МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Автоматизированных систем управления

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полное название кафедры)

# Утверждаю

#### Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, инициалы, фамилия)

«09» июня 2022 г.

Томилов И.Н.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**

*Раченкова Максима Павловича*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество студента – автора работы)

*Разработка информационной системы для организации проектной деятельности. Модуль “Анализ результатов проектной деятельности”*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(тема работы)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет автоматики и вычислительной техники

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полное название факультета)

09.03.01 – Информатика и вычислительная техника

Направление подготовки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование направления подготовки бакалавра)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Руководитель

### от НГТУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ученая степень, ученое звание)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

### Автор выпускной квалификационной работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(факультет, группа)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

*Лыгина Нина Ивановна*

*АВТФ, АВТ-812*

*Раченков Максим Павлович*

*к.п.н., доцент*

**Консультанты по разделам:**

*П.В. Терещенко*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(краткое наименование раздела)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(краткое наименование раздела)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, инициалы, фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, инициалы, фамилия)

*Оценка интерфейса*

*В.А. Астапчук*

*Организационно-экономический раздел*

Новосибирск 2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Автоматизированных систем управления

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полное название кафедры)

##### УТВЕРЖДАЮ

# Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

Доставалов Д.Н.

*Введите здесь свой текст*

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА**

*Раченкову Максиму Павловичу*

студенту *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(фамилия, имя, отчество)

09.03.01 – Информатика и вычислительная техника

Направление подготовки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование направления подготовки бакалавра)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет автоматики и вычислительной техники

(полное название факультета)

*Разработка информационной системы для организации проектной деятельности. Модуль “Анализ результатов проектной деятельности”*

Тема \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полное название темы выпускной квалификационной работы бакалавра)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Цель работы: разработка информационной системы для организации командной работы студентов старших курсов, позволяющей аккумулировать, анализировать и отражать данные по результатам проектной деятельности*

Исходные данные (или цель работы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. *характеристика предметной области;*
2. *обзор аналогов;*
3. *обоснование выбора инструментальных средств;*
4. *постановка задачи;*
5. *проектирование системы;*
6. *оценка пользовательского интерфейса;*
7. *организационно-экономическая часть*

Структурные части работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание согласовано и принято к исполнению.

### Руководитель

### от НГТУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ученая степень, ученое звание)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

### Студент

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(факультет, группа)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

*Раченков Максим Павлович*

*Лыгина Нина Ивановна*

*к.п.н., доцент*

*АВТФ, АВТ-812*

1084/2

марта

01

Тема утверждена приказом по НГТУ № \_\_\_\_\_\_\_\_\_ от «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2022 г.

изменена приказом по НГТУ № \_\_\_\_\_\_\_\_\_ от «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

ВКР сдана в ГЭК № \_\_\_\_\_\_\_, тема сверена с данными приказа

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись секретаря государственной экзаменационной комиссии по защите ВКР, дата)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество секретаря государственной

экзаменационной комиссии по защите ВКР)

**7**

**7**

**Консультанты по разделам:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(краткое наименование раздела)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(краткое наименование раздела)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, инициалы, фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, инициалы, фамилия)

*П.В. Терещенко*

*Оценка интерфейса*

*В.А. Астапчук*

*Организационно-экономический раздел*

**РЕФЕРАТ**

ВКРБ содержит 64 с., 17 рис., 34 табл., 26 источников, 2 прил., 3 формулы.

ДИСТАНЦИОННОЕ ОБУЧЕНИЕ, ОБРАЗОВАТЕЛЬНАЯ ПЛАТФОРМА, АНКЕТИРОВАНИЕ, КОМАНДНЫЕ РОЛИ, САМООЦЕНКА, ПРОЕКТНАЯ ДЕЯТЕЛЬНОСТЬ, КОНФЛИКТЫ.

Цель работы – разработка информационной системы для организации командной работы студентов старших курсов, позволяющей аккумулировать, анализировать и отражать данные по результатам проектной деятельности.

Произведено исследование командного подхода в обучении в ходе проектной деятельности студентов высших учебных заведений. Осуществлен сравнительный анализ существующих систем - аналогов. Выявлены основные требования заказчика, с их учетом выбраны инструментальные средства и спроектирована информационная система. Для функционирования информационной системы определен перечень входных и выходных данных, сделано описание структуры таблиц базы данных, реализованы алгоритмы, произведено тестирование системы и отладка.

Разработано руководство пользователя и приведен контрольный пример. Осуществлена оценка пользовательского интерфейса и трудозатрат на разработку программной системы.

Система реализована с использованием клиент-серверной СУБД PostgreSQL, веб-приложения pgAdmin4, языка программирования С#, среды разработки Microsoft Visual Studio 2019, библиотек Npgsql, ZedGraph, менеджера пакетов NuGet.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc104372869)

[1.1 Актуальность разработки 7](#_Toc104372870)

[1.2 Командная работа в высших учебных заведениях 8](#_Toc104372871)

[1.3 Командная работа на кафедре АСУ НГТУ 9](#_Toc104372872)

[2 Обзор существующих аналогичных систем, формулирование проблемы и определение задач 13](#_Toc104372873)

[2.1 Обзор существующих аналогов 13](#_Toc104372874)

[2.2 Формулирование проблемы и определение задач 16](#_Toc104372875)

[3 Обоснование выбора инструментальных средств 18](#_Toc104372876)

[4 Описание входной и выходной информации 19](#_Toc104372877)

[4.1 Входная информация системы 19](#_Toc104372878)

[4.2 Выходная информация системы 20](#_Toc104372879)

[5 Структура таблиц базы данных 24](#_Toc104372880)

[5.1 Концептуальная модель 24](#_Toc104372881)

[5.2 Логическая и физическая модели 25](#_Toc104372882)

[6 Разработанные основные алгоритмы задач 38](#_Toc104372883)

[7 Руководство пользователя и контрольный пример 42](#_Toc104372884)

[8 Оценка пользовательского интерфейса 51](#_Toc104372885)

[8.1 Измерение удовлетворенности посредством анкетирования 51](#_Toc104372886)

[8.2 Контрольный список для оценки качества интерфейса 53](#_Toc104372887)

[9 Оценка трудозатрат на разработку проекта 55](#_Toc104372888)

[9.1 Расчет параметров для оценки трудоемкости разработки 55](#_Toc104372889)

[9.2 Расчет предварительной оценки трудоемкости 57](#_Toc104372890)

[9.3 Расчет детальной оценки трудоемкости разработки 58](#_Toc104372891)

[9.4 Расчет длительности работы 60](#_Toc104372892)

[ЗАКЛЮЧЕНИЕ 61](#_Toc104372893)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 62](#_Toc104372894)

[ПРИЛОЖЕНИЕ А. Фрагмент программного кода анализа текстовых анкет 65](#_Toc104372895)

[ПРИЛОЖЕНИЕ Б. Фрагмент программного кода формы по конфликтам 75](#_Toc104372896)

ВВЕДЕНИЕ

Командность как компетенция, определяющая результативность взаимодействия с членами команды, умение соблюдать баланс интересов подразделения и личных интересов [1], по версии компании «Партнёрство навыки XXI века» - одна из значимых компетенций 21 века. Она отражает способность работать на совместный результат, понимать свою роль и свой вклад в общее дело [2]. Роль эффективной командной работы как обязательного ресурса IT-компаний для достижения своих целей непрерывно возрастает. Сложность информационных систем, поддерживающих все аспекты социальной жизни общества, налагает требование производительной координации на компании разработчиков.

