



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	09.04.02 Информационные системы и технологии
Отделение школы (НОЦ)	Информационных технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Разработки алгоритмического и программного обеспечения для укладки графов на плоскости

УДК 004.415:519.173.2:004.932.7

Студент

Группа	ФИО	Подпись	Дата
8ИМ72	Комаров Вячеслав Андреевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Буркатовская Юлия Борисовна	к.ф.-м.н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОСГН	Жаворонок Анастасия Валерьевна			

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД	Атепаева Наталья Александровна			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Шерстнёв Владислав Станиславович	к.т.н.		

Томск – 2019 г.

**ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ОСНОВНОЙ
ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ ПОДГОТОВКИ МАГИСТРОВ
09.04.02 «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»**

Код результатов	Результат обучения	Требования ФГОС, критерии АИОР
<i>Общепрофессиональные компетенции</i>		
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социальноэкономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.	Требования ФГОС 3+ (ОПК-1, ПК 8-12, ОК4), критерий 5 АИОР (п. 1.1), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.	Требования ФГОС 3+ (ОПК5, ПК-7, ОК-3), критерий 5 АИОР (п. 1.1, 1.2), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями	Требования ФГОС 3+ (ОПК2,6, ПК-1, ОК-1), критерий 5 АИОР (п. 1.2), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P4	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями	Требования ФГОС 3+ (ОПК2,6, ПК-1, ОК-1), критерий 5 АИОР (п. 1.2), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
<i>Профессиональные компетенции</i>		
P5	Разрабатывать стратегии и цели проектирования, критерии эффективности и ограничения применимости, новые методы, средства и технологии проектирования геоинформационных систем (ГИС) или промышленного программного обеспечения	Требования ФГОС 3+ (ПК-1,2,3, ОПК-2, ОК-1), критерий 5 АИОР (п.1.3), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P6	Планировать и проводить теоретические и экспериментальные исследования в области создания интеллектуальных ГИС и ГИС технологии или промышленного программного обеспечения с использованием методов системной инженерии.	Требования ФГОС 3+ (ПК-7-13, ОПК-1, ОК-4), критерий 5 АИОР (п. 1.4), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и сопровождения ГИС и ГИС технологий или промышленного программного обеспечения с использованием методов и средств системной инженерии, осуществлять подготовку и обучение персонала.	Требования ФГОС 3+ (ПК-4,17, ОПК-6, ОК-4,7), критерий 5 АИОР (п. 1.5), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P8	Формировать новые конкурентоспособные	Требования ФГОС 3+ (ПК-5,6,14,15,16,

	идеи в области теории и практики ГИС и ГИС технологий или системной инженерии программного обеспечения. Разрабатывать методы решения нестандартных задач и новые методы решения традиционных задач. Организовывать взаимодействие коллективов, принимать управленческие решения, находить компромисс между различными требованиями как при долгосрочном, так и при краткосрочном планировании.	ОПК-1,2, ОК-4), критерий 5 АИОР (п. 1.6), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
<i>Общекультурные компетенции</i>		
P9	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.	Требования ФГОС 3+ (ОК-4,7, ПК-8-12, ОПК-1,6), критерий 5 АИОР (п. 2.1, п. 2.3, п. 1.5), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P10	Свободно пользоваться русским и иностранным языками как средством делового общения.	Требования ФГОС 3+ (ОК-3, ПК-7, ОПК-4,5), критерий 5 АИОР (п. 2.2), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P11	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.	Требования ФГОС 3+ (ОК-1,5, ПК-1, ОПК-2), критерий 5 АИОР (п. 2.4, п. 2.5), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.
P12	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.	Требования ФГОС 3+ (ОК-2,6, ПК-2,3, ОПК-3), критерий 5 АИОР (п. 2.6), соответствующий международным стандартам EUR-ACE и FEANI. Запросы студентов, отечественных и зарубежных работодателей.

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

Школа
 Направление подготовки
 Отделение школы (НОЦ)

Информационных технологий и робототехники
 09.04.02 Информационные системы и технологии
 Информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Шерстнёв В.С.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

магистерской диссертации

Студенту:

Группа	ФИО
8ИМ72	Комарову Вячеславу Андреевичу

Тема работы:

Разработка алгоритмического и программного обеспечения для укладки графов на плоскости	
Утверждена приказом директора (дата, номер)	№ 1291/с от 19.02.2019 г

Срок сдачи студентом выполненной работы:

--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объектом исследования является процесс укладки графов на плоскость.
---------------------------------	---

Перечень подлежащих исследованию, проектированию и разработке вопросов	Исследование планарных графов, алгоритмов определения графов на планарность и укладки их на плоскость. Исследование существующих способов отображения не планарных графов на плоскость. Разработка алгоритма укладки не планарного графа на плоскость. Программная реализация приложения для отображения схем баз данных на плоскости с минимальным количеством пересечений связей (реализация алгоритмов для проверки графа на планарность, укладки планарного графа на плоскость, поиска минимального количества рёбер, после удаления которых граф станет планарным). Тестирование реализованного приложения.
Перечень графического материала	Пояснения по работе разработанного алгоритма для отображения не планарных графов на плоскости, макеты пользовательского интерфейса приложения, результаты тестирования приложения.

Консультанты по разделам выпускной квалификационной работы

(с указанием разделов)

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Жаворонок Анастасия Валерьевна
Социальная ответственность	Атепаева Наталья Александровна
Раздел на иностранном языке	Диденко Анастасия Владимировна Мирошниченко Евгений Александрович

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Разработка оптимального алгоритма укладки не планарного графа на плоскость, с минимальным количеством пересечений рёбер. (Development of an optimal algorithm for laying a non planar graph on a plane with a minimum number of intersections of edges).

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

Задание выдал руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Буркатовская Юлия Борисовна	к.ф.-м.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ72	Комаров Вячеслав Андреевич		

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	09.04.02 Информационные системы и технологии
Уровень образования	Магистратура
Отделение школы (НОЦ)	Информационных технологий
Период выполнения	Осенний/весенний семестр 2018/2019 учебного года

Форма представления работы:

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)
22.10.2018	Анализ предметной области	15
25.02.2019	Разработка алгоритма для укладки не планарного графа на плоскость	20
07.05.2019	Программная реализация приложения для отображения схем баз данных на плоскости с минимальным количеством пересечений рёбер	40
20.05.2019	Тестирование приложения	10
27.05.2019	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	5
03.06.2019	Социальная ответственность	5
10.06.2019	Раздел на иностранном языке	5

СОСТАВИЛ

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Буркатовская Юлия Борисовна	к.ф.-м.н		

СОГЛАСОВАНО

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Шерстнёв Владислав Станиславович	к.т.н.		

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

Студенту:

Группа	ФИО
8ИМ72	Комарову Вячеславу Андреевичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистр	Направление/специальность	09.04.02 Информационные системы и технологии

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

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	1. <i>Бюджет НИИ</i> 2. <i>Расходы на заработную плату</i> 3. <i>Отчисления на социальные нужды и прочее.</i>
2. <i>Нормы и нормативы расходования ресурсов</i>	<i>Тариф на электроэнергию – 5,8 руб.</i>
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	<i>Положения ФЗ РФ №212 от 24 июля 2009 г. «О страховых взносах в пенсионный фонд Российской Федерации, фонд социального страхования Российской Федерации, Федеральный фонд обязательного медицинского страхования»; Общая система налогообложения с учетом льгот для образовательных учреждений (27,1% - отчисления во внебюджетные фонды). Район. Коэф. – 30 %</i>

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Предпроектный анализ</i>	1. <i>Потенциальные потребители результатов исследования</i> 2. <i>QUAD анализ</i> 3. <i>SWOT-анализ</i> 4. <i>Определение возможных альтернатив проведения научных исследований</i>
2. <i>Инициация проекта</i>	1. <i>Определение заинтересованных сторон проекта</i> 2. <i>Определение целей и результатов проекта</i> 3. <i>Определение рабочей группы и ограничений проекта</i>
3. <i>План управления научно-техническим проектом</i>	1. <i>План проекта</i> 2. <i>Бюджет научного исследования</i> 3. <i>Организационная структура проекта</i> 4. <i>План управления коммуникациями проекта</i> 5. <i>Реестр рисков проекта</i> 6. <i>Оценка сравнительной эффективности проекта</i>

Перечень графического материала (с точным указанием обязательных чертежей):

Календарный план для выполнения научно-исследовательского проекта

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОСГН	Жаворонок Анастасия Валерьевна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ72	Комаров Вячеслав Андреевич		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ИМ72	Комаров Вячеславу Андреевичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Магистр	Направление/специальность	09.04.02 Информационные системы и технологии

Исходные данные к разделу «Социальная ответственность»:

Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является программная разработка алгоритмического и программного обеспечения для укладки графов на плоскости. Данное ПО может использоваться всеми программистами и администраторами фирм, в которых есть необходимость отображения больших схем баз данных.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; организационные мероприятия при компоновке рабочей зоны.	К основным нормативным документам относятся: ГОСТ 12.0.003-2015 ССБТ, ТОИ Р-45-084-01, СанПиН 2.2.4.548-96, СН 2.2.4/2.1.8.562-96, СНиП 23-03-2003, СП 52.13330.2016, СанПиН 2.2.2/2.4.1340-03, ГОСТ Р 50571.17-2000, ГОСТ 33875-2016, ГОСТ 12.3.031-83.
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Вредными факторами труда инженера-программиста являются: недостаточная освещенность рабочей зоны; отклонение параметров микроклимата в помещении; повышенный уровень шума. Опасными факторами труда инженера-программиста являются: опасность поражения электрическим током
3. Экологическая безопасность:	Утилизация используемой орг.техники и люминесцентных ламп.
4. Безопасность в чрезвычайных ситуациях:	Чрезвычайная ситуация техногенного характера для данного помещения – пожар. Установка общих правил поведения и рекомендаций во время пожара, план эвакуации.

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

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД	Атепаева Наталья Александровна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИМ72	Комаров Вячеслав Андреевич		

Реферат

Выпускная квалификационная работа содержит 82 страницы, 21 рисунок, 28 таблиц, 22 источника и 1 приложение.

Ключевые слова: визуализация графов, планарный граф, укладка графа на плоскости, отображение не планарного графа на плоскости, отображение диаграмм БД на плоскости с минимальным количеством пересечений связей.

Объектом исследования является процесс укладки графов на плоскость.

Цель работы – разработка алгоритмического и программного обеспечения для укладки графов на плоскости.

В процессе работы изучались способы укладки планарных и не планарных графов на плоскости.

В результате был программно реализован алгоритм для укладки произвольных графов на плоскости с минимальным количеством пересечений рёбер.

Область применения: приложение ориентировано на разработчиков ПО и администраторов БД и предназначено для визуализации диаграмм БД на плоскости в удобном для восприятия виде.

Определения, обозначения, сокращения, нормативные ссылки

Жордановы кривые – кривые без самопересечений.

Граф – объект, представляющий из себя множество вершин и набор ребер, то есть соединений между парами вершин.

Орграф – ориентированный граф, то есть граф, ребрам которого присвоено направление.

Подграф – подмножество вершин и ребер графа, содержащегося в начальном графе.

Порождённый подграф графа — граф, образованный из подмножества вершин графа вместе со всеми рёбрами, соединяющими пары вершин из этого подмножества.

Компонента связности – максимальный (по включению) связный подграф графа.

Точка сочленения – вершина графа, при удалении которой увеличится количество компонент связности.

Мост – ребро, при удалении которого, увеличится количество компонент связности.

Блок графа (или компонента двусвязности) – максимальный подграф графа, не имеющий собственных точек сочленения.

Инцидентность – отношение между ребром и его концевыми вершинами.

Смежные вершины – две вершины, инцидентные одному ребру.

Смежные рёбра – два ребра, инцидентные одной вершине.

Изоморфизм – отношение эквивалентности на классе структур. Два графа изоморфны, если вершинам одного графа можно сопоставить вершины другого графа, так чтобы соединённым вершинам первого графа соответствовали соединённые вершины второго графа и наоборот.

Степень вершины – количество ребер, инцидентных данной вершине.

Путь в графе – последовательность ребер, в которой конец каждого ребра (кроме последнего) совпадает с началом предыдущего.

Цикл в графе – замкнутый путь в графе.

Простой цикл – это цикл без самопересечений.

НИРС – научно-исследовательская работа студентов

Научное исследование – НИ

BFS – Алгоритм поиска в ширину в графе (Breadth-first search)

DFS – Алгоритм поиска в глубину (Depth-first search)

БД – База данных

Содержание

Введение.....	15
1 Проблема визуализации графа.....	17
1.1 Планарный граф	18
1.2 Алгоритмы проверки графа на планарность.....	19
1.3 Алгоритм укладки планарного графа на плоскость	20
1.4 Задача укладки произвольного (не планарного графа) на плоскость.....	21
2 Разработка оптимального алгоритма укладки не планарного графа на плоскость, с минимальным количеством пересечений рёбер.	22
2.1 Описание разработанного алгоритма.....	22
2.2 Алгоритм для нахождения и укладки планарных компонент.....	23
2.2.1 Описание алгоритма последовательного построения.....	23
2.3 Асимптотическая сложность разработанного алгоритма.....	24
2.4 Пример работы разработанного алгоритма.....	25
3 Реализация и тестирование программного продукта для укладки произвольного графа на плоскость	29
3.1 Проектирование интерфейса приложения	29
3.2 Реализация приложения	30
3.2.2 Основные классы и модули.	30
3.2.3 Алгоритм отрисовки изображения графа.....	31
3.3 Тестирование приложения	32
3.3.1 Тестирование парсера для преобразования sql скрипта БД в граф	32
3.3.2 Тестирование функционала определения графа на планарность	32
3.3.3 Тестирование отрисовки графа	34
3.3.4 Тестирование отрисовки диаграммы БД.....	36
4 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	37
4.1 Предпроектный анализ.....	38
4.1.1 Потенциальные потребители результатов исследования	38
4.1.2 QUAD анализ.....	38
4.1.3 SWOT-анализ	39

4.1.4	Определение возможных альтернатив проведения научных исследований	41
4.2	Инициация проекта	42
4.3	Планирование управления научно-техническим проектом	44
4.3.1	План проекта	44
4.3.2	Бюджет научно-технического исследования (НТИ)	47
4.3.3	Организационная структура проекта	50
4.3.4	План управления коммуникациями проекта.....	51
4.3.5	Реестр рисков проекта	52
4.4	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.....	53
5	Социальная ответственность	58
5.1	Правовые и организационные вопросы обеспечения безопасности	59
5.1.1	Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства.....	59
5.1.2	Организационные мероприятия при компоновке рабочей зоны	60
5.2	Профессиональная социальная безопасность	61
5.2.1	Анализ вредных факторов, которые могут возникнуть при проведении исследований	62
5.2.1.1	Отклонение показателей микроклимата.....	62
5.2.1.2	Превышение уровня шума	64
5.2.1.3	Недостаточная освещённость рабочей среды	65
5.2.2	Опасные факторы производственной среды.....	67
5.2.2.1	Опасность поражения электрическим током	67
5.3.	Экологическая безопасность.....	68
5.3.1	Анализ воздействия средств исследования на окружающую среду	69
5.4	Безопасность в чрезвычайных ситуациях	70
	Заключение	72
	Список использованных источников	73
	Приложение А. Раздел на иностранном языке.....	75

Введение

Теория графов является мощным инструментом моделирования и решения большого количества задач. Представление информации с помощью графов используется в самых различных областях точных и естественных наук.

Графы позволяют моделировать произвольные системы, представимые в виде набора объектов и связей между ними.

В настоящее время графовые модели применяются в:

- программировании, для проектирования диаграмм потоков данных и иерархии классов, для построения организационных IDEF-диаграмм при создании ПО и ИС;
- дизайне БД для отображения диаграмм сущностей-связей;
- моделировании компьютерных сетей;
- биологии для составления моделей размножения бактерий;
- химии для отображения структур сложных формул и цепочек преобразований;
- физике и электротехнике для изготовления электронных и печатных схем.

Вопрос визуализации графов очень актуален, так как от того, насколько удачно изображен граф, зависит удобство и простота восприятия информации, которая содержится в нём.

Так как постоянно увеличиваются объемы информации и сложности структур, возникающих в различных теоретических и практических задачах, системы визуализации информации становятся всё более востребованными

Целью работы является разработка алгоритмического и программного обеспечения для укладки графов на плоскости, с возможностью использовать данное ПО для визуализации плоского изображения диаграмм баз данных.

