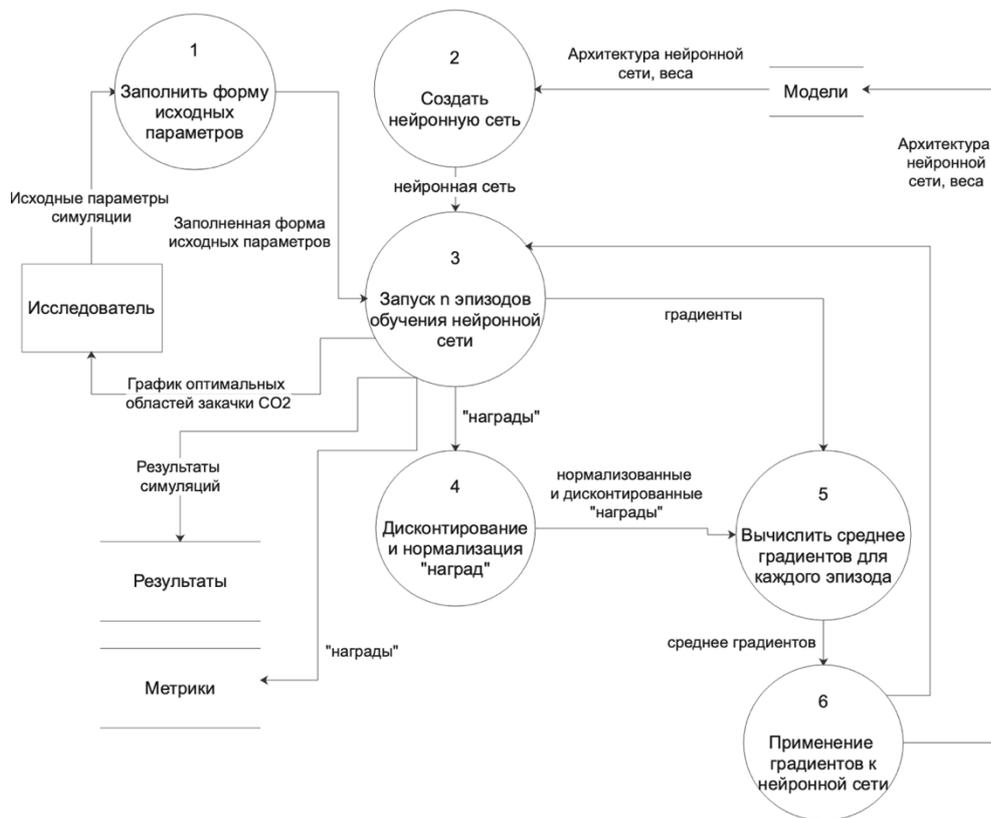


Рисунок 15 – Диаграмма DFD

## 5.6 Диаграмма BPMN

Диаграмма BPMN для процесса интуитивного поиска оптимального местоположения скважин представлена на рисунке 16. На данной диаграмме представлено взаимодействие между пользователем и процессом интуитивного поиска оптимальных местоположений скважин.

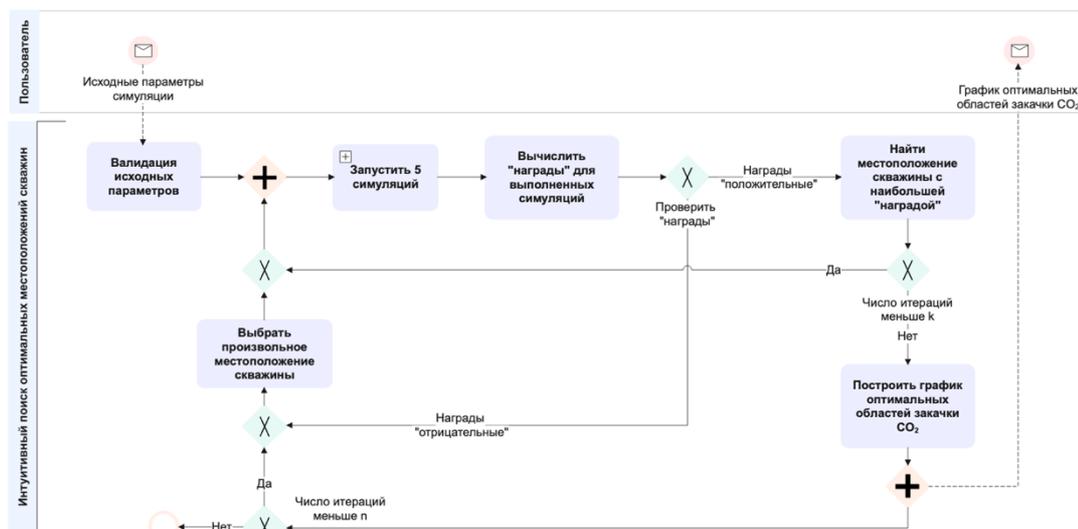


Рисунок 16 – Диаграмма BPMN

## 5.7 Диаграмма EPC

Диаграмма EPC для процесса симуляции долгосрочного хранения CO<sub>2</sub> представлена на рисунке 17. Данная диаграмма представляет собой упорядоченную комбинацию событий и функций, где процесс начинается с ввода исходных параметров симуляции пользователем и заканчивается построением графика инвентаризации захоронения CO<sub>2</sub>.

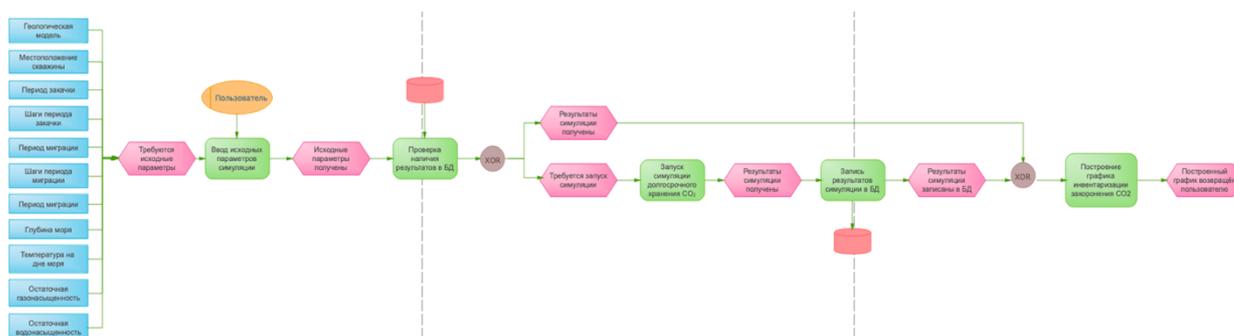


Рисунок 17 – Диаграмма EPC

## 5.8 Использование веб-приложения

При открытии стартовой страница веб-приложения загружается форма настроек (рис. 18), позволяющая пользователям вводить начальные параметры симуляции. Настройки должны вводиться при каждом запуске веб-приложения.

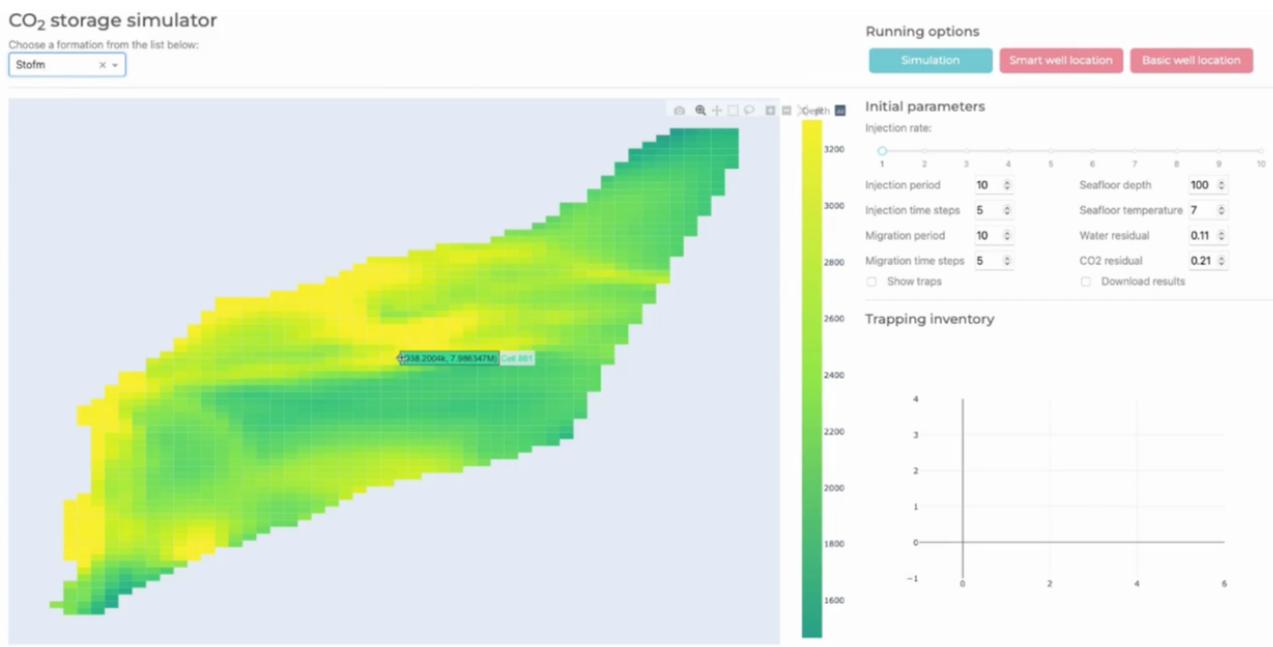


Рисунок 18 – Графический пользовательский интерфейс веб-приложения

- *Injection rate:*

Уровень закачки CO<sub>2</sub> в скважину в мегатоннах/год.

- *Injection period:*

Продолжительность периода закачки.

- *Injection time steps:*

Количество временных шагов моделирования, используемых для представления периода закачки.

- *Migration period:*

Продолжительность периода миграции, следующего за периодом закачки.

- *Migration time steps:*

Количество временных шагов моделирования, используемых для представления периода миграции.

- *Seafloor depth:*

Глубина моря в метрах.

- *Seafloor temperature:*

Температура на дне моря над рассматриваемым коллектором в градусах Цельсия.

- *Water residual:*

Остаточная водонасыщенность рассматриваемого коллектора.

- *CO<sub>2</sub> residual:*

Остаточная газонасыщенность (углекислый газ) рассматриваемого коллектора.

- *Run simulation:*

После того, как все начальные параметры заполнены в окне приложения, сценарии долгосрочного хранения CO<sub>2</sub> в масштабах всего водоносного горизонта запускаются нажатием кнопки «Run simulation». Результатом выполнения симуляции является график в правом нижнем углу на рисунке 19.

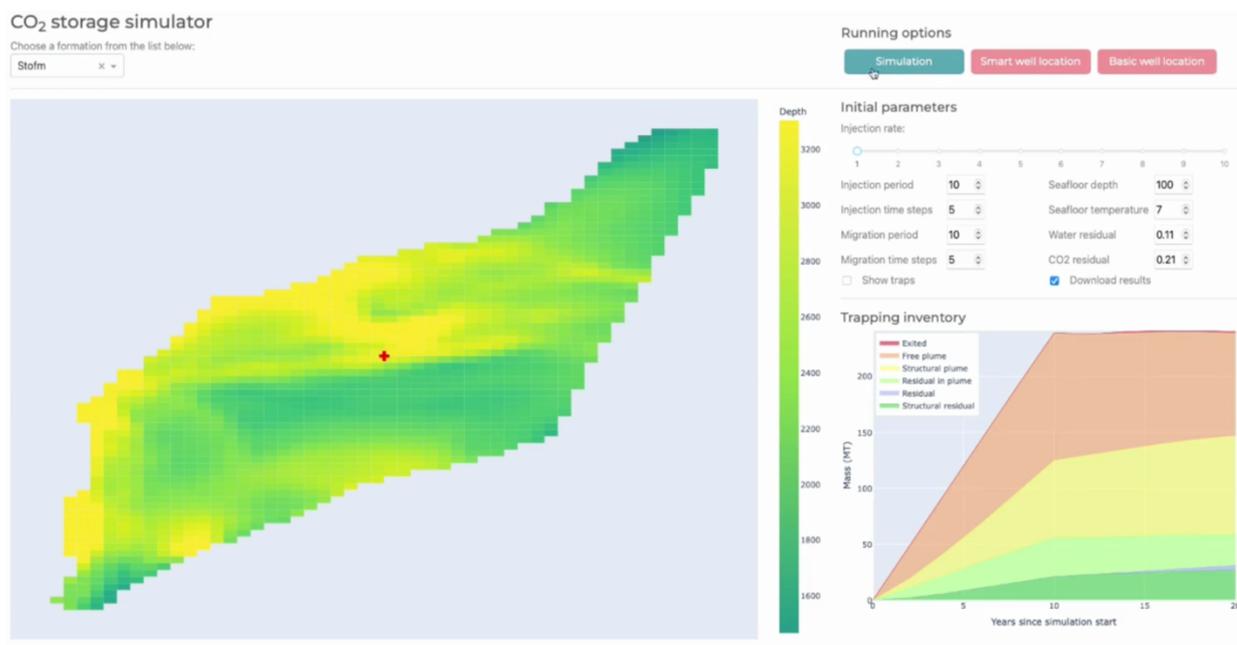


Рисунок 19 – Результат симуляции долгосрочного хранения CO<sub>2</sub>

- *Run smart well location:*

Для максимального хранения  $\text{CO}_2$  в структурной ловушке и минимизации утечки газа за пределы рассматриваемой геологической модели используется алгоритм обучения с подкреплением с политикой нейронной сети. Оптимизируя размещение точек закачки, оптимальные местоположения скважин обнаруживаются алгоритмом и отображаются при нажатии кнопки «Run smart well location». Для каждой исследуемой области местоположение скважины с максимальной «наградой» аннотируется соответствующим значением, равным разнице между массой  $\text{CO}_2$ , хранимой структурной ловушкой, и массой  $\text{CO}_2$ , вышедшей за границы геологической модели (рис. 20).

