TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа информационных технологий и робототехники
Направление подготовки 09.03.04 «Программная инженерия»
Отделение информационных технологий

## БАКАЛАВРСКАЯ РАБОТА

| Тема работы |
|---|
| **Агрегирование и обработка текстовой информации с нефиксированной структурой** |

УДК 004.4'242:004.912:004.422.63

Студент

| Группа | ФИО | Подпись | Дата |
|---|---|---|---|
| 8К51 | Ванюшин Иван Сергеевич | | |

Руководитель

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР ТПУ | Фофанов Олег Борисович | к.т.н. | | |

## КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОСТН ШБИП ТПУ | Подопригора Игнат Валерьевич | к.э.н. | | |

По разделу «Социальная ответственность»

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ООД ШБИП ТПУ | Винокурова Галина Фёдоровна | к.т.н. | | |

## ДОПУСТИТЬ К ЗАЩИТЕ:

| Руководитель ООП | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР ТПУ | Чердынцев Евгений Сергеевич | к.т.н. | | |

Томск - 2019

**Планируемые результаты обучения по ООП**

| Код результатов | Результат обучения (выпускник должен быть готов) | Требования ФГОС, критерии АИОР |
|---|---|---|
| Р1 | Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности. | Требования ФГОС (ОК-1, 10, ПК-4, 5, 6), критерий 5 АИОР (п. 1.1) |
| Р2 | Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач. | Требования ФГОС (ОК-11, 12, 13, ПК-1, 2, 11), критерий 5 АИОР (п.1.1, 1.2) |
| Р3 | Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей. | Требования ФГОС (ОК-1, 8, ПК-2, 4, 6), критерий 5 АИОР (п. 1.2) |
| Р4 | Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования. | Требования ФГОС (ОК-2, 3, ПК-3, 4, 5), критерий 5 АИОР (п. 1.3) |

| | | |
|---|---|---|
| Р5 | 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 аппаратных и программных средств информационных и автоматизированных систем. | Требования ФГОС (ОК-6, ПК-6, 7), критерий 5 АИОР (п.1.4) |
| Р6 | Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды. | Требования ФГОС (ОК-4, 15, 16, ПК-9, 10, 11), критерий 5 АИОР (п. 1.5) |
| | Универсальные компетенции | |
| Р7 | Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности. | Требования ФГОС (ОК-1, 4, ПК-1, 6, 7), критерий 5 АИОР (п. 2.1) |
| Р8 | Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности. | Требования ФГОС (ОК-14, ПК-7), критерий 5 АИОР (п. 2.2) |

| | | |
|---|---|---|
| Р9 | Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации. | Требования ФГОС (ОК-2, 3, 4), критерий 5 АИОР (п. 2.3, 2.4) |
| Р10 | Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности. | Требования ФГОС (ОК-1, 5, 9), критерий 5 АИОР (п. 2.5) |
| Р11 | Демонстрировать способность к самостоятельной к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии. | Требования ФГОС (ОК-6, 7), критерий 5 АИОР (п. 2.6) |

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа информационных технологий и робототехники
Направление подготовки 09.03.01 «Информатика и вычислительная техника»
Отделение информационных технологий

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** |
|---|---|
| 8К51 | Ванюшину Ивану Сергеевичу |

Тема работы:

| «Агрегирование и обработка текстовой информации с нефиксированной структурой» | |
|---|---|
| Утверждена приказом директора (дата, номер) | 26.02.2019 №1513/с |

| Срок сдачи студентом выполненной работы: | |
|---|---|

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

| **Исходные данные к работе**<br><br>*(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).* | Работа направлена на проведение исследования, в ходе которого будет сформирован подход обработки и агрегирования текстовой информации с применением индуктивного обучения, а также разработаны низкоуровневые модули создания множества моделей |
|---|---|

| | |
|---|---|
| **Перечень подлежащих исследованию, проектированию и разработке вопросов**<br>*(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).* | Изучение существующих подходов в описанной задаче<br>Изучение подходов индуктивного обучения<br>Проектирование решения задачи тематического моделирования при помощи индуктивного обучения<br>Реализация низкоуровневой части спроектированного решения<br>Расчет ресурсоэффективности и ресурсосбережения<br>Анализ вредных производственных факторов |
| **Перечень графического материала**<br>*(с точным указанием обязательных чертежей)* | 1. Блок-схемы<br>2. Презентация |

**Консультанты по разделам выпускной квалификационной работы**
*(с указанием разделов)*

| Раздел | Консультант |
|---|---|
| Финансовый менеджмент, ресурсоэффективность и ресурсосбережение | Подопригора Игнат Валерьевич |
| Социальная ответственность | Винокурова Галина Фёдоровна |

| | |
|---|---|
| **Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику** | |

**Задание выдал руководитель:**

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР | Фофанов Олег Борисович | к.т.н., доцент | | |

**Задание принял к исполнению студент:**

| Группа | ФИО | Подпись | Дата |
|---|---|---|---|
| 8К51 | Ванюшин Иван Сергеевич | | |

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа информационных технологий и робототехники
Направление подготовки 09.03.04 «Программная инженерия»
Уровень образования: Бакалавр
Отделение информационных технологий
Период выполнения осенний / весенний семестр 2018 /2019 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) |
|---|---|---|
| 01.03.2019 | Раздел 1. Аналитический обзор задачи | 25 |
| 17.03.2019 | Раздел 2. Проектирование предлагаемого решения | 25 |
| 20.04.2019 | Раздел 3. Реализация решения | 10 |
| 28.04.2019 | Раздел 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение | 20 |
| 16.05.2019 | Раздел 5. Социальная ответственность | 20 |

**СОСТАВИЛ:**
**Руководитель ВКР**

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР ТПУ | Фофанов Олег Борисович | к.т.н., доцент | | |

**СОГЛАСОВАНО:**

**Руководитель ООП**

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ОИТ ИШИТР ТПУ | Чердынцев Евгений Сергеевич | к.т.н. | | |

## Реферат

Выпускная квалификационная работа 66 с., 6 рис., 14 табл., 8 источников, 2 прил.

Ключевые слова: метод группового учёта аргументов, group method of data handling, тематическое моделирование, анализ данных, агрегирование информации, анализ текстов, индуктивное моделирование.

Объектом исследования является агрегирование текстов методами индуктивного обучения.

Цель работы – исследование возможности агрегирования и анализа текстов при помощи решения задачи тематического моделирования через построение иерархической модели методами индуктивного обучения, а также реализация низкоуровневого компонента иерархической системы.

**Определения, обозначения, сокращения, нормативные ссылки**

В данной работе применены следующие термины с соответствующими определениями:

1) Метод группового учёта аргументов (МГУА) — семейство индуктивных алгоритмов для математического моделирования мультипараметрических данных.
2) Токен (англ. token) — объект, создающийся из лексемы в процессе лексического анализа («токенизации», от англ. tokenizing).
3) Машинное обучение (Machine Learning) — обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться.
4) PLSA или вероятностный латентно-семантический анализ (ВЛСА)— это статистический метод анализа корреляции двух типов данных.
5) Обработка естественного языка (Natural Language Processing, NLP) — общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков.
6) Распределение Дирихле — это семейство непрерывных многомерных вероятностных распределений параметризованных вектором α неотрицательных вещественных чисел.
7) Размещение патинко (англ. pachinko allocation, PAM) — метод тематического моделирования, применяемый в машинном обучении и обработке естественного языка, позволяющий обнаружить скрытую тематическую структуру в коллекции документов.

**Оглавление**

## Введение

На данный момент информационных поток, создаваемых текстами различного вида, невероятно велик. Ежедневно публикуется более 2 миллионов статей, каждая из которых имеет некоторую информационную ценность и может быть обработана для получения некоторой информации.

Существенная проблема заключается в том, что крайне трудно выявить простые логические правила, по которым можно охарактеризовать новость. Простейший пример – это поиск человеком новости по интересующей его тематике, например, какой-нибудь фестиваль, проходящий в этом году на берегу Черного моря. Не представляется возможным выявить простые логические правила, по которым гарантированно можно определить, относится эта новость к категории искомых или нет. Человеческая логика в этом случае работает крайне сложным образом, обрабатывая синонимы и неоднозначные слова, извлекая взаимосвязь между различными деталями в тексте и терминами, интуитивно строя сложный логический граф исходя из информации в тексте. Имитировать такое поведение в тексте без длительного процесса обучения на масштабной выборке не представляется возможным, а без предварительного обучения на размеченных данных эту логику повторить практически невозможно.

Следует отметить, что ограничение на интерпретируемость ставит под вопрос очень большой диапазон применяемых подходов, от наивной байесовской модели до нейронных сетей, поскольку их интерпретируемость весьма ограничена и целиком зависит исключительно от тренировочной выборки данных.

В данной работе будет рассматриваться ставший уже классическим вероятностный подход для задачи тематического моделирования, а также будет предложена архитектура структурированного анализа при помощи метода группового учета аргументов (МГУА).

# Раздел 1. Аналитический обзор предметной области

## 1.1 Обзор задачи тематического моделирования

Тематическое моделирование — это метод построения модели набора документов, который соотносит принадлежность каждого из документов к некоторой теме. Задача тематического моделирования, как таковая, сводится к задаче формирования текстовых кластеров на данных, описывающих ту или иную особенность у рассматриваемых текстов.[1]

Тематическое моделирование активно используется для обнаружения скрытых тем среди набора текстовых данных. Тематическое моделирование представляет собой существенную часть машинного обучения и обработки естественного языка, представляя собой существенную часть подходов к анализу данных. Во многих случаях тематическое моделирование используется для анализа архивных данных, анализа задач, которые имеют лингвистическую интерпретацию.

Несколько упрощая, анализ текстов тематическим моделированием можно обозначить как набор текстовых характеристик, которые будут способствовать формированию некоторых категорий для рассматриваемых текстов. В случае уже описанных моделей, чаще всего рассматривается вероятностный подход рассмотрения уже существующих наборов текстов, чтобы выявить некоторые латентные зависимости использования слов в одном контексте. Тематическое моделирование пытается отобразить некоторую закономерность в использовании слов в определенных текстов и сформировать из этого некоторую математическую модель.

## 1.2 Существующие подходы тематического моделирования

Тематическая модель — модель коллекции текстовых документов, которая определяет, к каким темам относится каждый документ коллекции. Алгоритм построения тематической модели получает на входе коллекцию текстовых документов. На выходе для каждого документа выдаётся числовой

вектор, составленный из оценок степени принадлежности данного документа каждой из тем. Размерность этого вектора, равная числу тем, может либо задаваться на входе, либо определяться моделью автоматически.[4]

Подход, формирующий вероятностную тематическую модель, формирует некоторое дискретное распределение на основе рассматриваемых токенов текстов, формируя тем самым формальное описание темы. При этом каждый документ рассматривается, соответственно, некоторым дискретным распределение на основе тем, сформированных из распределения токенов. Поскольку документы не являются набором токенов, абсолютно случайно и независимо выбранных в тексте, ожидается, что они будут представлять собой некоторое распределение по взаимосвязанным событиям использования определенных токинов, задачу тематического моделирования можно рассмотреть как задачу восстановления распределения токенов по теме исходя из статистических данных

Впервые задача тематического моделирования описывается в работе Рагавана, Пападимитриу, Томаки и Вемполы 1998 году. В 1999 году было предложение ввести статистический подход для двухуровневого анализа текста, названный «вероятностное скрытое семантическое индексирование» (PLSI). Затем было предложение обобщение семантического индексирования, предложенного раннее — это латентное размещение Дирихле (LDA), эта модель была разработана Дэвидом Блейем, Эндрю ЫНом и Майклом Джорданом в 2002 году. Остальные модели, по большей части, являются только расширением относительно вышеописанной LDA-модели, к примеру, вместе с LDA используется размещение патинко для введения корреляционных коэффициентов для каждого термина, описывающего тему. [5]

Подавляющее большинство моделей описывают имплементацию латентного размещения Дирихле LDA. Этот подход сохраняет актуальность и на сегодняшний день, в противовес ему не было предложено существенно других моделей, которые никак не основывались бы на уже этом

существующем подходе. Однако этот подход имеет две глобальные проблемы, решение которых крайне проблематично для данного подхода.