В 2020 году сервис по поиску высокооплачиваемой работы SuperJob выяснил у работодателей, по каким критериям в их компаниях осуществляется подбор молодых специалистов [3]. В опросе приняли участие 1000 представителей предприятий и организаций из всех округов страны. В частности, респондентам было предложено определить две наиболее важные для их компании личностные компетенции выпускников высшей школы. В число «тройки лидеров» опроса вошли:

• мотивация достижений (48% респондентов назвали это качество);

• организованность (44%);

• командность (42%).

Причем показатель «мотивация достижений» потерял 14% голосов за семь лет, прошедших с первого опроса на данную тему, а показатель «командность», наоборот, получил за это время еще 14% голосов респондентов [3].

Теоретики командной работы обозначают разный набор необходимых навыков для формирования компетенции командность, например, коммуникабельность, системный анализ и оценка, организация рабочего времени [4]. По версии модели компетенций продуктивной команды Питера Хоукинса [5] центральная роль отводится навыку доверие, отражающему плодотворность, с которой происходит обмен рабочей информацией. Период пандемии, с одной стороны, сократил очный контакт сотрудников в офисе, с другой, вынудил людей заглянуть друг другу в личное пространство, поскольку дистанционное рабочее место организовано дома или на даче. Для лидера команды добавилось требование – обучить каждого сотрудника взять на вооружение сетевые инструменты, а именно, приложения для организации видеоконференций, онлайн доски, мессенджеры, чаты.

Работа в команде успешно используется в высшей школе как способ активизации деятельности обучающихся и условие развития их личностных компетенций [6,7]. На своем пути к действенной команде группа студентов проходит несколько стадий, каждая из которых характеризуется определенными показателями. Измерение и отслеживание динамики этих показателей позволяет ставить цели, определять точки приложения усилий, направленных на развитие команды, и получать количественные результаты этих усилий.

Помимо психологической, методологической составляющих в организации командной работы студентов, ее успешность определяется способом фиксации результатов работы команд студентов, удобным для всех участников работы, а также возможностью оперативного и эффективного отражения показателей деятельности в заданных срезах.

Цель работы: разработка информационной системы для организации командной работы студентов старших курсов, позволяющей аккумулировать, анализировать и отражать данные по результатам проектной деятельности.

В качестве исходных данных для проекта предполагается использовать результаты работы студентов старших курсов в учебной дисциплине «Программная инженерия» по направлению подготовки 09.03.01 «Информатика и вычислительная техника» в НГТУ за период, определенный руководителем [8]. **1 Описание предметной области**

1.1 Актуальность разработки

Определиться с предпочтительными ролями в команде разработчику программного обеспечения целесообразно и своевременно в период обучения в университете.

В ФГОС ВО 3+ сформирована компетенция УК-3 «Способность осуществлять социальное взаимодействие и реализовывать свою роль в команде» [8]. В основной образовательной программе 09.03.01 «Информатика и вычислительная техника» предусмотрены различные формы командной работы, относящиеся к активным, интерактивным методам обучения.

В требуемом варианте организации командной работы заложен характер предметной деятельности в команде при разработке обучающимися программного продукта в рамках каскадной модели жизненного цикла. Цели разработки на каждом этапе определены в соответствии с требованиями профессиональных стандартов [9].

Объектом исследования данной ВКР является командная работа.

Задачи проектирования: разработать и ввести в эксплуатацию модули информационной системы, позволяющие фиксировать условия командной работы, исходные данные, ход работы, промежуточные и конечную оценку деятельности команд, вывод данных по заявленным параметрам.

В моей зоне ответственности – модуль анализа результатов деятельности. Анкетирование и личные кабинеты разрабатываются другим студентом.

Актуальность работы: полученная в ходе ВКР информационная система позволит кафедре АСУ провести анализ накопленной информации по организации командной работы с целью выработки методических рекомендаций преподавателям кафедры по способу формирования, методам сопровождения эффективных команд студентов для выполнения проектной деятельности на старшем курсе обучения по специальности 09.03.01 Информатика и вычислительная техника; повысить личностную компетенцию командность выпускников кафедры АСУ.

Ограничения и допущения проектирования.

Разработка информационной системы осуществляется как выпускная квалификационная работа студента кафедры АСУ, работа исполнителя не оплачивается. Информационная система является собственностью кафедры АСУ. Данные для наполнения информационной системы предоставляются руководителем и не могут быть использованы исполнителем в коммерческих и иных целях.

1.2 Командная работа в высших учебных заведениях

В научной литературе приведены примеры методов командой работы, которые практикуются в высших учебных заведениях [10]:

- Метод «Мозгового штурма». Команда из 5–8 студентов 15-20 минут формирует идеи на заданную тему, затем следует коллективное обсуждение идей; часть отбрасывается, остальные берутся в проработку.

- Обучение в командных достижениях [11]: заключается в системе оценки индивидуальных работ. Оценка осуществляется по прогрессивно-сравнительному признаку: студент может пополнить копилку команды только в том случае, если его оценка за данную работу выше средней его оценки за предыдущие работы.

- Метод учебного турнира [12]: после изучения нового материала в турнирном поединке встречаются студенты из разных команд с одинаковым уровнем учебных достижений.

- Метод командной поддержки индивидуального обучения, описанный Селевко Г. К., заключается в предоставлении командным группам возможности продвигаться по учебной программе в индивидуальном темпе. Обучающиеся работают в малых группах над индивидуальными заданиями, в процессе выполнения которых они могут обращаться друг к другу и за консультацией [13]. Студенты также могут проверять работы друг друга, помогать исправлять допущенные ошибки. Преподаватель наблюдает за работой малых групп, а также постепенно выдает новый учебный материал группам, которые закончили работать над индивидуальными заданиями по предыдущему материалу. Индивидуальные задания проверяются специально назначенными преподавателем студентами из разных групп. Им выдаются листы с ответами для проверки индивидуальных работ. В это время преподаватель имеет возможность индивидуально работать с каждой малой группой. В конце недели подводится итог: сколько тем изучила каждая группа и каков суммарный учебный результат группы по индивидуальным заданиям.

Метод группового исследования заключается в том, что команды исследуют конкретный вопрос учебной дисциплины, готовят доклад и выступают с ним в группе. Вопросы по теме распределяются между командами так, чтобы в итоге выступлений охватить весь учебный материал темы. Внутри команды каждый обучающийся исследует свою часть, собирая необходимый материал, предоставляет его в группу, и далее на основе собранных частей формируется общий доклад группы [11].