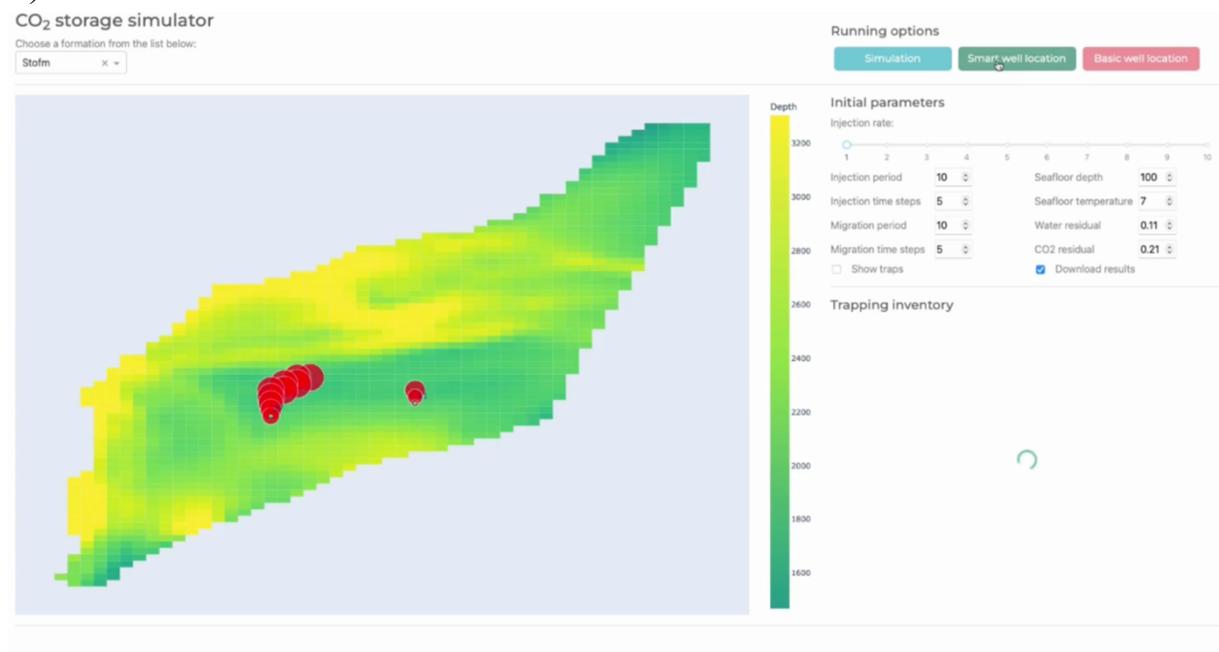


Рисунок 20 – Оптимальные области закачки  $\text{CO}_2$

- *Show traps:*

При выборе опции *Show traps* на геологическом модели отображаются структурные ловушки (рис. 21).

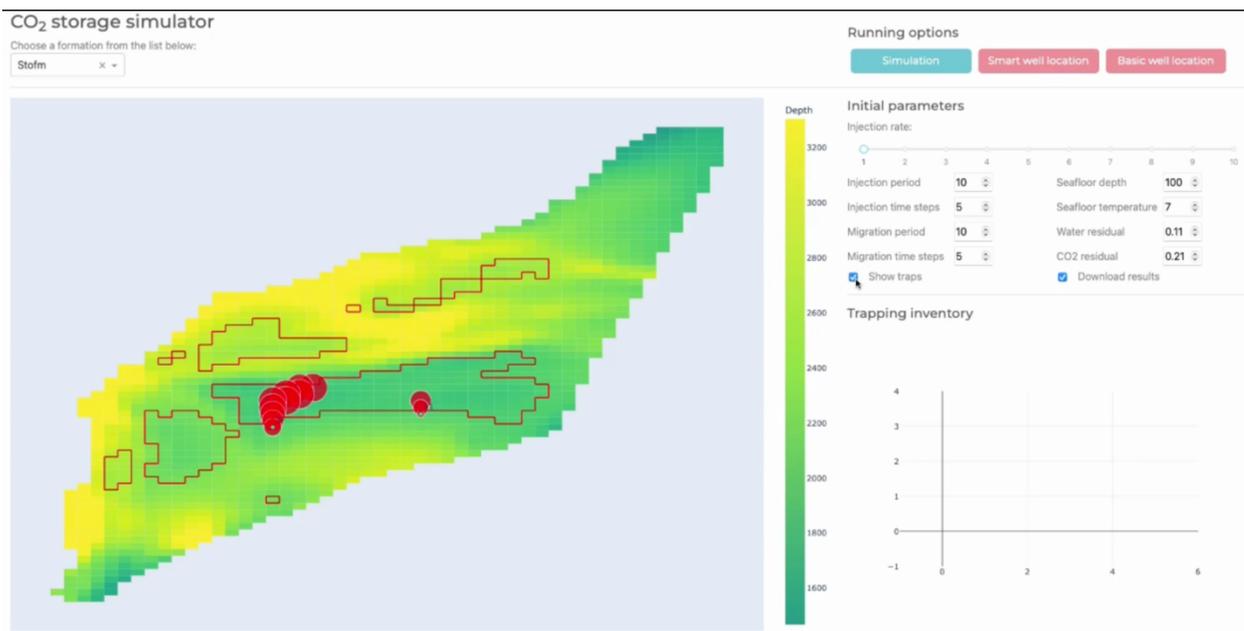


Рисунок 21 – Отображение структурных ловушек

- *Run basic well location:*

Разница между «Run smart well location» и «Run basic well location» заключается в политике, используемой алгоритмом обучения с подкреплением. При нажатии кнопки «Run basic well location» интуитивная «простая» политика используется для оптимизации размещения местоположений скважин. Результаты оптимизации местоположения скважин динамически отображаются на геологической модели (рис. 22).

## CO<sub>2</sub> storage simulator

Choose a formation from the list below:

Stofm x

### Running options

Simulation

Smart well location

Basic well location

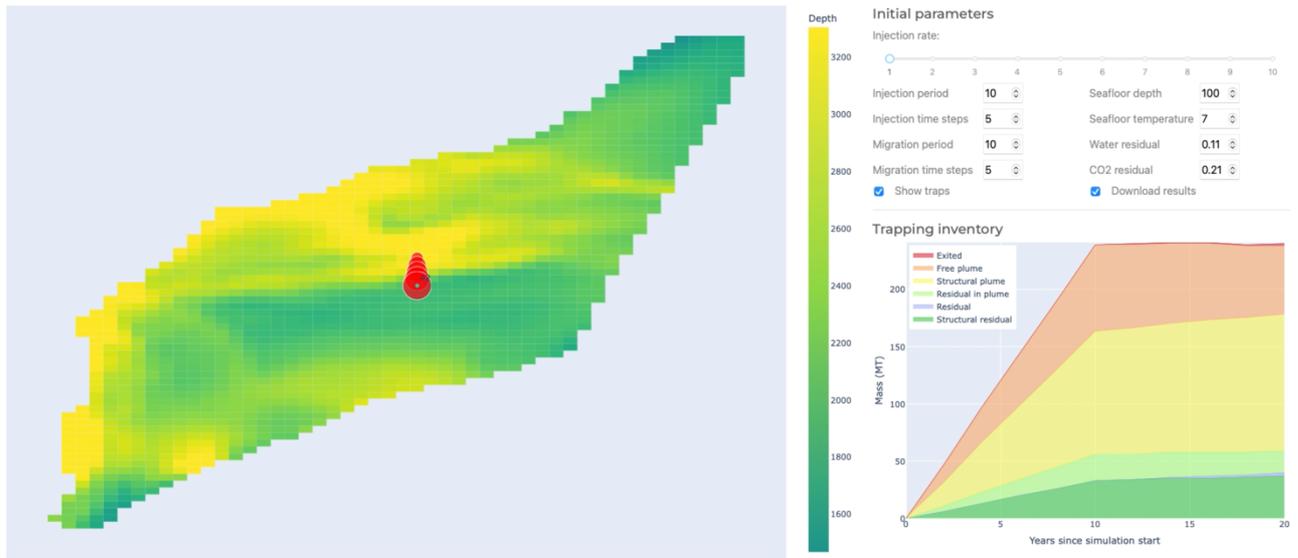


Рисунок 22 – Оптимальные области закачки CO<sub>2</sub>

## 5.9 Развертывание веб-приложения на «Яндекс.Облако»

На рисунках 23–27 показан процесс развёртывания веб-приложения на платформе «Яндекс.Облако».

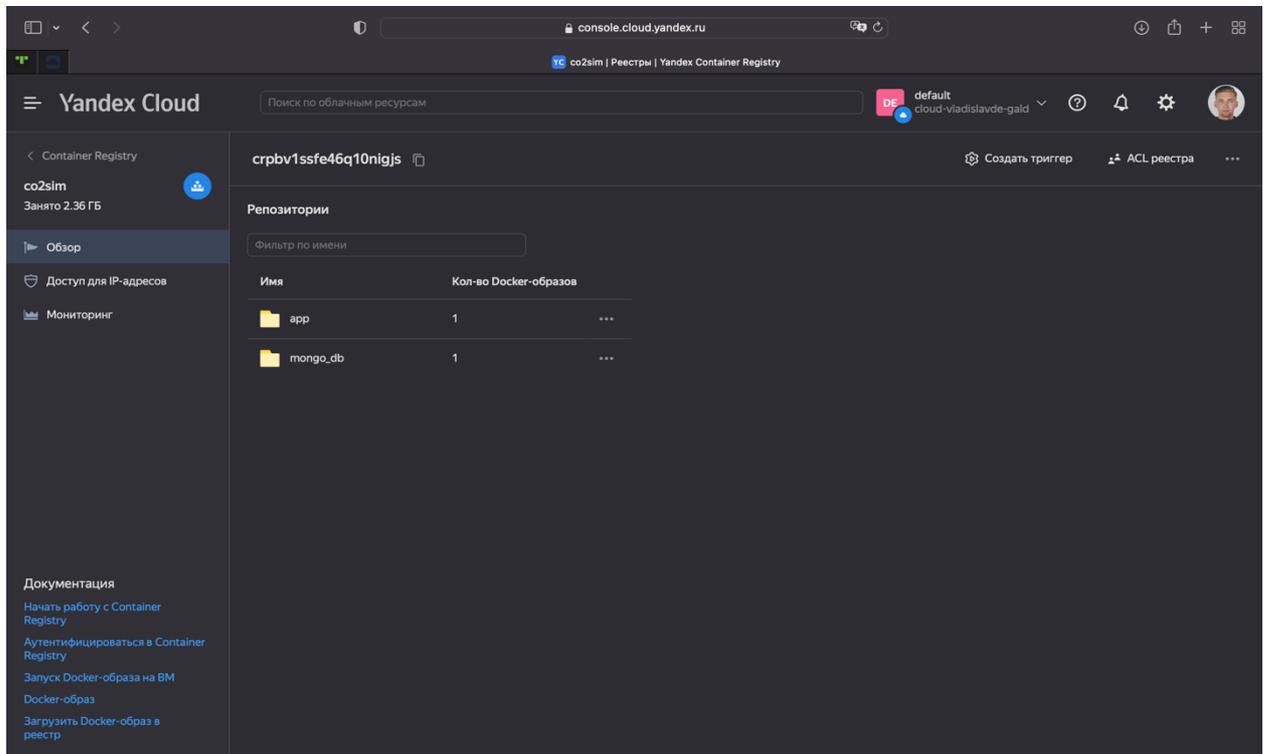


Рисунок 23 – Список docker-образов, загруженных на «Яндекс.Облако»