Для достижения цели следует решить следующие задачи:

1. проведение анализа предметной области, связанной с визуализацией графов;
2. изучение существующих алгоритмов определения графа на планарность и отображение его на плоскости;
3. изучение существующих способов отображения не планарных графов на плоскости;
4. разработка алгоритма укладки не планарного графа на плоскость минимизируя количество пересечений между рёбрами;
5. реализация программного продукта, для визуализации отображения диаграмм баз данных с минимальным количеством пересечений между связями.

1 Проблема визуализации графа

В математической теории графов граф – это совокупность непустого множества вершин и набор пар вершин (связей между вершинами) [1].

В графовой модели объекты являются вершинами, а связи между объектами – рёбрами. Виды графов сильно отличаются по своим свойствам в зависимости от области применения (например: ограничения количества вершин и рёбер, направленность рёбер, различные метаданные для вершин и рёбер). Поэтому существует множество различных способов отображения графов. Ниже перечислены основные способы [2]:

- произвольное;
- прямолинейное — в данном отображении рёбра представляются отрезками между вершинами.
- сеточное — все вершины, точки пересечения и сгибы рёбер имеют целочисленные координаты;
- полигональное или полилинейное — в данном отображении рёбра представлены в виде ломанных;
- ортогональное — рёбра являются ломанными, отрезки которых — вертикальные или горизонтальные линии;
- планарное или плоское — изображение графа предполагает отсутствие точек пересечения у линий, изображающих ребра;
- восходящее или нисходящее — восходящее/нисходящее каждое ребро является кривой, которая монотонно не убывает/не возрастает в вертикальном направлении. В частном случае изображение может быть строго восходящим (или строго нисходящим).

В данной работе рассматриваются различные способы отображения графов на плоскости.

1.1 Планарный граф

Граф G можно уложить в пространстве L , только в том случае, если он изоморфен графу, вершины которого – некоторые точки пространства, а ребра — жордановы кривые, соединяющие данные вершины, причем:

1. каждая кривая проходит только через вершины, которые она соединяет;
2. для каждой из пар кривых выполняется условие, что кривые, входящие в данную пару, могут пересекаться только в вершинах, инцидентных одновременно обоим этим ребрам [3].

Граф, который составлен из точек пространства и жордановых кривых из L – укладка исходного графа. Граф планарный, если он обладает укладкой на плоскости. Граф уложенный на плоскости – плоский.

Для плоских графов есть простое соотношение, называемое формулой Эйлера [4]:

$$V - E + F = 2,$$

где V — вершины, E — ребра, F — грани.

При помощи этого свойства можно доказать не планарность некоторых графов.

Покажем, что такое грань в графе. На рисунке 1 представлен граф G

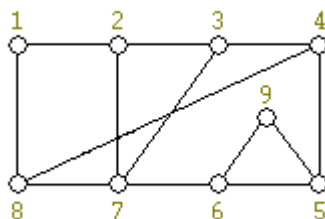


Рисунок 1 – граф G .

Выбираем любой простой цикл в графе (в данном случае $\{1, 2, 3, 4, 5, 6, 7, 8\}$) и получаем две грани: Γ_1 — внешнюю и Γ_2 — внутреннюю (Рисунок 2)

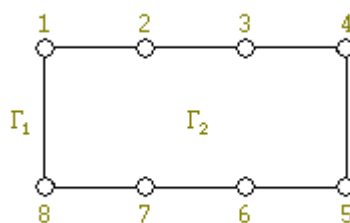


Рисунок 2 – грани Γ_1 и Γ_2

Согласно теореме о планарных графах (Понтрягин-Куратовский) [4], граф планарен тогда и только тогда, когда он не содержит в качестве частей графы K_5 и $K_{3,3}$ (быть может с добавочными вершинами степени 2) (K_5 – полный граф, состоящий из пяти вершин, $K_{3,3}$ – двудольный граф, состоящий из шести вершин и девяти рёбер).

1.2 Алгоритмы проверки графа на планарность

Существует большое количество алгоритмов определения графа на планарность: Наиболее часто используемые на практике представлены ниже.

- Метод добавления вершин
- Метод добавления рёбер
- Метод последовательного построения
- Генетический алгоритм

В базовом варианте метод добавления вершин и метод добавления ребер имеет время работы за $O(n^2)$ (n – количество вершин в графе), однако существуют улучшения основанные на PQ-дереве [5], что приводит асимптотику к $O(n)$

PQ-дерево – это структура данных для представления группы перестановок. Данное дерево является корневым планарным деревом. Висячие вершины – переставляемые элементы. Остальные вершины имеют пометку либо P , либо Q . Вершины с пометкой Q имеют по крайней мере 3 потомка, а вершины с пометкой P имеют по крайней мере 2 потомка. В PQ-дереве разрешается как угодно переставлять потомков вершины с пометкой P и обращать порядок потомков вершины с пометкой Q .

Метод добавления вершин создаёт структуру, хранящую все возможные вложения порождённого подграфа данного графа и добавления по одной новой вершине к этой структуре данных.

Метод добавления рёбер использует добавление рёбер к планарной структуре. Далее вычисляется наличие графа K_5 , либо графа $K_{3,3}$.

Гамма-алгоритм – алгоритм который предназначен для того, чтобы определить является ли текущий граф планарным, и если является – произвести его плоскую укладку. Данный алгоритм работает за $O(n)$.

Метод последовательного построения также работает за $O(n)$, и предназначен для быстрого определения графа на планарность в онлайн (для каждого нового ребра осуществляется одна проверка на его принадлежность грани текущего вложения).

1.3 Алгоритм укладки планарного графа на плоскость

Наиболее распространённым и эффективным алгоритмом для укладки планарного графа на плоскость является гамма-алгоритм [6].

Входными данными для данного алгоритма является граф, со следующими свойствами:

1. граф является связным;
2. в графе имеется хотя бы один цикл;
3. граф не имеет мостиков

В случае нарушения свойства (1) граф укладывается отдельно по компонентам связности.

В случае, если нарушено свойство (2) граф является деревом и нарисовать его плоскую укладку тривиально.

Если нарушено свойство (3), то мостики нужно разрезать, таким образом получив несколько компонент связности. Далее необходимо провести плоскую укладку каждой из компонент связности, а затем добавить ранее удалённые рёбра, соединив данные компоненты между собой. Для того, чтобы была возможность вернуть мостики в граф, не нарушив планарность, необходимо чтобы концевые вершины мостика не оказались внутри плоского графа. Для

этого необходимо нарисовать одну компоненту связности и последовательно присоединять к ней другие. Каждую новую компоненту связности необходимо рисовать в той грани, в которой лежит концевая вершина соответствующего мостика. Граф связности мостиками, компонент связности, полученных из исходного графа, является деревом, поэтому таким образом будет получена плоская укладка.

1.4 Задача укладки произвольного (не планарного графа) на плоскость

Многие графы не являются планарными, и поэтому их невозможно уложить на плоскость без пересечений рёбер. Существуют два основных способа решения данной задачи:

- сделать граф планарным, удалив из него как можно меньше рёбер (эта проблема является NP-полной) [9];
- сделать граф планарным, удалив такие рёбра, вставка которых впоследствии может создать наименьшее число пересечений. Данная задача также является NP-полной. Однако существуют некоторые эвристики, которые позволяют получать вполне удовлетворительные результаты [10].

В следующем разделе описан разработанный в ходе выполнения данной ВКР алгоритм, для решения задачи укладки не планарного графа на плоскость с минимизацией количества пересечений.

2 Разработка оптимального алгоритма укладки не планарного графа на плоскость, с минимальным количеством пересечений рёбер.

2.1 Описание разработанного алгоритма

Для того чтобы определить минимальное количество пересечений связей в таблице, был разработан алгоритм, описанный ниже.

Для упрощения описания разработанного алгоритма введем три понятия:

- Планарный компонент — подграф, являющийся планарным графом;
- Свободная вершина — вершина не входящая ни в один планарный компонент.
- Свободное ребро – ребро, которое не входит ни в один планарный компонент (например вершины находятся в разных компонентах).

Ниже приводится описание разработанного алгоритма.

Начало алгоритма.

Шаг 1. Создаём новый планарный компонент. Выбираем свободную вершину максимальной степени. Записываем её в данный компонент. Переходим к шагу 2.

Шаг 2. Среди свободных вершин оставшихся в графе выбираем такую, которая удовлетворяет следующим свойствам:

- а. После добавления данной вершины в данный планарный компонент он останется планарным.
- б. Среди всех возможных свободных вершин, удовлетворяющих условию “а” данная вершина имеет максимальное количество смежных ей вершин в данном компоненте.
- с. Из всех вершин, удовлетворяющих условиям “а” и “б” данная вершина имеет максимальную степень.

Если свободная вершина, удовлетворяющая данным свойствам, была найдена, переходим к шагу 3, в противном случае – к шагу 4.

Шаг 3. Добавляем найденную свободную вершину в текущий компонент. Переходим к шагу 2.

Шаг 4. В случае, если в графе ещё содержатся свободные вершины, переходим к шагу 1. В противном случае переходим к шагу 5.

Шаг 5. На оставшемся наборе свободных рёбер запускаем полный перебор. С его помощью находим набор рёбер максимальной величины, при добавлении которых планарность компонент не нарушится. Завершаем алгоритм.

Конец алгоритма.

После выполнения данного алгоритма остаётся набор рёбер, минимально возможного размера, после удаления из графа которых он станет планарным.

В разработанном алгоритме, на шаге 5 используется полный перебор, так как из-за специфики графов (алгоритм используется для работы с графами, составленными на основе диаграмм баз данных) количество свободных рёбер, оставшихся на последнем шаге не превышает 20.

2.2 Алгоритм для нахождения и укладки планарных компонент

В разделе 1.2 были рассмотрены существующие алгоритмы проверки графа на планарность. Для разработанного алгоритма необходимо проверять граф на планарность в онлайн.

Необходимо проверять граф на планарность при добавлении новых вершин к нему по одной, зная, что до добавления данной вершины граф являлся планарным.

Для реализации данного функционала был выбран алгоритм последовательного построения.

2.2.1 Описание алгоритма последовательного построения

Данный метод использует построение по индукции 3-связных графов для последовательного построения планарного вложения любой 3-связной компоненты графа G (а потому и планарного вложения самого графа G) [11].

Построение начинается с графа K_4 и определяется следующим образом:

- любой промежуточный граф является снова 3-связным.
- так как такие графы имеют единственное вложение (с точностью до выбора внешней грани), следующий граф после добавления нового ребра либо становится не планарным, либо является, уточнением предыдущего графа.

Данный способ сводит проверку планарности к проверке, будет ли следующее добавленное ребро иметь оба конца на внешней грани текущего вложения.

2.3 Асимптотическая сложность разработанного алгоритма

Рассмотрим асимптотическую сложность разработанного алгоритма.

На вход данному алгоритму подаётся граф, состоящий из n вершин и m рёбер.

- Для быстрого поиска вершины максимальной степени необходимо упорядочить список вершин по данному критерию. Для этого используется алгоритм быстрой сортировки, асимптотическая сложность которого $O(n \cdot \log(n))$

• В худшем случае шаг номер 2 будет вызываться $O(n)$ раз для каждой планарной компоненты. В худшем случае таких компонент в графе может быть $O(n/5)$ ($n/5$, так как не существует не планарных графов, количество вершин которых меньше пяти). Для того, чтобы среди вершин выбирать те, которые имеют максимальное количество смежных вершин входящих в компонент, необходимо проверять все рёбра, инцидентные текущей. Все данные операции можно сделать суммарно за $O(m \cdot \log(m))$, используя бинарное дерево для хранения принадлежности вершины ребру. Поэтому в худшем случае асимптотика данной части алгоритма будет составлять $O(n \cdot (n + m \cdot \log(m)))$.

- Последняя часть алгоритма – полный перебор оставшихся рёбер работает за $O(2^s)$, где s – количество ребер, которое осталось за пределами всех планарных компонент.

Суммарная асимптотика алгоритма составляет:

$$O(n \cdot \log(n) + n \cdot (n + m \cdot \log(m)) + 2^s).$$

2.4 Пример работы разработанного алгоритма

Рассмотрим работу разработанного алгоритма на примере графа состоящего из шести вершин и девяти рёбер (рисунок 3)

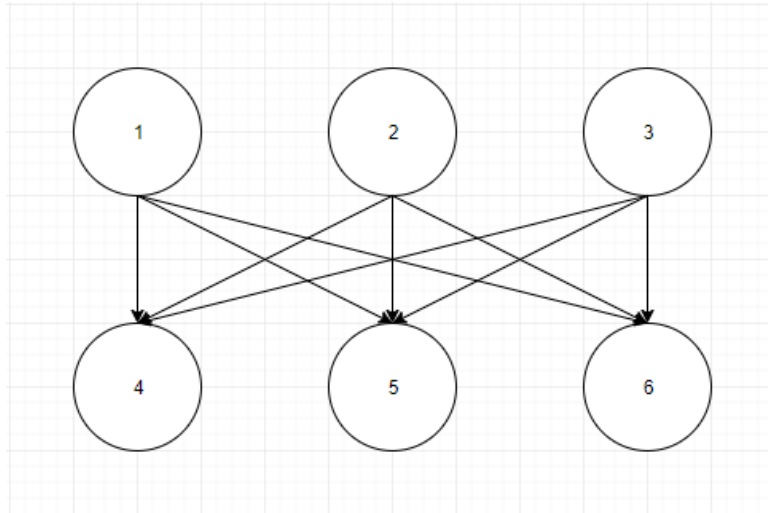


Рисунок 3 – Исходный граф

1. Изначально граф состоит из шести свободных вершин.
2. Создаётся первая компонента связности. В неё добавляется вершина 1 (на шаге 1 алгоритма). После этого дважды выполняется Шаг 2 в результате чего компонент будет состоять из вершин 1, 4 и 5 (рисунок 4).

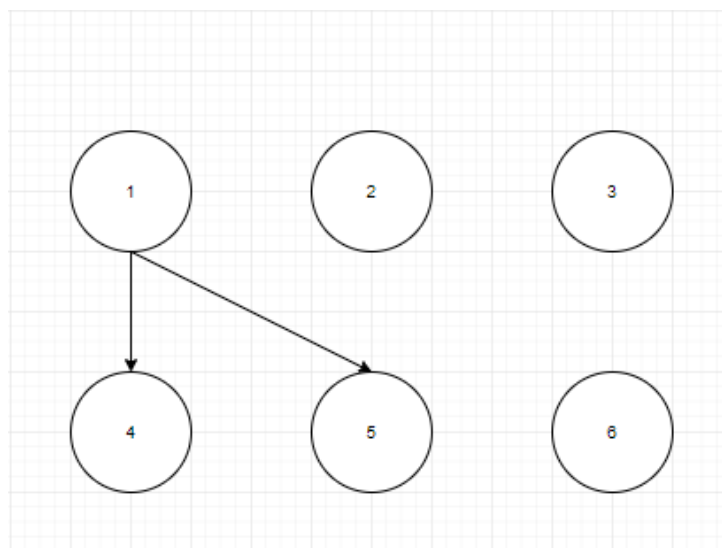


Рисунок 4 – Один компонент (1, 4, 5)

3. Далее к компоненте добавляется вершина 2 (рисунок 5), так как из трех оставшихся свободных вершин, данная вершина имеет максимальное количество смежных вершин находящихся внутри компоненты связности.