В первую очередь – это проблема механизма построения модели. Дело в том, что априорное распределение Дирихле (а также обобщений априорного распределения, процессы Питмана-Йора и Дирихле) не имеют достаточной лингвистической основы, поскольку частотный подход не формирует каких бы то ни было лингвистических обоснований. Лингвистические явление не формируются под руководством статистических данных, и имитация лингвистических правил через строго байесовский вывод апостериорной вероятности довольно сомнителен.

Второй проблемой является тот факт, что вероятностная модель слабо адаптируется под нужды формирования функциональных требований к предобработке данных и к самой модели.

В связи с вышеперечисленным, данное исследование имеет цель предложить альтернативный подход, отличающийся от уже существующего вероятностного подхода. Для этого предлагается рассматривать метод группового учёта аргументов (МГУА) — семейство индуктивных алгоритмов для математического моделирования мультипараметрических данных. Метод основан на рекурсивном селективном отборе моделей, на основе которых строятся более сложные модели. Точность моделирования на каждом следующем шаге рекурсии увеличивается за счет усложнения модели. Возможность адаптации модели под имеющиеся данные за счет итеративного усложнения и поиск оптимальной сложности позволяет адаптировать конечную модель математического моделирования так, чтобы она описывалась изначальной структурой предметной области рассматриваемого текста, а не апостериорной вероятностью нахождения терминов в одном тексте.

## 1.3 Дедуктивный и индуктивный методы построения модели

С точки зрения построения математических моделей, описывающих некоторые зависимости в данных, можно рассмотреть два принципиально отличающихся подхода к построению модели: индуктивный и дедуктивный.

Индуктивный подход рассматривает модель как незафиксированную структуру, нуждающуюся в адаптации и усовершенствовании, что должно находить отражение в рассматриваемых данных.

Дедуктивный подход, напротив, фиксирует структуру модели по некоторому усмотрению специалиста и в дальнейшем оптимизирует параметры этой модели для наилучшего соответствия рассматриваемым данным.

В данной работе мы рассматриваем применимость индуктивного подхода к поставленной задаче, поэтому рассмотрим тщательным образом алгоритм построения индуктивной модели, в частности, метод МГУА.

## 1.5 Индуктивное моделирование

Индуктивный алгоритм отыскания модели оптимальной структуры в состоит из следующих основных шагов.

1. Пусть имеется некоторая выборка $D = \{(x_n, y_n)\}_{n=1}^{N}, x \in \Re^m$ . Эта выборка делится на обучающую и тестовую. Пусть $\ell, C$ — множества индексов из $\{1, \dots, N\} = W$. Эти два множества удовлетворяют условиям разбиения $\ell \cup C = W, \ell \cap C = \emptyset$. Матрица $X_\ell$ состоит из векторов-строк $x_n$, где $n \in \ell$. Тогда $y_\ell$ состоит из объектов $y_n$, где индекс $n \in \ell$. Тогда разбиение выборки:

$$X_W = \left(\frac{X_\ell}{X_C}\right), y_W = \left(\frac{y_\ell}{y_C}\right), y_W \in \Re^{N\times 1}, X_W \in \Re^{N\times m}, |\ell| + |C| = N$$

2. Описывается базовая модель. Эта модель описывает отношение между зависимой переменной $y$ и свободными переменными $x$. Например,

используется функциональный ряд Вольтерра (или полином Колмогорова-Габора):

$$y = w_0 + \sum_{i=1}^{m} w_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} w_{ij} x_i x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} w_{ijk} x_i x_j x_k$$

В этой модели $x = \{x_i | i = 1, \dots, m\}$ — множество свободных переменных и $w$ — вектор параметров — весовых коэффициентов

$$w = \langle w_I, w_{i,j}, w_{i,j,k}, \dots | i, j, k, \dots = 1, \dots, m \rangle$$

Во многих случаях следует увеличивать размерность вектор свободной переменной $x$ посредством добавления нелинейных преобразований отдельно взятых переменных. Например, задано конечное множество нелинейных функций $G = \{g | \Re \longrightarrow \Re\}$. Дополнительная свободная переменная получается путем применения некоторого преобразования из $G$ к одной или к нескольким переменным из множества $\{x\}$. Базовая модель линейна относительно параметров $w$ и нелинейна относительно свободных переменных $x$.

3. Исходя из поставленных задач выбирается целевая функция — внешний критерий, описывающий качество модели. Ниже описаны несколько часто используемых внешних критериев.
4. Индуктивно порождаются модели-претенденты. Следует отметить, что при этом отдельно вводится некоторое ограничение на величину полинома. Предположим, что введено ограничение, что степень полинома базовой модели на не должно превышать заданное число $R$. Тогда модель представима в виде линейной комбинации заданного числа $F_0$ произведений свободных переменных:

$$y = f(x_1, x_2, \dots, x_1^2, x_1 x_2, x_2^2, \dots, x_m^R)$$

здесь $f$ — линейная комбинация. Аргументы этой функции переобозначаются следующим образом:

$$x_1 \longrightarrow a_1, x_2 \longrightarrow a_2, \dots, x_1^2 \longrightarrow a_\propto, x_1 x_2 \longrightarrow a_\beta, x_2^2 \longrightarrow a_\gamma, \dots, x_m^q \longrightarrow a_{F_0}$$

то есть,

$$y = f(a_1, a_2, \dots, a_{F_0})$$

Задается одноиндексная нумерация $w = \langle w_1, \dots, w_{F_0} \rangle$ для линейно входящих элементтов. Следовательно, модель представима как линейная комбинация

$$y = w_0 + \sum_{i=1}^{F_0} w_i a_i = w_0 + wa$$

Очевидно, что любая порождаемая модель задается линейной комбинацией элементов $\{(w_i, a_i)\}$, в которой множество индексов является подмножеством $\{1, \dots, F_0\}$.

Вектора $x_n$ — элемента выборки $D$ ставится в соответствие вектор $a^n$, алгоритм построения соответствия указан выше. Строится матрица $A_W$ — набор векторов-столбцов $a_i$. Матрица $A_W$ разбивается на подматрицы $A_l$ и $A_C$ . Наименьшую невязку $|y - \hat{y}|$, где $\hat{y} = A\widehat{w}$ доставляет значение вектора параметров $\widehat{w}$, который вычисляется методом наименьших квадратов:

$$\widehat{w}_G = (A_G^T A_G)^{-1} A_G^T y_{G'} \text{ где } G \in \{l, C, W\}$$

При этом в качестве внутреннего критерия выступает среднеквадратичная ошибка

$$\epsilon_G^2 = |y_G - A_G \widehat{w}_G|^2$$

В соответствии с критерием $\epsilon_G^2 \to min$ происходит настройка параметров $w$ и вычисление ошибки на тестовой подвыборке, обозначенной $G$, здесь

$G = l$. При усложнении модели внутренний критерий не дает минимума для моделей оптимальной сложности, поэтому для выбора модели он не пригоден.

5. Для выбора моделей вычисляется качество порожденных моделей. При этом используется контрольная выборка и назначенный внешний критерий. Ошибка на подвыборке $H$ обозначается

$$\Delta^2(H) = \Delta^2(H \backslash G) = |y_H - A_H \widehat{w}_G|^2,$$

где $H \in \{l, C\}$, $H \cap G = \emptyset$.. Это означает что ошибка вычисляется на подвыборке $H$ при параметрах модели, полученных на подвыборке $G$.

6. Модель, доставляющая минимум внешнему критерию, считается оптимальной.

Недостижение минимума внешнего критерия при генерации моделей более высокой сложности означает, что модель превысила необходимую сложность, и тогда выбирается лучшая модель из предыдущих моделей. Существуют следующие причины, по которым глобальный минимум может не существовать[6]:

- данные слишком зашумлены,
- среди данных нет необходимых для отыскания модели переменных,
- неверно задан критерий выбора,
- при анализе временных рядов существует значительная временная задержка отыскиваемой причинно-следственной связи.

Как результат, ожидается, что будет получена некоторая модель в результате перебора. Параметры. моделей должны быть настроены так, чтобы внешний критерий был достигнут минимума на каждой из моделей.

Все алгоритмы МГУА делятся на две большие подгруппы, однорядные и многорядные.

В результате массовой селекции алгоритм МГУА последовательно генерирует модели возрастающей сложности. Все модели являются настраиваемыми, поскольку результат обучения модели представлен коэффициентами в результате использования метода наименьших квадратов. Наилучшие из моделей выбираются при помощи внешнего критерия. Отличие многорядных алгоритмов заключается в том, что они способы вычислять остатки прочих регрессионных моделей при новом проходе селекции.

### 1.6 Объект, предмет исследования и требования к реализации

В этой работе будет показано, что есть возможность оттолкнуться от вероятностного подхода в задаче тематического моделирования и построить его на основе МГУА-модели. Такой метод построения модели позволить провести унификацию данных и дальнейшему построению онтологии на основе этих данных.

Объект исследования: методы тематического моделирования текстовых данных.

Предмет исследования: концепция построения тематического моделирования при помощи иерархической структуре с МГУА-элементами.

Цель работы: проектирование и разработка прикладного алгоритма для генерации МГУА-элементов для дальнейшего анализа в задаче тематического моделирования текстовых данных.

Для достижения цели необходимо выполнить следующие задачи:

1. Подготовка предварительной информации по иерархической модели и структурирование опорной информации для дальнейшего исследования;
2. Аналитический обзор индукционного подхода к созданию модели;
3. Проектирование информационной системы иерархической обработки текстовых данных с тематическим моделированием;

4. Разработка низкоуровневого элемента подбора модели

## 1.7 Анализ предъявляемых требований

Опишем основные идеи проектируемой системы:

1. Модель рассматривается как двухуровневая составная модель: множество моделей-генераторов, результаты которых управляются моделью-оркестратором.
2. Низкоуровневая модель представлена МГУА-моделью, генерирующая оптимальную модель в соответствии с заданными параметрами
3. Высокоуровневая модель оперирует результатами множества низкоуровневых моделей
4. Многообразие низкоуровневых моделей определяется сложностью предметной области
5. Рассматриваемое множество понятий формируется как графоподобная структура онтологии по набору данных
6. Любая сформированная тема из данных представляет собой какое-то рассматриваемое на онтологии поле, в которое укладываются рассматриваемые термины

Тщательная реализация вышеперечисленных требований должна привести к удовлетворению необходимых условий уровня качества продукта для комфортного и продуктивного использования.

# Раздел 2. Проектирование предлагаемого решения

## 2.1 Подбор инструментов для разработки

Поскольку данное исследование представляет собой алгоритм, способный выполнять поставленную перед ним функцию, целесообразнее всего имплементировать этот алгоритм в качестве переносимой библиотекой.

Ввиду крайне большого количества вычислительных операций, оптимизация работы с памятью которых серьезно отразится на скорости работы алгоритма, было принято решение после первичного проектирования реализовать библиотеку на языке C++ с последующей интеграции в язык Python в виде скомпилированного подключаемого модуля. Ручное управление памятью, работа с захэшированными значениями вычисленных степенных значений данных и возможность напрямую оптимизировать доступ к данным и многопоточность выполнения операций – всё это вместе позволяет оптимизировать процесс вычислений недостижимым для языка Python образом.

Используя уже рассмотренные требования к программе, можно выбрать соответствующие инструменты, с помощью которых станет возможным реализация проекта.

Поскольку проект подразумевает использование различных элементов и программного кода, целесообразно использовать такой программный комплекс, который позволит работать сразу со всеми такими элементами, собирая их в полноценную рабочую программу.

Такому требованию на данный момент соответствуют две среды разработки: JetBrains IDEA/CLion/Pycharm и Microsoft Visual Studio.

Следует отметить, что существенная часть разработки данного модуля, а также последующие исследования, будут проводиться на языке Python. Поддержка языка Python была введена в IDE Microsoft Visual Studio относительно недавно, и притом не поддерживает основной функционал IDE для этого языка целиком или явных ошибок при повседневном использовании.

В связи с этими фактами было принято решение использовать комплекс продуктов JetBrains.