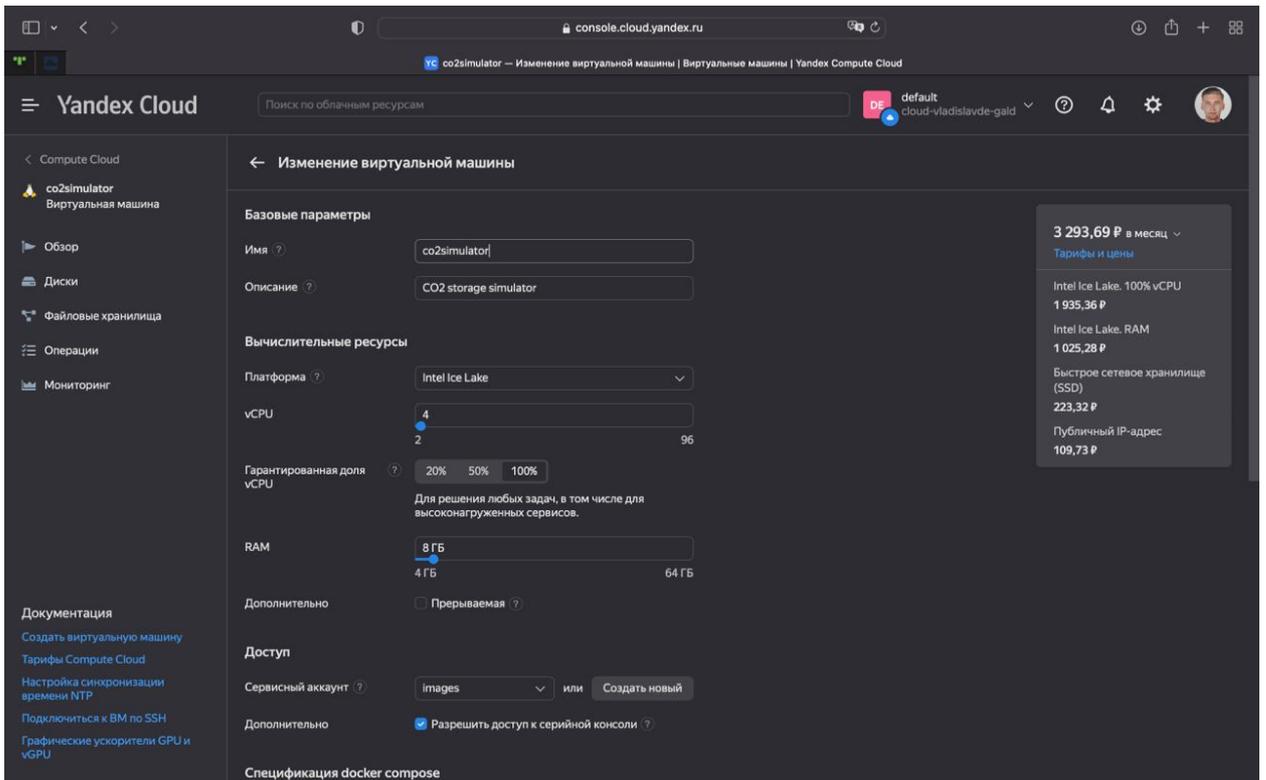


Рисунок 24 – Выбор параметров виртуальной машины

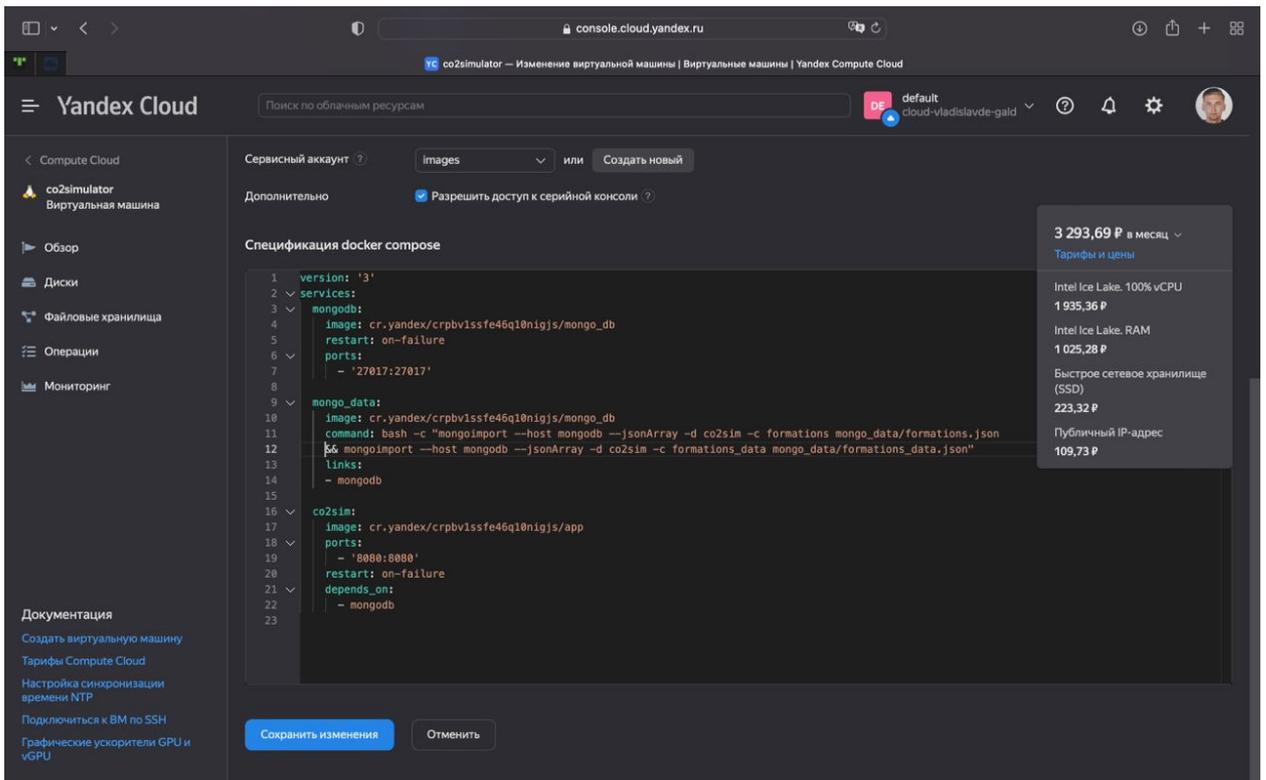


Рисунок 25 – Создание виртуальной машины на основе docker-compose

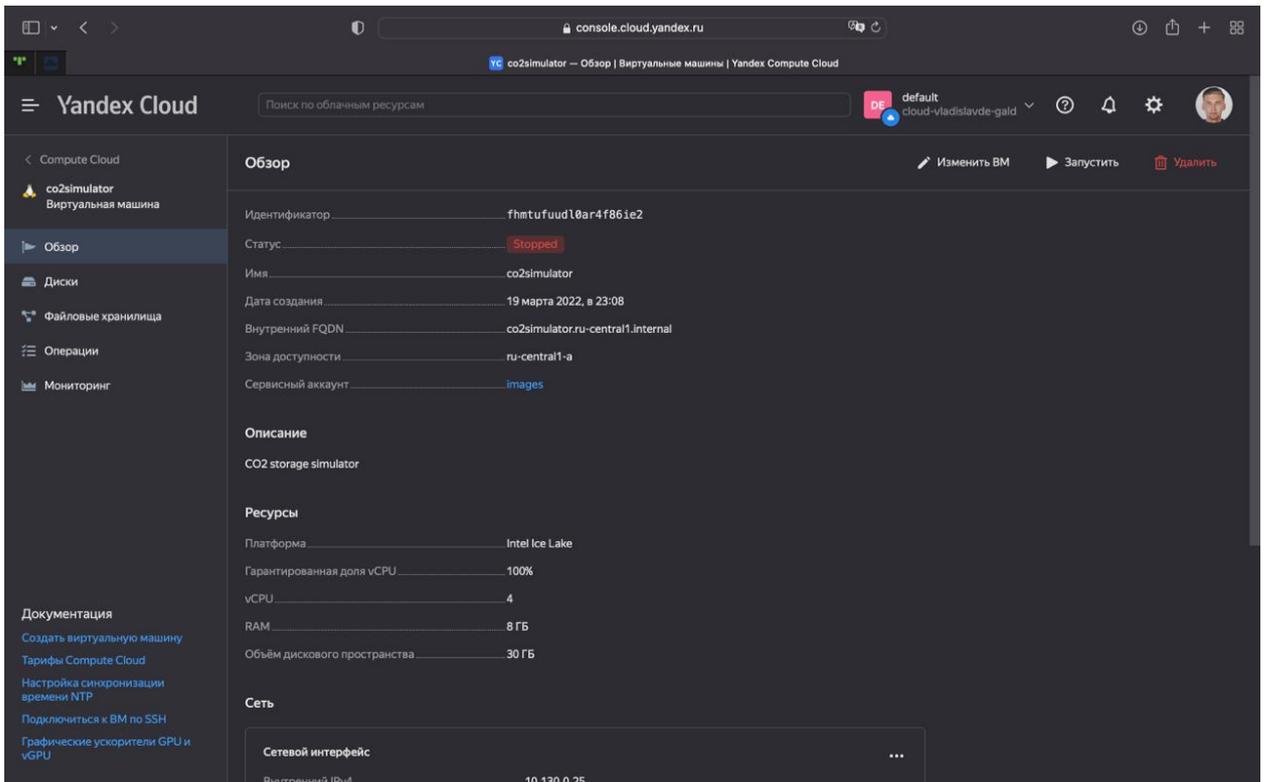


Рисунок 26 – Обзор параметров виртуальной машины

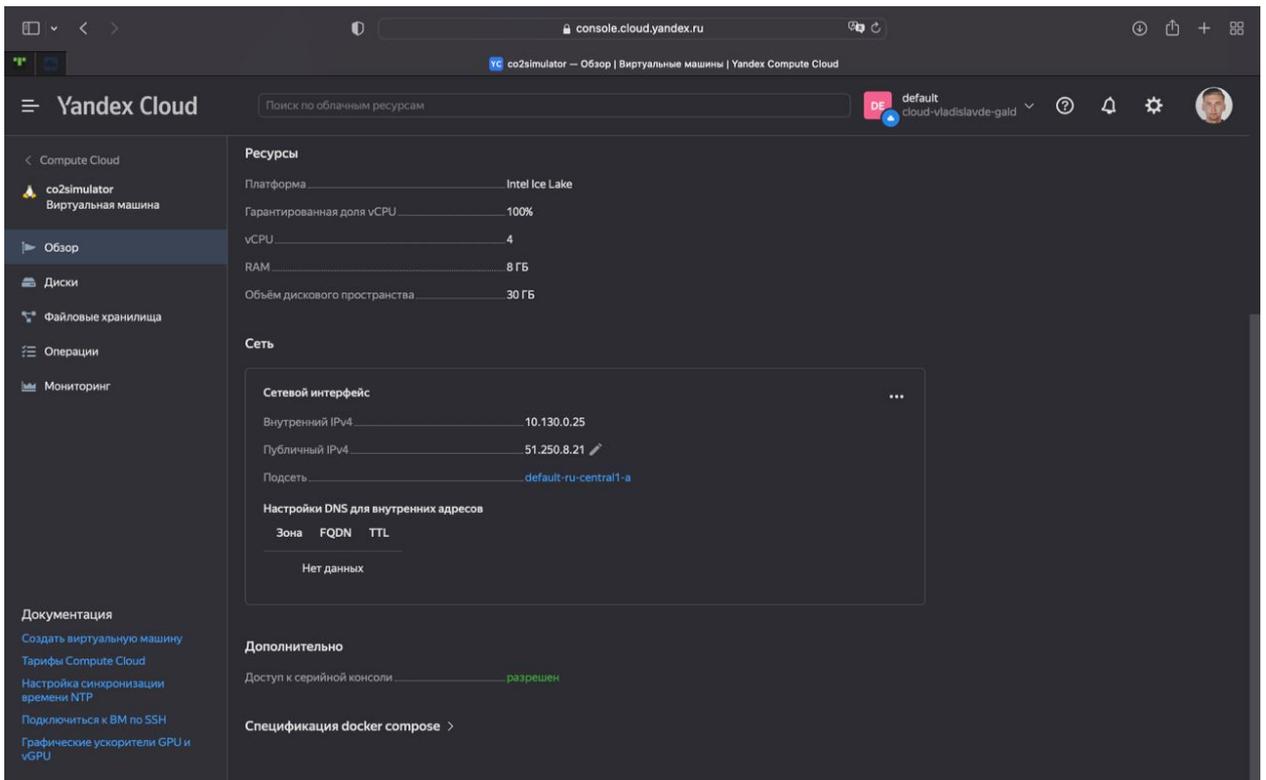


Рисунок 27 – Ресурсы и сеть виртуальной машины

## **6 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ**

Над разработкой программного обеспечения работают 2 человека – научный руководитель и студент.

Целью данной выпускной квалификационной работы является разработка открытого программного продукта, который позволит большему числу людей получить доступ к ресурсу, необходимому для изучения проблемы подземного захоронения CO<sub>2</sub>, а также «среду» для применения методов машинного обучения для решения проблем нефтегазового дела.

Раздел «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» содержит информацию о коммерческой оценке разрабатываемого программного обеспечения. Оценка дает возможность определить коммерческую привлекательность программного продукта.

### **6.1 Потенциальные потребители результатов исследования**

В качестве потенциальных потребителей разрабатываемого программного продукта «Симулятор долгосрочного хранения CO<sub>2</sub> с использованием машинного обучения» выступают нефтяные компании, занимающиеся подземным захоронением подземного газа, а также ученые, работающие над этой проблемой.

Примером предприятия потребителя является компания ООО «Газпромнефть НТЦ» г. Санкт-Петербург.

### **6.2 Анализ конкурентных технических решений**

Анализ конкурентных технических решений необходим, поскольку он помогает получить представление о конкурентах, их успешных решениях, а также распознать их неудачи с целью избежания их повторения. Оценочная карта конкурентных технических решений представлена в таблице 1.

Таблица 1 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		C1	C2	C3	K1	K2	K3
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Точность прогнозирования	0,2	4	5	4	0,8	1	0,8
2. Дизайн	0,15	3	4	3	0,45	0,6	0,45
3. Простота использования	0,15	3	3	3	0,45	0,45	0,45
4. Поддержка и сообщество	0,1	3	3	4	0,3	0,3	0,4
5. Производительность работы	0,2	4	4	5	0,8	0,8	1
Экономические критерии оценки эффективности							
6. Цена	0,1	3	4	3	0,3	0,4	0,3
7. Конкурентоспособность	0,1	2	4	4	0,2	0,4	0,4
<b>Итого</b>	<b>1</b>	<b>22</b>	<b>27</b>	<b>26</b>	<b>3,4</b>	<b>3,95</b>	<b>3,8</b>

Где C1 – программный продукт Eclipse CO2STORE;

C2 – программный продукт CMG-GEM;

C3 – программный продукт MRST-co2lab.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum V_i \times C_i, \quad (8)$$

где K – конкурентоспособность вида;

$V_i$  – вес критерия (в долях единицы);

$C_i$  – балл  $i$ -го показателя.

Данные оценочной карты показывают, что программный продукт CMG-GEM имеет наибольшую конкурентоспособность среди своих конкурентов, но в то же время не лишено таких минусов как: простота в использовании и поддержка сообщества. При разработке и продвижении разрабатываемого программного продукта следует учесть сильные стороны конкурентов и повысить собственную конкурентоспособность за счет качественного дизайна и понятной документации, а также следует развивать сообщество пользователей разрабатываемого приложения.

### 6.3 SWOT-анализ

SWOT — метод анализа в стратегическом планировании, заключающийся в разделении факторов и явлений на четыре категории: сильные стороны, слабые стороны, возможности и угрозы.

На первом этапе SWOT анализа в таблице 2 были описаны сильные и слабые стороны проекта, выявлены возможности и угрозы реализации программного обеспечения.

Таблица 2 – Матрица SWOT анализа

Сильные стороны	Возможности во внешней среде
С1. Доступность для широкого круга пользователей. С2. Скорость вычислений и прогнозирования за счет облачного решения. С3. Понятное руководство пользователя и документация по работе с системой. С4. Востребованность современных решений для хранения CO <sub>2</sub> .	В1. Интернет-потенциал для продвижения на зарубежном рынке. В2. Возрастающая потребность в цифровизации производственных процессов. В3. Адаптация приложения под популярные языки.
Слабые стороны	Угрозы внешней среды
Сл1. Зависимость от быстро меняющихся технологий. Сл2. Значительные временные и интеллектуальные затраты.	У1. Отсутствие спроса на данное решение; У2. Появление местных конкурентов У3. Развитие собственных сервисов российскими ВИНК;

На втором этапе была построена интерактивную матрицу проекта, представленная в таблице 3, где «+» означает сильное соответствие, «-» слабое соответствие и «0» трудность в ответе.

Таблица 3 – Интерактивная матрица сильных и слабых сторон и возможностей

Возможности проекта	Сильные стороны				Слабые стороны	
		C1	C2	C3	C4	Сл1
B1	+	+	+	+	+	0
B2	0	+	+	+	0	+
B3	+	0	+	0	-	+

Таблица 4 – Интерактивная матрица сильных сторон, слабых сторон и угроз

Угрозы проекта	Сильные стороны				Слабые стороны	
		C1	C2	C3	C4	Сл1
У1	+	0	+	+	+	0
У2	+	+	0	+	-	-
У3	+	+	+	+	-	-

Самой большой угрозой для проекта является развитие собственных сервисов российскими вертикально-интегрированными нефтяными компаниями.

Что касается слабых сторон, то разработка программного обеспечения для симуляции долгосрочного хранения CO<sub>2</sub> с использованием методов машинного обучения требует значительные временные и интеллектуальные затраты.

В рамках третьего этапа была составлена итоговая матрица SWOT анализа, представленная в таблице 5.