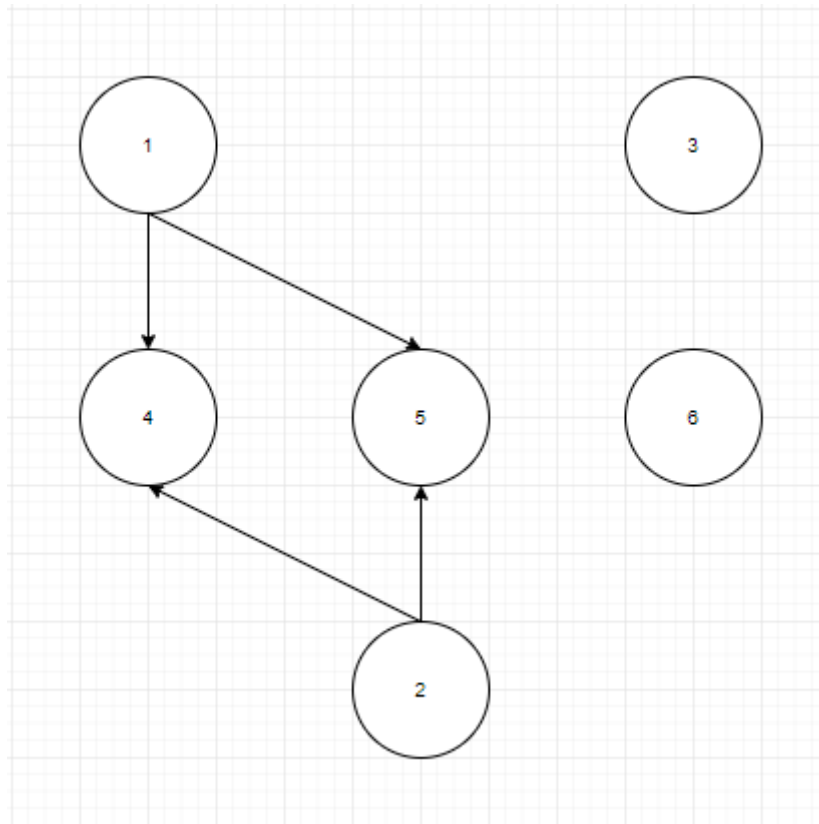


Рисунок 5 – Один компонент (1, 4, 5, 2)

4. Далее к компоненте добавляется вершина 3 (рисунок 6)

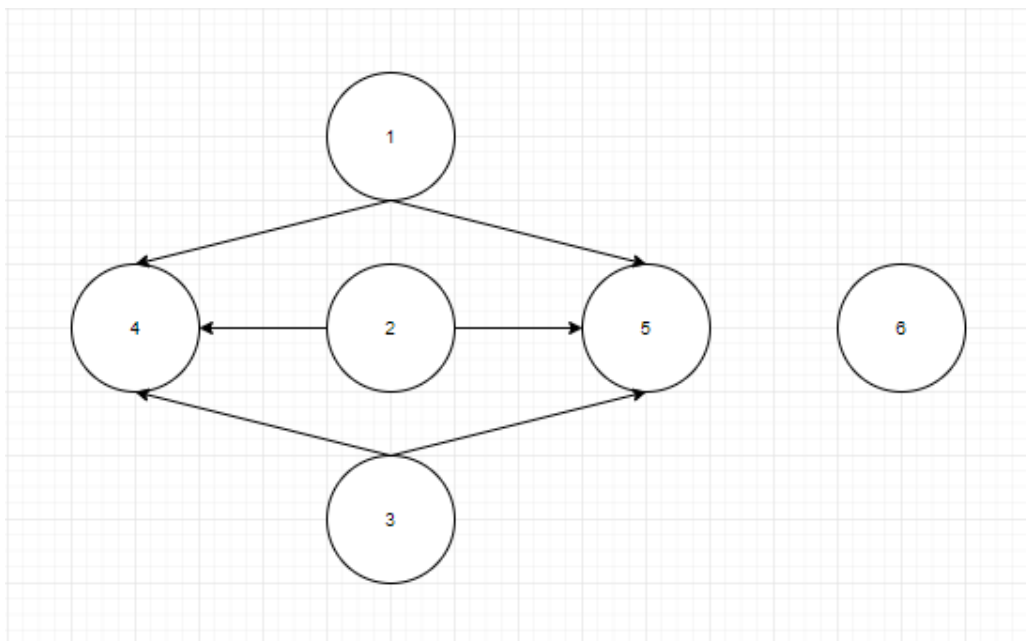


Рисунок 6 – Один компонент (1, 4, 5, 2, 3)

5. Вершина 6 не может быть добавлена в компонент, так как в этом случае нарушается планарность. Поэтому создаётся второй планарный компонент, состоящий из единственной вершины 6 (рисунок 7).

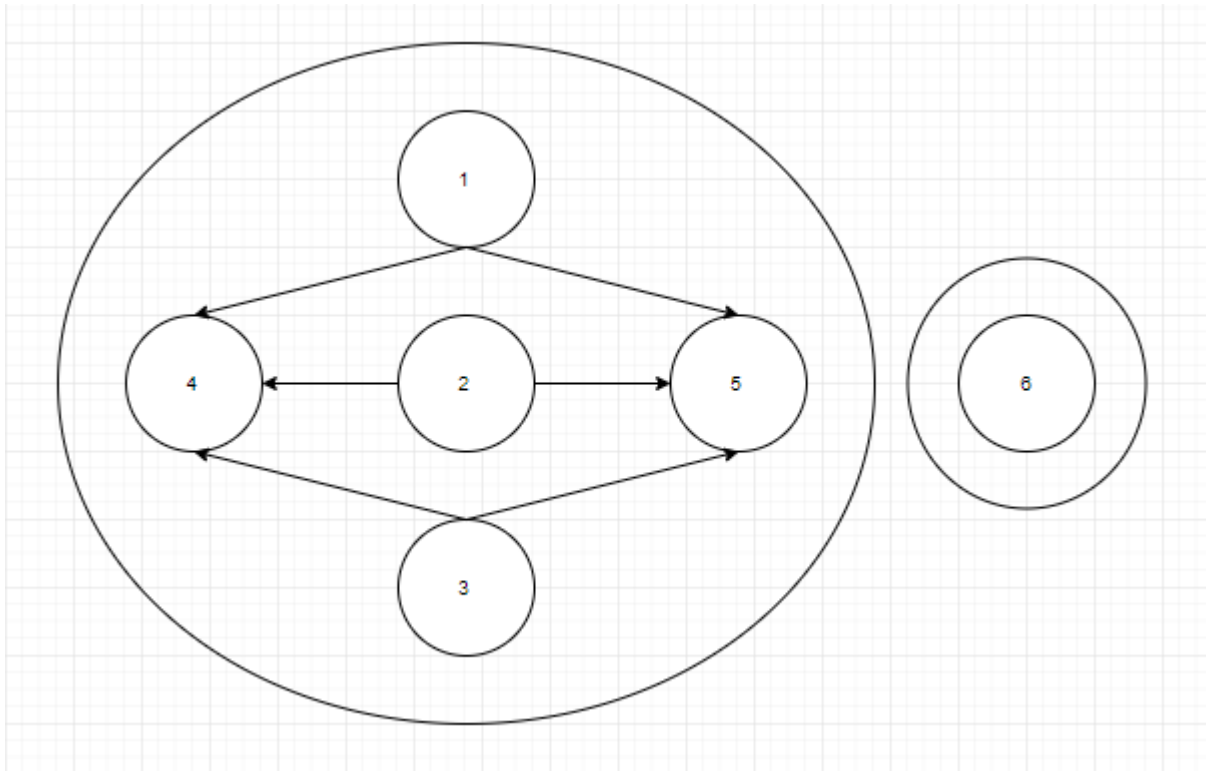


Рисунок 7 – Два компонента (1, 4, 5, 2, 3) и (6)

6. Далее при помощи полного перебора из оставшихся свободных рёбер (1-6, 2-6 и 3-6) выбираются максимально возможное количество рёбер, которое можно оставить в исходном графе, не нарушив планарность (рисунки 8 и 9).

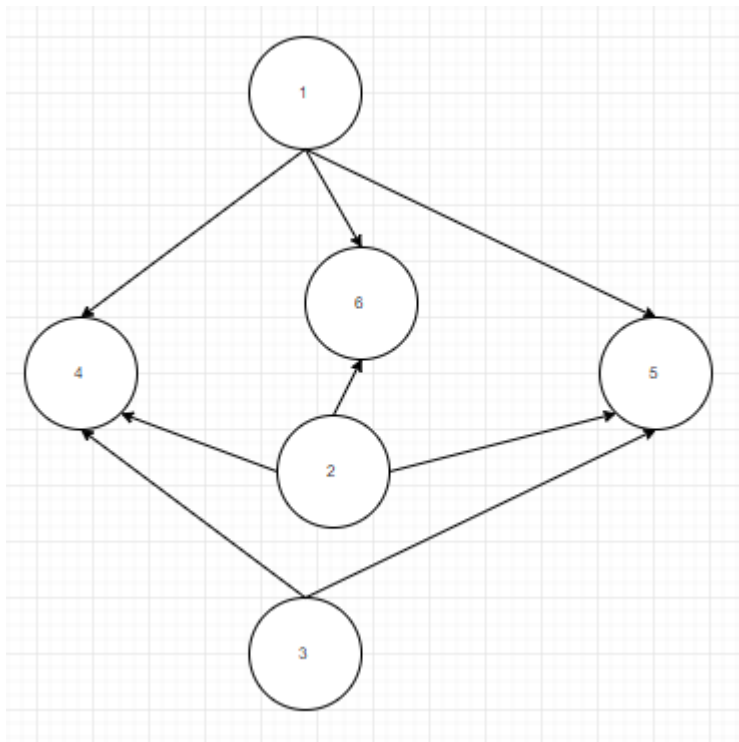


Рисунок 8 – Планарная укладка исходного графа без ребра 3-6.

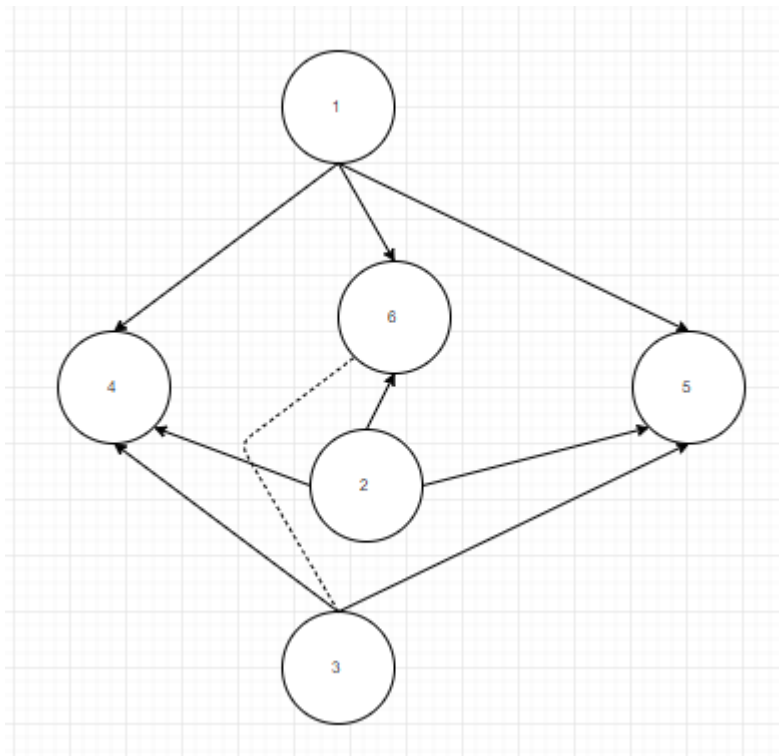


Рисунок 9 – Планарная укладка исходного графа с минимальным количеством пересечений рёбер

3 Реализация и тестирование программного продукта для укладки произвольного графа на плоскость

Данное приложение разрабатывалось в первую очередь для отображения графов, составленных на основе диаграмм БД. Исходя из этого, графы, для которых предназначено данное приложение имеют следующие характеристики:

- количество вершин не превышает 250;
- количество рёбер не превышает 800;
- рёбра имеют направленность (от таблицы с внешним ключом к таблице с первичным);
- между парами таблиц может быть несколько рёбер.

Данные характеристики были получены исходя из изучения множества БД, используемых при разработке ПО.

3.1 Проектирование интерфейса приложения

Перед реализацией был спроектирован интерфейс будущего приложения (рисунок 10) в инструменте для онлайн проектирования каркаса и макета приложения NinjaMock.

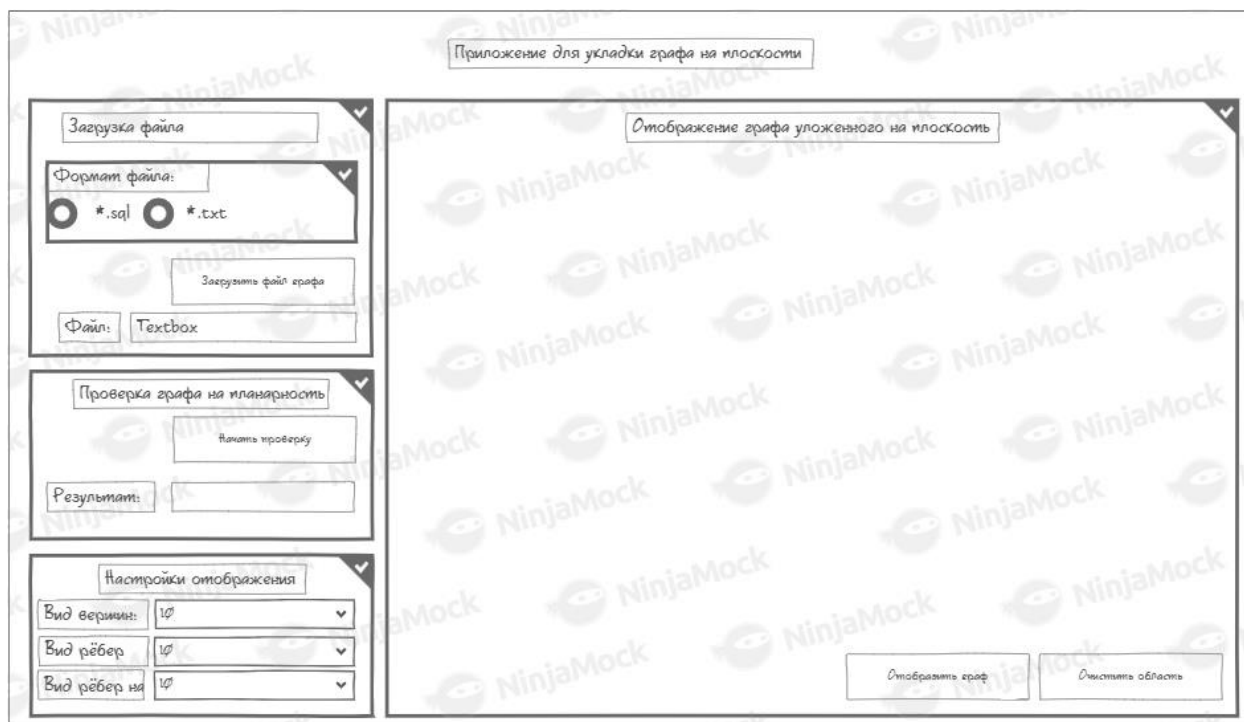


Рисунок 10 – Интерфейс приложения

Для упрощения работы с приложением было решено весь функционал разместить на единственном стартовом окне.

Левую часть окна приложения занимают компоненты для загрузки файла, содержащего граф, проверки графа на планарность и настройки отображения.

Правая часть – окно для отрисовки графа на плоскости.

3.2 Реализация приложения

Данное приложение было реализовано на языке С#.

Визуальная часть – wpf.

Основная страница приложения представлена на рисунке 11

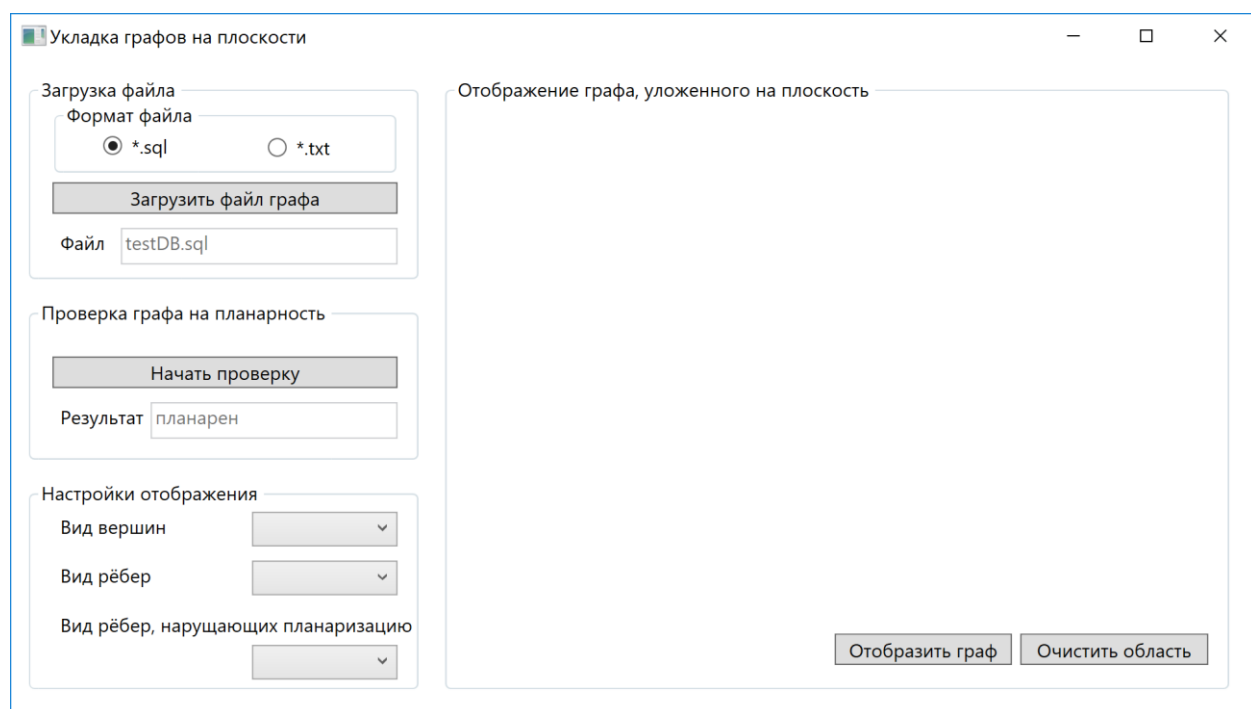


Рисунок 11 – Интерфейс приложения

3.2.2 Основные классы и модули.