Вдобавок к этому, среды разработки JetBrains предоставляют удобный интерфейс для написания функционала тестирования продукта. Этот функционал является крайне важным, поскольку это гарантирует корректность всех этапов разработки. Особенно это важно в контексте исследования, результаты которого нужно тщательно перепроверять на предмет ошибок.

Также крайне полезными инструментами IDE JetBrains в работе были признаны профилировщик и анализ покрытия кода тестами, интеграция Docker-контейнеризации и менеджер пакетов языка Python.

## 2.2 Проектирование логики приложения

Для минимизации проблем при разработке программы, следует максимально продумать логику работы программы.

Модули, написанные на C++ (или C), обычно используются для расширения возможностей интерпретатора Python, а также для доступа к низкоуровневым возможностям операционной системы. Существует три основных типа модулей:

- Модули акселератора: так как Python — это интерпретируемый язык, для повышения производительности некоторые фрагменты кода могут быть написаны на C++.
- Модули оболочки: предоставляют существующие интерфейсы C/C++ для кода Python или адаптированный API, который удобно использовать в Python.
- Модули низкоуровневого системного доступа: созданы для доступа к низкоуровневым функциям среды выполнения CPython, операционной системы и базового оборудования.

## 2.3 Предварительное проектирование модели высокого уровня

Важным моментом данного исследования является тот факт, что выполненное исследование и реализация не является конечным продуктом, готовым к использованию. Данная работа предусматривает исследовательски-подготовительную часть всего исследования в целом, а также первичную реализацию низкоуровневых модулей, являющихся вспомогательными элементами в дальнейшем исследовании.

Для оценки перспектив дальнейшего исследования, рассмотрим низкоуровневые модели и их границы применимости. Будучи атомарными простыми объектами, эти модели сильно ограничены в своих способностях – простые связи, математическая неполнота модели и недостаток реальных данных делает каждую из них по отдельности крайне слабой. Однако можно утверждать, что при корректном управлении особенностями каждой из этих моделей, можно наладить такую систему, где каждая из моделей будет работать в своем правильном месте и приносить наибольшую пользу.

Целью дальнейших исследований, вынесенных на магистерскую работу, является проектирование и реализация высокоуровневого модуля, способного провести такую агрегацию моделей.

В первом приближении, для ясности цельной картины применения этой структуры, представим, что низкоуровневые модели накладываются своими терминами на некоторую онтологию, представляющую определенную предметную область, интересующую специалиста-аналитика. За счет наложения каждой из моделей на онтологию, можно получить некоторую характеристики распространенности модели по онтологии, пересеченности с другими моделями, а также сцепливаемости терминов внутри поля на онтологии. Формально, если рассмотреть задачу оптимизации некоторой функции, которая сведет к минимуму сцепливаемость между полями, пересечения с другими полями, минимизирует площадь поля по онтологии (площадь как наибольшее из минимальных расстояний между всеми точками

взятой зоны), то минимизируя эту функцию, мы получим наилучшее распределение текста по тематикам. Дальнейшая интерпретация задачи целиком и полностью зависит исключительно от него самого, в особенности от его компетенций в данном вопросе.

# Раздел 3. Реализация решения

## 3.1 Общая характеристика решения

Как уже было сказано, общая архитектура решения имплементировалась на языке C++ с возможностью предоставления библиотеки для языка Python в дальнейшем. Отметим, что реализация модуля является исключительно подготовительной частью данной работы. Основной проблемой является возможность внедрения такого модуля в более высокоуровневую систему, а не конкретная реализация этого модуля. В данном разделе представлена реализация одного такого модуля, реализующего генерацию моделей, описывающих данные без специфики их применения, пользуясь признаками количества использованных в тексте слов. Пример итогового кода ядра библиотеки приведен ниже.

```
//------- Trainer -------
void PNNTrainer::train(const Common::tDataFramePtr& pDF, const Common::tDataSamplesPtr& pSamples, const Common::tDataFactorsPtr&
        std::vector< tModelDraftPtr >& drafts, std::vector< tModelPtr >& out) const
{
    if (pTarget->GetCount() != 1)
    {
        std::ostringstream ostr;
        ostr << "Single factor is accepted as target factor for model training; got " << pTarget->GetCount() << " : " << __func__
        throw ostr.str().c_str();
    }

    out.resize(drafts.size());

    for (size_t d = 0; d < drafts.size(); d++)
    {
        //References
        tModelDraftPtr pDraft = drafts[d];
        std::shared_ptr<PNNStructure> pStructure = std::dynamic_pointer_cast<PNNStructure>(pDraft->_structure);
        PNNBasis::tDimension nLayers = pStructure->GetNumberOfLayers();
        PNNBasis::tDimension neuronsPerLayer = pStructure->GetNumberOfNeuronsPerLayer();

        //DataFrame for PNN construction: {input factors} + {neurons outputs} + {target}
        Common::tDataFramePtr pNewDF;
        Common::tDataFactorsPtr pNewFactors, pNeurons, pNewTarget;
        SubsetAndExtendDataFrame(pDF, pSamples, pFactors, pTarget, neuronsPerLayer, pNewDF, pNewFactors, pNeurons, pNewTarget);
        Common::tDataFactorsPtr pCombinedFactors = Common::DirectDataFactors::MakeSingle(pNewFactors, pNeurons);
        std::valarray<Common::DataFactors::tIndex> idxsNeuronFactors = pNeurons->GetIndices();

        //Init analyzer
        _analyzer->Initialize(pNewDF, pSamples, pNewFactors, pNewTarget);

        //Layers
        std::vector<std::vector< tModelPtr >> structure;
        structure.reserve(nLayers);
        for (PNNBasis::tDimension iLayer = 0; iLayer < nLayers; iLayer++)
        {
            //Init
            if (iLayer == 0)    //First row
                _strategy->Initialize(pNewDF, pNewFactors, pNewTarget, pSamples, neuronsPerLayer);
            else                //Other rows
                _strategy->Initialize(pNewDF, pCombinedFactors, pNewTarget, pSamples, neuronsPerLayer);
```

Рис. 4.1. Пример кода функционала МГУА-модели

Основная идея архитектура заключается в цепочке зависимостей при сборке библиотек, каждый из этапов которой контролируется покрытием модульного тестирования, что позволяет избежать регрессионных ошибок при разработке. Пример тестирования со стороны функционального кода на языке C++ представлен на рисунке 4.2.

```
namespace GMDHTest
{
    static void InitializeDataFrame(Common::DataFrame& df)
    {
        std::string strInCSV(
            "X,Y,Z\n"
            "-1,10,0.100\n"
            "-2,-10,0.200\n"
            "-3,10,0.300\n"
            "-4,-10,0.400\n"
            "-5,10,0.100\n"
            "-6,-10,0.200\n"
            "-7,10,0.300\n"
            "-8,-10,0.400\n"
            "-9,10,0.100\n"
        );
        std::istringstream inCSV(strInCSV);
        df.LoadFromCSV(inCSV);
    }

    void PNNStructureGenerator_TestBody(std::valarray<Core::uint16> &trueLayersArray, std::valarray<Core::uint16> &trueNe
    {
        //Internal test consistency
        Assert::AreEqual(trueLayersArray.size(), trueNeuronsPerLayerArray.size());

        std::auto_ptr<Model::StructureGenerator> pGen(new Model::PNNStructureGenerator(trueLayersArray, trueNeuronsPerLay

        //TotalNumber works correctly - doesn't depend on factors number
        size_t nTotalTrue = trueLayersArray.size();
        Common::DataFactors::tCount nFactors1 = 1;
        Common::DataFactors::tCount nFactors = 3;
        Common::DataFactors::tCount nFactors2 = 10;
        size_t nTotal1 = pGen->TotalNumber(nFactors1);
        size_t nTotal = pGen->TotalNumber(nFactors);
        size_t nTotal2 = pGen->TotalNumber(nFactors2);
        Assert::AreEqual(nTotalTrue, nTotal1);
        Assert::AreEqual(nTotalTrue, nTotal);
        Assert::AreEqual(nTotalTrue, nTotal2);

        //Init Generator
        pGen->Initialize(nFactors);
        Assert::AreEqual(false, pGen->Finished());
```

Рис. 4.2. Тестирование функциональности кода на C++

Затем итоговая собранная библиотека интегрируется в виртуальную среду языка Python, где запускает тестовые модули и проверяет функционал библиотеки на описанных тестах. Пример тестирования обертки на языке Python представлен на рисунке 4.3.

```
class Test_PyGMDH_Common(unittest.TestCase):
    def test_DataFrame(self):
        pyGMDH_CPP.GMDHComputer_Reset()
        (N, x, y, z, x_ext, y_ext, z_ext, coeffs) = InitData()
        headersTrue = ['X', 'Y', 'Z']
        pyGMDH_CPP.GMDHComputer_AddData('DF', headersTrue, [list(a) for a in zip(x, y, z)])
        pyGMDH_CPP.GMDHComputer_AddData('DF_ext', headersTrue, [list(a) for a in zip(x_ext, y_ext,
        z_ext)])

        self.assertFalse(pyGMDH_CPP.GMDHComputer_DataExists('abc'))
        self.assertTrue(pyGMDH_CPP.GMDHComputer_DataExists('DF'))
        self.assertTrue(pyGMDH_CPP.GMDHComputer_DataExists('DF_ext'))

        (headers, values) = pyGMDH_CPP.GMDHComputer_GetData('DF')
        self.assertListEqual(headers, headersTrue)
        self.assertEqual(len(values), N)

        (headers, values) = pyGMDH_CPP.GMDHComputer_GetData('DF_ext')
        self.assertListEqual(headers, headersTrue)
        self.assertEqual(len(values), N)

    def test_DataFactors(self):
        trueIndices = [0,2]
        pyGMDH_CPP.GMDHComputer_CreateObject('DF', 'DirectDataFactors', indices=trueIndices)
        self.assertRaises(NameError, pyGMDH_CPP.GMDHComputer_CreateObject, 'DF', 'DirectDataFactors',
        indices=trueIndices)  #Object exists
        obj = pyGMDH_CPP.GMDHComputer_GetObject('DF', 'DataFactors')
        self.assertTrue(type(obj) == type({}))
        self.assertTrue('type' in obj)
```

Рис. 4.3. Тестирование оболочки на языке Python

На данный момент низкоуровневый модуль корректно генерирует множество возможных вариантов конфигураций ключевых слов для топика, исходя из заданной графовой структуры слов в онтологии.

Как результат, решение представляет собой модуль со следующими точками вызова API:

- Create()

  Создание объекта, который в последующем будет отвечать за сохранение множества моделей

- Fit()

  Генерация множества моделей, соответствующих полученным данным

- Predict()

  Сформировать предсказание на некоторой выборке данных в соответствии с какой-то моделью.