Наиболее значимым считается проект, где роли обучающихся в команде определяются их направлением подготовки [8].

1.3 Командная работа на кафедре АСУ НГТУ

При организации командной работы студентов, реализуемой на кафедре АСУ НГТУ в учебной дисциплине 4 курса «Программная инженерия» по направлению подготовки 09.03.01 «Информатика и вычислительная техника» [9], существуют определенные правила:

- Команды формируются студентами одного потока, специальности самостоятельно на добровольной основе.

- Количество членов в команде составляет 5-7 человек в зависимости от численности группы.

- Кафедрой утверждается перечень тем для реализации на текущий семестр и руководителей для каждой темы. Перечень передается лидерам уже сформированных команд.

- На семинарском занятии команды борются за право реализовать определенную тему. Решением заведующего кафедры закрепляется связка «тема- команда».

- Студенты потока, оставшиеся вне команды после добровольной разбивки, при наличии «вакантного» места присоединяются к команде, выполняющей определенный проект из списка запланированных выпускающей кафедрой.

В команде выделяют три категории ролей по их трудовым особенностям: роли действия, интеллектуальные роли, социально направленные роли [2]. Для гармонично слаженной команды необходимо наличие ролей всех трех типов.

Для определения командной роли используют различные психологические тесты. Одним из таких тестов является тест Р. Белбина [14]. Тест показывает, в какой мере человек использует в деятельности каждую из командных ролей, выявленных в процессе длительного исследования.

Р. Белбин выделил следующие роли, которые исполняет человек в зависимости от личных особенностей и качеств: председатель, формирователь, мыслитель, исполнитель, разведчик, оценщик, коллективист, доводчик.

По результатам прохождения теста Р. Белбина, участники команд договариваются о своей командной роли. Критичным является наличие в команде одного и только одного председателя и одного - двух исполнителей.

Наивысший балл по командной роли показывает, насколько хорошо студент может исполнять эту роль в команде. Следующий результат после наивысшего обозначает поддерживающую роль, на которую студент может переключиться, если основная командная роль занята в группе [9].

Когда роли определены, темы закреплены, промежуточные этапы и даты их обозначены, команды приступают к работе. В ходе проектирования используются ранее перечисленные методы групповой работы, например:

- метод турнира – соревновательный элемент при защите выбора темы проектирования;

- метод группового исследования при защите презентации по теме проекта;

- метод мозгового штурма при выборе алгоритмов для реализации.

При групповой работе возможны конфликты в команде, иногда приводящие к ее распаду. По условию проекта информационная система должна позволять, кроме прочего, комфортно фиксировать внутрикомандные конфликты с привязкой к ряду характеристик команды и участника.

По результатам применения командного подхода в ходе проектной деятельности студентов кафедры АСУ НГТУ можно сделать следующие выводы [8], [2]:

- Совместное проектирование вырабатывает у студентов принятие иного взгляда на задачу, снимает зажатость, позволяет раскрыть способности, улучшить мотивацию к обучению.

- Командный формат работы повышает личную ответственность участников – не хочется подвести «своих», что приводит к «подтягиванию» более слабых студентов за лидерами. Оценка однокурсников для молодых людей не менее весома, чем балл, выставленный преподавателем.

- У студентов вырабатывается самостоятельность в принятии управленческих решений, акцент освоения дисциплины смещается в зоны студент-студент и студент-учебный материал.

- Совместное проектирование способствует профессиональному самоопределению.

- Командные связки сохраняются при выполнении практических заданий по предметам в следующем семестре, что содействует основательному усвоению материалов всеми студентами.

Для удобства фиксации прогресса у студентов - участников проектной деятельности, по запросу заказчика, система должна включать возможности:

- анализа текстовых и балльных анкет;

- отображения оценок (итоговых и средней оценки за этапы) за проект по группе;

- отображение результатов теста Белбина по группе;

- анализа результатов самооценки команды за выбранный этап;

- учета и анализа конфликтов, в том числе закончившихся распадом команды.

Для подбора инструмента для сопровождения процесса проектной деятельности в условиях командной работы на кафедре АСУ НГТУ в следующем разделе будет проведено исследование существующих программных продуктов требуемого профиля.

2 Обзор существующих аналогичных систем, формулирование проблемы и определение задач

2.1 Обзор существующих аналогов

По результатам изучения рынка программного обеспечения можно выделить две группы продуктов:

- системы для дистанционного обучения eLearning, например, Moodle, ATutor, ILIAS, Diskurs, Classroom, DiSpace;

- системы управления проектами для командной работы, такие как Jira, GanttPro, Hygger, YouTrack.

Moodle (модульная объектно-ориентированная динамическая учебная среда)– это веб-система для организации дистанционного обучения, находится в открытом доступе. Она включает функционал создания и управление учебным курсом, управление пользователями внутри курса, отслеживание выполнения заданий, публикации учебных материалов, тестирования. Moodle позволяет настроить отчетность [15].

ATutor развивается за счет собственного сообщества на GitHub, похожа на Moodle, у нее меньше готовых модулей, но есть встроенный конструктор.

Решение для немецких университетов ILIAS включает встроенный конструктор тестов. Кастомизация осуществляется за счет имеющихся модулей и собственных разработок.

Отечественная платформа Diskurs представляет собой конструктор курсов на H5P. Техническая поддержка и добавление функций в Diskurs предполагают оплату [16].

Google Classroom - это интернет-сервис для онлайн-обучения. Google собрала в одном сервисе несколько своих инструментов: диск для хранения файлов, Google Docs для публикации текстовых лекций, презентации, опросы, сервис для видеовстреч и календарь для планирования обучения [17].

DiSpace – сайт для дистанционного обучения, используемый в НГТУ. Ресурс является базой для получения учебных материалов по курсам университета. В DiSpace реализована возможность отправлять работы студентов на проверку и проводить тестирование знаний, но платформа не имеет необходимой для командного обучения функциональности.

Достоинства и недостатки систем дистанционного обучения приведены в таблице 1.

Таблица 1. Достоинства и недостатки систем дистанционного обучения

|  |  |
| --- | --- |
| Достоинства | Недостатки |
| - Бесплатные или малобюджетные | Самостоятельная установка, настройка, обслуживание |
| - Гибкие, имеют открытый код | Отсутствует или слабая техническая поддержка |

Далее рассмотрены подробнее системы управления проектами для командной работы для групп разработчиков.

Jira - система для разработчиков, самая закупаемая в России по оценке компании YouGile [18]. Пользоваться ею может любое количество сотрудников. К плюсам Jira можно отнести: объемный функционал и возможности для кастомизации, мощный API, удобное планирование спринтов, Agile-бэклог, более 3000 приложений Jira Software, расчет скорости и объема работы команды за спринт. Прогресс работы по каждому проекту можно измерить и визуализировать. Минусом является то, что система сложна в освоении.