Приложение состоит из двух основных частей:

- Application – основной модуль приложения, содержащий классы для обработки действий пользователя и реализующие интерфейс.
- Core – модуль (библиотека классов), хранящий в себе все основные алгоритмы.

Модуль Core состоит из трёх основных классов:

- **Parcers** – хранит в себе алгоритмы для преобразования sql скрипт базы данных в граф.
- **Services** – хранит алгоритмы для проверки графа на планарность, определения минимального количества рёбер, удалив которые граф станет планарным, а так же реализацию гамма-алгоритма для укладки графа на плоскость.
- **GraphAlgoritms** – хранит в себе основные алгоритмы для работы с графом, а так же структуры данных, при помощи которого вершины и рёбра графа хранятся в памяти.

3.2.3 Алгоритм отрисовки изображения графа

После нажатия на кнопку «Отобразить граф», заданный граф обрабатывается при помощи алгоритма, описанного в разделе 2. После выполнения данного алгоритма на выходе имеется несколько планарных компонент и набор свободных рёбер (в случае, если изначально граф планарный, то один планарный компонент и ноль свободных рёбер).

Для каждой из планарных компонент запускается гамма-алгоритм. Гамма-алгоритм для каждой из компонент выдаёт список из n целочисленных координат, где n – количество вершин в планарной компоненте. Так как необходимо получить изображение вершин в виде фигуры, то данные координаты масштабируется. Для этого рассчитывается необходимая высота фигуры (которая равняется ширине). И каждая координата сдвигается на данную величину.

Высота и ширина фигуры рассчитывается по необходимому объему контента, который необходимо показывать. Размеры рассчитываются для каждой из вершин и среди всех величин выбирается максимальная.

Для соединения вершин, представленных в виде фигур на изображении, рёбрами используется следующий способ:

Каждая из фигур имеет по 4 точки для присоединения рёбер (середина каждой грани). Перебираются все возможные способы соединить две вершины ребром. Из этих способов выбирается тот, в котором длина ребра минимальна.

3.3 Тестирование приложения

3.3.1 Тестирование парсера для преобразования sql скрипта БД в граф

Для тестирования был использован скрипт базы данных (рисунок 12) сформированный при помощи MS SQL.

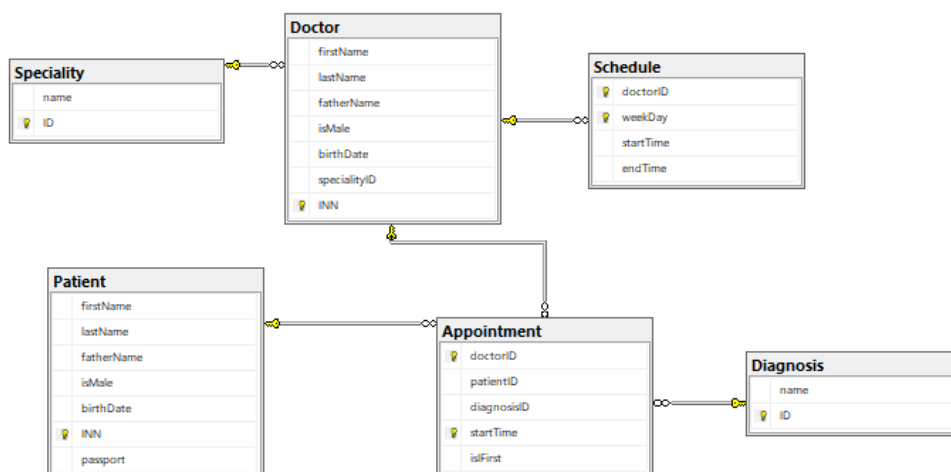


Рисунок 12 – Диаграмма БД для тестирования парсера

После применения парсера был получен граф, заданный следующим образом:

Перечисление вершин:

vertexName[“Appointment”].Number = 1

vertexName[“Diagnosis”].Number = 2

vertexName[“Doctor”].Number = 3

vertexName[“Patient”].Number = 4

vertexName[“Schedule”].Number = 5

vertexName[“Speciality”].Number = 6

Перечисление рёбер:

{ (3, 6), (5, 3), (1, 3), (1, 2), (1, 4) }

Набор вершин и рёбер графа совпадает.

3.3.2 Тестирование функционала определения графа на планарность

Было произведено два теста данной функции.

Первый тест. Передадим файл в котором задан заведомо планарный граф (рисунок 13).

```
firsrGraph.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
4 6
1 2
1 3
1 4
2 3
2 4
3 4
```

Рисунок 13 – Граф для первого теста на планарность

Данный граф состоит из четырёх вершин (с номерами 1, 2, 3, 4) и шести рёбер. Данный граф является планарным. Его можно отобразить на плоскости следующим образом (рисунок 14)

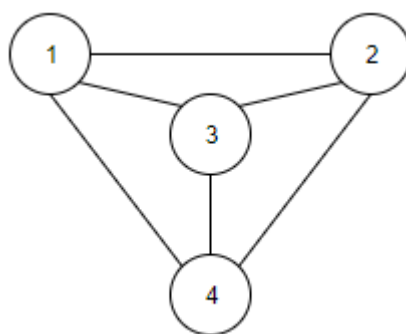


Рисунок 14 – Отображение графа на плоскости

При тестировании программа выдала результат, изображенный на рисунке 15.

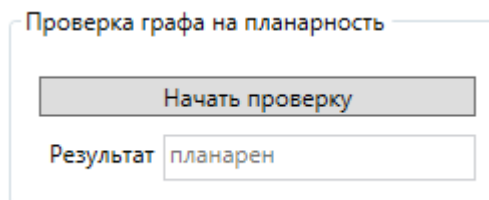
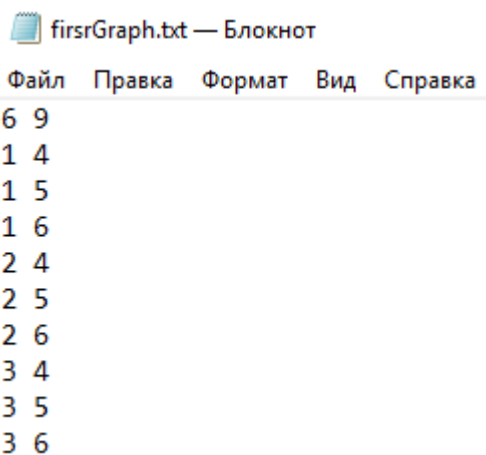


Рисунок 15 – Результат первой проверки на планарность

Второй тест. Передадим файл в котором задан заведомо не планарный граф (рисунок 16).



```
firsrGraph.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
6 9
1 4
1 5
1 6
2 4
2 5
2 6
3 4
3 5
3 6
```

Рисунок 16 – Граф для второго теста на планарность

Данный граф состоит из шести вершин и девяти рёбер. Данный граф невозможно отобразить на плоскости без пересечения рёбер, поэтому данный граф является не планарным.

При тестировании на данном графе программа выдала результат, изображенный на рисунке 17.

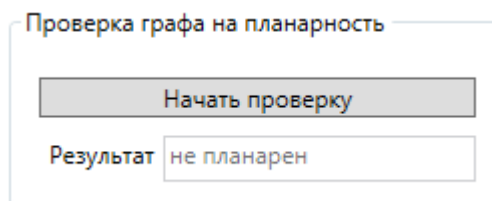


Рисунок 17 – Результат второй проверки на планарность

3.3.3 Тестирование отрисовки графа

Отрисовка планарного графа. Передадим файл в котором задан заведомо планарный граф (рисунок 13).

После нажатия кнопки «Отобразить граф» был получен результат изображенный на рисунке 18

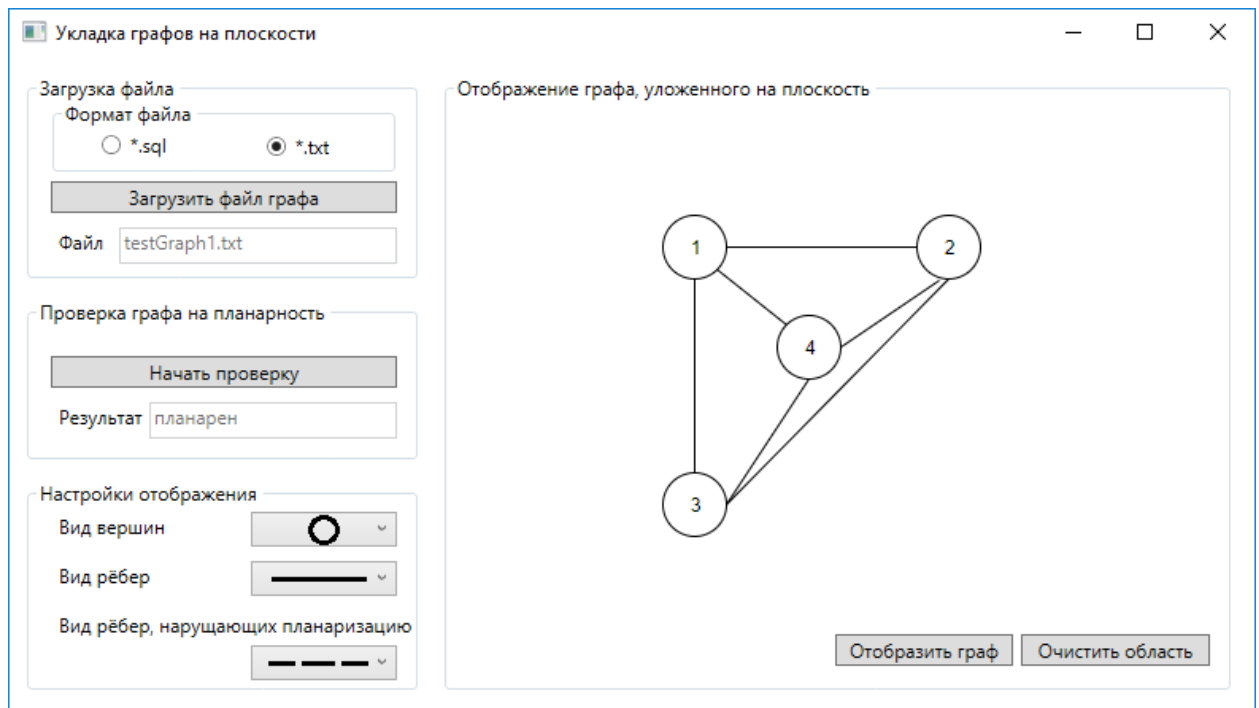


Рисунок 18 – Отрисовка планарного графа

Отрисовка не планарного графа. Передадим файл в котором задан заведомо планарный граф (рисунок 16).

После нажатия кнопки «Отобразить граф» был получен результат изображенный на рисунке 19

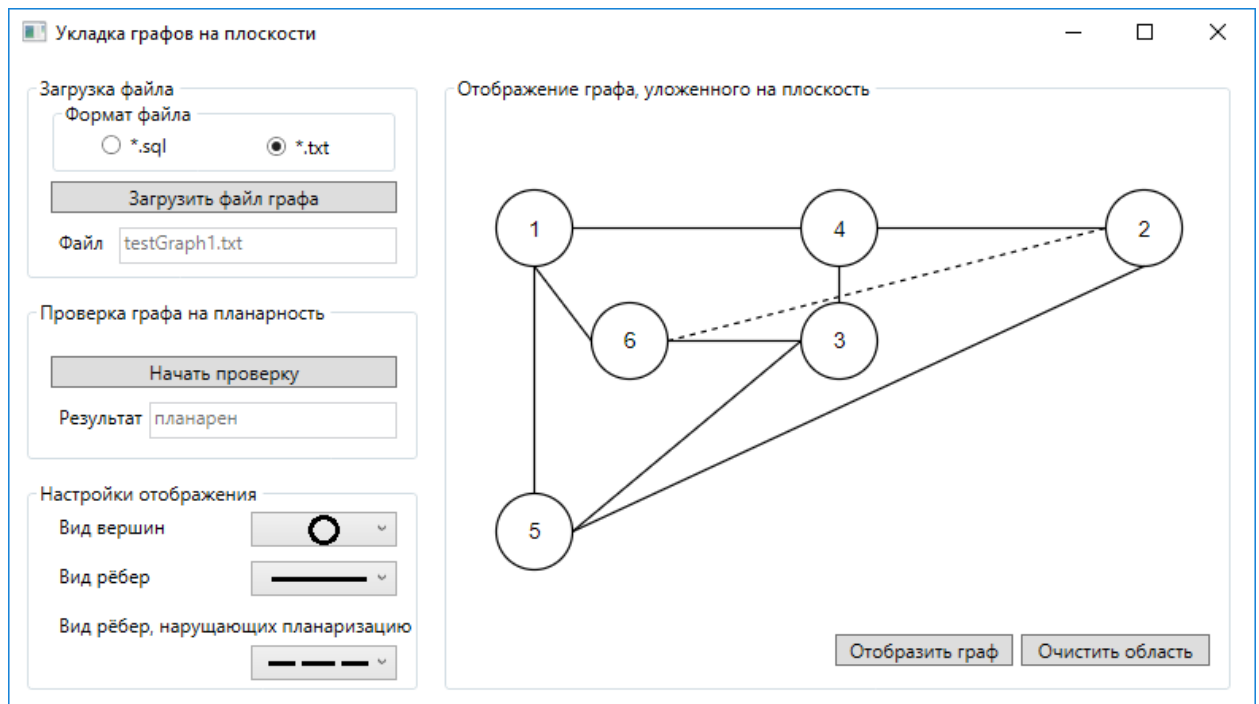


Рисунок 19 – Отрисовка не планарного графа

3.3.4 Тестирование отрисовки диаграммы БД

В качестве графа был передан скрипт БД, диаграмма которого представлена на рисунке 12. После нажатия кнопки «Отобразить граф» был получен результат изображенный на рисунке 20

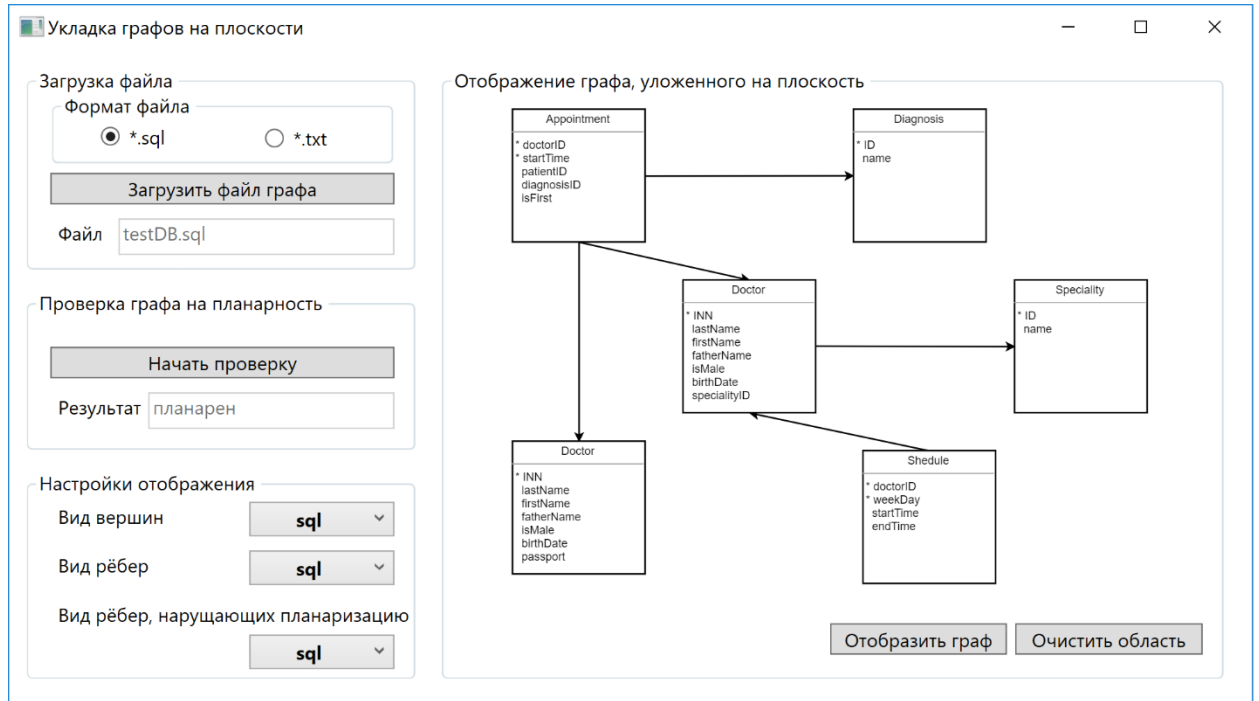


Рисунок 20 – Отрисовка диаграммы БД

4 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Введение. В настоящее время графовое представление информации все больше и больше используется в различных областях точных и естественных наук. В программировании графовые модели применяются при создании программного обеспечения (управляющие графы, иерархии классов, диаграммы потоков данных), дизайне баз данных (диаграммы сущностей-связей), разработке информационных систем и систем реального времени (модели компьютерных сетей, графы состояний, сети Петри), а также в системном программировании – при создании теории компиляции и преобразования программ. Вопрос визуализации подобных графовых структур является краеугольным камнем в процессе адекватного отображения информации.