К описанному API, предоставленному со стороны C++, добавляется внешний модуль языка Python, который использует этот API для расширения функционала. Рисунок 4.4 отображает фрагмент кода, отвечающий за стыковку преобразованных функций API, реализующих более сложную логику, характерную для конкретной реализации низкоуровневого элемента, с интерфейсом для методов в языке Python.

```
    }

    if (!pCommand.get())
    {
        std::ostringstream ostr;
        ostr << "Command is not recognized and though deserialized; type = " << pCommandLabel;
        PyErr_SetString(PyExc_NameError, ostr.str().c_str());
        return NULL;
    }
    pCommand->execute(*pComputer);

    Py_RETURN_NONE;
}

static PyMethodDef pyGMDH_CPP_methods[] = {
    { "GMDHComputer_AddData", (PyCFunction)GMDHComputer_AddData, METH_VARARGS, nullptr },
    { "GMDHComputer_DataExists", (PyCFunction)GMDHComputer_DataExists, METH_VARARGS, nullptr },
    { "GMDHComputer_GetData", (PyCFunction)GMDHComputer_GetData, METH_VARARGS, nullptr },
    { "GMDHComputer_CreateDataReference", (PyCFunction)GMDHComputer_CreateDataReference, METH_VARARGS, nullptr },
    { "GMDHComputer_CreateFactorList", (PyCFunction)GMDHComputer_CreateFactorList, METH_VARARGS, nullptr },
    { "GMDHComputer_TrainPNN", (PyCFunction)GMDHComputer_TrainPNN, METH_VARARGS, nullptr },
    { "GMDHComputer_UsePNN", (PyCFunction)GMDHComputer_UsePNN, METH_VARARGS, nullptr },
    { "GMDHComputer_Reset", (PyCFunction)GMDHComputer_Reset, METH_NOARGS, nullptr },
    { "GMDHComputer_SetThreads", (PyCFunction)GMDHComputer_SetThreads, METH_VARARGS, nullptr },
    { "GMDHComputer_GetThreads", (PyCFunction)GMDHComputer_GetThreads, METH_NOARGS, nullptr },
    { "GMDHComputer_CreateObject", (PyCFunction)GMDHComputer_CreateObject, METH_VARARGS | METH_KEYWORDS, nullptr },
    { "GMDHComputer_GetObject", (PyCFunction)GMDHComputer_GetObject, METH_VARARGS, nullptr },
    { "GMDHComputer_ExecuteCommand", (PyCFunction)GMDHComputer_ExecuteCommand, METH_VARARGS | METH_KEYWORDS, nullptr },
    // Terminate the array with an object containing nulls.
    { nullptr, nullptr, 0, nullptr }
};

static PyModuleDef pyGMDH_CPP_module = {
    PyModuleDef_HEAD_INIT,
    "pyGMDH_CPP",                       // Module name to use with Python import statements
    "Native interfact of GMDHComputer",     // Module description
    0,
    pyGMDH_CPP_methods                      // Structure that defines the methods of the module
};

PyMODINIT_FUNC PyInit_pyGMDH_CPP() {
    Core::MemObject::Construct();
    pComputer.reset(new GMDHComputing::GMDHComputer());
    return PyModule_Create(&pyGMDH_CPP_module);
}
```

Рис. 4.4, перенос реализованных надстроек над API в методы языка Python

Следует отметить, что в контексте данной реализации какие-либо графические отображения результатов принципиально невозможны, поскольку модуль заключается исключительно генерацией некоторых объектов, используемых далее внутри системы без предоставления доступа к пользователю. В связи с этим для отображения функциональности модуля

было введено дополнительное логгирование, чтобы отобразить процесс создания моделей и их валидации. На рисунке 4.5 изображены результаты логгирования запуска модуля с режимом расширенного логгирования.

```
2019-06-18 11:49:56,390 - root - DEBUG - Module initiated...
2019-06-18 11:49:56,390 - root - DEBUG - Model generating for data shape (13039 x 288) initiated
2019-06-18 11:49:56,464 - root - INFO - Validating data...
2019-06-18 11:49:56,769 - root - DEBUG - Done.
2019-06-18 11:49:56,769 - root - INFO - Starting model generator...
2019-06-18 11:49:56,769 - root - DEBUG - 20 models expected
2019-06-18 11:49:56,770 - root - INFO - Fitting model for a=0
2019-06-18 11:49:56,782 - root - DEBUG - Fitted. Coeffs: [0.54957291 0.0391326  0.79818901 0.8279688  0.70525137 0.55357596
 0.02410774]
2019-06-18 11:49:56,782 - root - INFO - Criterion value for model: 0.4509344863705709
2019-06-18 11:49:56,782 - root - INFO - Model for a=0 saved
2019-06-18 11:49:56,783 - root - INFO - Fitting model for a=1
2019-06-18 11:49:56,830 - root - DEBUG - Fitted. Coeffs: [0.19616343 0.97288289 0.05840419 0.84809988 0.42502245 0.06884314
 0.28743688]
2019-06-18 11:49:56,830 - root - INFO - Criterion value for model: 0.45349132718755214
2019-06-18 11:49:56,830 - root - INFO - Model for a=1 saved
2019-06-18 11:49:56,830 - root - INFO - Fitting model for a=2
2019-06-18 11:49:56,879 - root - DEBUG - Fitted. Coeffs: [0.36431564 0.74644299 0.44922257 0.29349647 0.57275843 0.05472064
 0.74979284]
2019-06-18 11:49:56,879 - root - INFO - Criterion value for model: 0.5249768497875618
2019-06-18 11:49:56,879 - root - INFO - Model for a=2 saved
2019-06-18 11:49:56,879 - root - INFO - Fitting model for a=3
2019-06-18 11:49:56,937 - root - DEBUG - Fitted. Coeffs: [0.86916915 0.07661961 0.55177052 0.8309042  0.81001493 0.01207062
 0.93595727]
2019-06-18 11:49:56,937 - root - INFO - Criterion value for model: 0.4903022087545875
2019-06-18 11:49:56,937 - root - INFO - Model for a=3 saved
2019-06-18 11:49:56,937 - root - INFO - Fitting model for a=4
2019-06-18 11:49:56,992 - root - DEBUG - Fitted. Coeffs: [0.24620129 0.26690046 0.12099907 0.5808971  0.32907108 0.52337878
 0.41240844]
2019-06-18 11:49:56,992 - root - INFO - Criterion value for model: 0.22232483411445672
2019-06-18 11:49:56,992 - root - INFO - Model for a=4 saved
2019-06-18 11:49:56,992 - root - INFO - Fitting model for a=5
2019-06-18 11:49:57,019 - root - DEBUG - Fitted. Coeffs: [0.9666375  0.37057645 0.72439509 0.85065247 0.79511544 0.99578203
 0.82529721]
2019-06-18 11:49:57,019 - root - INFO - Criterion value for model: 0.5582774970603029
2019-06-18 11:49:57,020 - root - INFO - Model for a=5 saved
2019-06-18 11:49:57,020 - root - INFO - Fitting model for a=6
2019-06-18 11:49:57,080 - root - DEBUG - Fitted. Coeffs: [0.03821524 0.11389532 0.42874108 0.33672035 0.99677254 0.47981513
 0.61850205]
2019-06-18 11:49:57,081 - root - INFO - Criterion value for model: 0.18388697839056245
2019-06-18 11:49:57,081 - root - INFO - Model for a=6 saved
```

Рис 4.5. Расширенное логгирование создания моделей.

Каждая модель обладает данными точками вызова, что обеспечивает низкоуровневый доступ к элементам.

В дальнейшем такая структура построения низкоуровневых моделей даст большое поле деятельности для построения высокоуровневой модели управления.

### 3.2 Перспектива исследования

В конечном счете эти модули предполагается использовать для существенно более сложной задачи, решение которой отводится на написание магистерской работы.

Дело в том, что реализованный в этой работе модуль не является независимым объектом, готовым к работе. Этот модуль представляет собой нечто вроде генератора, который используется, чтобы создать то множество моделей, среди которых потом будут подбираться наиболее полезные.

Как уже было описано выше, модель более высокого уровня в таком случае становится «ответственной» за поведение низкоуровневых моделей, их сочетание и взаимодействие.

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

Студенту:

| Группа | ФИО |
|---|---|
| 8К51 | Ванюшину Ивану Сергеевичу |

| | | | |
|---|---|---|---|
| **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 инженера – 21760 руб.<br>Оклад руководителя – 33664 руб. |
| 2. *Нормы и нормативы расходования ресурсов* | Премиальный коэффициент руководителя 30%;<br>Коэффициент доплат и надбавок руководителя 20%;<br>Районный коэффициент 30%;<br>Коэффициент дополнительной заработной платы 15%;<br>Накладные расходы 16%. |
| 3. *Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования* | Коэффициент отчислений на уплату во внебюджетные фонды 28% |
| **Перечень вопросов, подлежащих исследованию, проектированию и разработке:** | |
| 1. *Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения* | -Анализ конкурентных технических решений |
| 2. *Планирование и формирование бюджета научных исследований* | Формирование плана и графика разработки:<br>- определение структуры работ;<br>- определение трудоемкости работ;<br>- разработка графика Гантта.<br>Формирование бюджета затрат на научное исследование:<br>- материальные затраты;<br>- затраты на специальное оборудование;<br>- заработная плата (основная и дополнительная);<br>- отчисления на социальные цели;<br>- накладные расходы. |
| 3. *Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования* | - Определение потенциального эффекта исследования |
| **Перечень графического материала** *(с точным указанием обязательных чертежей)*: | |
| 1. *Оценочная карта конкурентных технических решений*<br>2. *Матрица SWOT*<br>3. *График Гантта*<br>4. *Расчет бюджета затрат* | |

| **Дата выдачи задания для раздела по линейному графику** | |
|---|---|

**Задание выдал консультант:**

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент | Подопригора Игнат Валерьевич | Кандидат экономических наук | | |

**Задание принял к исполнению студент:**

| Группа | ФИО | Подпись | Дата |
|---|---|---|---|
| 8К51 | Ванюшин И.С. | | |

# Раздел 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

## 4.1 Потенциальные потребители результатов исследования

Данное программное обеспечение позволяет анализировать содержание текста и группировать тексты по темам, которые описывают эти тексты. Основной целевой аудиторией данного программного обеспечения являются аналитики и прочие люди, которые нуждаются в анализе текстовых данных большого объема.

Актуальность данной работы заключается в возможности обрабатывания и анализа текстовой информации без существенных усилий со стороны специалиста. Данное программное обеспечение формирует новый подход к решению, что позволяет улучшить качество анализа текстов.

## 4.2 Технология QuaD

Данный вид анализа представляет собой гибкий инструмент измерения характеристик, описывающих качество новой разработки и ее перспективность на рынке и позволяющих принимать решение целесообразности вложения денежных средств в научно-исследовательский проект.

Оценочная карта, рассчитанная по технологии QuaD, представлена в таблице 1.

Таблица 1 – Оценочная карта по технологии QuaD

| **Критерии оценки** | **Вес крит ерия** | **Средн ий балл** | **Макси-мальный балл** | **Относительное значение (3/4)** | **Средневзвеш енное значение (5x2)** |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| **Показатели оценки качества разработки** | | | | | |
| 1. Удобство в использовании | 0.1 | 80 | 100 | 0.8 | 0.08 |

Продолжение таблицы 1

| | | | | | |
|---|---|---|---|---|---|
| 2. Функциональные возможности | 0.05 | 65 | 100 | 0.65 | 0.0325 |
| 3. Скорость работы программы | 0.15 | 95 | 100 | 0.95 | 0.1425 |
| 4. Интерфейс | 0.05 | 75 | 100 | 0.75 | 0.0375 |
| 5. Требовательность к системе | 0.15 | 95 | 100 | 0.95 | 0.1425 |
| 6. Возможность оперативного обновления | 0.1 | 95 | 100 | 0.95 | 0.095 |
| 7. Справочная информация | 0.05 | 60 | 100 | 0.6 | 0.03 |
| 8. Независимость от сторонних расчетов | 0.1 | 75 | 100 | 0.75 | 0.075 |
| 9. Масштабируемость | 0.05 | 80 | 100 | 0.8 | 0.04 |
| **Показатели оценки коммерческого потенциала разработки** | | | | | |
| 10. Цена | 0.1 | 90 | 100 | 0.9 | 0.09 |
| 11. Перспективность рынка | 0.05 | 55 | 100 | 0.55 | 0.0275 |
| 12. Послепродажное обслуживание | 0.05 | 85 | 100 | 0.85 | 0.0425 |
| **Итого** | **1** | | | | **0.835** |

Показатель конкурентоспособности, рассчитанный по технологии QuaD, равен 0.835, что говорит о хорошей перспективности данной разработки.

## 4.3 SWOT-анализ

SWOT-анализ представляет собой комплексный анализ научно-исследовательского проекта. Данный анализ проводится для исследования внутренней и внешней среды проекта и позволяет оценить сильные и слабые стороны, возможности и угрозы проекта.

Результаты анализа представлены в таблице 2.