GanttPro оптимальна для сферы IT для команд до 20 человек. Инструмент для аналитики и визуализации данных по проектам позволяет распределять нагрузку на сотрудников с учетом ресурсов и времени, которое требуется для выполнения задач. В GanttPro предусмотрена возможность сохранить визуализированные данные в формате PDF. Снижает привлекательность системы то, что она сложна в освоении, GanttPro не хватает интеграций с другими сервисами, например, Microsoft Outlook.

Hygger создавалась как упрощенная Jira. подходит для Agile-команд разработчиков. Интерфейсы у Hygger простые, имеются удобный бэклог, качественная приоритизация фич, карта развития продукта. Из недостатков можно отметить отсутствие массовых действий.

YouTrack применяется как баг-трекер и система управления задачами для команд разработки. В системе реализованы быстрая генерация отчетов и обработка проблем, удобный интеллектуальный поиск при помощи поисковых запросов с подсказками и подсветкой ошибок. Рабочими процессами в ней легко управлять при помощи клавиатуры. Гибкость продукта проявляется в том, что можно настроить атрибуты задач под себя, есть возможность пакетно изменять поля задач, назначать исполнителей, связывать между собой задачи. К минусам можно отнести меньшее число возможностей для интеграции по сравнению с Jira и необходимость длительного изучения интерфейса [18].

Сравнение возможностей обозначенных систем управления проектами для командной работы представлены в таблице 2.

Таблица 2. Сравнение возможностей систем управления проектами

|  |  |  |  |
| --- | --- | --- | --- |
| Название системы | Рекомендуемая численность команды, человек | Преимущества | Недостатки |
| Jira | От 100 | >3000 приложений Jira, мощный API, гибкая кастомизация | Сложная |
| GanttPro | До 20 | Удобна для аналитики, визуализации данных | Слабая интеграция |
| Hygger | До 10 | Простой интерфейс, приоритизация задач | Нет массовых действий |
| YouTrack | До 100 | Интеллектуальный поиск, настройка атрибутов задач под исполнителя | Слабая интеграция |

Недостатками всех систем из таблицы 2 являются: сложный интерфейс, требующий значительного времени на его освоение и то, что все они платные, что затрудняет их использование для задач высших учебных заведений.

Таким образом, некоторые продукты первой группы содержат элементы командной работы, но не позволяют реализовать весь объем задач командной работы в проектном обучении на кафедре АСУ НГТУ. Продукты второй группы требуют значительных материальных и временных вложений. Можно сделать вывод, что ни один из обозначенных программных продуктов не обеспечивает все необходимые функции, требуется разработка информационной системы.

2.2 Формулирование проблемы и определение задач

Для повышения качества разрабатываемой системы в заданные сроки было принято решение о разделении разработки системы на следующие задачи:

1. Проектирование и программирование БД, содержащей:
   * ответы студентов на вопросы анкет;
   * результаты теста Р. Белбина;
   * списки групп;
   * списки команд;
   * списки проектов;
   * статистику этапов проектирования: промежуточные и итоговые оценки (и самооценки) по проектам;
   * статистику конфликтов в команде.
2. Анализ текстовых анкет:
   * создание и редактирование списка слов для анализа;
   * подсчет количества и частоты слов в ответах;
   * отображение результатов в виде столбиковой и лепестковой диаграммы, таблицы.
3. Анализ балльных анкет:
   * поиск 3-х вопросов с максимальными и минимальными баллами;
   * отображение средних баллов по анкете в виде графика и гистограммы;
   * отображение результатов по группе/ команде/ году и семестру/ предмету.
4. Отображение оценок (итоговых и средней оценки за этапы) за проект по группе.
5. Отображение результатов теста Р. Белбина по группе.
6. Анализ результатов самооценки по команде.
7. Создание, редактирование и просмотр комментариев преподавателя о командах.
8. Анализ конфликтов:
   * создание и редактирование причин конфликтов;
   * добавление записей о конфликте в команде;
   * анализ числа распавшихся и не распавшихся команд по типам конфликта;
   * график числа распавшихся команд по годам;
   * таблица соотношения распавшихся и нераспавшихся команд.

Для эффективного решения поставленных задач в следующей главе будет произведен выбор инструментальных средств.

3 Обоснование выбора инструментальных средств

В рамках проекта требуется организовать хранение данных, поэтому для разработки информационной системы нужно проектировать базу данных.

Для разработки данной информационной было решено использовать клиент-серверную СУБД – PostgreSQL, администрирование СУБД PostgreSQL веб-приложением pgAdmin4, язык программирования С#, среду разработки – Microsoft Visual Studio 2019, библиотеки Npgsql, ZedGraph, менеджер пакетов NuGet.

PostgreSQL – реляционная СУБД с открытым исходным кодом. Для некоммерческих целей использование PostgreSQL бесплатно. СУБД PostgreSQL поддерживает большое количество языков программирования. Главным преимуществом является скорость обработки информации и функциональность СУБД. Для управления СУБД был выбран инструмент pgAdmin4. Программа является простой в использовании, таблицы экспортируются в удобный для дальнейшего использования формат [19].

C# – объектно-ориентированный язык программирования. Язык программирования имеет большое количество библиотек. Такое разнообразие облегчит написание кода [20].

Microsoft Visual Studio - наиболее подходящая среда разработки для выбранного языка программирования. Она содержит в себе множество платформ для разработки графического приложения и не требует сложной настройки дополнительных пакетов.

ZedGraph - бесплатная и богатая библиотека вывода различных видов графиков. Отличается упрощенной графикой, простотой в использовании [21]. Также имеется поддержка настраивания через дизайнер форм Visual Studio.

Npgsql – библиотека, используемая для подключения к базе данных.

NuGet - система управления пакетами для платформ разработки Microsoft, в первую очередь библиотек .NET Framework. Управляется .NET\_Foundation. При установке пакета через NuGet, система копирует файлы библиотеки и автоматически обновляет проект.

4 Описание входной и выходной информации

4.1 Входная информация системы

В таблице 3 представлен перечень входной информации для информационной системы.

Таблица 3 – Входные данные

|  |  |  |
| --- | --- | --- |
| Обозначение | Название | Этап формирования |
| D1 | Студент | Берется из БД, формируется в другом модуле |
| D2 | Группа | Берется из БД, формируется в другом модуле |
| D3 | Дисциплина | Берется из БД, формируется в другом модуле |
| D4 | Проект | Берется из БД, формируется в другом модуле |
| D5 | Этап | Берется из БД, формируется в другом модуле |
| D6 | Оценка этапа | Берется из БД, формируется в другом модуле |
| D7 | Результаты теста Белбина | Берется из БД, формируется в другом модуле |
| D8 | Конфликт | Добавление, удаление преподавателем |
| D9 | Тип конфликта | Добавление, удаление преподавателем |
| D10 | Оценка сокомандников | Берется из БД, формируется в другом модуле |
| D11 | Анкета | Берется из БД, формируется в другом модуле |

Продолжение таблицы 3

|  |  |  |
| --- | --- | --- |
| D12 | Вопрос текстовой анкеты | Берется из БД, формируется в другом модуле |
| D13 | Ответ на вопрос текстовой анкеты | Берется из БД, формируется в другом модуле |
| D14 | Вопрос балльной анкеты | Берется из БД, формируется в другом модуле |
| D15 | Ответ на вопрос балльной анкеты | Берется из БД, формируется в другом модуле |
| D16 | Категория слова | Добавление, удаление преподавателем |
| D17 | Слово для анализа ответов на текстовые анкеты | Добавление, удаление преподавателем |
| D18 | Комментарий к команде | Добавление, изменение, удаление преподавателем |
| D19 | Данные для подключение к БД | Добавление, изменение преподавателем |

4.2 Выходная информация системы

В таблице 4 представлен перечень выходной информации.