В ходе данной работы была создана программа, которая позволяет представлять граф на плоскости с минимальным количеством пересечений рёбер.

Цели. Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является проектирование и создание конкурентоспособных разработок, технологий, отвечающих современным требованиям в области ресурсоэффективности и ресурсосбережения.

В рамках выбранной тематики проведения научного исследования «Разработка алгоритмического и программного обеспечения для укладки графов на плоскости» целесообразным является:

- Определить востребованность разработки;
- Определить преимущества и недостатки продукта;
- Оценить возможные варианты продвижения разработки на рынке;
- Осуществить детальное планирование разработки приложения;
- Выбрать наиболее эффективный способ реализации приложения.

Задачи. Для достижения описанных выше целей необходимо решить следующие задачи:

- Определение потенциальных потребителей путем рассмотрения целевого рынка и проведения его сегментирования;
- QUAD анализ для корректировки собственного проекта;
- Проведение SWOT-анализа для комплексной оценки угроз, возможностей, сильных и слабых сторон научно-исследовательского проекта, а также для анализа внешней и внутренней среды проекта;
- Расчёт затрат на реализацию проекта;
- Расчёт эффективности различных вариантов реализации приложения.

4.1 Предпроектный анализ

4.1.1 Потенциальные потребители результатов исследования

Произведём анализ рынка потенциальных потребителей. Потенциальными потребителями являются разработчики и администраторы баз данных, в состав которых входят множество таблиц и связей между ними, и необходимо отображать схему таблиц базы данных.

4.1.2 QUAD анализ

Так как у данной разработки нет никаких конкурентов на рынке, был произведен QUAD для оценки перспективности данного приложения.

Таблица 1 – QUAD

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение	Средневзвешенное значение (*100)
1	2	3	4	5	
Показатели оценки качества разработки					
1. Потребность в ресурсах памяти	0,15	90	100	0,9	13,8
2. Функциональность	0,1	95	100	0,95	9,5
3. Простота эксплуатации	0,07	75	100	0,75	5,25
4. Скорость работы	0,15	80	100	0,8	12
5. Надежность	0,05	100	100	1	5
6. Удобство эксплуатации	0,07	70	100	0,7	4,9
7. Точность	0,2	85	100	0,75	17
Показатели оценки коммерческого потенциала разработки					
8. Конкурентоспособность продукта	0,04	80	100	0,8	3,2

9. Уровень проникновения на рынок	0,08	0	100	0	0
10. Законченность работы	0,04	80	100	0,8	3,2
11. Послепродажное сопровождение	0,05	75	100	0,75	3,75
Итого	1	830	1100	8,2	77,6

Средневзвешенное значение показателя качества и перспективности данной научной разработки равна 77,6. Это означает, что перспективность данной разработки выше среднего (находится в пределах от 60-79).

4.1.3 SWOT-анализ

SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта. Сильные и слабые стороны являются факторами внутренней среды объекта анализа, возможности и угрозы являются факторами внешней среды.

Далее приведена матрица SWOT-анализа, реализованная для разрабатываемого приложения (Таблица 2).

Таблица 2 – Матрица SWOT

	Сильные стороны: С1. Наличие возможности определения графа на планарность С2. Богатый функционал. С3. Дружественный интерфейс.	Слабые стороны: Сл1. Сложность преобразования sql БД в формат графа, пригодного для работы алгоритма Сл2. Большая асимптотика работы приложения (низкая скорость работы).
Возможности: В1. Появление новых алгоритмов укладки непланарного графа на плоскость с минимальным количеством	Необходимо вносить модификации в работу программы, оптимизировать алгоритмы в приложения, после появления новых	Необходимо прописать системные требования к данному приложению. Это позволит пользователям использовать приложение не

пересечений рёбер. В2. Наличие отзывов от пользователей. В3. Выход новых версий и модификаций приложения. В4. Увеличение мощностей устройств.	разработок в этой области. Так как постоянно растёт быстродействие устройств необходимо реализовать более трудоёмкие алгоритмы, дающие лучший результат. Необходимо выпускать модификации приложения, чтобы разнообразить его функционал	испытывая трудностей с его быстродействием. Необходимо сделать сайт на котором пользователи смогут оставлять отзывы о данном приложении. Эти отзывы могут преподнести новые идеи модификаций, которые помогут улучшить данное приложение.
Угрозы: У1. Появление/увеличение конкуренции. У2. Проблемы с размещением продукта в магазине приложений. У3. Отсутствие спроса	Необходимо рекламировать богатый функционал и дружелюбный интерфейс чтобы справиться с конкуренцией.	Необходимо доработать недостающий функционал приложения для увеличения его конкурентоспособности. Необходимо добиться возможности размещения данного продукта в магазине приложений.

Далее была построена интерактивная матрица проекта (таблица ,
таблица 3 и 4).

Таблица 3 – Интерактивная матрица проекта (сильные стороны)

	С1	С2	С3
В1	+	-	-
В2	-	+	+
В3	-	+	+
В4	+	-	-

Таблица 4 – Интерактивная матрица проекта (слабые стороны)

	Сл1	Сл2
У1	+	+
У2	-	-
У3	-	-

Итак, на основе интерактивных матриц и матрицы SWOT можно сделать вывод о том, какие сильные и слабые стороны имеются у разработки, а также оценить возможности и угрозы, исходящие из внешней среды. В данном случае можно увидеть, что у приложения есть перспективные направления для развития, такие как добавление новых языков и добавление нового функционала.

4.1.4 Определение возможных альтернатив проведения научных исследований

Для создания стратегии разработки и продвижения проекта очень важно рассмотреть все возможные альтернативы создания продукта и выбрать из них наиболее оптимальные по тому или иному критерию. Получить все возможные варианты помогает морфологический подход (таблица 5)

Таблица 5 – Морфологическая матрица проекта

	1	2	3	4
А. Платформа приложения	Мобильное приложение	Веб-приложение	Десктопное приложение	
Б. Язык приложения	Русский	Английский		
В. Алгоритм для минимизации пересечений после выбора «планарных» компонент	Полное пересечение. Не обрабатывать последнюю стадию.	Рандомный выбор вершин, с попыткой соединить их ребрами без пересечений.	Полный перебор всех возможных вариантов.	
Г. Язык реализации	C++	C#	Python	Java

В качестве наиболее вероятных для реализации вариантов были выбраны следующие:

- АЗБ1В3Г1;
- АЗБ1В3Г3;
- АЗБ1В2Г1.

4.2 Инициация проекта

Для начала определим заинтересованные стороны проекта. В нашем случае заказчиком является «Rubius Group», заинтересованный в разработке приложения, позволяющего отображать диаграммы БД с минимально возможным пересечением связей между таблицами (таблица 6).

Таблица 6 – Заинтересованные стороны проекта

Заинтересованные стороны проекта	Ожидания заинтересованных сторон
«Rubius Group»	Создания приложения для персонального компьютера, позволяющего отображать диаграммы БД с минимально возможным пересечением связей между таблицами

Далее в таблице 7 указаны цели и результаты проекта.

Таблица 7 – Цели и результаты проекта

Цели проекта:	Создания десктоп-приложения для отображения диаграммы БД с минимально возможным пересечением связей между таблицами
Ожидаемые результаты проекта:	Программного обеспечения, позволяющее находить минимальное количество ребер, после удаления, которых граф станет планарным.
Критерии приемки результатов проекта:	<ol style="list-style-type: none"> 1. Многократное тестирование работы созданного ПО; 2. Проверен весь функционал созданного ПО;

Требования к результату проекта:	<ol style="list-style-type: none"> 1. Приложение должно обладать требуемому заказчиком функционалу; 2. Приложение должно быть протестировано на случай возникновения внезапных ошибок; 3. Приложение должно задавать для работа начальные данные и получать от него результат выполнения алгоритма; 4. Приложение должно позволять выбирать алгоритм для выполнения;
---	--

Организационная структура проекта представляет собой рабочую группу, приведенную в таблице 8

Таблица 8 – Рабочая группа проекта

№ п/п	ФИО, основное место работы, должность	Роль в проекте	Функции	Трудозатраты, час.
1	Комаров, В.А.	Разработчик (Р)	Разработка ПО, разработка алгоритмов управления	2352
2	Абрамов А.О.	Руководитель проекта от заказчика	Постановка промежуточных задач, тестирование	606
3	Буркатовская Ю. Б.	Руководитель от ТПУ	Постановка задачи НИ, проверка пояснительной записки	768
ИТОГО:				3726

Факторы, которые могут послужить ограничением степени свободы участников команды проекта приведены в таблице 9.

Таблица 9 – Ограничения проекта

Фактор	Ограничения/допущения
3.1 Бюджет проекта, руб.	1250000
3.1.1 Источник финансирования проекта	ООО «Rubius Group»
3.2 Сроки проекта:	
3.2.1 Дата утверждения плана управления проектом	Январь 2018 г.
3.2.2 Дата завершения проекта	Июль 2019 г.

4.3 Планирование управления научно-техническим проектом

4.3.1 План проекта

Линейный график плана проекта приведен в таблице 10. В данной таблице И – исполнитель – разработчик, РП – руководитель проекта, НР – научный руководитель.

Таблица 10 – Календарный план проекта в рабочих днях

Код и название этапа работы	Длительность, раб. дни	Дата начала работы	Дата окончания работы	Состав участников (ФИО ответственных исполнителей)
1. Постановка задачи	9	11.01.2018	21.01.2018	Разработчик (Р) НР, РП
2. Утверждение технического задания	10	22.01.2018	01.02.2018	РП
3. Разработка алгоритма укладки непланарного графа на плоскость с минимальным количеством пересечений рёбер	108	02.02.2018	14.06.2018	Р
4. Изучение предметной	27	01.08.2018	01.09.2018	Р, НР

области				
5. Разработка desktop- приложения	146	01.09.2018	01.03.2018	Р
6. Тестирование и исправление ошибок	92	01.03.2019	21.06.2019	Р, РП, НР
Итого:	392			





Построим диаграмму Ганта – гистограмма, позволяющая наглядно увидеть календарный план проекта (Таблица 11)

Таблица 11 – Календарный план-график проведения НИОКР по теме

Код этапа работы	Исполни- тели	Т _к , кал, дн.	Продолжительность выполнения работ																										
			2018																										
			январь			февраль			Март			апрель			май			Июнь			Июль			Август			Сентябрь		
			1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	Р, НР, РП	10																											
2	РП	11																											
3	Р	133																											
4	Р, НР	31																											
5	Р	181																											
6	Р, НР, РП	118																											

Продолжение календарного плана:

Код этапа работы	Исполни- тели	Т _к , кал, дн.	Продолжительность выполнения работ																										
			2018												2019														
			Октябрь			Ноябрь			Декабрь			Январь			Февраль			Март			Апрель			Май			июнь		
			1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	Р, НР, РП	10																											
2	РП	11																											
3	Р	133																											
4	Р, НР	31																											
5	Р	181																											
6	Р, НР, РП	118																											

НР,Р,РП –		РП –	
Р –			
Р, НР –			

4.3.2 Бюджет научно-технического исследования (НТИ)

В таблице ниже (таблица 12) указаны затраты на специальное оборудование и программного обеспечение, необходимое для разработки и проведения научного исследования.

Таблица 12 – Расчет затрат по статье «Спецоборудование для научных работ».

№ п/п	Наименование оборудования	Кол-во единиц оборудования	Цена единицы оборудования, тыс. руб.	Общая стоимость оборудования, тыс. руб.
1	Системный блок и комплектующие для работы компьютера	1	62	62
2	Контроллер «Мышь»	1	0,75	0,75
3	Контроллер «Клавиатура»	1	1,25	1,25
4	Монитор	2	17	34
Итого:				98

Бюджет научного исследования складывается из таких составляющих как материальные затраты, затраты за заработную плату руководителя и инженера.

Оклад доцента (руководителя) от университета – 33366 руб. (без учета районного коэффициента).

Оклад инженера-сотрудника (программиста) – 11300 руб. (без учета районного коэффициента) (принято на основе данных с окладов профессорско-преподавательского состава и дипломников-студентов).

Баланс рабочего времени представлен в таблице 13.

Таблица 13 – Баланс рабочего времени

Показатели рабочего времени	Руководитель/сотрудник
Календарное число	365
Количество нерабочих дней	66
Потери рабочего времени (отпуск/невыходы по болезни)	48
Действительный годовой фонд рабочего времени	251

Рассчитаем месячную заработную плату с учетом тридцатипроцентного районного коэффициента:

$$З_{\text{м}}^{\text{рук}} = 33366 * 1,3 = 43375 \text{ руб.}$$

$$З_{\text{м}}^{\text{инж}} = 11300 * 1,3 = 14690 \text{ руб.}$$

Исходя из месячной заработной платы, можно рассчитать среднедневную заработную плату. Так как отпуск составляет 48 дня, количество рабочих месяцев принято считать за 10,4.

$$З_{\text{дн}}^{\text{рук}} = \frac{43375 * 10,4}{251} = 1797,21 \text{ руб.,}$$

$$З_{\text{дн}}^{\text{инж}} = \frac{14690 * 10,4}{251} = 608,67 \text{ руб.}$$

Расчет основной заработной платы представлен в таблице ниже (таблица 14). Дополнительная заработная плата равна 10% от основной.

Таблица 14 – Расчёт основной и дополнительной заработной платы

Исполнители	З _б , руб.	k _{пр}	k _д	k _р	З _м , руб	З _{дн} , руб.	Т _р , раб. дн.	З _{осн} , руб.	З _{доп} , руб
Руководитель от ТПУ	33366	-	-	1,3	43375	1797,21	128	230042,88	23004,3
Руководитель проекта	33366	-	-	1,3	43375	1797,21	101	181518,21	18151,8
Инженер	11300	-	-	1,3	14690	608,67	392	238598,64	23859,9

В ходе данного научного исследования не было сделано ни одной производственной или научной командировки.

Отчисления на социальные нужды (на внебюджетные фонды) высчитывается по формуле:

$$C_{\text{внеб}} = k_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}})$$

где коэффициент отчисления на внебюджетные фонды составляет 0,271 (27% от общей и доп. Заработной платы).