Таблица 5 – Итоговая матрица SWOT-анализа

	<p>Сильные стороны научно-исследовательского проекта:</p> <p>C1. Доступность для широкого круга пользователей.</p> <p>C2. Скорость вычислений и прогнозирования за счет облачного решения.</p>	<p>Слабые стороны научно-исследовательского проекта:</p> <p>Сл1. Зависимость от быстро меняющихся технологий.</p> <p>Сл2. Значительные временные и интеллектуальные затраты.</p>
--	--	--

	<p>С3. Понятное руководство пользователя и документация по работе с системой.</p> <p>С4. Востребованность современных решений для хранения CO<sub>2</sub>.</p>	
<p>Возможности:</p> <p>В1. Интернет-потенциал для продвижения на зарубежном рынке.</p> <p>В2. Возрастающая потребность в цифровизации производственных процессов.</p> <p>В3. Адаптация приложения под популярные языки.</p>	<p>Большой потенциал проекта в области хранения CO<sub>2</sub> в России и за рубежом поможет проекту развиваться и совершенствовать вычислительные методы для увеличения точности прогнозирования.</p>	<p>Проект предполагает значительные временные затраты, использование современных технологий для прогнозирования хранения CO<sub>2</sub>, а также использование ИИ для оптимизации расстановки скважин.</p>
<p>Угрозы:</p> <p>У1. Отсутствие спроса на данное решение.</p> <p>У2. Появление местных конкурентов.</p> <p>У3. Развитие собственных сервисов российскими ВИНК.</p>	<p>Ставка на доступность и вычислительную точность приложения вместе с понятной документацией позволит привлечь пользователей среди специалистов нефтяников и учёных крупных нефтегазовых компаний.</p>	<p>Самой большой угрозой для проекта является появление крупных конкурентов среди ВИНК, из-за ограниченности ресурсов и некоторых каналов продвижения будет непросто развивать и распространять проект среди крупных «игроков».</p>

## **6.4 Планирование работ по научно-техническому исследованию**

### **6.4.1 Структура работ в рамках научного исследования**

Для дальнейших расчёты было выделено три варианта решения технической задачи:

1. Исп. 1 – технологии Java и Matlab.
2. Исп. 2 – технологии Python и GNU Octave.
3. Исп. 3 – технологии C+ и Matlab.

Проекты по разработке программного обеспечения, как правило, сложны и многогранны. Без надлежащего планирования разработки программного обеспечения проект по разработке программного обеспечения может оказаться под угрозой невыполнения поставленных задач или даже провала. Исполнителями проекта по разработке программного обеспечения для симуляции долгосрочного хранения CO<sub>2</sub> с использованием методов машинного обучения являются научный руководитель и студент. Перечень работ по проекту приведен в таблице 6.

Таблица 6 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Выбор направления исследований	1	Составление и утверждение темы бакалаврской работы	Руководитель Бакалавр
Содержание проекта	2	Определение содержания проекта	Руководитель Бакалавр
Техническое задание	3	Подготовка требований к программному обеспечению	Руководитель Бакалавр
	4	Разработка бюджета проекта	Бакалавр
	5	Составление календарного плана-графика	Руководитель Бакалавр
Проектирование программного обеспечения	6	Проектирование десктопного приложения	Бакалавр
	7	Проектирование базы данных	Бакалавр
	8	Проектирование веб-приложения	Бакалавр
	9	Разработка десктопного приложения	Бакалавр

Разработка программного обеспечения	10	Применение методов машинного обучения для оптимизации расположения скважин	Бакалавр
	11	Проектирование веб-приложения	Бакалавр
Тестирование	12	Разработка плана тестирования на основе технического задания	Бакалавр
	13	Модульное тестирование десктопного приложения	Бакалавр
	14	Интеграционное тестирование десктопного приложения	Бакалавр
	15	Модульное тестирование веб-приложения	Бакалавр
	16	Интеграционное тестирование веб-приложения	Бакалавр
Документация	17	Подготовка документации	Бакалавр
Оформление отчета по НИР	18	Оформление пояснительной записки	Руководитель Бакалавр

#### 6.4.2 Определение трудоёмкости выполнения работ

Большая часть бюджета при разработке программного обеспечения состоит из заработной платы разработчиков, поэтому очень важно оценить трудоёмкость выполнения работ по разработке ПО.

Ожидаемая трудоёмкость  $t_{ож\ i}$  рассчитывается по следующей формуле:

$$t_{ож\ i} = \frac{3t_{\min\ i} + 2t_{\max\ i}}{5}, \quad (9)$$

где  $t_{\min\ i}$  – минимально возможная трудоёмкость выполнения  $i$ -ой работы (чел.-дн.);

$t_{\max\ i}$  – максимально возможная трудоёмкость выполнения  $i$ -ой работы (чел.-дн.).

Продолжительность работы в рабочих днях  $T_p$  рассчитывается по следующей формуле:

$$T_{pi} = \frac{t_{ожі}}{ч_i}, \quad (10)$$

где  $t_{ожі}$  – ожидаемая трудоемкость выполнения одной работы (чел.-дн.);

$ч_i$  – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе (чел).

### 6.4.3 Разработка графика проведения научного исследования

Удобным способом визуальной демонстрации графика проведения научного исследования является диаграмма Ганта. Она представляет собой столбчатую диаграмму, которая обеспечивает визуальное представление задач проекта, запланированных с течением времени.

Чтобы обеспечить удобство построения диаграммы Ганта, необходимо длительность этапов работ перевести из рабочих дней в календарные, используя следующую формулу:

$$T_{ki} = T_{pi} \cdot k_{кал}, \quad (11)$$

где  $T_{pi}$  – продолжительность выполнения  $i$ -й работы в рабочих днях;

$k_{кал}$  – коэффициент календарности.

Коэффициент календарности равен:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - (T_{вых} + T_{пр})} = \frac{365}{365 - 118} = 1,48, \quad (12)$$

где  $T_{кал}$  – количество календарных дней в году;

$T_{вых}$  – количество выходных дней в году;

$T_{пр}$  – количество праздничных дней в году.

Временные показатели проведения научного исследования представлены в таблице 7.

Таблица 7 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ									Исполнители	Длительность работ в рабочих днях $T_{pi}$			Длительность работ в календарных днях $T_{ki}$		
	$T_{min}$ , чел–дни			$T_{max}$ , чел–дни			$T_{ожи}$ , чел– дни				Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3							
Составление и утверждение темы бакалаврской работы	1	1	1	2	2	2	1,4	1,4	1,4	Студент, научный руководитель	1	1	1	1	1	1
Определение содержания проекта	1	1	1	2	2	2	1,4	1,4	1,4	Студент, научный руководитель	1	1	1	1	1	1
Подготовка требований к программному обеспечению	4	5	5	6	6	7	4,8	5,4	5,8	Студент, научный руководитель	2	2	2	2	2	2
Разработка бюджета проекта	7	7	8	10	9	10	8,2	7,8	8,8	Студент	8	7	8	11	10	11
Составление календарного плана-графика	4	4	5	6	6	7	4,8	4,8	5,8	Студент, научный руководитель	2	2	2	2	2	2

Проектирование десктопного приложения	15	15	16	18	18	17	16,2	16,2	16,4	Студент	16	16	16	23	23	23
Проектирование базы данных	8	8	9	10	11	11	8,8	9,2	9,8	Студент	8	9	9	11	13	13
Проектирование веб-приложения	15	16	16	18	18	19	16,2	16,8	17,2	Студент	16	16	17	23	23	25
Разработка десктопного приложения	50	52	54	63	60	60	55,2	55,2	56,4	Студент	55	55	56	81	81	82
Применение методов машинного обучения для оптимизации расположения скважин	30	30	34	42	40	42	34,8	34	37,2	Студент	34	34	37	50	50	54
Разработка плана тестирования на основе технического задания	8	9	9	10	10	11	8,8	9,4	9,8	Студент	8	9	9	11	13	13
Разработка веб-приложения	16	18	18	21	21	20	18	19,2	18,8	Студент	18	19	18	26	28	26
Модульное тестирование десктопного приложения	3	4	4	5	5	6	3,8	4,4	4,8	Студент	3	4	4	4	5	5

Интеграционное тестирование десктопного приложения	2	2	3	4	5	5	2,8	3,2	3,8	Студент	2	3	3	2	4	4
Модульное тестирование веб-приложения	3	3	4	5	5	5	3,8	3,8	4,4	Студент	3	3	4	4	4	5
Интеграционное тестирование веб-приложения	2	3	3	4	4	5	2,8	3,4	3,8	Студент	2	3	3	2	4	4
Составление документации	3	4	4	5	5	6	3,8	4,4	4,8	Студент	3	4	4	4	5	5
Оформление пояснительной записки	3	3	4	5	6	6	3,8	4,2	4,8	Студент, научный руководитель	1	2	2	1	2	2

Для иллюстрации календарного плана была использована диаграмма Ганта, которая представлена в таблице 8.

Таблица 8 – Календарный план-график проведения научного исследования

№ раб	Название работы	Исполнители	Продолжительность выполнения работ								
			Сентябрь	Октябрь	Ноябрь	Декабрь	Январь	Февраль	Март	Апрель	Май
1	Составление и утверждение темы бакалаврской работы	Ст. НР	■								
2	Определение содержания проекта	Ст. НР	■								
3	Подготовка требований к программному обеспечению	Ст. НР	■	■							
4	Разработка бюджета проекта	Ст.		■							
5	Составление календарного плана-графика	Ст. НР		■							
6	Проектирование десктопного приложения	Ст.			■						
7	Проектирование базы данных	Ст.			■						
8	Проектирование веб-приложения	Ст.				■					
9	Разработка десктопного приложения	Ст.				■	■				
10	Применение методов машинного обучения для оптимизации	Ст.					■	■			

	расположения скважин									
11	Разработка веб-приложения	Ст.								
12	Разработка плана тестирование на основе технического задания	Ст.								
13	Модульное тестирование десктопного приложения	Ст.								
14	Интеграционное тестирование десктопного приложения	Ст.								
15	Модульное тестирование веб-приложения	Ст.								
16	Интеграционное тестирование веб-приложения	Ст.								
17	Подготовка документации	Ст.								
18	Составление пояснительной записки	Ст. НР								

– научный руководитель;
 – студент.

## 6.5 Бюджет научно-технического исследования (НТИ)

Бюджетное планирование научно-технического исследования является очень важным этапом разработки программного обеспечения.

В процессе формирования бюджета НТИ используется следующая группировка затрат по статьям:

1. Материальные затраты.
2. Затраты на приобретение спецоборудования для разработки ПО.
3. Основная и дополнительная заработные платы исполнителей проекта.
4. Отчисления во внебюджетные фонды (страховые отчисления).
5. Накладные расходы.

### 6.5.1 Расчет материальных затрат НТИ

Статья материальных затрат включает в себя стоимость всех материалов, используемых при работе над проектом. Материальные затраты для разработки программного обеспечения перечислены в таблице 9.

Таблица 9 – Материальные затраты

Наименование	Единица измерения	Количество			Цена за ед., руб.	Затраты на материалы, (Зм), руб.		
		Исп.1	Исп.2	Исп.3		Исп.1	Исп.2	Исп.3
Аренда виртуальной машины	Шт.	1	1	1	3600	3600	3600	3600
Интернет	Мбит/с	1	1	1	450	450	450	450
Электроэнергия	кВт*ч	250	300	280	3,85	962,5	1155	1078
Итого, руб.						5012,5	5205	5128

Общие материальные затраты составили 5012,5 рублей.

### 6.5.2 Расчет затрат на специальное оборудование для научных работ

Данная статья включает все затраты, связанные с приобретением специального оборудования необходимого для разработки текущего проекта, которые представлены в таблице 10.

Таблица 10 – Расчет бюджета затрат на приобретение спецоборудования для научных работ

Наименование	Единица измерения	Количество			Цена за ед., тыс. руб.	Затраты на материалы, (Зм), тыс. руб.		
		Исп. 1	Исп. 2	Исп. 3		Исп. 1	Исп. 2	Исп. 3
Персональный компьютер	Шт.	1	1	0	60	60	60	-
Ноутбук MacBook Pro 2019, 128 Гб	Шт.	1	1	1	90	90	90	90
Программное обеспечение PyCharm	Шт.	1	1	1	2	2	2	2
Программное обеспечение (MATLAB)	Шт.	1	1	1	5	5	5	5
Итого:						157	157	97

### 6.5.3 Основная заработная плата исполнителя темы

Данная статья расходов включает основную заработную плату научного руководителя и студента. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20–30% от тарифа или оклада. Расчет основной заработной платы приводится в таблице 11.

Таблица 11 – Расчет основной заработной платы

№ п/п	Наименование этапов	Исполнители по категориям	Трудоемкость, чел.-дн.			Заработная плата, приходящаяся на один чел.-дн.			Всего заработная плата по тарифу (окладам), тыс. руб.		
			Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
1.	Составление и утверждение темы	Студент НР	1	1	1	3,1			3,1	3,1	3,1

	бакалаврской работы								
2.	Определение содержания проекта	Студент НР	1	1	1	3,1	3,1	3,1	3,1
3.	Подготовка требований к программному обеспечению	Студент НР	2	2	2	2	4	4	4
4.	Разработка бюджета проекта	Студент	8	7	8	3,1	24,8	21,7	24,8
5.	Составление календарного плана-графика	Студент НР	2	2	2	2	4	4	4
6.	Проектирование десктопного приложения	Студент	16	16	16	3,1	49,6	49,6	49,6
7.	Проектирование базы данных	Студент	8	9	9	2	16	18	18
8.	Проектирование веб-приложения	Студент	16	16	17	3,1	49,6	49,6	52,7
9.	Разработка десктопного приложения	Студент	55	55	56	3,1	170,5	170,5	173,6
10.	Применение методов машинного обучения для оптимизации расположения скважин	Студент	34	34	37	3,1	105,4	105,4	114,7
11.	Проектирование веб-приложения	Студент	8	9	9	3,1	24,8	27,9	27,9
12.	Разработка плана тестирования на основе технического задания	Студент	18	19	18	2	36	38	36
13.	Модульное тестирование	Студент	3	4	4	2	6	8	8

	десктопного приложения								
14.	Интеграционное тестирование десктопного приложения	Студент	2	3	3	2	4	6	6
15.	Модульное тестирование веб-приложения	Студент	3	3	4	3,1	9,3	9,3	12,4
16.	Интеграционное тестирование веб-приложения	Студент	2	3	3	3,1	6,3	9,3	9,3
17.	Подготовка документации	Студент	3	4	4	2	6	8	8
18.	Составление пояснительной записки	Студент НР	1	2	2	2	2	4	4
Итого							524, 5	539, 5	559, 2

Статья включает основную заработную плату работников, непосредственно занятых выполнением проекта, (включая премии, доплаты) и дополнительную заработную плату и рассчитывается по формуле:

$$Z_{зп} = Z_{осн} + Z_{доп}, \quad (13)$$

где  $Z_{осн}$  – основная заработная плата;

$Z_{доп}$  – дополнительная заработная плата (12–20 % от  $Z_{осн}$ ).