Таблица 4 – Выходные данные

|  |  |  |
| --- | --- | --- |
| Обозначение | Название | Этап формирования |
| D1 | Список команд | Переход в окно «Самооценка команд», «Обработка балльных анкет», «Редактор конфликтов» |
| D2 | Список анкет | Переход в окно «Обработка текстовых анкет», «Обработка балльных анкет» |

Продолжение таблицы 4

|  |  |  |
| --- | --- | --- |
| D3 | Список слов для анализа | Переход в окно «Обработка текстовых анкет», «Редактор словаря» |
| D4 | Список категорий слов | Переход в окно «Обработка текстовых анкет», «Редактор словаря» |
| D5 | Список групп | Переход в окно «Оценки», «Обработка балльных анкет», «Результаты теста Белбина» |
| D6 | Список годов | Переход в окно «Конфликты», «Обработка балльных анкет» |
| D7 | Список дисциплин | Переход в окно «Конфликты», «Обработка балльных анкет» |
| D8 | Список конфликтов | Переход в окно «Конфликты», «Редактор конфликтов» |
| D9 | Список типов конфликтов | Переход в окно«Конфликты», «Редактор конфликтов» |
| D10 | Список этапов проекта | Переход в окно «Самооценка команд» |
| D11 | Таблица «Количество выбранных слов в ответах на текстовую анкету» | Нажатие на кнопку «Таблица» окна «Обработка текстовых анкет» |
| D12 | Столбиковая диаграмма «Количество выбранных слов в ответах на текстовую анкету» | Нажатие на кнопку «Столбиковая диаграмма» окна «Обработка текстовых анкет» |

Продолжение таблицы 4

|  |  |  |
| --- | --- | --- |
| D13 | Лепестковая диаграмма «Количество выбранных слов в ответах на текстовую анкету» | Нажатие на кнопку «Лепестковая диаграмма» окна «Обработка текстовых анкет» |
| D14 | Три вопроса балльной анкеты с минимальным средним баллом | Нажатие на кнопку «Мин/макс» окна «Обработка балльных анкет» |
| D15 | Три вопроса балльной анкеты с максимальным средним баллом | Нажатие на кнопку «Мин/макс» окна «Обработка балльных анкет» |
| D16 | График «Средний балл балльной анкеты/число студентов» | Нажатие на кнопку «График» окна «Обработка балльных анкет» |
| D17 | Гистограмма «Средний балл балльной анкеты/число студентов» | Нажатие на кнопку «Гистограмма» окна «Обработка балльных анкет» |
| D18 | Таблица «Оценки группы» (итоговая оценка и среднее за этапы) | Выбор группы из выпадающего списка в окне «Оценки» |
| D19 | Таблица «Результаты теста Р.Белбина» по группе | Выбор группы из выпадающего списка в окне «Результаты теста Р.Белбина» |
| D20 | Таблица «Самооценка команды» | Выбор команды и этапа из выпадающих списков в окне «Самооценка команд» |
| D21 | Таблица «Средняя оценка сокомандников» | Выбор команды и этапа из выпадающих списков в окне «Самооценка команд» |

Продолжение таблицы 4

|  |  |  |
| --- | --- | --- |
| D22 | График «Число распавшихся команд» | Нажатие на кнопку «График» окна «Конфликты» |
| D23 | Соотношение распавшихся и не распавшихся команд | Нажатие на кнопку «Соотношение» окна «Конфликты» |
| D24 | Таблица «Статистика конфликтов» | Нажатие на кнопку «Статистика конфликтов» окна «Конфликты» |
| D25 | Таблица «Комментарии к командам» | Нажатие на кнопку «Комментарии» окна «Конфликты» |

5 Структура таблиц базы данных

Проектирование базы данных включает следующие этапы:

- концептуальный этап;

- логический этап;

- физический этап.

5.1 Концептуальная модель

Инструментом для построения модели данных на начальном этапе проектирования является модель «сущность-связь» [22, 23].

Для построения ER-модели необходимо выделить все типы сущностей, которые будут храниться в базе данных и связи между ними. Взаимодействие между сущностями определяется одним из трех видов связей: «один к одному» (1:1), «один ко многим» (1:M), «многие ко многим» (M:N).

ER-модель информационной системы представлена на рисунке 1.

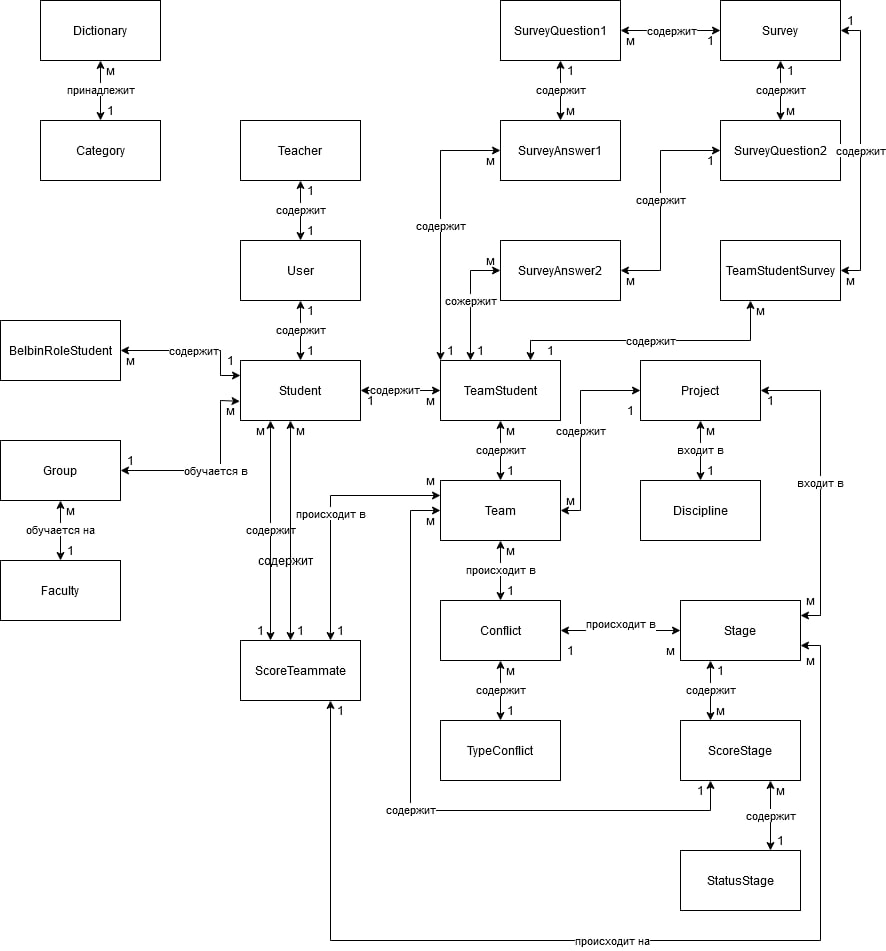


Рисунок 1 – ER-диаграмма (концептуальная модель данных)

5.2 Логическая и физическая модели

Построение логической модели осуществляется преобразованием концептуальной модели. На этом этапе определим структуры таблиц и атрибуты сущностей.