Рассчитаем общие расходы на социальные нужды:

$$(230042,88 * 1,1 * 0,271) + (181518,21 * 1,1 * 0,271) + (238598,64 * 1,1 * 0,271) = 193812,65 \text{ руб.}$$

Накладные расходы высчитываются с коэффициентом, равным 30% (K_{накл} = 0,3):

$$C_{\text{накл}} = k_{\text{накл}} \cdot (Z_{\text{осн}} + Z_{\text{доп}})$$

Рассчитаем данные расходы для каждого сотрудника и сложим полученный результат, получив общие затраты на накладные расходы проекта:

$$(230042,88 * 1,1 * 0,3) + (181518,21 * 1,1 * 0,3) + (238598,64 * 1,1 * 0,3) = 214552,72 \text{ руб.}$$

Прямыми расходами в данном исследовании и разработке являются затраты на электроэнергию. Возьмем тариф равный 5,8 рубля за 1 кВт электроэнергии. Так как все три исполнителя использовали в качестве электро-питающего оборудования лишь персональные компьютеры, зная их мощность 0,284 (у руководителя от ТПУ) и 0,355 кВт (у прочих исполнителей) кВт, рассчитаем общие затраты электроэнергии в кВт:

$$(0,284 * 128 * 6) + (0,355 * (392 + 101) * 6) = 1268 \text{ кВт}$$

Умножив данное число на трафик, узнаем общие прямые затраты:

$$1268 * 5,8 = 7354,4 \text{ руб.}$$

В таблице 15 приведена группировка всех затрат по вышеперечисленным статьям.

Таблица 15 – Группировка затрат по статьям

Спец. оборудование для научных работ	Основная заработная плата	Дополнительная заработная плата	Отчисления на социальные нужды	Прочие прямые расходы	Накладные расходы	Итого плановая себестоимость
98000	650159,73	65016	193812,65	7354,4	214552,72	1228795,5

4.3.3 Организационная структура проекта

В практике используется несколько базовых вариантов организационных структур: функциональная, проектная, матричная.

Для выбора наиболее подходящей организационной структуры можно использовать таблицу 16.

Таблица 16 – Выбор организационной структуры научного проекта

Критерии выбора	Функциональная	Матричная	Проектная
Степень неопределенности условий реализации проекта	Низкая	Высокая	Высокая
Технология проекта	Стандартная	Сложная	Новая
Сложность проекта	Низкая	Средняя	Высокая
Взаимозависимость между отдельными частями проекта	Низкая	Средняя	Высокая
Критичность фактора времени (обязательства по срокам завершения работ)	Низкая	Средняя	Высокая
Взаимосвязь и взаимозависимость проекта от организаций более высокого уровня	Высокая	Средняя	Низкая

В данном случае выбор лежит к проектной структуре проекта из-за особенностей разработки. Составляющая проекта является модульные системы, работающие в постоянном взаимодействии с другими модулями. Также основной причиной выбора проектной структуры является то, что технология проекта является новой, и имеются ограниченные сроки реализации.

4.3.4 План управления коммуникациями проекта

План управления коммуникациями отражает требования к коммуникациям со стороны участников проекта. Пример плана управления коммуникациями приведен в таблице 17.

Таблица 17 – Пример плана управления коммуникациями

№ п/п	Какая информация передается	Кто передает информацию	Кому передается информация	Когда передает информацию
1.	Статус проекта	Руководитель проекта	Представителю заказчика	Ежеквартально (первая декада квартала)

2.	Обмен информацией о текущем состоянии проекта	Исполнитель проекта	Участникам проекта	Еженедельно (пятница)
3.	Документы и информация по проекту	Ответственное лицо по направлению	Руководителю проекта	Не позже сроков графиков и к. точек
4.	О выполнении контрольной точки	Исполнитель проекта	Руководителю проекта	Не позже дня контрольного события по плану управления

4.3.5 Реестр рисков проекта

Идентифицированные риски проекта включают в себя возможные неопределенные события, которые могут возникнуть в проекте и вызвать последствия, которые повлекут за собой нежелательные эффекты. Сведённые результаты представлены в таблице 18.

Таблица 18 – Реестр рисков

№	Риск	Потенциальное воздействие	Вероятность наступления (1-5)	Влияние риска (1-5)	Уровень риска*	Способы смягчения риска	Условия наступления
1	Потеря актуальности		2	5	средний	Внедрение нового функционала в процессе жизненного цикла ПО	Появление новых способов/алгоритмов представления не планарных графов на плоскость
2	Неточность ПО		4	5	высокий	Модификация алгоритма ПО	Появление новых алгоритмов укладки не планарных графов на плоскость, которые будут находить укладку с меньшим количеством пересечений
3	Застой в исследовании и разработке		2	4	низкий	Правильный выбор ПО для разработки, раннее планирование	Неправильно выбранные программные или аппаратные средства,

						финансирования проекта	приостановление финансирования проекта
4	Невозможность добавления требуемого функционала		4	4	средний	Грамотное планирование всех этапов разработки проекта	Необдуманность и неправильная последовательность этапов разработки

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

Определение эффективности проекта происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Расчёт финансовой эффективности проводится с помощью расчёта интегрального финансового показателя (таблица 19)

Таблица 19 – Расчёт интегрального финансового показателя

	Исп.1	Исп.2	Исп.3
Бюджет затрат	1228795	1683280	875306
Интегральный финансовый показатель	0.73	1	0.52

Интегральный показатель ресурсоэффективности считается как сумма балльных оценок каждого из вариантов по критериям с учётом их весовых коэффициентов (таблица 20).

Таблица 20 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Объект исследования	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3

Потребность в ресурсах памяти	0.20	4	5	4
Функциональность	0.15	5	4	3
Скорость работы	0.20	5	4	3
Надёжность	0.06	4	4	4
Удобство в эксплуатации	0.07	5	3	2
Простота эксплуатации	0.07	4	5	5
Точность	0.25	5	2	2
ИТОГО	1	4.67	3.7	3.08

Далее с помощью рассчитанных показателей были найдены интегральные показатели эффективности (таблица 21).

Таблица 21 – Сравнительная эффективность разработки

№ п/п	Показатели	Исп.1	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0.73	1	0.52
2	Интегральный показатель ресурсоэффективности разработки	4.67	3.7	3.28
3	Интегральный показатель эффективности	6.39	3.7	5.92
4	Сравнительная эффективность вариантов исполнения	1.73	1	1,6

Исходя из проведенного анализа можно отметить, что проект с разработкой данного варианта приложения менее выгоден чем вариант Исп. 3, однако намного эффективнее этого варианта. Выбранный для разработки вариант и выгоднее и эффективнее варианта Исп. 3. Таким образом, реализованное в данной работе приложение является экономически выгодным и достаточно качественным вариантом реализации данного проекта.

Оценки общей экономической эффективности

Для оценки общей экономической эффективности инноваций необходимо посчитать чистый доход (ЧД) и чистый дисконтированный доход (ЧДД). Для этого составим план денежных потоков (таблица 22).

Продолжительность шага расчёта равна одному году.

Поступления денежных средств занесены в таблицу со знаком «+», а расходование (оттоки) – со знаком «-».

Таблица 22 – План денежных потоков

№	Показатель, тыс руб	Номер шага (периода)расчета (t)					
		0	1	2	3	4	5
Операционная деятельность							
1	Выручка без НДС	0	2459	3073	3196	3577	4057
2	Полные текущие издержки, в том числе:	0	- 1966	-2387	-2468	-2727	-3058
3	прямые материальные затраты	0	-516	-644	-671	-749	-850
4	ФОТ основных рабочих, включая взносы во внебюджетные фонды	0	-609	-762	-791	-885	-1005
5	Силовая энергия	0	-196	-246	-255	-285	-324
6	Общепроизводстве	0	-361	-383	-383	-393	-405

	нные расходы						
7	Общехозяйственные расходы	0	-120	-125	-125	-130	-135
8	Коммерческие расходы	0	-76	-92	-95	-108	-122
9	Прочие расходы	0	-88	-135	-148	-177	-217
10	Денежный поток от производственной (операционной) деятельности (п.1-п.2)	0	493	686	728	850	999
Инвестиционная деятельность							
11	Поступление инвестиций	0	0	0	0	0	0
12	Капиталовложения, обслуживание инвестиций	-1228	-368	0	0	0	0
13	Сальдо от инвестиционной деятельности (п.11+п.12)	-1228	-368	0	0	0	0
14	Сальдо суммарного потока (п.10+п.13)	-1228	125	686	728	850	999
15	Сальдо накопленного потока	-1228	-1103	-417	311	1161	2160
16	Коэффициент дисконтирования при ставке дохода 10%	1,00	0,909	0,826	0,751	0,683	0,621

17	Дисконтированное сальдо суммарного потока (стр.14*´стр.16)	- 1228	113	567	546	580	620
----	--	-----------	-----	-----	-----	-----	-----

Чистый доход указан в последнем столбце 15 строки (Сальдо накопленного потока). ЧД=2160 тыс. руб.

Посчитаем Чистый дисконтированный доход при норме дисконта E=10%. ЧДД равен сумме строки 17. ЧДД=1198 тыс. руб.

Таким образом данный проект эффективен.

Заключение. В ходе написания раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» для работы «Разработка алгоритмического и программного обеспечения для укладки графов на плоскости» была оценена конкурентоспособность разрабатываемого приложения.

5 Социальная ответственность

Введение. В настоящее время графовое представление информации все больше и больше используется в различных областях точных и естественных наук. В программировании графовые модели применяются при создании программного обеспечения (управляющие графы, иерархии классов, диаграммы потоков данных), дизайне баз данных (диаграммы сущностей-связей), разработке информационных систем и систем реального времени (модели компьютерных сетей, графы состояний, сети Петри), а также в системном программировании - при создании теории компиляции и преобразования программ. Вопрос визуализации подобных графовых структур является краеугольным камнем в процессе адекватного отображения информации

В ходе данной работы была создана программа, которая позволяет представлять граф на плоскости с минимальным количеством пересечений рёбер.

Научно-исследовательский проект представляет собой разработку программного продукта и предполагает большой объем работы с ПК, поэтому важным критерием безопасности является организация рабочего места и режима трудовой деятельности. К вредным факторам труда инженера-программиста относятся: недостаточная освещенность рабочей зоны, отклонение параметров микроклимата в помещении и уровень шума, а к опасным факторам – опасность поражения электрическим током [12].

5.1 Правовые и организационные вопросы обеспечения безопасности

5.1.1 Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства

Проектируемая рабочая ситуация представляет собой непрерывную в течении всего рабочего дня работу за ПК. Правовыми нормами трудового законодательства для этого варианта рабочей зоны являются:

1. ТК РФ;
2. типовая инструкция ТОИ Р-45-084-01;
3. СанПиН 2.2.2. / 2.4. 1340-03.

Согласно типовой инструкции ТОИ Р-45-084-01 [13] особенностью трудового законодательства для рассматриваемой области является разделение видов трудовой деятельности за ПК на три группы:

1. А – работа по считыванию информации с экрана компьютера с предварительным запросом;
2. Б – работа по вводу информации;
3. В – творческая работа в режиме диалога с компьютером.

Разработку ПО можно относить к группе В. Разделение категорий работ по уровню нагрузки распределяется в этой группе представлено в таблице 23.

Таблица 23 – Распределение уровня нагрузки по категориями работ для группы В

Категория работы	Уровень нагрузки за рабочую смену, час.
I	до 2,0
II	до 4,0
III	до 6,0

То есть, в данном случае имеем категорию работ III группы В. В этом случае необходимо устраивать перерывы через 1,5 - 2,0 часа от начала рабочей смены и через 1,5 - 2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы для того, чтобы исключить повышенную утомляемость и снижение работоспособности.

5.1.2 Организационные мероприятия при компоновке рабочей зоны

В типовой инструкции ТОИ Р-45-084-01 [13] для рабочей зоны инженера-программиста устанавливаются следующие правила:

1. рабочие места с компьютерами должны размещаться таким образом, чтобы расстояние от экрана одного видеомонитора до тыла другого было не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м;
2. рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, преимущественно слева;
3. оконные проемы в помещениях, где используются персональные компьютеры, должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.;

4. рабочая мебель для пользователей компьютерной техникой должна отвечать следующим требованиям:

4.1. высота рабочей поверхности стола должна регулироваться в пределах 680 - 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм;

4.2. рабочий стол должен иметь пространство для ног высотой не менее 600 мм, глубиной на уровне колен не менее 450 мм и на уровне вытянутых ног не менее 650 мм;

4.3. рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также по расстоянию спинки от переднего края сиденья;

4.4. рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов; поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

5.2 Профессиональная социальная безопасность

В таблице 24 представлены вредные и опасные факторы при разработке ПО.

Таблица 24 – Возможные вредные и опасные факторы

Факторы (ГОСТ 12.0.003-2015) [1]	Вид работ		Нормативные документы
	Разработка ПО	Применение ПО	
Вредные			
1. Отклонение показателей	+	+	СанПиН 2.2.4.548–96 [14]

микроклимата			
2. Превышение уровня шума	+	+	СН 2.2.4/2.1.8.562–96 [15], СНиП 23-03-2003 [16]
3. Недостаточная освещенность рабочей зоны	+	+	СП 52.13330.2016 [17], СанПиН 2.2.2/2.4.1340-03 [18]
Опасные			
4. Опасность поражения электрическим током	+	+	СанПиН 2.2.2/2.4.1340-03 [18] ГОСТ Р 50571.17-2000 [19]

5.2.1 Анализ вредных факторов, которые могут возникнуть при проведении исследований

5.2.1.1 Отклонение показателей микроклимата

Использование персональных компьютеров может привести к повышению температуры и снижению относительной влажности в рабочем помещении, то есть к изменению микроклимата. Отклонение показателей микроклимата от комфортных может повлиять на здоровье персонала.

Оптимальные параметры микроклимата определяются в зависимости от категории работ по уровню энергозатрат. Инженер-программист относится к категории Ia [9]. В таблице 25 приведены оптимальные параметры для рабочего помещения персонала категории Ia, а в таблице 26 допустимые параметры для этой же категории.

Таблица 25 – Оптимальные величины показателей микроклимата на рабочих местах производственных помещений для категории Ia [14]

Период года	Категория работ по уровню энергозатрат, Вт	Температура воздуха, °С	Температура поверхности, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia (до 139)	22-24	21-25	60-40	0,1
Теплый	Ia (до 139)	23-25	22-26	60-40	0,1

Таблица 26 – Допустимые величины показателей микроклимата на рабочих местах производственных помещений для категории Ia [14]

Период года	Категория работ по уровню энергозатрат, Вт	Температура воздуха, °С	Температура поверхности, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia (до 139)	20-25	19-26	15-75	0,1
Теплый	Ia (до 139)	21-28	20-29	15-75	0,1-0,2

Для поддержания соответствующих микроклиматических параметров используются системы отопления и вентиляции, а также проводится кондиционирование воздуха в помещении.

5.2.1.2 Превышение уровня шума

Использование персональных компьютеров, наличие центральной системы вентиляции и кондиционирования воздуха приводят к повышению уровня шума на рабочем месте. Повышенный уровень шума затрудняет разборчивость речи, снижает работоспособность, повышает утомляемость.

В таблице 27 представлены предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для инженера-программиста.

Таблица 27 – Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для инженера-программиста [15]

Вид трудовой деятельности, рабочее место	Уровни звукового давления, дБ в октавных полосах со среднегеометрическими частотами, Гц						
	1,5	3	25	50	100	200	500
Творческая деятельность, руководящая работа с повышенными требованиями, научная деятельность, конструирование и проектирование, программирование, преподавание и обучение, врачебная деятельность. Рабочие места в помещениях дирекции, проектно-конструкторских бюро, расчетчиков, программистов вычислительных машин, в лабораториях для теоретических работ и обработки данных, приема больных в здравпунктах	6	1	1	4	9	5	2

Для уменьшения воздействий шума можно использовать следующие методы [16]:

1. экранирование рабочих мест, то есть установка перегородок между рабочими местами;
2. установка оборудования, производящего минимальный шум.

Для снижения уровня шума, производимого персональными компьютерами, рекомендуется регулярно проводить их техническое обслуживание: чистка от пыли, замена смазывающих веществ; а также применять звукопоглощающие материалы.