Основная заработная плата руководителя рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} \cdot T_p, \quad (14)$$

где  $Z_{осн}$  – основная заработная плата одного работника;

$T_p$  – продолжительность работ, выполняемых научно-техническим работником, раб. дн.;

$Z_{дн}$  – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_{\text{м}} \cdot M}{F_{\text{д}}}, \quad (15)$$

где  $Z_{\text{м}}$  – месячный должностной оклад работника, руб.;

$M$  – количество месяцев работы без отпуска в течение года:

при отпуске в 24 раб. дня  $M = 11,2$  месяца, 5–дневная неделя;

при отпуске в 48 раб. дней  $M = 10,4$  месяца, 6–дневная неделя;

$F_{\text{д}}$  – действительный годовой фонд рабочего времени научно–технического персонала, раб. дн.

Таблица 12 – Баланс рабочего времени

Показатели рабочего времени	Руководитель	Студент
Календарное число дней	365	365
Количество нерабочих дней - выходные дни - праздничные дни	118	118
Потери рабочего времени - отпуск - невыходы по болезни	48 0	64 0
Действительный годовой фонд рабочего времени	199	183

Месячный должностной оклад работника (руководителя):

$$Z_{\text{м}} = Z_{\text{тс}} \cdot (1 + k_{\text{пр}} + k_{\text{д}}) \cdot k_{\text{р}}, \quad (16)$$

где  $Z_{\text{тс}}$  – заработная плата по тарифной ставке, руб.;

$k_{\text{пр}}$  – премиальный коэффициент, равный 0,3 (т. е. 30 % от  $Z_{\text{тс}}$ );

$k_{\text{д}}$  – коэффициент доплат и надбавок составляет примерно 0,2–0,5;

$k_{\text{р}}$  – районный коэффициент, равный 1,3 (для Томска).

Тарифная заработная плата  $Z_{\text{тс}}$  находится из произведения тарифной ставки работника 1-го разряда  $T_{\text{с1}} = 600$  руб. на тарифный коэффициент  $k_{\text{т}}$  и учитывается по единой для бюджетной организации тарифной сетке.

Тарифный коэффициент для НР = 1,866; для С = 1,407.

Расчет основной заработной платы представлен в таблице 13.

Таблица 13 – Расчет основной заработной платы

Исполнители	Разряд	к <sub>Т</sub>	З <sub>тс</sub> , руб.	к <sub>пр</sub>	к <sub>д</sub>	к <sub>р</sub>	З <sub>м</sub> , руб.	З <sub>дн</sub> , руб.	Т <sub>р</sub> , раб. дн.	З <sub>осн</sub> , руб.
Научный руководитель	к.т.н. доцент	1,866	30000	0,3	0,4	1,3	66300	3731,45	7	26120,15
Студент	Программист	1,407	15000	0,3	0,3	1,3	31200	1909,5	183	349438,5
Итого										375558,65

#### 6.5.4 Расчет дополнительной заработной платы

Согласно трудовому законодательству на территории России предусмотрена возможность выплаты дополнительной заработной платы в зависимости от условий труда, которая предоставляется работнику в качестве поощрения, например, премиальные отчисления или при переработке установленных нормативов.

Расчет дополнительной заработной платы рассчитывается по формуле:

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}}, \quad (17)$$

где  $k_{\text{доп}}$  – коэффициент дополнительной заработной платы, принятый на стадии проектирования за 0,15.

#### 6.5.5 Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из формулы:

$$Z_{\text{внеб}} = k_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}}), \quad (18)$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

В соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30,2%.

Отчисления во внебюджетные фонды представлены в таблице 14.

Таблица 14 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель проекта	26120,15	29851,6	29851,6	3918	4477,74	4477,74
Студент	349438,5	343710	374262	52415,77	51556,5	56139,3
Коэффициент отчислений во внебюджетные фонды	0,302					
Итого						
Исполнение 1	130431,51					
Исполнение 2	129737,94					
Исполнение 3	140348,65					

#### 6.5.6 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов. Их величина определяется по формуле:

$$Z_{\text{накл}} = (\sum \text{статей}) \cdot k_{\text{нр}}, \quad (19)$$

где  $k_{\text{нр}}$  – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов можно взять в размере 16%.

Накладные расходы для исполнения 1 составили:

$$Z_{\text{накл}} = (5012,5 + 157000 + 26120,15 + 349438,5 + 3918 + 52415,77 + 130431,51) \cdot 0,16 = 115893,83 \text{ руб.}$$

Накладные расходы для исполнения 2 составили:

$$Z_{\text{накл}} = (5205 + 157000 + 29851,6 + 343710 + 4477,74 + 51556,5 + 129737,94) \cdot 0,16 = 115446,2 \text{ руб.}$$

Накладные расходы для исполнения 3 составили:

$$Z_{\text{накл}} = (5128 + 97000 + 29851,6 + 374262 + 4477,74 + 56139,3 + 140348,65) \cdot 0,16$$

$$= 113153,16 \text{ руб.}$$

### 6.5.7 Формирование бюджета затрат научно-исследовательского проекта

Определение бюджета затрат на научно–исследовательский проект приведено в таблице 15.

Таблица 15 – Расчет бюджета затрат НТИ

Наименование статьи	Сумма, руб.			Примечание
	Исп.1	Исп.2	Исп.3	
1. Материальные затраты НТИ	5012,5	5205	5128	Пункт 4.5.1
2. Затраты на специальное оборудование для научных (экспериментальных) работ	157000	157000	97000	Пункт 4.5.2
3. Затраты по основной заработной плате исполнителей темы	349438,5	343710	374262	Пункт 4.5.3
4. Затраты по дополнительной заработной плате исполнителей темы	52415,77	51556,5	56139,3	Пункт 4.5.4
5. Отчисления во внебюджетные фонды	130431,51	129737,94	140348,65	Пункт 4.5.5
6. Затраты на научные и производственные командировки	-	-	-	Отсутствуют
7. Контрагентские расходы	-	-	-	Отсутствуют
8. Накладные расходы	115893,83	115446,2	113153,16	Пункт 4.5.6
9. Бюджет затрат НТИ	810192,11	802655,64	786031,11	

### 6.6 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования.

Интегральный показатель финансовой эффективности научного исследования определяется как:

$$I_{\text{фин.р}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (20)$$

где  $\Phi_{pi}$  – стоимость  $i$ -го варианта исполнения;

$\Phi_{\text{max}}$  – максимальная стоимость исполнения научно-исследовательского проекта.

$$I_{\text{фин.р}}^{\text{исп1}} = \frac{115893,83}{115893,83} = 1$$

$$I_{\text{фин.р}}^{\text{исп2}} = \frac{115446,2}{115893,83} = 0,996$$

$$I_{\text{фин.р}}^{\text{исп3}} = \frac{113153,16}{115893,83} = 0,976$$

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum_{i=1}^n a_i \times b_i, \quad (21)$$

где  $a_i$  – весовой коэффициент  $i$ -го варианта исполнения разработки;

$b_i^a, b_i^p$  – бальная оценка  $i$ -го варианта исполнения разработки;

$n$  – число параметров сравнения.

Таблица 16 – Сравнительная оценка характеристик вариантов исполнения проекта

Объект исследования Критерии	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
1. Способствует снижению стоимости разработки	0,1	3	4	5
2. Удобство в эксплуатации	0,2	5	5	3
3. Порог входа	0,15	5	5	5
4. Безопасность	0,15	5	5	5
5. Надежность	0,1	5	5	5
6. Производительность	0,2	5	5	3

7. Доступность	0,1	4	4	4
Итого	1	4,7	4,8	4,1

$$I_{p-исп1} = 0,1 \cdot 3 + 0,2 \cdot 5 + 0,15 \cdot 5 + 0,15 \cdot 5 + 0,1 \cdot 5 + 0,2 \cdot 5 + 0,1 \cdot 4 = 4,7;$$

$$I_{p-исп2} = 0,1 \cdot 4 + 0,2 \cdot 5 + 0,15 \cdot 5 + 0,15 \cdot 5 + 0,1 \cdot 5 + 0,2 \cdot 5 + 0,1 \cdot 4 = 4,8;$$

$$I_{p-исп3} = 0,1 \cdot 5 + 0,2 \cdot 3 + 0,15 \cdot 5 + 0,15 \cdot 5 + 0,1 \cdot 5 + 0,2 \cdot 3 + 0,1 \cdot 4 = 4,1.$$

Интегральный показатель эффективности вариантов исполнения разработки ( $I_{испi}$ ) определяется по следующей формуле:

$$I_{исп1} = \frac{I_{p-исп1}}{I_{фин.р}} = \frac{4,7}{1} = 4,7$$

$$I_{исп2} = \frac{I_{p-исп2}}{I_{фин.р}} = \frac{4,8}{0,996} = 4,81$$

$$I_{исп3} = \frac{I_{p-исп3}}{I_{фин.р}} = \frac{4,1}{0,976} = 4,2$$

Сравнительная эффективность проекта ( $\mathcal{E}_{ср}$ ):

$$\mathcal{E}_{ср} = \frac{I_{исп1}}{I_{исп2}} \quad (22)$$

Таблица 17 – Сравнительная эффективность разработки

№	Показатели	Исп. 1	Исп. 2	Исп. 3
1	Интегральный финансовый показатель разработки	1	0,996	0,976
2	Интегральный показатель ресурсоэффективности разработки	4,7	4,8	4,1
3	Интегральный показатель эффективности	4,7	4,81	4,2
4	Сравнительная эффективность вариантов исполнения	0,97	1	0,87

Сравнив значения интегральных показателей эффективности, можно сделать вывод, что реализация технологии во 2-м исполнении, то есть посредством использования технологий Python и GNU Octave, по сравнению с Java, Matlab (Исп. 1) и C+, Matlab (Исп. 3), является более эффективным вариантом решения задачи, поставленной в данной работе с позиции финансовой и ресурсной эффективности.

## 7 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

### Введение

Согласно Парижскому соглашению, которое является международным договором об изменении климата, улавливание и хранение углерода является одной из важнейших технологий, необходимых для достижения нулевых выбросов и достижения климатических целей, изложенных в соглашении. Сейчас наше общество как никогда сталкивается с проблемой глобального потепления и выбросов CO<sub>2</sub> в атмосферу, ледники тают с каждым годом, уровень мирового океана повышается, животные сталкиваются с проблемой выживания в районах вечной мерзлоты. Одним из способов смягчения последствий глобального потепления является подземное хранение CO<sub>2</sub>.

Целью данной работы является разработка веб- и десктопного приложений для долгосрочного хранения CO<sub>2</sub> с использованием методов машинного обучения, которые позволят большему числу людей получить доступ к программному обеспечению, необходимому для изучения проблемы подземного захоронения углекислого газа, а также «среду» для применения методов машинного обучения для решения проблем нефтегазовой отрасли.

Разрабатываемые веб- и десктопное приложения могут быть использованы как учёными, так и специалистами-нефтяниками, заинтересованными проблемой подземного захоронения CO<sub>2</sub>.

Местом выполнения работ по проектированию и разработке веб- и десктопного приложений является офисное пространство жилого помещения, расположенное в городе Томске. Офисное пространство имеет площадь 21 м<sup>2</sup> и оборудовано персональным компьютером для осуществления проектирования и разработки программного обеспечения.

## **7.1 Правовые и организационные вопросы обеспечения безопасности**

### **7.1.1 Специальные правовые нормы трудового законодательства**

В процессе разработки программного обеспечения для долгосрочного хранения углекислого газа необходимо учитывать правовые нормы трудового законодательства. Целями трудового законодательства являются установление государственных гарантий трудовых прав и свобод граждан, создание благоприятных условий труда, защита прав и интересов работников и работодателей [15].

В соответствии с Трудовым кодексом РФ 197-ФЗ:

1. Продолжительность рабочего времени не может превышать 40 часов в неделю.
2. Предоставление перерывов для отдыха и питания продолжительностью не более двух часов и не менее 30 минут.
3. Защита персональных данных работника от неправомерного их использования или утраты должна быть обеспечена работодателем за счет его средств.
4. Выплата заработной платы не реже чем каждые полмесяца.
5. Работодатель обязан возмещать вред, причиненный работникам в связи с исполнением ими трудовых обязанностей.

Каждый сотрудник, при устройстве на работу, должен пройти вводный инструктаж специалистом по охране труда и первичный инструктаж непосредственным руководителем.

### **7.1.2 Основные эргономические требования к правильному расположению и компоновке рабочей зоны**

Согласно ГОСТ 12.2.032–78 ССБТ [4], устанавливающим общие эргономические требования к рабочим местам при выполнении работ в

положении сидя, СП 2.2.3670–20 [12] и СП 52.13330.2016 [13] рабочее место для создания программного обеспечения должно соответствовать следующим требованиям:

1. При работе двумя руками органы управления размещают с таким расчетом, чтобы не было перекрещивания рук.

2. Очень часто используемые средства отображения информации, требующие точного и быстрого считывания показаний, следует располагать в вертикальной плоскости под углом  $\pm 15^\circ$  от нормальной линии взгляда и в горизонтальной плоскости под углом  $\pm 15^\circ$  от сагиттальной плоскости.

3. Конструкцией рабочего места должно быть обеспечено выполнение трудовых операций в пределах зоны досягаемости моторного поля.

4. Оснащение светопроницаемых конструкций и оконных проёмов должно позволять регулировать параметры световой среды в помещении.

5. Персональные компьютеры следует размещать таким образом, чтобы показатели освещенности не превышали установленных гигиенических нормативов утвержденных в соответствии с пунктом 2 статьи 38 Федерального закона от 30.03.1999 N 52-ФЗ "О санитарно-эпидемиологическом благополучии населения".

6. Помещения с постоянным пребыванием людей должны иметь естественное освещение.

7. Не допускается применение для освещения ламп накаливания общего назначения мощностью 100 Вт и более.

Рабочее помещение представлено офисным пространством жилого помещения площадью 21 м<sup>2</sup>, где на одно рабочее место приходится площадь 7 м<sup>2</sup>. В помещении имеется естественное и искусственное освещение. Эргономические параметры рабочего стола и стула соответствуют нормативным требованиям. При работе над созданием программного

обеспечения были учтены вышеупомянутые требования с целью предотвращения неблагоприятного влияния вредных факторов производственной среды и трудового процесса на здоровье человека.

## 7.2 Производственная безопасность

ГОСТ 12.0.003–2015 [1] устанавливает вредные и опасные факторы, которые могут воздействовать на сотрудника. В таблице 1 перечислены факторы, которые могут возникнуть при работах по проектированию, разработке и тестированию веб- и десктопного приложений долгосрочного хранения CO<sub>2</sub> с применением методов машинного обучения.