Таблица 2 – Матрица SWOT

| | **Сильные стороны научно-исследовательского проекта:** | **Слабые стороны научно-исследовательского проекта:** |
|---|---|---|
| | С1. Использование движка Cython для работы с параллельными вычислениями.<br>С2. Быстрота разработки.<br>С3. Легкая масштабируемость.<br>С4. Оперативность добавления нововведений.<br>С5. Нетребовательность к квалификации разработчиков. | Сл1. Техническая ограниченность возможностями движка.<br>Сл2. Зависимость от сторонних программ при создании нововведений.<br>Сл3. Визуальное улучшение программы сильно сказываются на требованиях к ЭВМ пользователя.<br>Сл4. Пользователь не может сам добавлять нужные ему нововведения.<br>Сл5. Вовлеченность разработчика в создание всех частей программы. |
| **Возможности:**<br>В1. Привлечение новых потребителей.<br>В2. Легкая возможность ухода разработки в иную предметную область для расширения целевой аудитории.<br>В3. Большие возможности по модернизации проекта.<br>В4. Возможность включения в проект новых технологических решений.<br>В5. Привлечение новых кадров к работе над проектом. | Направления развития:<br>В1С2С4<br>Быстрота создания продукта и оперативность добавления нововведений позволят привлечь новую аудиторию и подстроиться под ее запросы<br>В2С2<br>Возможность добавления необходимого функционала для совершенно новой аудитории позволить увеличить спрос без ущерба для прошлой целевой аудитории<br>В4В3С3С4<br>Имеется возможность относительно оперативного добавления новых технологический решений в проект, что позволит дольше поддерживать проект актуальным<br>В5С5<br>Достаточно просто найти необходимого кандидата на роль разработчика | Сдерживающие факторы:<br>В1Сл3Сл4<br>Определенные программные ограничения и требования не всем понравятся<br>В3Сл2Сл5<br>В случае необходимости добавления нововведений разработчики будут сильно нагружены работой сразу над множеством задач<br>В4Сл1<br>Имеется существенная вероятность, что некоторые новые технологические решения не будут доступны в движке изначально |

Продолжение таблицы 2

| | | |
|---|---|---|
| **Угрозы:**<br>У1. Изначально слабый спрос на продукт.<br>У2. Простота разработки может быстро привлечь конкурентов.<br>У3. Падение спроса из-за роста требований к ЭВМ пользователя.<br>У4. Изначально очень слабое финансирование проекта может критично сказаться на дальнейшем развитии проекта.<br>У5. Полное отсутствие маркетинга может критично сказаться на увеличении изначально слабого спроса. | Угрозы развития:<br>С4У4<br>Для оплаты труда разработчиков при создании и добавления нововведений может не хватит финансовых средств<br><br>С5У5<br>Затруднительно привлечь новые кадры с проекту с низким спросом. | Уязвимости:<br>У1Сл4<br>Части потенциальных потребителей может не понравиться идея ожидания добавления необходимого им контента в программу<br><br>У3Сл3<br>Добавление/улучшение визуальной части сократит часть аудитории, которая обладает относительно слабыми ЭВМ<br>У4Сл1Сл2Сл5<br>Малого объема финансовых средств может не хватит для найма новых кадров и добавления нового функционала |

Анализ показал, что проект множество направлений для своего развития, при этом имея несколько немаловажных уязвимостей.

## 4.4 Планирование научно-исследовательских работ

### 4.4.1 Структура работ в рамках научного исследования

Планирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках научного исследования;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения научных исследований.

В данном разделе составлен перечень работ в рамках проведенного научного исследования, проведено распределение исполнителей по видам работ. Порядок составленных работ и распределение исполнителей приведен в таблице 3.

Таблица 3 – Перечень работ и распределение исполнителей.

| № работы | Наименование работы | Исполнители работы |
|---|---|---|
| 1 | Выбор научного руководителя бакалаврской работы | Ванюшин И.С. |
| 2 | Составление и утверждение темы бакалаврской работы | Фофанов О.Б., Ванюшин И.С. |
| 3 | Составление календарного плана-графика выполнения бакалаврской работы | Фофанов О.Б. |
| 4 | Подбор и изучение литературы по теме бакалаврской работы | Ванюшин И.С., Фофанов О.Б. |
| 5 | Анализ предметной области | Ванюшин И.С., Фофанов О.Б. |
| 6 | Разработка необходимых элементов программы | Ванюшин И.С., Фофанов О.Б. |
| 7 | Разработка необходимых скриптов | Ванюшин И.С., Фофанов О.Б. |
| 8 | Сборка рабочей программы | Ванюшин И.С., Фофанов О.Б. |
| 9 | Тестирование и исправление ошибок работы программы | Ванюшин И.С., Фофанов О.Б. |
| 10 | Согласование выполненной работы с научным руководителем | Фофанов О.Б., Ванюшин И.С. |
| 11 | Выполнение других частей работы (финансовый менеджмент, социальная ответственность) | Ванюшин И.С. |
| 12 | Подведение итогов, оформление работы | Ванюшин И.С. |

Таким образом, сформирован перечень работ с распределением всех исполнителей.

### 4.4.2 Определение трудоемкости выполнения работ

Для определения ожидаемой продолжительности работ $t_{ож}$ с помощью экспертных оценок были использованы следующие формулы:

$$t_{ожi} = \frac{3t_{\min i} + 2t_{\max i}}{5}, \quad (1)$$

$t_{\min i}$ – минимально возможная трудоемкость выполнения заданной $i$-ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;

$t_{\max i}$ – максимально возможная трудоемкость выполнения заданной $i$-ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях $T_р$, учитывающая параллельность выполнения работ несколькими исполнителями. Такое вычисление необходимо для обоснованного расчета заработной платы, так как удельный вес зарплаты в общей сметной стоимости научных исследований составляет около 65 %.

$$T_{р_i} = \frac{t_{ожi}}{Ч_i}, \quad (2)$$

где $T_{рi}$ – продолжительность одной работы, раб. дн.;

$t_{ожi}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.

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

### 4.4.3 Разработка графика проведения научного исследования

Для построения графика, длительность каждого этапа работ из рабочих дней переведена в календарные дни, для этого использована формула:

$$T_{кi} = T_{рi} \cdot k_{кал}, \quad (3)$$

где $T_{кi}$– продолжительность выполнения *i*-й работы в календарных днях;

$T_{рi}$ – продолжительность выполнения *i*-й работы в рабочих днях;

$k_{кал}$ $k_{кал}$– коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - T_{вых} - T_{пр}}, \quad (4)$$

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

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

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

Согласно производственному календарю (для 6-дневной рабочей недели) в 2019 году 365 календарных дней, 299 рабочих дней, 66 выходных/праздничных дней.

$$k\text{кал} = \frac{\text{Ткал}}{\text{Ткал} - \text{Твых} - \text{Тпр}} = \frac{365}{365 - 66} = 1{,}22$$

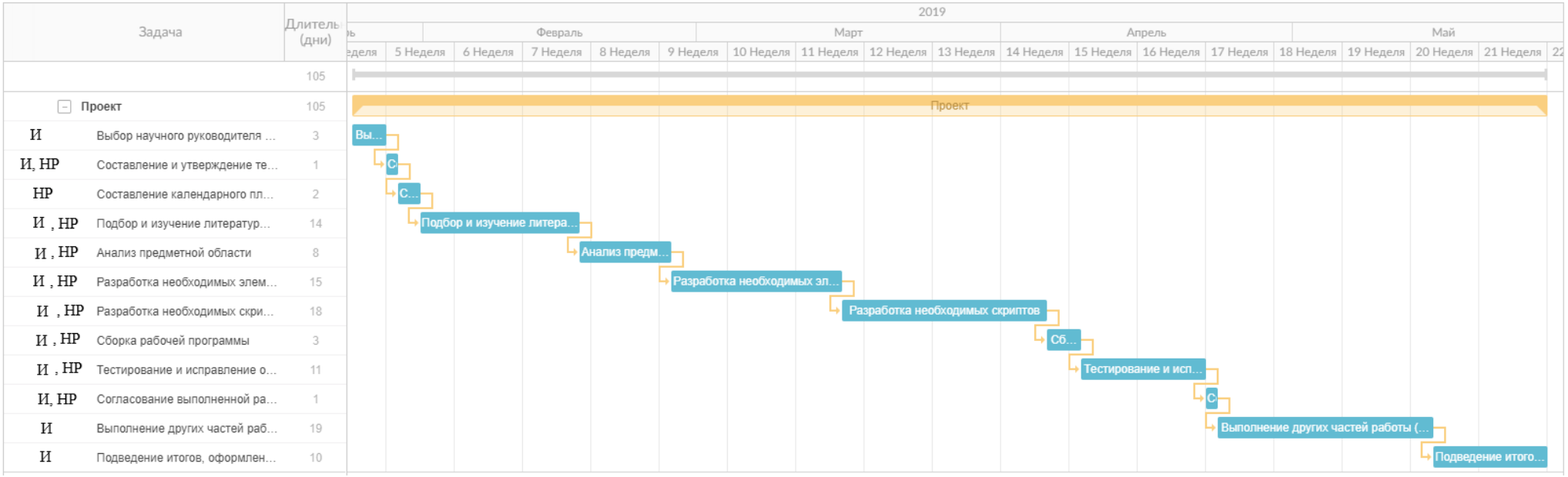
Просчитанные значения в календарных днях приведены в таблице 4

Таблица 4 – Временные показатели проведения научного исследования

| Наименование работы | Исполнители работы | Трудоемкость работ, чел-дни | | | Длительность работ, дни | |
|---|---|---|---|---|---|---|
| | | tmin | tmax | tож | Тр | Тк |
| Выбор научного руководителя бакалаврской работы | Ванюшин И.С. | 1 | 3 | 1.8 | 2 | 3 |
| Составление и утверждение темы бакалаврской работы | Ванюшин И.С. | 2 | 4 | 2.8 | 1 | 1 |
| | Фофанов О.Б. | 1 | 2 | 1.4 | 1 | 1 |
| Составление календарного плана-графика выполнения бакалаврской работы | Фофанов О.Б. | 2 | 3 | 2.4 | 2 | 2 |
| Подбор и изучение литературы по теме бакалаврской работы | Ванюшин И.С. | 10 | 14 | 11.6 | 12 | 15 |
| | Фофанов О.Б. | 1 | 2 | 1.4 | 1 | 1 |
| Анализ предметной области | Ванюшин И.С. | 7 | 10 | 8.2 | 8 | 10 |
| | Фофанов О.Б. | 1 | 2 | 1.4 | 1 | 1 |
| Разработка необходимых элементов программы | Ванюшин И.С. | 12 | 18 | 14.4 | 14 | 17 |
| | Фофанов О.Б. | 2 | 3 | 2.4 | 2 | 2 |
| Разработка необходимых скриптов | Ванюшин И.С. | 14 | 21 | 16.8 | 17 | 21 |
| | Фофанов О.Б. | 2 | 3 | 2.4 | 2 | 2 |
| Сборка рабочей программы | Ванюшин И.С. | 2 | 4 | 2.8 | 3 | 4 |
| | Фофанов О.Б. | 2 | 3 | 2.4 | 2 | 2 |
| Тестирование и исправление ошибок | Ванюшин И.С. | 7 | 14 | 9.8 | 10 | 12 |
| | Фофанов О.Б. | 2 | 3 | 2.4 | 2 | 2 |
| Согласование выполненной работы с научным руководителем | Ванюшин И.С. | 1 | 2 | 1.4 | 1 | 1 |
| | Фофанов О.Б. | 1 | 2 | 1.4 | 1 | 1 |
| Выполнение других частей работы | Ванюшин И.С. | 14 | 21 | 16.8 | 17 | 21 |
| Подведение итогов, оформление работы | Ванюшин И.С. | 7 | 12 | 9 | 9 | 11 |
| **Итого** | **Ванюшин И.С.** | | | | **94** | **120** |
| | **Фофанов О.Б.** | | | | **14** | **14** |

Для наглядности представленных данных построена Диаграмма Гантта для данных расчётов. В качестве условных обозначений использовались следующие установки: И – исполнитель (Ванюшин И.С.), НР – научный руководитель (Фофанов О.Б.) Работа над проектом велась с 24 января 2019 по 25 мая 2019 года.

Таблица 5 – Диаграмма Гантта



## 4.5 Бюджет научно-технического исследования

### 4.5.1 Расчет материальных затрат научно-технического исследования

Данная статья затрат включает в себя затраты на приобретение сырья, материалов, полуфабрикатов и комплектующих со стороны. Также в эту статью включаются транспортные расходы, равные 15% от общей стоимости материальных затрат.