5.2.1.3 Недостаточная освещённость рабочей среды

Отсутствие достаточного количества источников освещения в рабочей зоне является причиной такого вредного фактора, как недостаточная освещённость. Недостаточная освещённость приводит к понижению работоспособности, а также может вызвать проблемы со здоровьем, а именно может повлиять на качество зрения персонала.

Согласно СП 52.13330.2016 [17] зрительную работу инженера-программиста можно характеризовать как работу разряда Б – высокой точности (наименьший эквивалентный размер объекта различения составляет 0,3-0,5 мм), подразряда 1 (относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%). В таблице 28 представлены требования к освещению рабочего помещения для указанного разряда.

Таблица 28 – Требования к освещению рабочего помещения для разряда Б1 [17]

Искусственное освещение				Естественное освещение	
Освещенность на рабочей поверхности и от системы общего освещения, лк	Цилиндрическая освещенность, лк	Объединенный показатель дискомфорта, не более	Коэффициент пульсации освещенности, K_n , %, не более	Коэффициент естественной освещенности, %, при	
				верхнем или комбинированном	боковом
300	100	21	15	3	1

Для снижения влияния фактора недостаточной освещенности необходимо, чтобы уровень естественного освещения рабочего места и яркость дисплея персонального компьютера были приблизительно одинаковыми, так как яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение, что приводит к их быстрой утомляемости.

Также следует соблюдать следующие нормативы, установленные в СанПиН 2.2.2/2.4.1340-03 [18]:

1. следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране персонального компьютера не должна превышать 40 кд/м² и яркость потолка не должна превышать 200 кд/м²;

2. дизайн персонального компьютера должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света; корпус персонального компьютера, клавиатура и другие блоки и устройства персонального компьютера должны иметь матовую поверхность с коэффициентом отражения 0,4-0,6 и не иметь блестящих деталей, способных создавать блики;

3. для освещения помещений с персонального компьютера следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пускорегулирующими аппаратами; допускается использование многоламповых светильников с электромагнитными пускорегулирующими аппаратами, состоящими из равного числа опережающих и отстающих ветвей.

5.2.2 Опасные факторы производственной среды

5.2.2.1 Опасность поражения электрическим током

Электрические установки, к которым относится персональный компьютер, представляют для человека большую потенциальную опасность, так как в процессе эксплуатации или проведения профилактических работ человек может коснуться комплектующих компьютера, находящихся под напряжением. Кроме того, корпуса персонального компьютера и прочего оборудования, оказавшегося под напряжением в результате повреждения или пробоя изоляции, не подают каких-либо сигналов, которые предупреждают человека об опасности. Причинами электропоражений являются: провода с поврежденной изоляцией, розетки сети без предохранительных кожухов. Поражение персонала электрическим током является опасным для жизни и здоровья.

Согласно СанПиН 2.2.2/2.4.1340-03 [18] рабочее место инженера-программиста должно находиться в безопасной зоне, которое не характеризуется наличием таких условий, как повышенная влажность (относительная влажность воздуха длительно превышает 75%), высокая

температура (более 35°C), токопроводящая пыль, токопроводящие полы, возможность одновременного соприкосновения к имеющим соединения с землей металлическим элементам и металлическим корпусам электрооборудования.

Для защиты от поражения электрическим током все токоведущие части должны быть защищены от случайных прикосновений кожухами, корпус устройства должен быть заземлен. Заземление выполняется изолированным медным проводом сечением 1,5 мм, который присоединяется к общей шине заземления с общим сечением 48 мм при помощи сварки. Общая шина присоединяется к заземлению, сопротивление которого не должно превышать 4 Ом.

Согласно ГОСТ Р 50571.17-2000 [19] питание устройства в рабочем помещении инженера-программиста должно осуществляться от силового щита через автоматический предохранитель, который срабатывает при коротком замыкании нагрузки. Для снижения величин возникающих разрядов должны применяться покрытия из антистатического материала. Рабочие место инженера-программиста должно быть оборудовано таким образом, чтобы исключить взаимное соприкосновение кабелей и шнуров питания соседних компьютеров.

К организационно-техническим мероприятиям относится первичный инструктаж по технике безопасности, который является обязательным условием для допуска к работе в данном помещении.

5.3. Экологическая безопасность

Разрабатываемое программное обеспечение не оказывает влияния на окружающую среду, так как разрабатывается и используется внутри персональных компьютеров.

Тем не менее, некоторые средства исследования могут стать источником загрязнения окружающей среды, а именно:

1. персональные компьютеры;
2. люминесцентные лампы.

5.3.1 Анализ воздействия средств исследования на окружающую среду

Для уменьшения вредного воздействия от утилизации ПК и люминесцентных ламп на окружающую среду необходимо соблюдать правила утилизации.

Согласно «Методике проведения работ по комплексной утилизации вторичных драгоценных металлов из отработанных средств вычислительной техники» [20] утилизация персональных компьютеров предусматривает 4 этапа, в рамках которых производится разборка и подготовка к утилизации внутренних частей ПК.

В результате выполнения первых 3-х этапов формируется партия сырья, включающая ячейки и элементы для объединений микросхем и электрорадиоэлементов, содержащие драгоценные металлы, а также партии черных и цветных металлов и сплавов (медь, сталь, никель, латунь, бронза, алюминий, дюралюминий, свинцово-оловянные припой), направляемых на переработку.

На последнем этапе осуществляется разборка ячеек и элементов для объединений микросхем и электрорадиоэлементов до уровня отдельных компонентов.

Согласно ГОСТ 12.3.031-83 [21] все ртутьсодержащие отходы и вышедшие из строя приборы, содержащие ртуть, подлежат сбору и возврату для последующей регенерации ртути в специализированных организациях. К работе по замене и сбору отработанных ртутьсодержащих ламп допускаются только электромонтеры. Главным условием при замене и сборе отработанных ртутьсодержащих ламп является сохранение герметичности. Факт сдачи ртутьсодержащих отходов подтверждается возвращением

паспорта на вывоз отходов с отметкой о приеме представителя специализированного предприятия.

5.4 Безопасность в чрезвычайных ситуациях

Одними из наиболее вероятных и разрушительных видов чрезвычайных ситуаций являются пожар или взрыв на рабочем месте.

Всякий работник при обнаружении пожара должен (ППБ 01-03 [22]):

- незамедлительно сообщить об этом в пожарную охрану;
- принять меры по эвакуации людей, каких-либо материальных ценностей согласно плану эвакуации;
- отключить электроэнергию, приступить к тушению пожара первичными средствами пожаротушения.

При возникновении пожара должна сработать система пожаротушения, передав на пункт пожарной станции сигнал о ЧС. В случае если система не сработала, то необходимо самостоятельно произвести вызов пожарной службы по телефону 101, сообщить точный адрес места возникновения ЧС и ожидать приезда специалистов.

Рабочее место располагается в 10 корпусе ТПУ 402а аудитория. На рисунке 1 представлен план эвакуации четвертого этажа 10 корпуса.

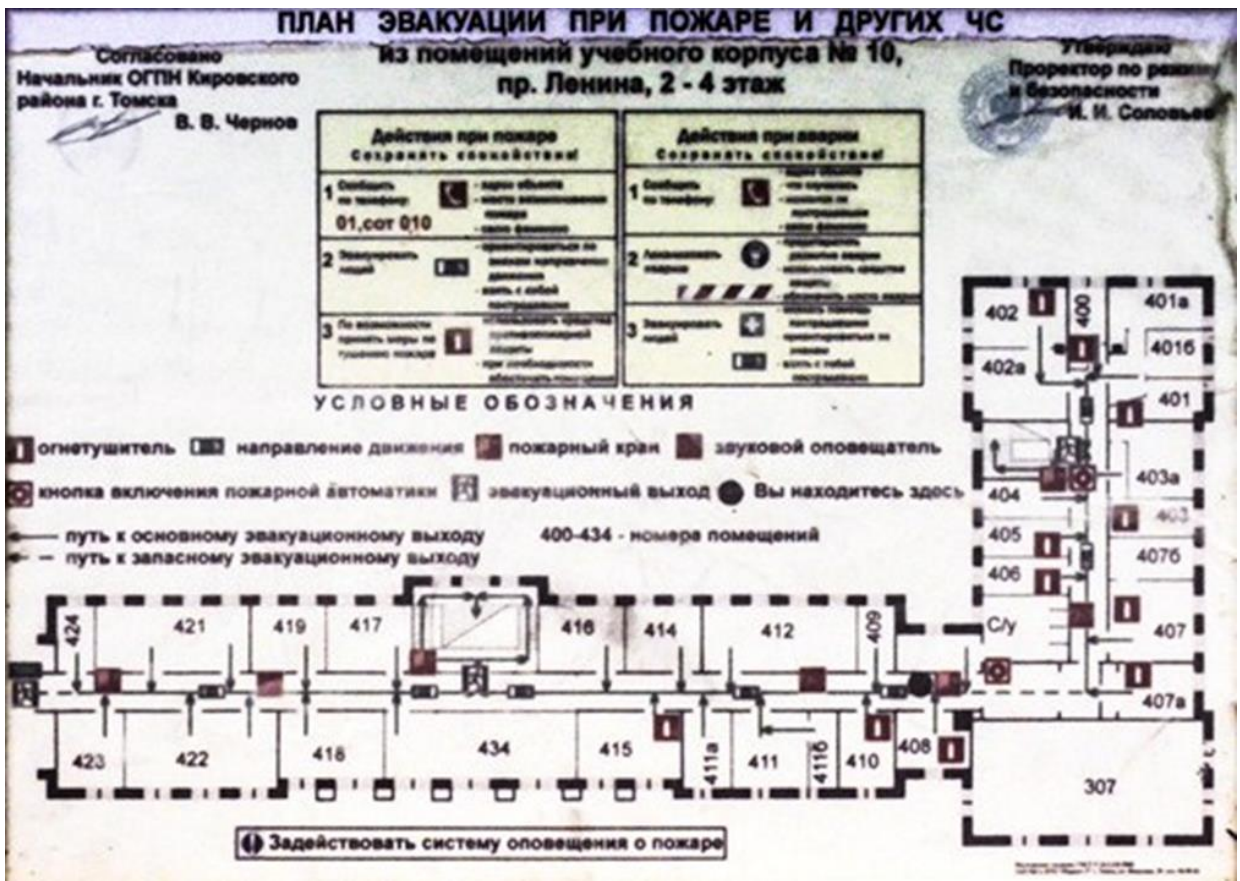


Рисунок 21 – План эвакуации при чрезвычайных ситуациях

Заключение

В ходе выполнения выпускной квалификационной работы были получены следующие результаты:

1. произведён анализа предметной области, связанной с визуализацией графов;
2. рассмотрены существующие алгоритмы определения графа на планарность и отображения его на плоскости;
3. изучены существующие способы отображения не планарных графов на плоскости;
4. разработан алгоритм укладки не планарного графа на плоскости минимизируя количество пересечений между рёбрами;
5. реализован программный продукт, для визуализации отображения диаграмм баз данных с минимальным количеством пересечений между связями.

Список использованных источников

1. Граф (математика) // ru.wikipedia.org: Википедия – свободная энциклопедия. URL: [http://ru.wikipedia.org/wiki/Граф_\(математика\)](http://ru.wikipedia.org/wiki/Граф_(математика)) (дата обращения 25.05.2019)
2. Визуализация графов // ru.wikipedia.org: Википедия – свободная энциклопедия. URL: http://ru.wikipedia.org/wiki/Визуализация_графов (дата обращения 25.05.2019)
3. Р. Уилсон. Введение в теорию графов. – М.Мир, 1977
4. А. Ю. Ольшанский. Плоские графы. // СОЖ, 1996, No 11, с. 117—122.
5. Booth, Kellogg S. and Lueker, George S. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms (англ.) // Journal of Computer and System Sciences. — 1976. — Vol. 13, no. 3. — P. 335–379.
6. Емеличев Р.И., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. — М.: Наука, 1990.
7. Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., Wagner, D. (2008) On modularity clustering. IEEE Transactions on Knowledge and Data Engineering 20 (2), 172-187.
8. Емеличев Р.И., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. — М.:Наука, 1990.
9. List of NP-complete problems // en.wikipedia.org: Википедия – свободная энциклопедия. URL: https://en.wikipedia.org/wiki/List_of_NP-complete_problems (дата обращения 25.05.2019)
10. D3.js. Визуализация графов // habrahabr.ru: Хабрахабр. URL: <https://habr.com/ru/post/302968/> (дата обращения 10.06.2019)
11. Jens M. Schmidt. The Mondschein Sequence // Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP'14). — 2014. — С. 967–978.

12. ГОСТ 12.0.003-2015 ССБТ. Опасные и вредные производственные факторы. Классификация.

13. ТОИ Р-45-084-01. Типовая инструкция по охране труда при работе на персональном компьютере" (утв. Приказом Минсвязи РФ от 02.07.2001 N 162).

14. СанПиН 2.2.4.548–96. Гигиенические требования к микроклимату производственных помещений.

15. СН 2.2.4/2.1.8.562–96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки.

16. СНиП 23-03-2003. Строительные нормы и правила российской федерации. Защита от шума.

17. СП 52.13330.2016. Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*.

18. СанПиН 2.2.2/2.4.1340-03. "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы".

19. ГОСТ Р 50571.17-2000 (МЭК 60364-4-482-82). Электроустановки зданий. Требования по обеспечению безопасности. Выбор мер защиты в зависимости от внешних условий. Защита от пожара.

20. Методика проведения работ по комплексной утилизации вторичных драгоценных металлов из отработанных средств вычислительной техник, утвержденная Государственным Комитетом РФ по телекоммуникациям от 19 октября 1999 г.

21. ГОСТ 12.3.031-83. Система стандартов безопасности труда (ССБТ). Работы со ртутью. Требования безопасности.

22. ППБ 01-03. Правила пожарной безопасности в Российской Федерации. – М.: ФГУ ВНИИПО МЧС России, 2003. – 111 с.

Приложение А. Раздел на иностранном языке

2 Development of an optimal algorithm for plotting a non-planar graph on a plane with a minimum number of intersections of edges.

Студент:

Группа	ФИО	Подпись	Дата
8ИМ72	Комаров Вячеслав Андреевич		

Консультант отделения информационных технологий ИШИТР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Мирошниченко Евгений Александрович	к.т.н.		

Консультант – лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИЯ	Диденко Анастасия Владимировна	к.фил.н.		

2 Development of an optimal algorithm for plotting a non-planar graph on a plane with a minimum number of intersections of edges.

2.1 Description of the developed algorithm

In order to determine the minimum number of intersections of links in the table, the algorithm described below is developed. To simplify the description of the developed algorithm, we introduce three concepts:

- Planar component – a subgraph that is a planar graph;
- Free top – the top is not a member of a single planar component.
- Free edge – an edge that is not part of any planar component (for example, vertices are in different components).

The beginning of the algorithm.

Step 1. Create a new planar component. Choose a free vertex of maximum degree. Write it in this component. Go to step 2.

Step 2. Among the free vertices remaining in the graph, choose one that satisfies the following properties:

- a. After adding this vertex to this planar component, the component remains planar.
- b. Among all possible free vertices satisfying the condition “a” this vertex has the maximum number of adjacent vertices in this component.
- c. Of all the vertices that satisfy the conditions “a” and “b”, this vertex has the maximum degree.

If a free vertex satisfying these properties has been found, proceed to step 3, otherwise – to step 4.

Step 3. Add the found free vertex to the current component. Go to step 2.

Step 4. In the case that the graph still contains a free vertex, go to step 1. Otherwise, go to step 5.

Step 5. On the remaining set of free edges run a full search. With its help, a set of edges of the maximum value is found, the addition of which the planarity of the component is not disturbed. Completing the algorithm.

The end of the algorithm.

After executing this algorithm, a set of edges of the minimum possible size remains. After removing the set from the graph, the graph becomes planar.

In the developed algorithm, step 5 uses a full search, because due to the specificity of graphs (the algorithm is used to work with graphs based on database diagrams), the number of free edges remaining in the last step does not exceed 20.

2.2 Algorithm for finding and stacking planar components

In section 1.2 existing verification algorithms for graph planarity are considered. For the developed algorithm, it is necessary to check the graph for planarity online.

It is necessary to check the planarity of the graph when adding new vertices to it one by one, knowing that before adding this vertex the graph is planar. To implement this functionality, a sequential construction algorithm is chosen.