Таблица 18 – Возможные опасные и вредные производственные факторы на рабочем месте программиста

Факторы (ГОСТ 12.0.003–2015)	Нормативные документы
Недостаточная освещенность рабочей зоны	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95* (с Изменениями N 1, 2).
Отсутствие или недостаток естественного света	
Отклонение параметров микроклимата	СанПиН 1.2.3685–21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания
Умственное переутомление	Трудовой кодекс Российской Федерации от 30.12.2001 №197-ФЗ (ред. от 01.03.2022)
Повышенный уровень электромагнитных излучений	ГОСТ 12.1.019-2017 Система стандартов безопасности труда (ССБТ).
Электрический ток	Электробезопасность. Общие требования и номенклатура видов защиты (с Поправкой)

### 7.2.1 Недостаточная освещенность рабочей зоны

Плохое освещение, особенно при работе с ПЭВМ, связано с целым рядом неблагоприятных последствий для здоровья, как физических, так и психофизических, таких как напряжение глаз, головные боли, усталость, а также стресс и беспокойство в условиях напряженной рабочей среды.

Согласно пункту 251 СП 2.2.3670–20 [12], компьютеры должны быть размещены таким образом, чтобы показатели освещенности не превышали установленных в СанПиН 1.2.3685–21 [10] гигиенических нормативов, которые представлены в таблице 2.

Таблица 19 – Требования к освещению на рабочих местах

Показатель	Норма
Освещенность поверхности рабочего стола	500 лк
Освещенность экрана монитора	Не более 200 лк
Коэффициент пульсации освещенности	Не более 5%

Чтобы освещение в офисе соответствовало всем нормам, лучше всего его сделать многоуровневым. В этом случае лампы с широкими углами рассеивания будут обеспечивать общее освещение, а точечные светильники — подсветку отдельных рабочих зон. В офисах рекомендуется применять светильники с люминесцентными лампами.

### 7.2.2 Отсутствие или недостаток естественного света

Зрительная способность и комфорт имеют огромное значение с позиции безопасности труда. Недостаток солнечного света может вызвать неприятные побочные эффекты, такие как депрессия, вялость, болезни сердца или снижение производительности.

Работа за ПК относится к зрительным работам высокой точности для любого типа помещений. Помещения с ПК должны иметь естественное и искусственное освещение. При боковом освещении коэффициент естественной освещенности (КЕО) рабочего стола должен составлять 1,2 % [13].

Рабочий стол следует размещать таким образом, чтобы экран компьютера был ориентирован боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева. Освещение не должно создавать бликов на поверхности экрана.

### **7.2.3 Отклонение параметров микроклимата**

Температура, влажность и скорость движения воздуха в помещении составляют микроклимат. Микроклимат помещения, в котором человек находится долго, играет существенную роль в формировании иммунитета, работоспособности, способности к восстановлению организма.

При регулярной работе в условиях повышенной температуры могут наблюдаться функциональные расстройства со стороны нервной и сердечно-сосудистой систем, желудочно-кишечного тракта, почек, а работе в помещениях с пониженной температурой возможно развитие заболеваний периферической нервной и сердечно-сосудистых систем. Повышенная влажность способствует развитию заболеваний дыхательной и мочевыделительной системы, в то время как недостаточная влажность воздуха приводит к пересыханию слизистой оболочки носа, повышению вязкости крови и ухудшению общего самочувствия.

Работа программиста относится к категории Ia тяжести труда, поскольку работы выполняются сидя и сопровождаются незначительным физическим напряжением с интенсивностью энергозатрат до 139 Вт/час [10]. На рабочих местах пользователей персональных компьютеров должны обеспечиваться оптимальные параметры микроклимата в соответствии с СанПиН 1.2.3685–21 [10] для категории тяжести работ Ia (таблица 3).

Таблица 20 – Допустимые величины параметров микроклимата

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный	22–24	21-25	15-75	0,1
Тёплый	24,1–25,0	19,0–26,0		0,2

Для обеспечения перечисленных параметров требуется в зависимости от погодных условий обеспечить либо кондиционирование воздуха, либо отопление, либо естественную вентиляцию. Фактические значения параметров микроклимата на рабочем месте соответствуют допустимым значениям.

#### **7.2.4 Умственное переутомление**

Работа программиста требует обработки большого количества информации, непрерывного повышения профессиональных компетенций и умения быстро обучаться. Это является сложной задачей, которая очень сильно сказывается на психофизическом состоянии человека, качестве сна, эмоциональном состоянии.

Согласно СанПиН 1.2.3685–21 [10], длительность сосредоточенного наблюдения должна составлять от 26 до 50% от времени смены, то есть не более 4-х часов при 8-и часовом рабочем дне.

Для снижения умственного перенапряжения необходимо проводить физические упражнения, медитации и консультации со специалистами как во время рабочих перерывов, так и после окончания рабочего дня. Данные меры позволяют увеличить производительность труда и предотвратить появление профессиональных болезней.

#### **7.2.5 Повышенный уровень электромагнитных излучений**

Работающий компьютер является основным источником электромагнитного излучения на рабочем месте программиста. Рассматривая вредное воздействие от монитора, следует учитывать, что боковые стороны и

задняя поверхность монитора генерируют гораздо больше вредного излучения, так как лицевая сторона монитора защищена специальным покрытием.

Излучение от компьютера имеет ряд негативных воздействий на организм человека: возрастает риск патологических заболеваний, нарушается общий гормональный фон организма, увеличивается потенциальная возможность депрессивных состояний, нарушения работы высшей нервной деятельности.

Согласно ГОСТ 12.1.045–84 [3], допустимые уровни напряженности электростатических полей устанавливаются в зависимости от времени пребывания персонала на рабочих местах, а при напряженности электростатических полей менее 20 кВ/м время пребывания в электростатических полях не регламентируется.

На производстве в качестве средств защиты от облучений применяются поглощающие (защитные) экраны. При работе в офисе их использование не целесообразно, так как излучение крайне маловероятно может превышать установленную норму в 20 кВ/м. Учитывая исходящую от ЭВМ опасность, следует соблюдать соответствующие меры безопасности при работе с ними:

1. Необходимо находиться на безопасном расстоянии во время длительных работ с ЭВМ, оно составляет 30 см.
2. Важно сокращать время подвержения излучению.
3. Необходимо отключать ЭВМ после завершения рабочего дня.

### **7.2.6 Электрический ток**

Так как программист имеет дело с электрооборудованием, то вопросам электробезопасности на его рабочем месте должно уделяться особое внимание. Электрический ток, проходящий через тело человека, может привести к внутренним повреждениям, остановке сердца или другим травмам.

Требования к защите от поражения электрическим током рассмотрены в ГОСТ 12.1.019–2017 [2].

Разработка веб- и десктопного приложений проводилась в офисном пространстве жилого помещения, которое согласно ПУЭ [9] относится к помещениям без повышенной опасности, так как отсутствуют следующие факторы: сырость, токопроводящая пыль, токопроводящие полы, высокая температура, возможность прикосновения человека к механизмам и металлическим корпусам электрооборудования.

Для защиты от прямого прикосновения необходимо применять [2]:

1. Безопасное расположение токоведущих частей, размещение их вне зоны досягаемости частями тела, конечностями.
2. Основная изоляция.
3. Защитное заземление.
4. Обеспечение надежными и быстродействующими автоматами и устройствами защитного отключения.
5. Электрозащитные средства и другие средства индивидуальной защиты.

Программисты относятся к неэлектротехническому персоналу, выполняющему работы, при которых может возникнуть опасность поражения электрическим током, следовательно им присваивается группа I по электробезопасности [7].

### **7.3 Экологическая безопасность**

При разработке веб- и десктопного приложений долгосрочного хранения CO<sub>2</sub> с использованием методов машинного обучения, источником загрязнения литосферы может стать неправильная утилизация комплектующих компьютеров, бумаги, иной техники. Косвенным источником загрязнения атмосферы является использование электричества, генерируемого тепловыми электростанциями.

Для обеспечения экологической безопасности необходимо соблюдать правила утилизации перечисленных выше отходов, переработка отходов электронной промышленности регулируется ГОСТ 30775–2001 [5].

Потребляемое компьютерами электричество в городе Томске вырабатывается тепловыми электростанциями, которые используют 3–4 вида топлива: природный газ, уголь (каменный и бурый), мазут и торф. Образующие в процессе эксплуатации тепловых электростанций дымовые газы являются основным источником загрязнения атмосферы. Для снижения косвенного влияния программиста на процесс загрязнения атмосферы необходимо стремиться к использованию электричества, получаемого из возобновляемых источников энергии, применять электросберегающие лампы и использовать электроприборы только при необходимости.

Персональный компьютер в рабочем помещении используется только с целью исследований и разработки нового программного обеспечения, следовательно рабочее помещение относится к IV категории объектов, оказывающих негативное воздействие на окружающую среду [6].

#### **7.4 Безопасность в чрезвычайных ситуациях**

Возможными чрезвычайными ситуациями на рабочем месте могут быть пожары, аварии коммунальных систем, взрывы, наводнения, ураганы, террористические акты, эпидемии, пандемии. Наиболее типичной чрезвычайной ситуацией является пожар, так как специфика работы заключается во взаимодействии с электронной вычислительной техникой, что может стать причиной возгорания и угрожать жизни людей.

Причинами возгорания при работе с вычислительной техникой могут быть:

1. Короткое замыкание.
2. Перегрев и дальнейшее воспламенение.
3. Неисправность работы техники.

Короткое замыкание может быть вызвано следующими причинами: старые или поврежденные электрические устройства, жидкости, контактирующие с электропроводкой, износ оболочки электрического кабеля и т. п.

Перегрев процессора является очень частым явлением при работе с ПЭВМ, что может спровоцировать дальнейшее воспламенение. Основными причинами перегрева компьютеров являются: плохое охлаждение, скопление пыли, заблокированные вентиляционные отверстия, вредоносные программы и большое количество процессов, запущенных одновременно.

Находясь на рабочем месте, для предупреждения возникновения пожара необходимо строго соблюдать следующие требования пожарной безопасности на рабочем месте, рекомендуемые главным управлением МЧС России по Томской области [8], а именно, необходимо:

1. Выключать все электрооборудование, когда оно не используется.
2. Регулярно проверять техническое состояние оборудования, в особенности кабелей. Плохая эксплуатация или загрязнение офисной техники может привести к пожару.
3. Соблюдать чистоту на рабочем месте. Это поможет потушить пожар на ранней стадии или предотвратит быстрое распространение пожара.
4. Проверить помещение на пожароопасность в конце рабочего дня.
5. Курить в специально отведенных для этого местах. Перед выбрасыванием окурки нужно тщательно затушить.

Так как рабочее помещение содержит твердые горючие и трудногорючие вещества и материалы (в том числе пыли и волокна), вещества и материалы, способные при взаимодействии с кислородом воздуха только гореть, то данное помещение является пожароопасным и относится к категории В [11]. Согласно классификации пожаров по виду горючего материала, установленной статьей 8 Федерального закона от 22.07.2008 N 123-

ФЗ [14], возможный пожар относится к классу Е, а именно пожару горючих веществ и материалов электроустановок, находящихся под напряжением.

Этаж и рабочее помещение оборудовано следующими средствами пожаротушения [14]: переносные огнетушители, покрывала для изоляции очага возгорания, пожарные краны и средства обеспечения их использования.

При появлении пожара, каждый гражданин, обязан:

1. Незамедлительно заявить о данном происшествии в пожарную службу по телефонному номеру 01 или 112.
2. Принять по возможности меры по эвакуации людей, тушению пожара и сохранению материальных ценностей.

### **Вывод по разделу**

В результате работы над данным разделом были выявлены опасные и вредные факторы, являющиеся потенциально опасными для программиста, работающего над созданием приложений для долгосрочного хранения CO<sub>2</sub> с использованием методов машинного обучения.

Возможные вредные и опасные факторы соответствуют нормативным значениям. Микроклиматические условия соблюдаются за счет регулярного проветривания помещения, отопления и искусственной вентиляции. Во время работы делаются периодические перерывы и минутные упражнения для глаз.

Рабочее помещение оборудовано в соответствии с требованиями электро- и пожарной безопасности. Работа программиста относится к категории тяжести труда Ia и требует 1 группы по электробезопасности.

Рабочее помещение относится к:

1. Помещениям без повышенной опасности в отношении опасности поражения людей электрическим током.
2. Категории В в отношении пожароопасности.
3. IV категории объектов, оказывающих негативное воздействие на окружающую среду.

## ЗАКЛЮЧЕНИЕ

– В данной работе был изучен модуль `co2lab MRST` (MATLAB Reservoir Simulation Toolbox). Был извлечен модуль для вычисления нелинейной физической модели, используя автоматическое дифференцирование, что является «ядром» для симуляции долгосрочного хранения  $\text{CO}_2$ . Полученный модуль был частично преобразован в Python, используя GNU Octave интерпретатор для выполнения непосредственно самих вычислений.

– Разработаны и применены алгоритмы обучения с подкреплением для оптимизации размещения скважин, чтобы предотвратить утечку газа и добиться максимального захоронения  $\text{CO}_2$  в структурных ловушках.

– Спроектированы и разработаны веб- и десктопное приложения для симуляции долгосрочного хранения  $\text{CO}_2$  с использованием методов машинного обучения. Веб-приложение размещено на облачной платформе Яндекс.Облако.

– Разработанные приложения имеют следующий функционал: изменение параметров симуляции, визуальное отображение результатов симуляции в виде графика инвентаризации захоронения  $\text{CO}_2$ , интеллектуальный поиск эффективной расстановки скважин, «интуитивный» поиск оптимальной расстановки скважин, сохранение результатов симуляции в .csv формате, отображение структурных ловушек на геологической модели.

– Созданные веб- и десктопное приложения имеют открытую лицензию и могут быть использованы как для изучения реалистичных сценариев закачки и миграции углекислого газа, так и функционировать в качестве платформы для быстрого прототипирования новых моделей и вычислительных методов, а также использоваться как «среда» для применения методов машинного обучения в нефтегазовой отрасли.

## СПИСОК ЛИТЕРАТУРЫ

1. Документация MRST [Электронный ресурс] // SINTEF Digital. – Режим доступа: <https://www.sintef.no/projectweb/mrst/documentation>, свободный. – Загл. с экрана (Дата обращения: 05.04.2022).
2. Изображение «Operations done by a neuron» [Электронный ресурс] // Towards Data Science, Arnx A.. – Режим доступа: <https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e0beaf>, свободный. – Загл. с экрана (Дата обращения: 05.04.2022).
3. Изображение «Overview of the solvers» [Электронный ресурс] // SINTEF Digital. – Режим доступа: [https://www.sintef.no/contentassets/2551f5f85547478590сеса14bc13ad51/\\_images/relations.png](https://www.sintef.no/contentassets/2551f5f85547478590сеса14bc13ad51/_images/relations.png), свободный. – Загл. с экрана (Дата обращения: 05.04.2022).
4. Ларман, К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку / К. Ларман. - М.: Вильямс, 2013. - 736 с.
5. Трусов, Б.Г. Программная инженерия [Текст]: учебник для вузов / под. ред. Б.Г. Трусова. – М.: Академия, 2017. – 72 с.
6. Geİron, A. Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2nd ed. [Текст] /A. Geİron. — М.: O’Reilly Media, 2019. – 1065 с.
7. Goldie, A. Placement Optimization with Deep Reinforcement Learning. [Текст] / A. Goldie, A. Mirhoseini // International Symposium on Physical Design. – 2020. — С. 3-7.
8. Islam, J. A holistic review on artificial intelligence techniques for well placement optimization problem. [Текст] / J. Islam, P. Vasant, B. Negash, M. Laruccia, M. Myint, J. Watada // Advances in Engineering Software. – 2020. — №141.