Логическая модель данных представлена на рисунке 2.

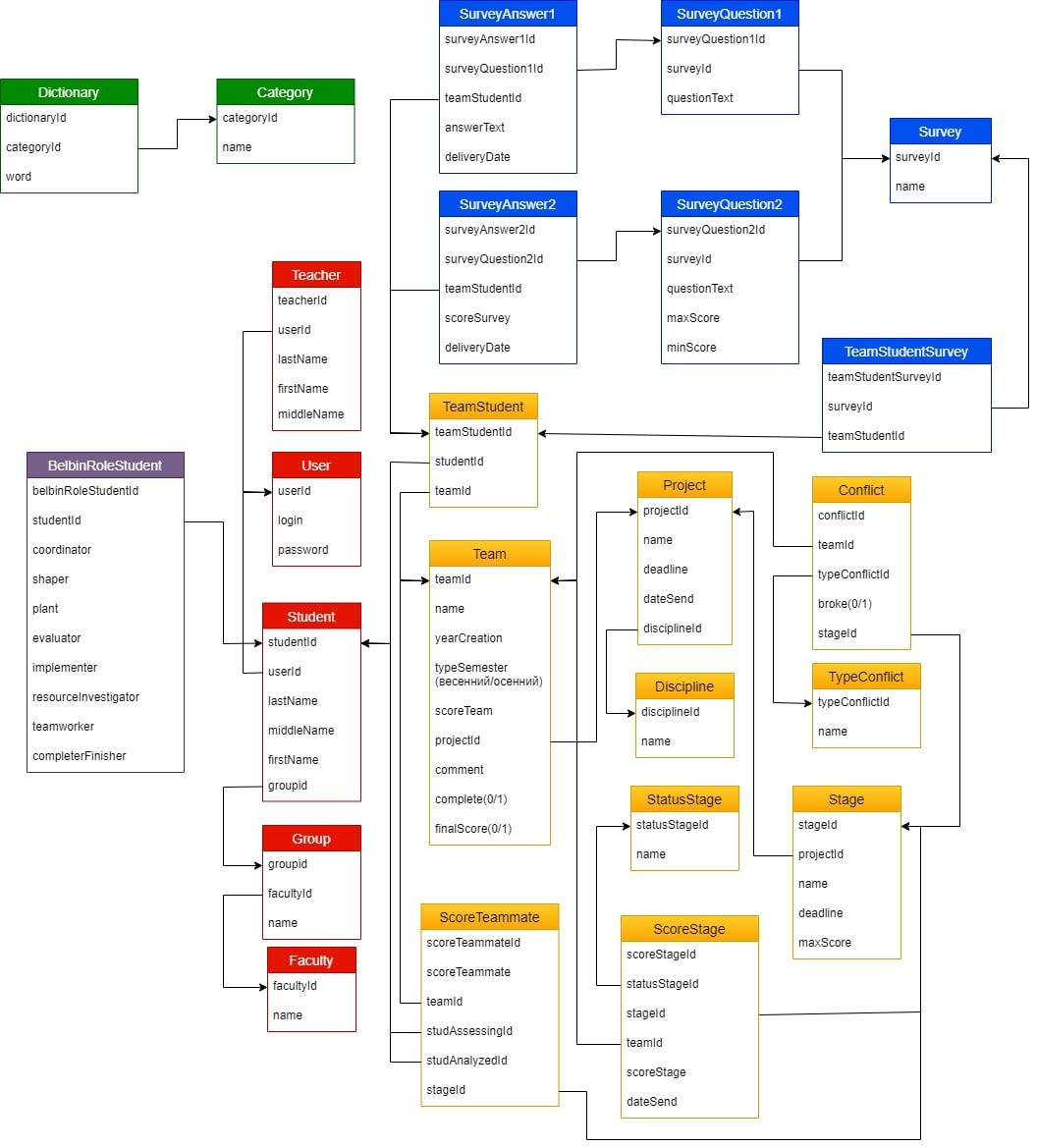


Рисунок 2 – Логическая модель данных

Описание таблиц, представленных в логической модели, приведено ниже.

В таблице 5 содержится информация о студентах.

Таблица 5 – Структура таблицы Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| studentId | integer | Идентификатор | V |  |

Продолжение таблицы 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| userId | integer | Идентификатор пользователя |  | V |
| lastName | character varying(100) | Фамилия |  |  |
| middleName | character varying(100) | Отчество |  |  |
| firstName | character varying(100) | Имя |  |  |
| groupId | integer | Идентификатор группы |  | V |

В таблице 6 содержится информация о пользователях.

Таблица 6 – Структура таблицы User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| userId | integer | Идентификатор | V |  |
| login | character varying(100) | Логин |  |  |
| password | character varying(100) | Пароль |  |  |

В таблице 7 содержится информация о группах.

Таблица 7 – Структура таблицы Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| groupId | integer | Идентификатор | V |  |
| facultyId | integer | Идентификатор факультета |  | V |
| name | character varying(100) | Название |  |  |

В таблице 8 содержится информация о преподавателях.

Таблица 8 – Структура таблицы Teacher

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| teacherId | integer | Идентификатор | V |  |
| userId | integer | Идентификатор пользователя |  | V |
| lastName | character varying(100) | Фамилия |  |  |
| firstName | character varying(100) | Имя |  |  |
| middleName | character varying(100) | Отчество |  |  |

Таблица 9 включает данные о факультете.

Таблица 9 – Структура таблицы Faculty

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| facultyId | integer | Идентификатор | V |  |
| name | character varying(100) | Название |  |  |

Таблица 10 определяет категорию характеристики для текстовых анкет.

Таблица 10 – Структура таблицы Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| categoryId | integer | Идентификатор | V |  |
| name | character varying(100) | Название |  |  |

В таблице 11 представлены данные о баллах студента для роли из теста Р. Белбина.

Таблица 11 – Структура таблицы BelbinRoleStudent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| belbinRoleStudentId | integer | Идентификатор | V |  |
| studentId | integer | Идентификатор студента |  | V |
| coordinator | integer | Баллы роли председатель |  |  |
| shaper | integer | Баллы роли формирователь |  |  |
| plant | integer | Баллы роли мыслитель |  |  |
| evaluator | integer | Баллы роли оценщик |  |  |
| implementer | integer | Баллы роли исполнитель |  |  |
| resourceInvestigator | integer | Баллы роли разведчик |  |  |
| teamworker | integer | Баллы роли коллективист |  |  |
| completerFinisher | integer | Баллы роли доводчик |  |  |

Таблица 12 содержит данные об анкетах.