Расчет материальных затрат осуществляется по следующей формуле:

$$З_м = (1 + k_T) \cdot \sum_{i=1}^{m} Ц_i \cdot N_{расхi}, \quad (5)$$

где $m$ – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расхi}$ – количество материальных ресурсов $i$-го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м$^2$ и т.д.);

$Ц_i$ – цена приобретения единицы $i$-го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м$^2$ и т.д.);

$k_T$ – коэффициент, учитывающий транспортно-заготовительные расходы.

Для выполнения данной работы были приобретены только канцелярские принадлежности в сумме на 1260 рублей.

### 4.5.2 Расчет материальных затрат на специальное оборудование для научных (экспериментальных) целей

В ходе выполнения научно-исследовательской работы использовалось оборудование, имеющееся лично у студента. Далее приведены расчеты амортизации используемого оборудования за время работы.

Срок полезного использования для офисных машин (код 330.28.23.23) составляет 3 года, ПК был использован на протяжении 4 месяцев, его цена составляет 75000 рублей.

Норма амортизации:

$$Ан = \frac{1}{n} * 100\% = \frac{1}{3} \times 100\% = 33{,}33\%$$

Годовые амортизационные отчисления:

$$Аг = 75000 \times 0{,}33 = 27450 \text{ рублей}$$

Ежемесячные амортизационные отчисления:

$$Ам = \frac{27450}{12} = 2287{,}5 \text{ рублей}$$

Итоговая сумма амортизации основных средств:

$$А = 2287{,}5 \times 4 = 9150 \text{ рублей}$$

Также для выполнения проекта была приобретена платная версия ПО, сроком на 6 месяцев стоимостью 48750 рублей.

Норма амортизации:

$$Ан = \frac{1}{n} * 100\% = \frac{1}{0.5} \times 100\% = 200\%$$

Годовые амортизационные отчисления:

$$Аг = 48750 \times 2 = 97500 \text{ рублей}$$

Ежемесячные амортизационные отчисления:

$$Ам = \frac{27450}{12} = 8125 \text{ рублей}$$

Итоговая сумма амортизации нематериальных активов:

$$А = 8125 \times 4 = 32500 \text{ рублей}$$

Итоговый расчет затрат на амортизацию представлен в таблице 6.

Таблица 6 – Расчет затрат на амортизацию

| Наименование | Затраты, руб. |
|---|---|
| Амортизация ПК | 9150 |
| Амортизация ПО | 32500 |

### 4.5.3 Основная заработная плата исполнителей темы

Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя. Месячный оклад (МО) НР, занимающего должность доцента и

имеющего степень кандидата технических наук, составляет 33664 руб./мес. МО исполнителя, являющегося студентом, составляет 21760 руб./мес.

Статья включает основную заработную плату работников, непосредственно занятых выполнением НТИ, (включая премии, доплаты) и дополнительную заработную плату:

$$З_{зп} = З_{осн} + З_{доп}, \quad (6)$$

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

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

Основная заработная плата ($З_{осн}$) руководителя рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \cdot T_р, \quad (7)$$

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

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

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

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_м \cdot М}{F_д}, \quad (8)$$

где $З_м$ – месячный должностной оклад работника, руб.;

М – количество месяцев работы без отпуска в течение года:

при отпуске в 48 раб. дней М=10,4 месяца, 6-дневная неделя;

$F_д$ – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн.

Таблица 7 – Баланс рабочего времени для 6-дневной недели

| Показатели рабочего времени | Дни |
|---|---|
| Календарные дни | 365 |
| Нерабочие дни (праздники/выходные) | 66 |
| Потери рабочего времени (отпуск/невыходы по болезни) | 56 |
| Действительный годовой фонд рабочего времени | 243 |

Расчет дневной заработной платы студента и руководителя:

(21760*10.4)/243 = 931.29 рублей (для студента)

(33664*10.4)/243 = 1440.76 рублей (для научного руководителя)

Расчет основной заработной платы студента и руководителя:

931.29*1.3*94 = 113 803.64 рублей (для студента)

1440.76*14*1.3*(1+0.3+0.2) = 39332,75 рублей (для руководителя)

Таблица 8 – Расчет основной заработной платы

| Исполнители | Здн, руб. | Кпр | Кд | Кр | Тр | Зосн |
|---|---|---|---|---|---|---|
| Студент | 931,29 | 0 | 0 | 1,3 | 94 | 113803,64 |
| Научный руководитель | 1440,76 | 0,3 | 0,2 | 1,3 | 14 | 39332,75 |
| | | | | | Итого: | 153136,388 |

### 4.5.4 Дополнительная заработная плата исполнителей темы

Расчет дополнительной заработной платы ведется по следующей формуле:

$$З_{доп} = k_{доп} \cdot З_{осн} \quad (9)$$

где $k_{доп}$ – коэффициент дополнительной заработной платы (0,12 – 0,15).

Расчет дополнительной заработной платы студента и руководителя:

113803,64*0,15 = 17070,55 рублей (для студента)

39332,75*0,15 = 5899,91 рублей (для руководителя)

Таким образом, для студента дополнительная заработная плата составит 17070,55 рублей, для научного руководителя – 5899,91 рубля

### 4.5.5 Отчисления во внебюджетные фонды (страховые отчисления)

Эта статья включает обязательные отчисления по установленным законодательством РФ нормам органам ФСС, ПФ и ФФОМС от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп}), \quad (10)$$

Расчет отчислений для студента и руководителя:

0,28*(113803,64 + 17070,55) = 36 644,77 рублей (для студента)

0,28*(39332,75 + 5899,91) = 12 665,14 рублей (для руководителя)

Таким образом, отчисления для студента составят 36 644,77 рублей, для научного руководителя – 12 665,14 рублей.

### 4.5.6 Накладные расходы

Накладные расходы учитывают прочие затраты, не попавшие в предыдущие статьи расходов.

Их величина определяется по следующей формуле:

$З_{накл}$ = (сумма статей с 1 по 5) * $k_{нр}$ (11)

где $k_{нр}$ – коэффициент, учитывающий накладные расходы (16%).

Расчет накладных расходов:

0,16*(1260+9150+32500+113803,64+39332,75+17070,55+5899,91+3626 2,26+13569,80) = 43015,83 рублей

В итоге, величина накладных расходов составляет 43015,83 рублей

### 4.5.7 Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно-исследовательской работы (темы) является основой для формирования бюджета затрат проекта.

Бюджет затрат на выполняемый проект приведен в таблице 9.

Таблица 9 – Расчет бюджета затрат НТИ

| Наименование | Сумма, руб. | Удельный вес, % |
|---|---|---|
| Материальные затраты | 1260 | 0,40 |
| Затраты на специальное оборудование | 41650 | 13,36 |
| Затраты на основную заработную плату | 153136,388 | 49,10 |
| Затраты на дополнительную заработную плату | 22970,46 | 7,37 |
| Страховые взносы | 49832,06 | 15,98 |
| Накладные расходы | 43015,83 | 13,79 |
| **Общий бюджет** | **311864,74** | **100%** |

### 4.6 Определение потенциального эффекта исследования

Определим эффективность научного исследования на основе расчета интегрального показателя эффективности научного исследования.

Коэффициент научной (научно-технической) результативности определяется по формуле $E = \sum_{i=1}^{k} E_i K_i$, где:

k – число оцениваемых параметров;

Ei – коэффициент значимости фактора (влияние i-го параметра на научную (научно-техническую) результативность);

Ki – коэффициент достигнутого уровня i-го параметра.

Воспользуемся характеристиками факторов из таблицы 10 для оценки коэффициента научно-технический результативности.

Таблица 10 – Характеристики факторов и признаков научно-технической результативности прикладной НИР

| **Фактор научно-технической результативности** | **Коэффициент значимости фактора** | **Качество фактора** | **Характеристика фактора** | **Коэффициент достигнутого уровня** |
|---|---|---|---|---|
| Перспективность использования результатов | 0,5 | Первостепенная | Результаты могут найти применение во многих научных направлениях | 1,0 |
| | | Важная | Результаты будут использованы при разработке новых технических решений | 0,8 |
| | | Полезная | Результаты будут использованы при последующих НИР и разработках | 0,5 |
| Масштаб реализации результатов | 0,3 | Национальная экономика | Время реализации: до 3 лет<br>до 5 лет<br>до 10 лет<br>свыше 10 лет | 1,0 0,8 0,6 0,4 |
| | | Отрасль | Время реализации: до 3 лет<br>до 5 лет<br>до 10 лет<br>свыше 10 лет | 0,8 0,7 0,5 0,3 |
| | | Отдельные фирмы и предприятия | Время реализации: до 3 лет<br>до 5 лет<br>до 10 лет<br>свыше 10 лет | 0,4 0,3 0,2 0,1 |
| Завершенность результатов | 0,2 | Высокая | Техническое задание на опытно- конструкторские работы | 1 |
| | | Средняя | Рекомендации, развернутый анализ, предложения | 0,6 |
| | | Недостаточная | Обзор, информация | 0,4 |

Проведем необходимые расчеты исходя из оценок экспертов, непосредственно участвовавших в процессе проведения исследования:

$$E = E_1K_1 + E_2K_2 + E_3K_3 = = 0.5 \cdot 1.0 + 0.3 \cdot 0.4 + 0.2 \cdot 0.6 =$$
$$= 0.5 + 0.12 + 0.12 = 0.74$$

Показатель научно технической результативности $E = 0.74$, что говорит о довольно высокой качественной характеристике проведенного исследования.

Данный проект потребует финансовых затрат в сумме 311864,74 рублей и займет 120 календарный дней на выполнение.

Сам проект изначально имеет узкий круг потребителей, но способен выделиться из-за отсутствия конкурентов в данной области. Если же результаты окажутся положительными, то имеются весомые шансы увеличения спроса на данный программный продукт, и, как следствие, продолжится развитие проекта в новых направлениях.

В результате использования данный проект позволить упростить процесс анализа текстовых данных любого вида за счет отсутствия необходимости ручного анализа каждого текста в отдельности.

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА
## «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

| Группа | ФИО |
|---|---|
| 8К51 | Ванюшину Ивану Сергеевичу |

| Школа | ИШИТР | Отделение (НОЦ) | ОИТ |
|---|---|---|---|
| Уровень образования | бакалавриат | Направление/специальность | 09.03.04 Программная инженерия |

Тема ВКР:

| | |
|---|---|
| **Визуализация движения подводного аппарата по имеющейся траектории** | |
| **Исходные данные к разделу «Социальная ответственность»:** | |
| 1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения | *Программное обеспечение, предназначенное для использования на персональных компьютерах для облегчения проведения анализа текстовых данных. Рабочим местом выступает письменный стол, ПК и соответствующее оборудование* |
| Перечень вопросов, подлежащих исследованию, проектированию и разработке: | |
| **1. Правовые и организационные вопросы обеспечения безопасности:**<br>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;<br>– организационные мероприятия при компоновке рабочей зоны. | – Рабочее место при выполнении работ сидя регулируется ГОСТом 12.2.032 –78<br>– Организация рабочих мест с электронно-вычислительными машинами регулируется СанПиНом 2.2.2/2.4.1340 – 03<br>– Рациональная организация труда в течение рабочего времени предусмотрена Трудовым Кодексом РФ ФЗ-197 |
| **2. Производственная безопасность:**<br>2.1. Анализ выявленных вредных и опасных факторов<br>2.2. Обоснование мероприятий по снижению воздействия | – Повышенный уровень электромагнитных излучений<br>– Отклонение показателей микроклимата<br>– Недостаточная освещенность рабочей зоны<br>– Повышенный уровень шума на рабочем месте<br>– Монотонность труда<br>– Эмоциональные перегрузки<br>– Электробезопасность |
| **3. Экологическая безопасность:** | – Загрязнение окружающей среды при утилизации ПК и его частей |
| **4. Безопасность в чрезвычайных ситуациях:** | – Возникновение пожара |

| | |
|---|---|
| **Дата выдачи задания для раздела по линейному графику** | |

**Задание выдал консультант:**

| Должность | ФИО | Ученая степень, звание | Подпись | Дата |
|---|---|---|---|---|
| Доцент ООД | Винокурова Г.Ф. | К.т.н. | | |

**Задание принял к исполнению студент:**

| Группа | ФИО | Подпись | Дата |
|---|---|---|---|
| 8К51 | Ванюшин И.С. | | |

# Раздел 5. Социальная ответственность

## Введение

Данный раздел включает в себя выявление, анализ вредных и опасных факторов, связанных с разработкой (этап изготовления отсутствует в силу специфики работы) и эксплуатацией программного обеспечения. Рассмотрены вопросы организации безопасных условий труда, охраны окружающей среды.