9. Jang, Il Sik. Well-placement optimisation using sequential artificial neural networks. [Текст] / Il S. Jang, S. Oh, Y. Kim, C. Park, H. Kang // Energy Exploration & Exploitation. – 2017. — №36.
10. Practical Reinforcement Learning [Электронный ресурс] // Coursera, Свечников П., Панин А. – Режим доступа: <https://www.coursera.org/learn/practical-rl>, свободный. – Загл. с экрана (Дата обращения: 05.12.2021).
11. Reinforcement learning [Электронный ресурс] // Wikipedia. – Режим доступа: [https://en.wikipedia.org/wiki/Reinforcement\\_learning](https://en.wikipedia.org/wiki/Reinforcement_learning), свободный. – Загл. с экрана (Дата обращения: 05.04.2022).
12. Reservoir Simulation for CO2 Studies [Электронный ресурс] // Schlumberger. – Режим доступа: <https://www.software.slb.com/blog/reservoir-simulation-for-co2-storage-studies>, свободный. – Загл. с экрана (Дата обращения: 10.05.2022).
13. Документация Dash [Электронный ресурс] // Plotly. – Режим доступа: <https://dash.plotly.com>, свободный. – Загл. с экрана (Дата обращения: 11.04.2022).
14. Описание GNU Octave [Электронный ресурс] // John W. Eaton. – Режим доступа: <https://www.gnu.org/software/octave/index>, свободный. – Загл. с экрана (Дата обращения: 11.04.2022).
15. Преимущества использования Dash [Электронный ресурс] // В.В. Костерин – Режим доступа: <http://chel-center.ru/python-yfc/2019/11/06/dash-nbsp-mdash-peredovoj-web-8209-frejmwork-python/>, свободный. – Загл. с экрана (Дата обращения: 11.04.2022).
16. Документация Яндекс.Облако [Электронный ресурс] // Yandex. – Режим доступа: <https://cloud.yandex.ru/docs>, свободный. – Загл. с экрана (Дата обращения: 11.04.2022).

17. ГОСТ 12.0.003–2015 «Опасные и вредные производственные факторы. Классификация» [Текст] – Взамен ГОСТ 12.0.003–74; введ. 01-03-2017. – Москва: Стандартинформ, 2016. – 10 с.

18. ГОСТ 12.1.019–2017 Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты (с Поправкой) [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/1200161238>, свободный. – Загл. с экрана (Дата обращения: 03.05.2022).

19. ГОСТ 12.1.045–84 Система стандартов безопасности труда (ССБТ). Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/1200161238>, свободный. – Загл. с экрана (Дата обращения: 21.05.2022).

20. ГОСТ 12.2.032–78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/1200003913>, свободный. – Загл. с экрана (Дата обращения: 03.05.2022).

21. ГОСТ 30775–2001 Ресурсосбережение. Обращение с отходами. Классификация, идентификация и кодирование отходов. Основные положения [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/1200028877>, свободный. – Загл. с экрана (Дата обращения: 03.05.2022).

22. Об утверждении критериев отнесения объектов, оказывающих негативное воздействие на окружающую среду, к объектам I, II, III и IV категорий (с изменениями на 7 октября 2021 года) [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/573292854>, свободный. – Загл. с экрана (Дата обращения: 14.05.2022).

23. Об утверждении Правил по охране труда при эксплуатации электроустановок [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/573264184>, свободный. – Загл. с экрана (Дата обращения: 14.05.2022).

24. Памятка по пожарной безопасности на рабочем месте [Электронный ресурс] // МЧС России. Главное управление по Томской области. – Режим доступа: [https://static.mchs.gov.ru/upload/site72/document\\_file/ТІКWZadUte.pdf](https://static.mchs.gov.ru/upload/site72/document_file/ТІКWZadUte.pdf), свободный. – Загл. с экрана (Дата обращения: 05.05.2022).

25. Правила устройства электроустановок (ПУЭ). Глава 1.1 Общая часть (Издание седьмое) [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/1200030216>, свободный. – Загл. с экрана (Дата обращения: 14.05.2022).

26. СанПиН 1.2.3685–21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/573500115>, свободный. – Загл. с экрана (Дата обращения: 03.05.2022).

27. СП 12.13130.2009 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности (с Изменением N 1) [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/1200071156>, свободный. – Загл. с экрана (Дата обращения: 14.05.2022)

28. СП 2.2.3670–20 Санитарно-эпидемиологические требования к условиям труда [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/573230583>, свободный. – Загл. с экрана (Дата обращения: 03.05.2022).

29. СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95\* (с Изменениями N 1, 2) [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/456054197>, свободный. – Загл. с экрана (Дата обращения: 03.05.2022).

30. Технический регламент о требованиях пожарной безопасности (с изменениями на 30 апреля 2021 года) [Электронный ресурс] // Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/902111644>, свободный. – Загл. с экрана (Дата обращения: 14.05.2022).

31. Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ (ред. от 01.03.2022) // Собрание законодательства РФ. - 07.01.2002. - № 1 (ч. 1). - Ст. 3.

## ПРИЛОЖЕНИЕ А. КОД DDL

*Коллекция «formations\_data»:*

```
db.createCollection('formations_data', {
  validator: {
    $jsonSchema: {
      bsonType: 'object',
      title: 'formations_data',
      required: ['formation_id', 'coarsening', 'depths', 'faces',
'faces_trapping', 'vertices', 'formations'],
      properties: {
        formation_id: {
          bsonType: 'double'
        },
        coarsening: {
          bsonType: 'double'
        },
        depths: {
          bsonType: 'array',
          items: {}
        },
        faces: {
          bsonType: 'array',
          items: {}
        },
        faces_trapping: {
          bsonType: 'array',
          items: {}
        },
        vertices: {
          bsonType: 'array',
          items: {}
        },
        formations: {
          bsonType: 'double'
        }
      }
    }
  });
db.formations_data.createIndex({
  "_id": 1
})
```

*Коллекция «formations»:*

```
db.createCollection('formations', {
  validator: {
    $jsonSchema: {
      bsonType: 'object',
      title: 'formations',
      required: ['formation'],
      properties: {
        formation: {
          bsonType: 'string'
        }
      }
    }
  }
});
db.formations.createIndex({
  "_id": 1
})
```

*Коллекция «results»:*

```
db.createCollection('results', {
  validator: {
    $jsonSchema: {
      bsonType: 'object',
      title: 'results',
      required: ['formation_id', 'well_location', 'simulation_parameters',
        'result'],
      properties: {
        formation_id: {
          bsonType: 'double'
        },
        well_location: {
          bsonType: 'array',
          items: {}
        },
        simulation_parameters: {
          bsonType: 'object',
          title: 'results.simulation_parameters',
          required: ['rho_cref', 'grid_coarsening', 'seafloor_depth',
            'seafloor_temp', 'temp_gradient', 'water_density', 'dis_max',
            'max_num_wells', 'default_rate', 'max_rate', 'water_compr_val', 'pv_mult',
```

```
'water_residual', 'co2_residual', 'inj_time', 'inj_steps', 'mig_time',  
'mig_steps', 'well_radius', 'c', 'outside_distance', 'use_dissolution',  
'use_trapping', 'use_cap_fringe'],
```

```
  properties: {  
    rho_cref: {  
      bsonType: 'double'  
    },  
    grid_coarsening: {  
      bsonType: 'double'  
    },  
    seafloor_depth: {  
      bsonType: 'double'  
    },  
    seafloor_temp: {  
      bsonType: 'double'  
    },  
    temp_gradient: {  
      bsonType: 'double'  
    },  
    water_density: {  
      bsonType: 'double'  
    },  
    dis_max: {  
      bsonType: 'double'  
    },  
    max_num_wells: {  
      bsonType: 'double'  
    },  
    default_rate: {  
      bsonType: 'double'  
    },  
    max_rate: {  
      bsonType: 'double'  
    },  
    water_compr_val: {  
      bsonType: 'double'  
    },  
    pv_mult: {  
      bsonType: 'double'  
    },  
    water_residual: {  
      bsonType: 'double'
```

```

    },
    co2_residual: {
      bsonType: 'double'
    },
    inj_time: {
      bsonType: 'double'
    },
    inj_steps: {
      bsonType: 'double'
    },
    mig_time: {
      bsonType: 'double'
    },
    mig_steps: {
      bsonType: 'double'
    },
    well_radius: {
      bsonType: 'double'
    },
    c: {
      bsonType: 'double'
    },
    outside_distance: {
      bsonType: 'double'
    },
    use_dissolution: {
      bsonType: 'bool'
    },
    use_trapping: {
      bsonType: 'bool'
    },
    use_cap_fringe: {
      bsonType: 'bool'
    }
  }
},
result: {
  bsonType: 'object',
  title: 'results.result',
  required: ['masses', 'time'],
  properties: {
    masses: {

```

```

        bsonType: 'array',
        items: {}
    },
    time: {
        bsonType: 'array',
        items: {}
    }
}
}
}
}
}
});
db.results.createIndex({
    "_id": 1
})

```

*Коллекция «models»:*

```

db.createCollection('models', {
    validator: {
        $jsonSchema: {
            bsonType: 'object',
            title: 'models',
            required: ['model', 'weights'],
            properties: {
                model: {
                    bsonType: 'object',
                    title: 'models.model',
                    required: ['class_name', 'config', 'keras_version', 'backend'],
                    properties: {
                        class_name: {
                            bsonType: 'string'
                        },
                    },
                    config: {
                        bsonType: 'object',
                        title: 'models.model.config',
                        required: ['name', 'layers'],
                        properties: {
                            name: {
                                bsonType: 'string'
                            },
                        },
                        layers: {

```

```

bsonType: 'array',
items: {
  title: 'models.model.config.layers',
  required: ['class_name', 'config'],
  properties: {
    class_name: {
      bsonType: 'string'
    },
  },
  config: {
    bsonType: 'object',
    title: 'models.model.config.layers.config',
    required: ['dtype', 'sparse', 'ragged', 'name', 'trainable',
'units', 'activation', 'use_bias', 'kernel_initializer', 'bias_initializer'],
    properties: {
      batch_input_shape: {
        bsonType: 'array',
        items: {
          bsonType: ["null", "double"]
        }
      },
      dtype: {
        bsonType: 'string'
      },
      sparse: {
        bsonType: 'bool'
      },
      ragged: {
        bsonType: 'bool'
      },
      name: {
        bsonType: 'string'
      },
      trainable: {
        bsonType: 'bool'
      },
      units: {
        bsonType: 'double'
      },
      activation: {
        bsonType: 'string'
      },
      use_bias: {

```

```

        bsonType: 'bool'
    },
    kernel_initializer: {
        bsonType: 'object',
        title:
'models.model.config.layers.config.kernel_initializer',
        required: ['class_name', 'config'],
        properties: {
            class_name: {
                bsonType: 'string'
            },
            config: {
                bsonType: 'object',
                title:
'models.model.config.layers.config.kernel_initializer.config',
                properties: {
                    seed: {
                        bsonType: 'null'
                    }
                }
            }
        }
    },
    bias_initializer: {
        bsonType: 'object',
        title: 'models.model.config.layers.config.bias_initializer',
        required: ['class_name', 'config'],
        properties: {
            class_name: {
                bsonType: 'string'
            },
            config: {
                bsonType: 'object',
                title:
'models.model.config.layers.config.bias_initializer.config',
            }
        }
    },
    kernel_regularizer: {
        bsonType: 'null'
    },
    bias_regularizer: {

```

```

        bsonType: 'null'
      },
      activity_regularizer: {
        bsonType: 'null'
      },
      kernel_constraint: {
        bsonType: 'null'
      },
      bias_constraint: {
        bsonType: 'null'
      }
    }
  }
}
},
keras_version: {
  bsonType: 'string'
},
backend: {
  bsonType: 'string'
}
},
weights: {
  bsonType: 'array',
  items: {}
}
}
});
db.models.createIndex({
  "_id": 1
})

```

*Коллекция «metrics»:*

```

db.createCollection('metrics', {
  validator: {
    $jsonSchema: {

```

```

bsonType: 'object',
title: 'metrics',
required: ['iteration_rewards', 'mean_reward', 'sum_reward'],
properties: {
  iteration_rewards: {
    bsonType: 'array',
    items: {}
  },
  mean_reward: {
    bsonType: 'array',
    items: {
      bsonType: 'double'
    }
  },
  sum_reward: {
    bsonType: 'array',
    items: {
      bsonType: 'double'
    }
  }
}
});
db.metrics.createIndex({
  "_id": 1
})

```

## ПРИЛОЖЕНИЕ Б. ИСХОДНЫЙ КОД ОСНОВНОГО МОДУЛЯ ВЕБ-ПРИЛОЖЕНИЯ

```
from typing import Optional, List, Dict
from copy import deepcopy

import dash
from dash import dcc, html
import dash_bootstrap_components as dbc
from dash.dependencies import Input, Output, State
from dash.exceptions import PreventUpdate
import plotly.graph_objects as go
import numpy as np
import threading
import pandas as pd

from python.db_client.mongo_client import MongoClient
from python.web.plotting.plot_trapping_distribution_web import LABELS
from python.web.reinforcement_learning.basic_policy_web import
run_basic_policy_web
    from python.web.reinforcement_learning.nn_policy_web import
run_nn_policy_web

    from python.web.plotting.plot_formation_web import plot_formation_web
    from python.web.plotting.plot_trapping_distribution_web import
plot_trapping_distribution

    from python.web.simulation.explore_simulation import
explore_simulation, YEAR

FORMATIONS = [
    'Arefm', 'Bjarmelandfm', 'Brentgrp', 'Brynefm', 'Fensfjordfm', 'Garnfm',
    'Gassumfm', 'Ilefsm',
    'Johansenfm', 'Krossfjordfm', 'Nordmelafm', 'Pliocenesand', 'Sandnesfm',
    'Skadefm', 'Sognefjordfm',
    'Statfjordfm', 'Stofm', 'Tiljefm', 'Tubaenfm', 'Ulafm', 'Utsirafm'
]

OPTIONS = [
    {'label': formation, 'value': formation}
    for formation in FORMATIONS
]
```

```

]
DEFAULT_FORMATION = FORMATIONS[16]
COLORS = ['primary', 'secondary']

MONGO_CLIENT = MongoClient('co2sim')

app = dash.Dash(external_stylesheets=[dbc.themes.MINTY])
app.title = 'CO2 storage simulator'
server = app.server

previous_formation_graph = {}
formation_graph = {}

previous_trapping_masses = np.array([])
trapping_masses = np.array([])
trapping_time = np.array([])

run_smart_well_location = 0
run_basic_well_location = 0

stop_smart_well_location = [True]
stop_basic_well_location = [True]

count = 0
initial_call = True

current_well_loc = (0, 0)

app.layout = dbc.Container(
    [
        dbc.Row(
            [
                dbc.Col(
                    [
                        html.H3([
                            'CO',
                            html.Sub(2),
                            ' storage simulator'
                        ]),
                        html.Span(
                            'Choose a formation from the list below:'
                        )
                    ]
                )
            ]
        )
    ]
)