Таблица 12 – Структура таблицы Survey

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| surveyId | integer | Идентификатор | V |  |
| name | character varying(100) | Название |  |  |

Таблица 13 включает информацию по словарю для текстовых анкет.

Таблица 13 – Структура таблицы Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| dictionaryId | integer | Идентификатор | V |  |
| categoryId | integer | Идентификатор категории |  | V |
| word | character varying(100) | Слово |  |  |

Таблица 14 включает ответы студентов на вопросы текстовых анкет.

Таблица 14 – Структура таблицы SurveyAnswer1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| surveyAnswer1Id | integer | Идентификатор | V |  |
| surveyQuestion1Id | integer | Идентификатор вопроса |  | V |
| teamStudentId | integer | Идентификатор связки студент-команда |  | V |
| answerText | character varying(2000) | Текст ответа |  |  |
| deliveryDate | date | Дата ответа |  |  |

Таблица 15 содержит информацию по вопросам текстовых анкет.

Таблица 15 – Структура таблицы SurveyQuestion1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| surveyQuestion1Id | integer | Идентификатор | V |  |
| surveyId | integer | Идентификатор анкеты |  | V |
| questionText | character varying(2000) | Текст вопроса |  |  |

Таблица 16 содержит информацию по вопросам балльных анкет.

Таблица 16 – Структура таблицы SurveyQuestion2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| surveyQuestion2Id | integer | Идентификатор | V |  |
| surveyed | integer | Идентификатор анкеты |  | V |
| questionText | character varying(2000) | Текст вопроса |  |  |
| maxScore | integer | Максимальный допустимый балл |  |  |
| minScore | integer | Минимальный допустимый балл |  |  |

Таблица 17 включает ответы студентов на вопросы балльных анкет.

Таблица 17 – Структура таблицы SurveyAnswer2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| surveyAnswer2Id | integer | Идентификатор | V |  |
| surveyQuestion2Id | integer | Идентификатор вопроса |  | V |
| teamStudentId | integer | Идентификатор связки студент-команда |  | V |
| scoreSurvey | integer | Балл ответа |  |  |
| deliveryDate | date | Дата ответа |  |  |

В таблице 18 приводятся данные по проектам.

Таблица 18 – Структура таблицы Project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| projectId | integer | Идентификатор | V |  |
| name | character varying(100) | Название |  |  |
| deadline | date | Дедлайн |  |  |
| dateSend | date | Дата |  |  |
| disciplineId | integer | Идентификатор дисциплины |  | V |

Таблица 19 содержит параметры команд.

Таблица 19 – Структура таблицы Team

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| teamId | integer | Идентификатор | V |  |

Продолжение таблицы 19

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| name | character varying(100) | Название |  |  |
| yearCreation | integer | Год создания |  |  |
| typeSemester | boolean | Тип семестра |  |  |
| scoreTeam | integer | Оценка |  |  |
| projectId | integer | Идентификатор проекта |  | V |
| comment | character varying(2000) | Комментарий |  |  |
| complete | boolean | Метка завершения работы |  |  |
| finalScore | boolean | Метка наличия финальной оценки |  |  |

Таблица 20 включает информацию по конфликтам.

Таблица 20 – Структура таблицы Conflict

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| conflictId | integer | Идентификатор | V |  |
| teamId | character varying(100) | Название |  | V |
| typeConflictId | integer | Идентификатор типа конфликта |  | V |
| broke | boolean | Метка распада команды |  |  |
| staged | integer | Идентификатор этапа |  | V |

В таблице 21 содержится информация по предметам.

Таблица 21 – Структура таблицы Discipline

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| disciplineId | integer | Идентификатор | V |  |
| name | character varying(100) | Название |  |  |

Таблица 22 содержит связку «студент-команда».

Таблица 22 – Структура таблицы TeamStudent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| teamStudentId | integer | Идентификатор | V |  |
| studentId | integer | Идентификатор студента |  | V |
| teamId | integer | Идентификатор команды |  | V |

Таблица 23 определяет параметры этапа.

Таблица 23– Структура таблицы Stage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| stageId | integer | Идентификатор | V |  |
| projectId | integer | Идентификатор проекта |  | V |
| name | character varying(100) | Название |  |  |
| deadline | date | Дедлайн |  |  |
| maxScore | integer | Максимальный балл за этап |  |  |

Таблица 24 содержит информацию по этапам проектирования.

Таблица 24 – Структура таблицы ScoreStage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| scoreStageId | integer | Идентификатор | V |  |
| statusStageId | integer | Идентификатор статуса этапа |  | V |
| stageId | integer | Идентификатор статуса |  | V |
| teamId | integer | Идентификатор команды |  | V |
| scoreStage | integer | Оценка этапа |  |  |
| dateSend | date | Дата отправки |  |  |

Таблица 25 включает список типов конфликтов.

Таблица 25 – Структура таблицы TypeConflict

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| typeConflictId | integer | Идентификатор | V |  |
| name | character varying(100) | Название |  |  |

Таблица 26 содержит типы статусов для описания прогресса проектирования.

Таблица 26 – Структура таблицы StatusStage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| statusStageId | integer | Идентификатор | V |  |
| teamId | character varying(100) | Название |  |  |

Таблица 27 включает данные по самооценке команд.

Таблица 27 – Структура таблицы ScoreTeammate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первичный ключ | Внешний ключ |
| scoreTeammateId | integer | Идентификатор | V |  |
| scoreTeammate | integer | Оценка сокомандника |  |  |
| teamId | integer | Идентификатор команды |  | V |
| studAssessingId | integer | Идентификатор оценивающего студента |  | V |
| studAnalyzedId | integer | Идентификатор оцениваемого студента |  | V |
| stageId | integer | Идентификатор статуса |  | V |

Таблица 28 содержит информацию, что студенту заданной команды назначена определенная анкета.

Таблица 28 – Структура таблицы TeamStudentSurvey

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | Описание | Первич-ный ключ | Внеш-ний ключ |
| teamStudentSurveyId | integer | Идентификатор | V |  |
| surveyId | integer | Идентификатор анкеты |  | V |

Продолжение таблицы 28

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| teamStudentId | integer | Идентификатор связки студент-команда |  | V |

Этап проектирования **физической модели** заключается в разворачивании разработанной ранее логической модели в конкретной системе управления баз данных [24], а именно PostgreSQL.

6 Разработанные основные алгоритмы задач

В данной главе описаны некоторые алгоритмы, реализованные в разрабатываемом модуле.

При переходе в окно «Анализ текстовых анкет» пользователь видит содержимое справочника слов – характеристик. После выбора необходимых характеристик, группы и анкеты и нажатие на кнопку «Таблица» выполняется подсчет количества выбранных характеристик в ответах студентов, алгоритм приведен на рисунке 3.