2.2.1 Description of the sequential construction algorithm

This method uses the construction by induction of 3-connected graphs for the sequential construction of the planar embedding of any 3-connected component of the graph G (and therefore the planar embedding of the graph G). The construction starts with K_4 and is defined in such a way that any intermediate graph on the path to the full component is 3-connected again. Since such graphs have a single embedding (up to the selection of the outer face), the next larger graph, if it remains planar, must be a clarification of the previous graph. This allows to reduce the verification of planarity to just checking whether the next added edge to have both ends on the outer face of current nesting.

2.3 Asymptotic complexity of the developed algorithm

The asymptotic complexity of the algorithm is described below. The input to this algorithm is a graph consisting of n vertices and m edges.

- To quickly find the maximum degree of a vertex, arrange the list of vertices according to this criterion. To do this, use a fast sorting algorithm whose asymptotic complexity is $O(n \cdot \log(n))$

- In the worst case, step number 2 will be called $O(n)$ times for each planar component. In the worst case, such components in the graph can be $O(n/5)$ ($n/5$, since there are no non-planar graphs whose number of vertices is less than five). In order to select among the vertices those that have the maximum number of adjacent vertices included in the component, it is necessary to check all edges incident to the current one. All these operations can be done in total in $O(m \cdot \log(m))$, using a binary tree to store the belonging of the vertex to the edge. Therefore, in the worst case, the asymptotics of this part of the algorithm will be $O(n \cdot (n + m \cdot \log(m)))$.

- The last part of the algorithm is a complete search of the remaining edges works for $O(2^s)$, where s is the number of edges that remain outside all planar components.

The total complexity of the algorithm is:

$$O(n \cdot \log(n) + n \cdot (n + m \cdot \log(m)) + 2^s).$$

2.4 Example of the developed algorithm

An algorithm work example is described below. The considered graph consists of six vertices and nine edges (Figure 3).

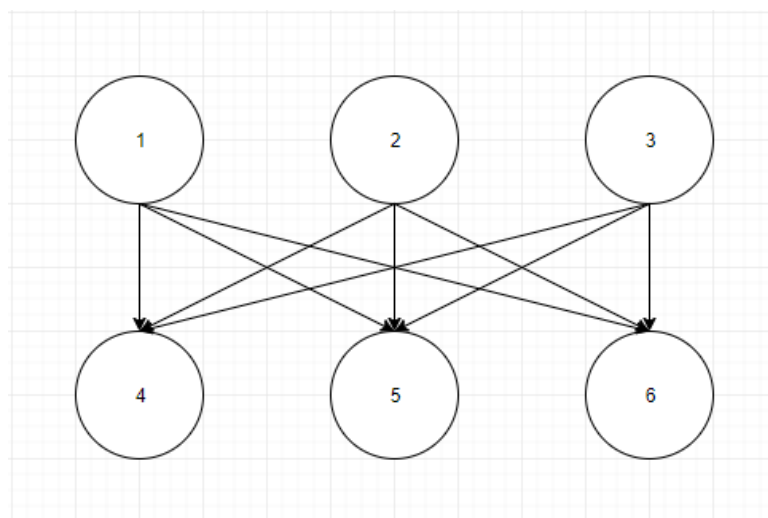


Figure 3 – Initial graph

1. Initially, the graph consists of six free vertices.
2. The first connectivity component is created. It adds a vertex 1 (in step 1 of the algorithm). After that, step 2 is executed twice, resulting in the component consisting of vertices 1, 4 and 5 (Figure 4).

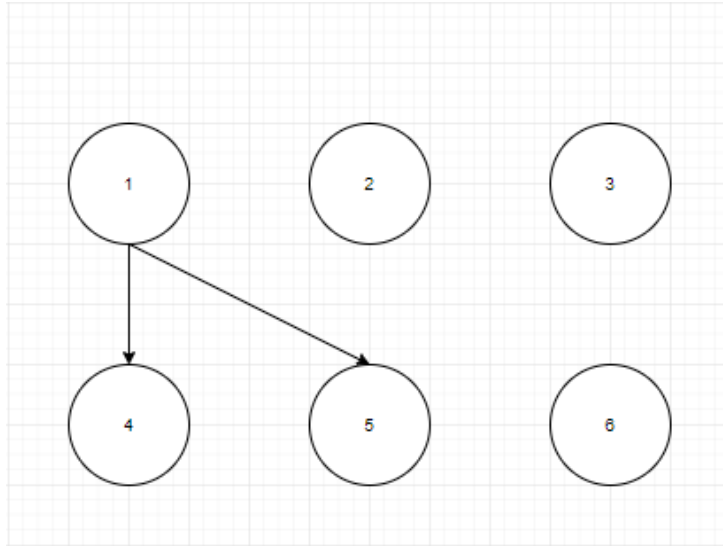


Figure 4 – One component (1, 4, 5)

3. Next, a vertex 2 is added to the component (Figure 5), because of the three remaining free vertices, this vertex has the maximum number of adjacent vertices inside the connectivity component.

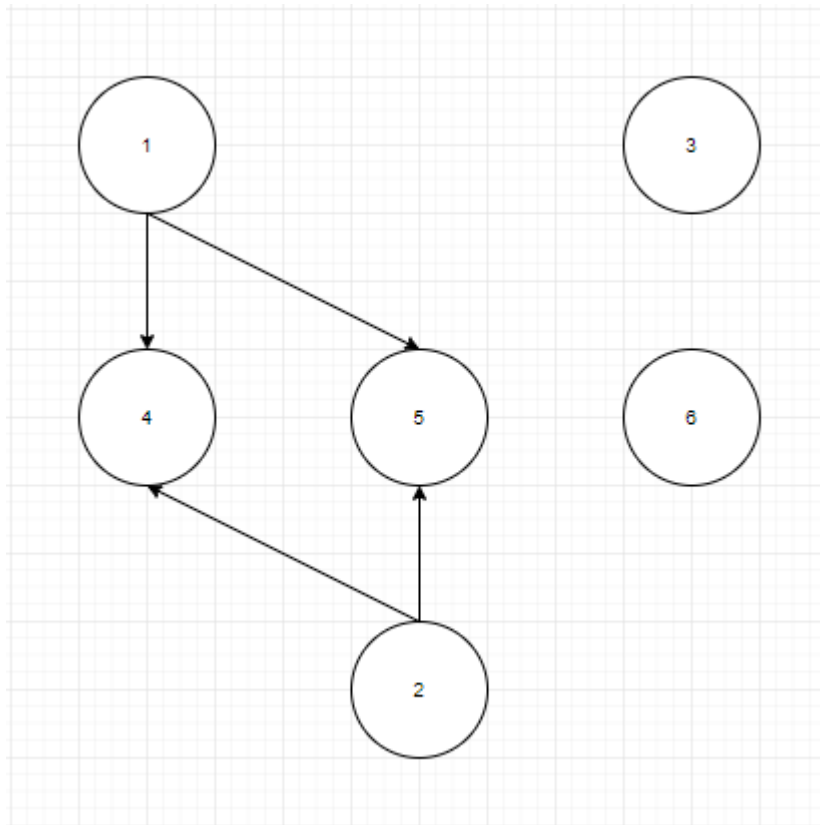


Figure 5 – One component (1, 4, 5, 2)

4. Next, a vertex 3 is added to the component (Figure 6)

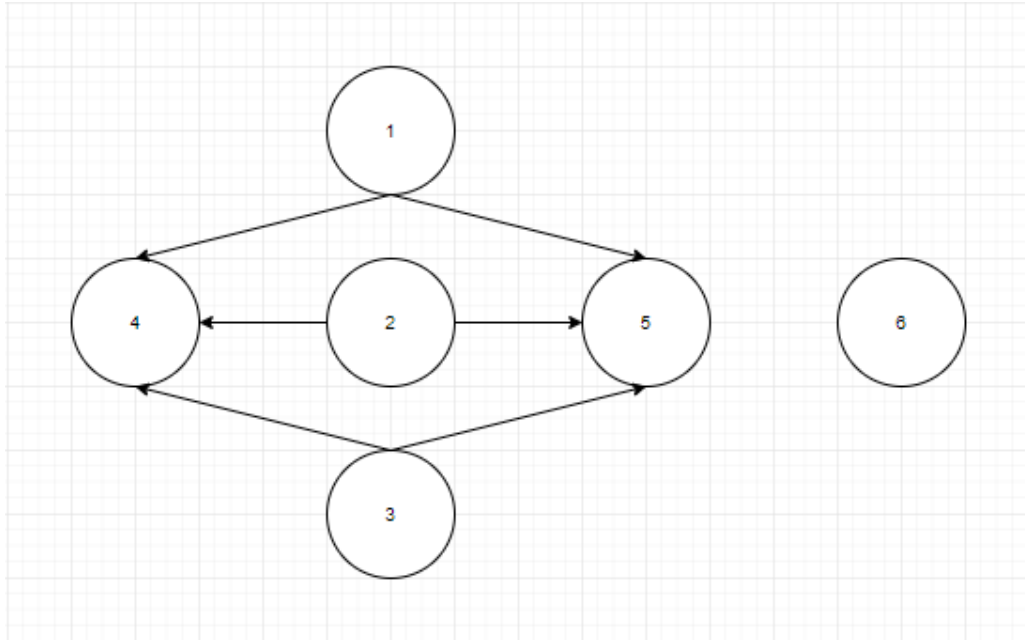


Figure 6 – One component (1, 4, 5, 2, 3)

5. A vertex 6 cannot be added to the component because it breaks the planarity. Therefore, a second planar component consisting of a single vertex 6 is created (Figure 7).

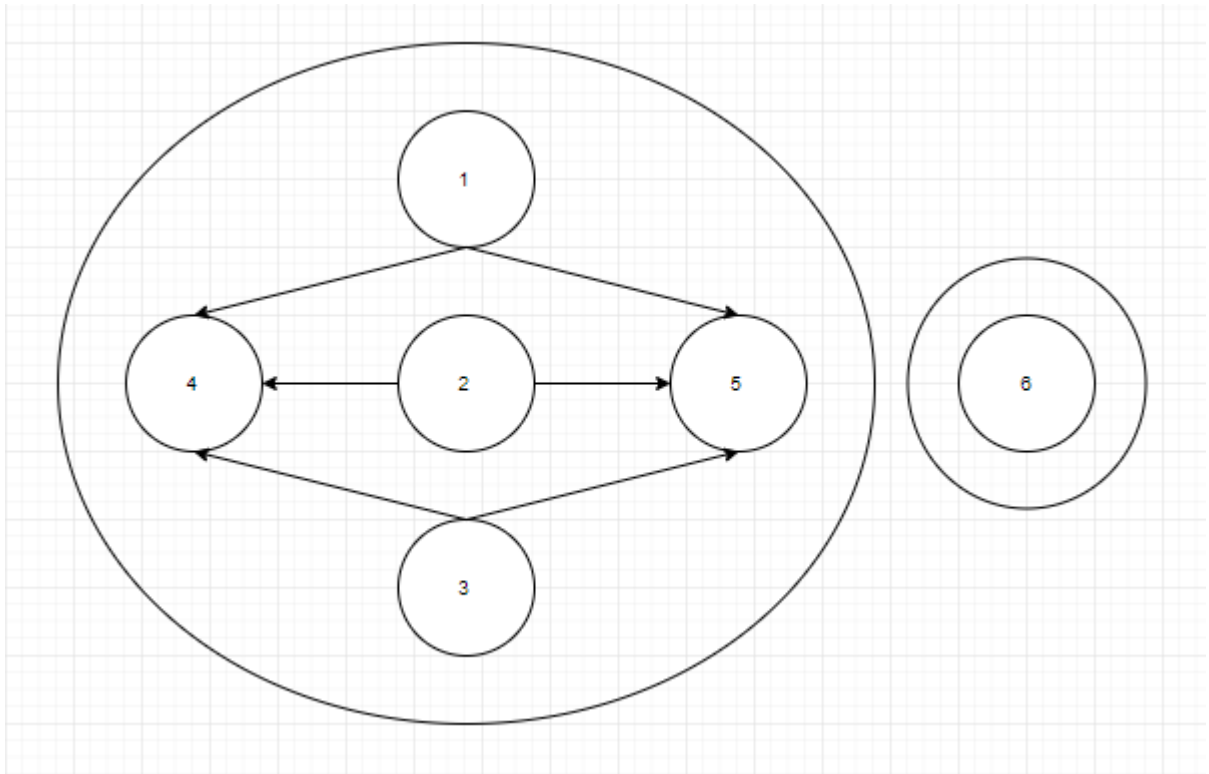


Figure 7 – Two components (1, 4, 5, 2, 3) and (6)

6. Further, using a full search of the remaining free edges (1-6, 2-6 and 3-6), the maximum possible number of edges is selected, which can be left in the original graph without violating the planarity (Figures 8 and 9).

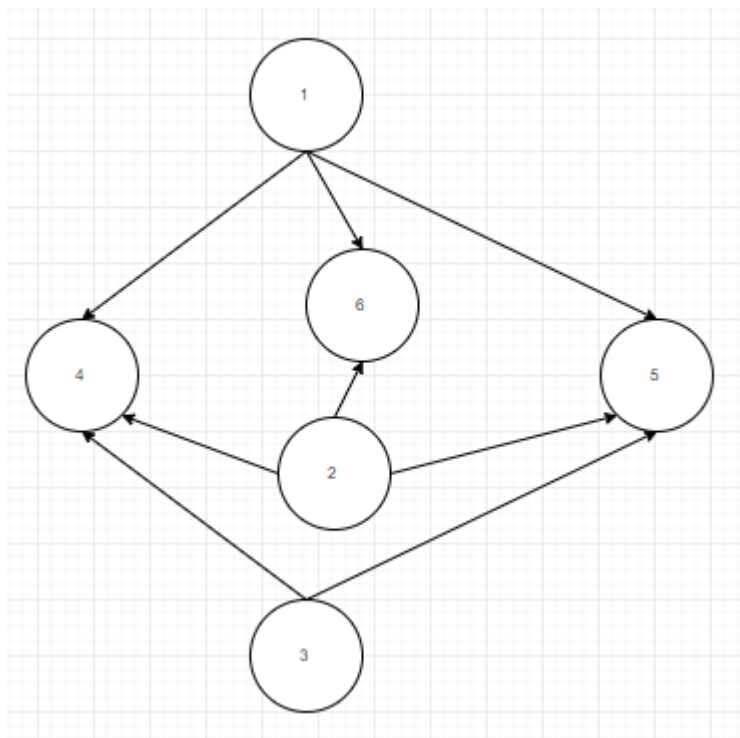


Figure 9 – Planar stacking of the original graph without edge 3-6.

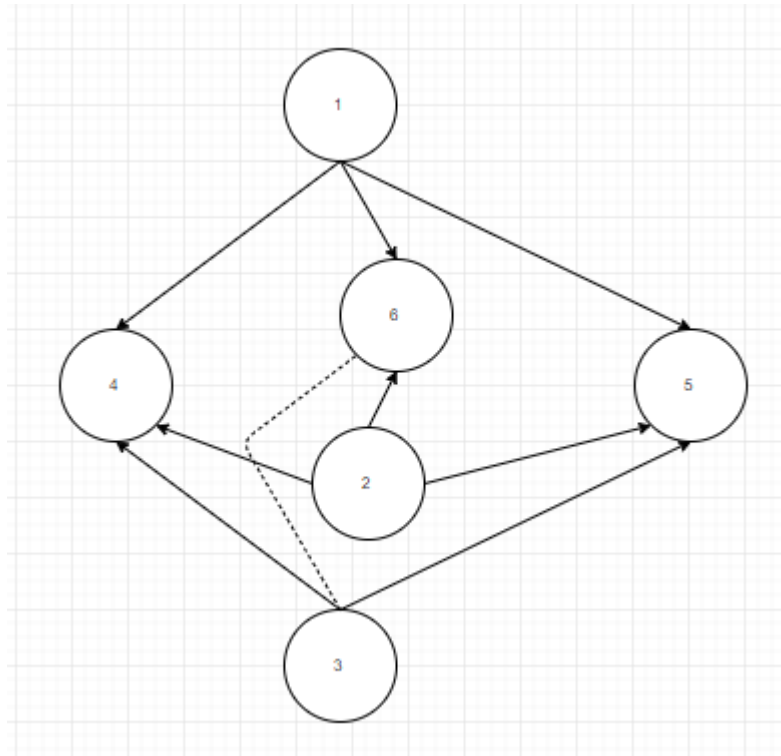


Figure 9 – Planar stacking of the original graph with minimal number of edge intersections