```

```

        dcc.DropDown(
            id='formation_dropdown',
            options=OPTIONS,
            value=DEFAULT_FORMATION,
            style={
                'width': '175px',
            }
        ),
    ],
),
dbc.Col(
    [
        html.H5('Running options'),
        dbc.Button(
            'Simulation',
            id='simulation',
            color='info',
            n_clicks_timestamp='0',
        ),
        dbc.Button(
            'Smart well location',
            id='smart_well_location',
            color=COLORS[1],
            n_clicks_timestamp='0',
        ),
        dbc.Button(
            'Basic well location',
            id='basic_well_location',
            color=COLORS[1],
            n_clicks_timestamp='0',
        ),
    ],
    width=4
)
],
justify='between',
align='end'
),
html.Hr(),
dbc.Row(
    [
        dbc.Col(

```

```

dbc.Spinner(
  dcc.Graph(
    id='formation_graph',
    style={'height': '80vh'},
  ),
  color='primary',
),
width=8,
),
dbc.Col(
  [
    html.Div(
      [
        html.H5('Initial parameters'),
        html.P('Injection rate:'),
        dcc.Slider(
          id='injection_rate',
          min=1,
          max=10,
          step=1,
          value=1,
        ),
        dbc.Row(
          [
            dbc.Col(
              [
                dbc.Label('Injection period', size='md'),
                dcc.Input(
                  id='injection_period',
                  value=10,
                  type='number'
                ),
                dbc.Label('Injection time steps', size='md'),
                dcc.Input(
                  id='injection_time_steps',
                  value=5,
                  type='number'
                ),
                dbc.Label('Migration period', size='md'),
                dcc.Input(
                  id='migration_period',
                  value=10,

```

```

        type='number'
    ),
    dbc.Label('Migration time steps', size='md'),
    dcc.Input(
        id='migration_time_steps',
        value=5,
        type='number'
    ),
    dcc.Checklist(
        id='traps_checkbox',
        options={
            'traps': 'Show traps',
        }
    )
],
width=6
),
dbc.Col(
[
    dbc.Label('Seafloor depth', size='md'),
    dcc.Input(
        id='seafloor_depth',
        value=100,
        type='number'
    ),
    dbc.Label('Seafloor temperature', size='md'),
    dcc.Input(
        id='seafloor_temperature',
        value=7,
        type='number'
    ),
    dbc.Label('Water residual', size='md'),
    dcc.Input(
        id='water_residual',
        value=0.11,
        type='number'
    ),
    dbc.Label('CO2 residual', size='md'),
    dcc.Input(
        id='co2_residual',
        value=0.21,
        type='number'

```

```

    ),
    dcc.Checklist(
        id='download_checkbox',
        options={
            'download': 'Download results',
        }
    )
],
width=6
),
]
)
]
),
html.Hr(),
html.Div(
    [
        html.H5('Trapping inventory'),
        dbc.Spinner(
            dcc.Graph(id='trapping_graph'),
            color='primary'
        )
    ]
),
html.Div(
    [
        dcc.Interval(
            id='formation_updater',
            interval=3 * 1000,
            n_intervals=0,
            disabled=True
        ),
        dcc.Interval(
            id='trapping_graph_updater',
            interval=5 * 1000,
            n_intervals=0,
            disabled=True
        ),
        dcc.Store(id='local_formation'),
        dcc.Store(id='local_trapping'),
        dcc.Download(id='download_simulation_results')
    ]
)

```

```

        )
    ],
    width=4,
),
],
),
html.Hr(),
],
fluid=True,
)

```

```

@app.callback(
[
    Output('trapping_graph', 'figure'),
    Output('download_simulation_results', 'data'),
    Output('formation_updater', 'disabled'),
    Output('trapping_graph_updater', 'disabled'),
    Output('simulation', 'n_clicks_timestamp'),
    Output('smart_well_location', 'color'),
    Output('smart_well_location', 'n_clicks_timestamp'),
    Output('basic_well_location', 'color'),
    Output('basic_well_location', 'n_clicks_timestamp'),
    Output('local_trapping', 'clear_data')
],
[
    Input('simulation', 'n_clicks_timestamp'),
    Input('smart_well_location', 'n_clicks_timestamp'),
    Input('basic_well_location', 'n_clicks_timestamp'),
    Input('local_trapping', 'data')
],
[
    State('formation_dropdown', 'value'),
    State('injection_rate', 'value'),
    State('injection_period', 'value'),
    State('injection_time_steps', 'value'),
    State('migration_period', 'value'),
    State('migration_time_steps', 'value'),
    State('seafloor_depth', 'value'),
    State('seafloor_temperature', 'value'),
    State('water_residual', 'value'),
    State('co2_residual', 'value'),

```

```

        State('download_checkbox', 'value')
    ],
    prevent_initial_call=True
)
def run_simulation(
    simulation: float,
    smart_well_location: float,
    basic_well_location: float,
    figure_dict: Dict[str, any],
    formation: str,
    injection_rate: int,
    injection_period: int,
    injection_time_steps: int,
    migration_period: int,
    migration_time_steps: int,
    seafloor_depth: float,
    seafloor_temperature: float,
    water_residual: float,
    co2_residual: float,
    download_checkbox: List[str]
) -> [go.Figure, bool]:
    button_pressed = np.argmax(
        np.array(
            [
                0,
                float(simulation),
                float(smart_well_location),
                float(basic_well_location)
            ]
        )
    )
)

if button_pressed == 1:
    masses, time = explore_simulation(
        current_well_loc,
        formation=formation,
        default_rate=injection_rate,
        inj_time=injection_period * YEAR,
        inj_steps=injection_time_steps,
        mig_time=migration_period * YEAR,
        mig_steps=migration_time_steps,
        seafloor_depth=seafloor_depth,

```

```

        seafloor_temp=seafloor_temperature,
        water_residual=water_residual,
        co2_residual=co2_residual,
        mongo_client=MONGO_CLIENT
    )
    output = plot_trapping_distribution(masses, time)

    download = dash.no_update
    if download_checkbox:
        download = _create_download_csv(masses, time)

    return (
        output,
        download,
        dash.no_update,
        dash.no_update,
        0,
        dash.no_update,
        dash.no_update,
        dash.no_update,
        dash.no_update,
        dash.no_update,
    )

elif button_pressed == 2:
    global run_smart_well_location
    global stop_smart_well_location

    run_smart_well_location = bool((run_smart_well_location + 1) % 2)

    if run_smart_well_location:
        trapping_graph = dash.no_update
        local_trapping = False

        reset_formation_graph()
        reset_trapping_graph()
        stop_smart_well_location.pop()

    smart_well_location_thread = threading.Thread(
        target=run_nn_policy_web,
        name='smart_well_location',
        kwargs={

```

```

        'formation': formation,
        'formation_graph_callback': set_formation_graph_callback,
        'trapping_graph_callback': set_trapping_graph_callback,
        'stop_smart_well_location': stop_smart_well_location,
        'default_rate': injection_rate,
        'inj_time': injection_period * YEAR,
        'inj_steps': injection_time_steps,
        'mig_time': migration_period * YEAR,
        'mig_steps': migration_time_steps,
        'seafloor_depth': seafloor_depth,
        'seafloor_temp': seafloor_temperature,
        'water_residual': water_residual,
        'co2_residual': co2_residual,
        'mongo_client': MONGO_CLIENT
    }
)
    smart_well_location_thread.start()
else:
    stop_smart_well_location.append(True)
    reset_formation_graph()
    reset_trapping_graph()
    trapping_graph = 'Empty graph'
    local_trapping = True

return (
    trapping_graph,
    dash.no_update,
    not run_smart_well_location,
    not run_smart_well_location,
    dash.no_update,
    COLORS[not run_smart_well_location],
    0,
    dash.no_update,
    dash.no_update,
    local_trapping
)

elif button_pressed == 3:
    global run_basic_well_location
    global stop_basic_well_location

run_basic_well_location = bool((run_basic_well_location + 1) % 2)

```

```

if run_basic_well_location:
    trapping_graph = dash.no_update
    local_trapping = False

    reset_formation_graph()
    reset_trapping_graph()
    stop_basic_well_location.pop()

    basic_well_location_thread = threading.Thread(
        target=run_basic_policy_web,
        name='basic_well_location',
        kwargs={
            'formation': formation,
            'formation_graph_callback': set_formation_graph_callback,
            'trapping_graph_callback': set_trapping_graph_callback,
            'stop_basic_well_location': stop_basic_well_location,
            'default_rate': injection_rate,
            'inj_time': injection_period * YEAR,
            'inj_steps': injection_time_steps,
            'mig_time': migration_period * YEAR,
            'mig_steps': migration_time_steps,
            'seafloor_depth': seafloor_depth,
            'seafloor_temp': seafloor_temperature,
            'water_residual': water_residual,
            'co2_residual': co2_residual
        }
    )
    basic_well_location_thread.start()
else:
    stop_basic_well_location.append(True)
    reset_formation_graph()
    reset_trapping_graph()
    trapping_graph = 'Empty graph'
    local_trapping = True

return (
    trapping_graph,
    dash.no_update,
    not run_basic_well_location,
    not run_basic_well_location,
    dash.no_update,

```

```

        dash.no_update,
        dash.no_update,
        COLORS[not_run_basic_well_location],
        0,
        local_trapping
    )

if figure_dict:
    return (
        go.Figure(**figure_dict),
        dash.no_update,
        dash.no_update,
        dash.no_update,
        dash.no_update,
        dash.no_update,
        dash.no_update,
        dash.no_update,
        dash.no_update,
        dash.no_update
    )
else:
    raise PreventUpdate

def _create_download_csv(
    masses: np.array,
    time: np.array
) -> Dict[str, any]:
    df = pd.DataFrame(masses, columns=time // YEAR, index=LABELS)
    return dcc.send_data_frame(df.to_csv, "masses.csv")

@app.callback(
    Output('formation_graph', 'figure'),
    [
        Input('formation_graph', 'clickData'),
        Input('formation_dropdown', 'value'),
        Input('local_formation', 'data'),
        Input('traps_checkbox', 'value')
    ],
    State('formation_graph', 'figure')
)

```

```

def _plot_formation_with_well(
    click_data: Optional[Dict[str, List[Dict[str, float]]]],
    formation: str,
    figure_dict: Dict[str, any],
    traps: List[str],
    current_figure: Dict[str, any]
) -> [go.Figure, bool]:
    ctx = dash.callback_context
    triggered_input = ctx.triggered

    use_trapping = True if traps else False

    global previous_formation_graph

    if triggered_input[0]['prop_id'] == 'local_formation.data' and figure_dict:
        return go.Figure(**figure_dict)

    elif triggered_input[0]['prop_id'] == 'formation_dropdown.value':
        previous_formation_graph = plot_formation_web(formation,
MONGO_CLIENT).to_dict()
        return plot_formation_web(formation, MONGO_CLIENT,
use_trapping=use_trapping)

    elif triggered_input[0]['prop_id'] == 'formation_graph.clickData':
        x = click_data['points'][0]['x']
        y = click_data['points'][0]['y']
        marker = (x, y)

        global current_well_loc
        current_well_loc = marker

        previous_formation_graph = plot_formation_web(formation,
MONGO_CLIENT, marker=marker).to_dict()
        return plot_formation_web(formation, MONGO_CLIENT,
marker=marker, use_trapping=use_trapping)

    elif triggered_input[0]['prop_id'] == 'traps_checkbox.value':
        if use_trapping:
            previous_formation_graph = current_figure
            return plot_formation_web(formation, MONGO_CLIENT,
use_trapping=use_trapping, current_figure=current_figure)
        else:

```

```

        return go.Figure(**previous_formation_graph)

    else:
        return plot_formation_web(formation, MONGO_CLIENT,
use_trapping=use_trapping)

@app.callback(
    Output('local_formation', 'data'),
    Input('formation_updater', 'n_intervals'),
    prevent_initial_call=True
)
def dynamic_figure_update(
    n_intervals: int
) -> Dict[str, any]:
    global previous_formation_graph
    formation_graph_with_colorbar = {}

    if formation_graph:
        formation_graph_with_colorbar = deepcopy(formation_graph)
        del formation_graph['data'][0] # delete first cell with colorbar

    if formation_graph != previous_formation_graph:
        previous_formation_graph = formation_graph
        return formation_graph_with_colorbar
    else:
        return dash.no_update

@app.callback(
    Output('local_trapping', 'data'),
    Input('trapping_graph_updater', 'n_intervals'),
    prevent_initial_call=True
)
def dynamic_trapping_graph_update(
    n_intervals: int
) -> Dict[str, any]:
    global previous_trapping_masses
    global count
    count += 1
    print(f'Attempt {count}')
    if not np.array_equal(trapping_masses, previous_trapping_masses):

```

```

        print(f'Attempt succeeded')
        previous_trapping_masses = trapping_masses
        return plot_trapping_distribution(trapping_masses,
trapping_time).to_dict()
    else:
        return dash.no_update

```

```

def reset_formation_graph() -> None:
    global previous_formation_graph
    global formation_graph
    previous_formation_graph = {}
    formation_graph = {}

```

```

def reset_trapping_graph() -> None:
    global previous_trapping_masses
    global trapping_masses
    previous_trapping_masses = []
    trapping_masses = []

```

```

def set_formation_graph_callback(
    fig: Dict[str, any]
) -> None:
    global formation_graph
    formation_graph = fig

```

```

def set_trapping_graph_callback(
    masses: np.array,
    time: np.array
) -> None:
    global trapping_masses
    global trapping_time
    trapping_masses = masses
    trapping_time = time

```

```

if __name__ == '__main__':
    app.run_server(debug=False, host='0.0.0.0', port=8080)

```