Объектом исследования в данном разделе выступает разрабатываемое программное обеспечение, предназначенное для использования на персональном компьютере для облегчения проведения анализа текстовых данных.

Рабочие места как разработчика, так и пользователя программного обеспечения, идентичны и представляют собой стационарное место, оборудованное персональным компьютером и оргтехникой.

В качестве рабочей зоны используется учебная аудитория в Кибернетическом центре ТПУ с системой отопления, кондиционером, естественным и искусственным освещением.

## 5.1 Правовые и организационные вопросы обеспечения безопасности

Для организации комфортного и безопасного рабочего места используются требования ГОСТ 12.2.032-078 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования», ГОСТ 12.2.061-81 «ССБТ. Оборудование производственное. Общие требования безопасности к рабочим местам».

Работа с персональным компьютером сопровождается значимыми зрительными, нервно-психологическими нагрузками. Рабочая мебель должна иметь возможность индивидуальной регулировки для соответствия росту рабочего, организации удобного положения тела.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количественных и конструктивных особенностей, а также характера выполняемой работы. Высота рабочей поверхности стола должна регулироваться в пределах 680-800мм, или же 725мм, если нет возможности регулировки.

Согласно требований СанПиН 2.2.2/2.4.1340 – 03 при организации работы на ПЭВМ должны выполняться следующие условия:

-персональный компьютер (ПК), и, соответственно, рабочее место должно располагаться так, чтобы свет падал сбоку, лучше слева;

-расстояние от ПК до стен должно быть не менее 1 м, поэтому по возможности следует избежать расположения рабочего места в углах помещения либо лицом к стене;

-ПК лучше установить так, чтобы, подняв глаза от экрана, можно было увидеть какой-нибудь удаленный предмет в помещении или на улице. Перевод взгляда на дальнее расстояние является одним из наиболее эффективных способов разгрузки зрительного аппарата при работе на ПК;

-окна в помещениях с ПЭВМ должны быть оборудованы регулируемыми устройствами (жалюзи, занавески, внешние козырьки и т.д.);
-монитор, клавиатура и корпус компьютера должны находиться прямо перед оператором; высота рабочего стола с клавиатурой должна составлять 680 – 800 мм над уровнем стола; а высота экрана (над полом) –900–1280см;

-монитор должен находиться от оператора на расстоянии 60 – 70 см на 20 градусов ниже уровня глаз;

-рабочее кресло должно иметь мягкое сиденье и спинку, с регулировкой сиденья по высоте, с удобной опорой для поясницы; 71 - положение тела пользователя относительно монитора должно соответствовать направлению просмотра под прямым углом или под углом 75 градусов.

## 5.2 Анализ опасных и вредных производственных факторов и обоснование мероприятий по снижению их воздействия

В данном разделе рассмотрены вредные и опасные факторы, которые могут возникать при разработке и эксплуатации программного обеспечения.

Для их идентификации использован ГОСТ 12.0.003-2015 «Опасные и вредные производственные факторы. Классификация». Их перечень представлен в таблице 11.

Таблица 11 – Возможные опасные и вредные факторы

| Факторы (ГОСТ 12.0.003-2015) | Этапы работ | | Нормативные документы |
|---|---|---|---|
| | Разработка | Эксплуатация | |
| 1.Отклонение показателей микроклимата | + | + | СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»<br>СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений» |
| 2. Превышение уровня шума | + | + | СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»<br>ГОСТ 12.1.003-2014 «Система стандартов безопасности труда. Шум. Общие требования безопасности» |
| 3.Недостаточная освещенность рабочей зоны | + | + | СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» |
| 4.Возможность возникновения короткого замыкания | + | + | ГОСТ IEC 60950-1-2014 «Оборудование информационных технологий. Требования к безопасности. Часть 1. Общие требования» |
| 5.Повышенный уровень электромагнитных излучений | + | + | СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы» |

**Отклонение показателей микроклимата**

Отклонение показаний микроклимата может привести к возникновению заболеваний органов дыхания, сердечно-сосудистой системы (если показатели меньше оптимальных), негативно сказаться на двигательной реакции, координации и качеству выполнения точечных операций работником. В случае превышения оптимальных показателей также снижаются работоспособность и производительность труда, возможно появление головной боли, слабости, головокружения и теплового удара.

Воздушная среда в рабочем помещении должна обеспечивать тепловой комфорт работникам в течение всего рабочего времени, не вызывать каких-либо отклонений в состоянии здоровья. Энергетические затраты человеческого организма измеряются в ккал/ч или же в Вт. Работа программиста относится к категории 1а с энергозатратами до 120 ккал/ч или же 139 Вт.

Программист работает сидя и подвергается небольшим физическим напряжениям. Согласно СанПиН 2.2.2/2.4.1340-03 нормы микроклимата для таких работ должны соответствовать оптимальным показателям на рабочих местах производственных помещений. Оптимальные и допустимые показатели приведены в таблице 12 и таблице 13.

Таблица 12 – Оптимальные показатели микроклимата по СанПиН 2.2.4.548-96

| Период года | Категория работ | Температура воздуха °С | Температура поверхностей °С | Относительная влажность воздуха % | Скорость движения воздуха м/с |
|---|---|---|---|---|---|
| Холодный | 1а | 22-24 | 21-25 | 60-40 | 0.1 |
| Теплый | 1а | 23-25 | 22-26 | 60-40 | 0.1 |

Таблица 13 – допустимые показатели микроклимата по СанПиН 2.2.4.548-96

| Период года | Категория работ | Температура воздуха °С | Температура поверхностей °С | Относительная влажность воздуха % | Скорость движения воздуха м/с |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| Холодный | 1а | 20-21.9/24.1-25.0 | 19-26 | 15-75 | 0.1/0.1 |
| Теплый | 1а | 21.0-22.9/25.1-28.0 | 20-29 | 15-75 | 0.1/0.2 |

В таблице 13 для температуры и скорости воздуха параметры указаны в следующем формате: диапазон ниже оптимальных величин/выше оптимальных величин.

Параметры микроклимата в рабочем помещении регулируются системой центрального отопления, кондиционером, естественной вентиляцией с параметрами: влажность 40%, скорость движения воздуха 0.1 м/с, температура 22-25 градусов Цельсия, что соответствует нормам.

К методам оздоровления и поддержания необходимого состояния микроклимата относятся: грамотная организация системы вентиляции помещения, отопление.

**Превышение уровня шума**

Воздействие шума оказывает значительную нагрузку на нервную систему работника, что может привести к нервозу или стрессу. Длительное воздействие шумов приводит к ухудшению состояния слуховых органов человека, возникновению головных болей.

Источником шума на рабочем месте в данном случае может выступать персональный компьютер, система вентиляции или иное оборудование.

Характеристикой шума выступает уровень звукового давления в децибелах в октавных полосах частот со среднегеометрическими частотами 31,5; 63; 125; 250; 500; 1000; 2000; 4000; 8000 Гц, определяемые по формуле:

$$L=20\times\lg(P/P0), \quad (12)$$

где P – среднеквадратичная величина звукового давления, Па;

P0 = 2 × 10-5 Па – исходное значение звукового давления в воздухе. Шум от работающего компьютера создаёт P = 0,05 Па. Таким образом, L = 68 дБА, что не превышает требованию ГОСТ 12.1.003-2014 в 75 дБА.

В качестве мероприятий по снижению уровня вредного воздействия шума следует применять звукоизоляцию, рациональный режим труда и отдыха, подавления шума в источниках.

### Недостаточная освещенность рабочей зоны

Правильное освещение рабочей зоны является важным условием для поддержания безопасных и комфортных условий труда. Недостаток света может вызвать слепоту, привести к быстрому утоплению и снизить работоспособность человека. Помимо этого, растет вероятность ошибочных действий.

Для создания равномерной освещенности рабочей зоны используются источники искусственного света, которые желательно располагать в непрерывный сплошной ряд вдоль длинной стороны помещения, а также окна для организации естественного света.

В соответствии с СанПиН 2.2.2/2.4.1340-03 освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк, освещенность поверхности экрана не должна быть больше 300 лк.

Реальная освещенность в аудитории Кибернетического центра, где выполнялась работа, соответствовали указанным требованиям.

Для поддержания необходимых показателей рекомендуется поддерживать окна и светильники в чистоте, своевременно проводить замену перегоревших ламп.

### Повышенный уровень электромагнитных излучений

Используемый в работе ПК производит электромагнитное излучение, воздействие которого на организм человека зависит от напряженности электрического и магнитного полей, потока энергии, частоты колебаний, размеров облучаемого тела.

Нарушения в человеческом организме от действия электромагнитного поля обратимы, но при превышении показателей наносится достаточный ущерб нервной системе, сердечно-сосудистой системе и органам ЖКТ.

Согласно требованиям СанПиН 2.2.2/2.4.1340-03 уровни ЭМП на рабочих местам должны соответствовать уровням, приведенным в таблице 14.

Таблица 14 – Временные допустимые уровни ЭМП

| Название параметров | | ВДУ |
|---|---|---|
| Напряженность электрического поля | В диапазоне частот 5 Гц – 2 кГц | 25 В/м |
| | В диапазоне частот 2 кГц – 400 кГц | 2.5 В/м |
| Плотность магнитного потока | В диапазоне частот 5 Гц – 2 кГц | 250 нТл |
| | В диапазоне частот 2 кГц – 400 кГц | 25 нТл |
| Напряженность электростатического поля | | 15 кВ/м |

Уровень электромагнитного излучения на рабочем месте не превышает требований, следовательно, соответствует требованиям СанПиН 2.2.2/2.4.1340-03.

В качестве методов по нейтрализации вредного воздействия ЭМП на работника используется увеличение расстояния работника от основных источников излучения (системный блок, кабели, монитор должен располагаться на расстоянии больше 50см), использование средств индивидуальной защиты, чередование работы и отдыха.

**Возможность возникновения короткого замыкания**

Короткое замыкание происходит чаще всего из-за нарушения изоляции токопроводящих частей в результате разного рода воздействий. При

возникновении короткого замыкания резко возрастает сила тока, и, как следствие, количество выделяемого тепла, что приводит к возгоранию.

Согласно ГОСТ IEC 60950-1-2014 для предотвращения пробоя изоляции и недопущения короткого замыкания следует предусматривать основную изоляцию с заземлением доступных токопроводящих частей. Также возможно использование двойной или усиленной изоляции между частями, находящимися при нормальной работе под опасным напряжением и доступными токопроводящими частями.

Для предотвращения огнеопасных ситуаций, связанных с коротким замыканием, данный ГОСТ рекомендует следующие меры:

- защита от перегрузки по току
- использование материалов соответствующего класса воспламеняемости
- правильный выбор компонентов для предотвращения повышения температуры
- ограничение использования горючих материалов
- экранирование
- использование для корпусов оборудования соответствующих материалов.

## 5.4 Экологическая безопасность

Разрабатываемое программное обеспечение не несет какого-либо вреда частям окружающей среды, однако его использование невозможно без персонального компьютера, поэтому наносимый разработкой вред следует рассматривать с точки зрения вреда персонального компьютера окружающей среде.

Согласно ГОСТ Р 56397-2015 «Техническая экспертиза работоспособности радиоэлектронной аппаратуры, оборудования информационных технологий, электрических машин и приборов. Общие требования» пункт 5.8.1, после проведения технической экспертизы, если

оборудование не поддается ремонту, то оно признается неработоспособным и рекомендуется к списанию (замене); в случае отказа оборудования и нецелесообразности его ремонта и модернизации даются рекомендации о необходимости его списания и утилизации.

Согласно «Методики проведения работ по комплексной утилизации вторичных драгоценных металлов из отработанных средств вычислительной техники», утвержденной Государственным Комитетом РФ по телекоммуникациям от 19 октября 1999 г. В п.3.1.3. «Технология разборки универсальных ЭВМ» расписаны 4 этапа разборки и подготовки к утилизации внутренних частей ПК.

### 5.5 Безопасность в чрезвычайных ситуациях

При разработке и эксплуатации рассматриваемого программного обеспечения могут возникнуть определенные чрезвычайные ситуации, связанные с персональным компьютером, используемых для разработки и эксплуатации ПО.

Использование персонального компьютера может привести возгоранию, спровоцированного различными опасными факторами. Например, источниками пожара могут стать провода, внутренние части работающего устройства, периферийные устройства.

В случае угрозы возникновения ЧС необходимо отключить электропитание, вызвать по телефону пожарную команду, эвакуировать людей из помещения согласно плану эвакуации. При наличии небольшого очага пламени можно воспользоваться подручными средствами с целью прекращения доступа воздуха к объекту возгорания. В качестве подручных средств можно использовать углекислотные огнетушители ОУ-5 высокого давления с зарядом жидкой двуокиси углерода (по ГОСТ 8050-85 [11]), расположение которых можно найти на плане эвакуации людей при пожаре и других ЧС из помещения Кибернетического центра ТПУ.

В качестве превентивных мер может служить проверка состояния оборудования и его частей на наличие повреждений или неисправностей,

своевременное их исправление; использование систем звукового и визуального оповещения персонала об опасности, обучение персонала методам работы с ПК, наличие средств пожаротушения, информационных досок с планами эвакуации.

**Выводы по разделу**

В данной главе были рассмотрены особенности рабочей зоны и рабочего место программиста-разработчика и пользователя разрабатываемого программного обеспечения.

Указаны опасные факторы, связанные с использованием/разработкой программного комплекса, подробно рассмотрено их опасное воздействие, а также перечислены методы для профилактики/ликвидации вреда этих факторов человеку.

Рассмотрены вопросы, касающиеся влияния разработки на экологию и возникновения определенных чрезвычайных ситуаций.

В результате анализа было установлено, что используемая аудитория Кибернетического центра ТПУ удовлетворяет всем необходимым требованиям, соответственно, ее можно использовать в качестве рабочего места.

## Заключение

В результате было проведено исследование, которое показало теоретическую возможность составления цельной модели, решающую задачу агрегирования текстов и тематического моделирования.

Результатом исследования стал проект иерархической модели на основе алгоритма МГУА, который способен агрегировать тексты для решения задачи тематического моделирования. Был реализован элемент низкоуровневого моделирования, генерирующий множество подмоделей, описывающих исходные данные. В дальнейшем это множество моделей будет руководиться более высокоуровневой моделью, создание которой ставится целью будущей магистерской работы.

Процесс разработки подразумевал такие этапы как:

- Рассмотрение МГУА подхода, теоретическое обоснование применимости этой модели к данной задачи.
- Проектирование решения, включающего в себя МГУА подход для агрегирования текстов
- Реализация низкоуровневого модуля создания моделей

Исследование имеет большой потенциал для дальнейшего продолжения работы, что позволит реализовать проект до конца и предоставить новый класс моделей тематического моделирования для использования.

## Список использованных источников


1. Коршунов А., Гомзин А. Тематическое моделирование текстов на естественном языке //Труды Института системного программирования РАН. – 2012. – Т. 23.
2. Карпович С. Н. Русскоязычный корпус текстов SCTM-RU для построения тематических моделей //Труды СПИИРАН. – 2015. – Т. 2. – №. 39. – С. 123-142.
3. Воронцов К. В. Вероятностное тематическое моделирование [Электронный ресурс] //URL: http://www. machinelearning. ru/wiki/images/2/22/Voron-2013-ptm. pdf (дата обр. 16.04. 2016). – 2013.Sue Blackman. Unity for absolute beginners. Apress publishing, June 2014. – 575 с.
4. Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman. Indexing by Latent Semantic Analysis // JASIS (41) 1990 pp. 391-407.
5. Белов С. П., Плискин Е. Л., Усков А. В. Опыт получения и использования наукометрической информации в системах управления научной деятельностью //Труды Института системного анализа Российской академии наук. – 2015. – Т. 65. – №. 2. – С. 33-38.
6. Васильев В. И. Взаимодополняемость метода группового учета аргументов (МГУА) и метода предельных упрощений (МПУ) //Искусственный интеллект. – 2001. – Т. 1. – С. 29-42.
7. Орлов А. А. Принципы построения архитектуры программной платформы для реализации алгоритмов метода группового учета аргументов //Управляющие системы и машины. – 2013.
8. Степашко В. С., Булгакова А. С. Обобщенный итерационный алгоритм метода группового учета аргументов //Управляющие системы и машины. – 2013.

# Приложение А

Model.cpp

```
#include "Model.h"
#include <sstream>

namespace Model
{
    size_t CompleteModelDraftGenerator::TotalNumber() const
    {
        _state.AssertInitialized();

        return _totalDrafts;
    }

    void CompleteModelDraftGenerator::Initialize(Common::DataFactors::tCount
numFactors)
    {
        //Generate factors
        _factorsGen->Generate(numFactors, _genFactorsArray);

        //Impl
        InitializeImplementation();
    }
    void CompleteModelDraftGenerator::Initialize(std::shared_ptr<Common::DataFactors>
pFactors)
    {
        //Generate factors
        _factorsGen->Generate(pFactors, _genFactorsArray);

        //Impl
        InitializeImplementation();
    }
    void CompleteModelDraftGenerator::InitializeImplementation()
    {
        //Extract and shuffle factors counts
        Common::DataFactors::tCount nGenFactors = _genFactorsArray.size();
        std::valarray<Common::DataFactors::tCount>
factorsCount((Common::DataFactors::tCount)0, nGenFactors);
        _factorsCountMap.clear();
        for (Common::DataFactors::tCount f = 0; f < nGenFactors; f++)
        {
            factorsCount[f] = _genFactorsArray[f]->GetCount();
            _factorsCountMap[factorsCount[f]]++;
        }

        _factorsIndicesMap.clear();
        _structuresNumberMap.clear();
        _totalDrafts = 0;
        for (auto i = _factorsCountMap.begin(); i != _factorsCountMap.end(); i++)
        {
            const Common::DataFactors::tCount& count = i->first;
            std::valarray<size_t>& v = _factorsIndicesMap[count];
            v.resize(i->second);
            size_t w = 0;
            for (Common::DataFactors::tCount f = 0; f < nGenFactors; f++)
                if (factorsCount[f] == count)
                    v[w++] = f;

            _structuresNumberMap[count] = _structureGen->TotalNumber(count);

            _totalDrafts += _factorsCountMap[count] * _structuresNumberMap[count];
        }

        //go to state
        _state.GoToInitialized();
```

```
    }

    size_t CompleteModelDraftGenerator::Generate(size_t maxDrafts, std::vector<
tModelDraftPtr >& out)
    {
        _state.GoToGenerating();

        //Simple implementation
        if (_totalDrafts > maxDrafts)
        {
            std::ostringstream ostr;
            ostr << "Can't generate " << _totalDrafts << " drafts: limit is " <<
maxDrafts << " : " << __func__;
            throw ostr.str().c_str();
        }

        out.resize(_totalDrafts);
        size_t draftIdx = 0;

        //Generate drafts
        for (auto i = _factorsCountMap.begin(); i != _factorsCountMap.end(); i++)
        {
            const Common::DataFactors::tCount& curFactorsCount = i->first;
            const Common::DataFactors::tCount& curFactorsNumber = i->second;
            const std::valarray<size_t>& curFactorsIndices =
_factorsIndicesMap[curFactorsCount];
            const size_t& curStructuresNumber = _structuresNumberMap[curFactorsCount];

            //Init generator
            _structureGen->Initialize(curFactorsCount);

            //Generate structures
            std::vector<tStructurePtr> genStructuresArray;
            _structureGen->Generate(curStructuresNumber, genStructuresArray);

            //Combine
            for (Common::DataFactors::tCount f = 0; f < curFactorsNumber; f++)
            {
                for (size_t s = 0; s < genStructuresArray.size(); s++)
                {
                    out[draftIdx] = std::make_shared<ModelDraft>();
                    out[draftIdx]->_factors = _genFactorsArray[curFactorsIndices[f]];
                    out[draftIdx]->_structure = genStructuresArray[s];
                    draftIdx++;
                }
            }
        }

        _state.GoToFinished();

        return _totalDrafts;
    }

}
```

## Приложение Б

Architecture.cpp

```
// Architecture.cpp : Defines the entry point for the console application.
//
#include <iostream>

#include "../../3rdParty/pugixml/pugixml.hpp"
#include "GMDHComputing.h"
#include "Core/Mem.h"
#include "Core/Task.h"
#include <string.h>

using namespace std;

//=============================================================================================
=======================
class MainTask : public Core::Task {
public:
    MainTask(int _argc, char* _argv[]) : Task(), argc(_argc), argv(_argv) {
    }
    ~MainTask() {
    }
    void exec();
private:
    int argc;
    char** argv;
};

void MainTask::exec() {

#if !defined(_WIN64) && !defined(_LINUX64)
    _set_SSE2_enable(1);
#endif // _WIN64

    //GMDH

    //Deprecating old-styled way of configuration, when experiment ID is provided from
keyboard
    // - when no command-line parameters are provided
    if (argc == 1)
    {
        throw("Deprecated");
    }
   else {
        string input_cfg_filename;
        bool helpRequested = false;

        for (int a = 0; a < argc; a++) {
            if ((a + 1) < argc && (!strcmp(argv[a], "-c") || !strcmp(argv[a], "--
config"))) {
                input_cfg_filename = argv[a + 1];
            }
            if (!strcmp(argv[a], "-h") || !strcmp(argv[a], "--help"))
                helpRequested = true;
        }

        if (helpRequested) {
            cout << "Specific command line options:" << endl;
            cout << "-c or --config [input config file name]" << endl;
            cout << endl;
        }
        else {
            if (input_cfg_filename.length() == 0)
                throw("Input config filename is not provided");
```

```
            pugi::xml_document doc;
            pugi::xml_parse_result result = doc.load_file(input_cfg_filename.c_str());

            if (result) {
                std::cout << "XML [" << input_cfg_filename << "] parsed without
errors" << endl;

                GMDHComputing::GMDHComputer gmdhComputer;
                gmdhComputer.ExecuteConfig(doc);
            }
            else {
                std::cout << "XML [" << input_cfg_filename << "] parsed with errors"
<< endl;
                std::cout << "Error description: " << result.description() << endl;
                std::cout << "Error offset: " << result.offset << endl;
                throw "Error reading XML config";
            }

        }
    }

    std::cout << "Press any key..." << std::endl;
    char ccc;
    std::cin >> ccc;
}

//=============================================================================
=======================
int main(int argc, char* argv[]) {

    cout << "Hello, World!" << endl;

    using namespace Core;
    MemObject::Construct();

    cout << sizeof(MemPool::Block) << endl;
    cout << sizeof(MemPool) << endl;
    cout << MemObject::MaxESize << endl;

    //Default is 4 cores
    size_t nThreads = 4;

    bool helpRequested = false;
    //Search for -t or --threads option
    for (int a = 0; a < argc; a++)
    {
        if ((a + 1) < argc && ( !strcmp(argv[a], "-t") || !strcmp(argv[a], "--
threads") ))
        {
            size_t t = atoi(argv[a + 1]);
            if (t > 0)
                nThreads = t;

            break;
        }
        if (!strcmp(argv[a], "-h") || !strcmp(argv[a], "--help"))
            helpRequested = true;
    }

    if (helpRequested) {
        cout << "Global command line options:" << endl;
        cout << "-t or --threads [number of threads]" << endl;
        cout << "-h or --help for help" << endl;
        cout << endl;
    }

    Task::Execute(nThreads, new MainTask(argc, argv));
```

```
    cout << "SUCCESS!" << endl;
    cin.get();
    //Sleep(500);
    //MemObject::Destruct();
    return 0;
}
```