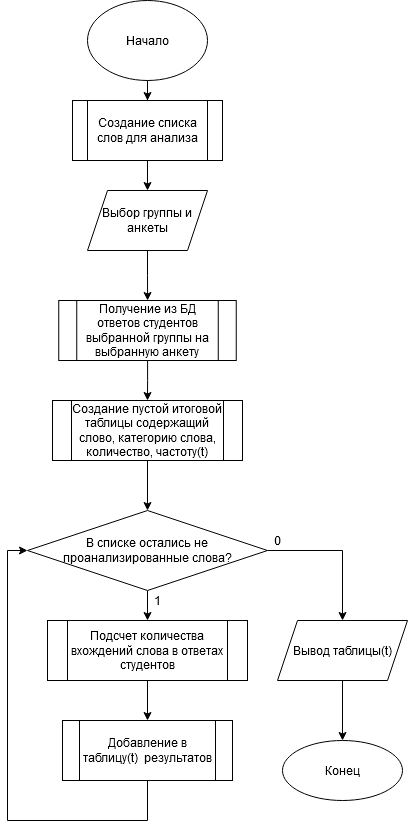


Рисунок 3 - Алгоритм подсчета количества выбранных характеристик в ответах студентов

По требованиям заказчика в форме «Анализ текстовых анкет» реализована возможность прорисовки лепестковой диаграммы с переменным числом лепестков по выбранным пользователем элементам. Алгоритм построения диаграммы приведен на рисунке 4.

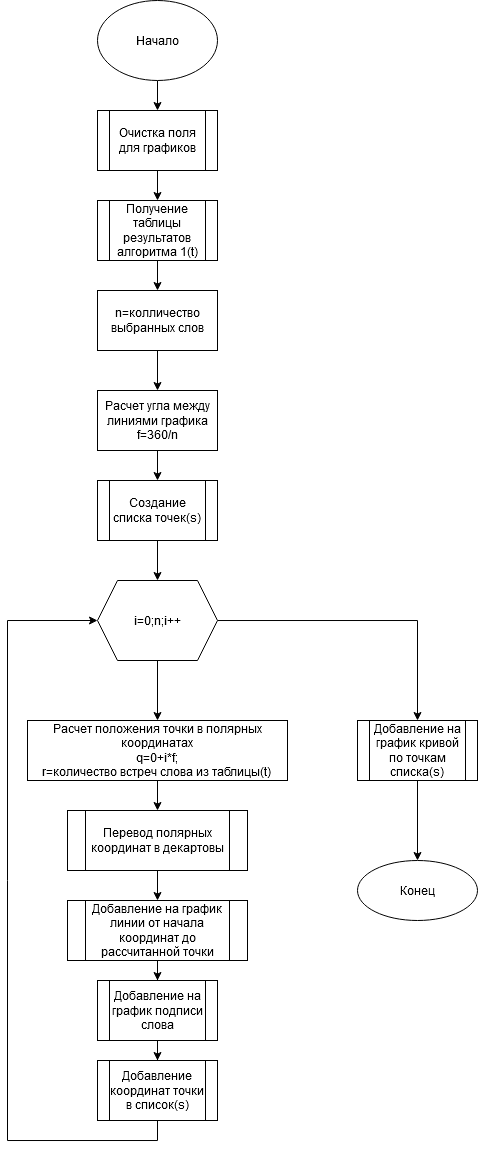


Рисунок 4 – Алгоритм построения лепестковой диаграммы

Алгоритм расчета среднего балла студента сокомандниками выбранной команды за выбранный этап работы, входящий в перечень необходимого функционала реализуемой системы, приведен на рисунке 5.

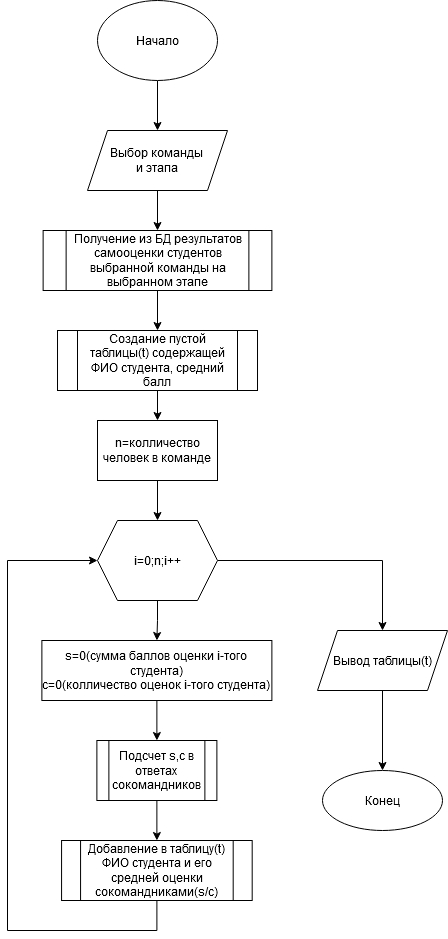


Рисунок 5 – Алгоритм расчета среднего балла студента сокомандниками

7 Руководство пользователя и контрольный пример

Документация на ПО включает в себя руководство пользователя, содержащее описание последовательности действий в системе для достижения заданного результата и скриншоты экрана для наглядности.

По условиям задачи в информационной системе реализована одна роль – преподавателя, которая включает в себя полномочия администратора.

Вид главного окна приведен на рисунке 6.

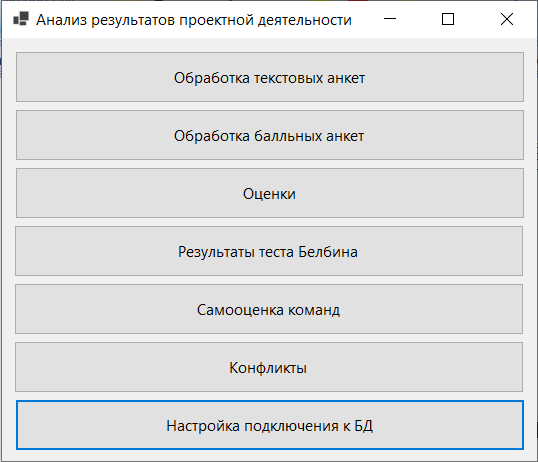


Рисунок 6 – Главное окно системы

При первом входе в систему следует нажать на кнопку «Настройка подключения к БД» главного окна и указать параметры для подключения. Вид отрывающегося диалогового окна приведен на рисунке 7.

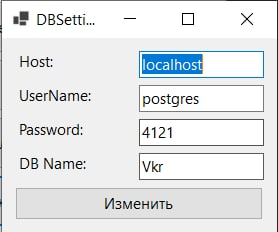


Рисунок 7 – Диалоговое окно «Настройка подключения к БД»

Если в последующем работа ведется с той же БД, повторная настройка не требуется.

Нажатие на кнопку «Обработка текстовых анкет» запускает модуль работы с тестовыми анкетами и отрывает окно, приведенное на рисунке 8.

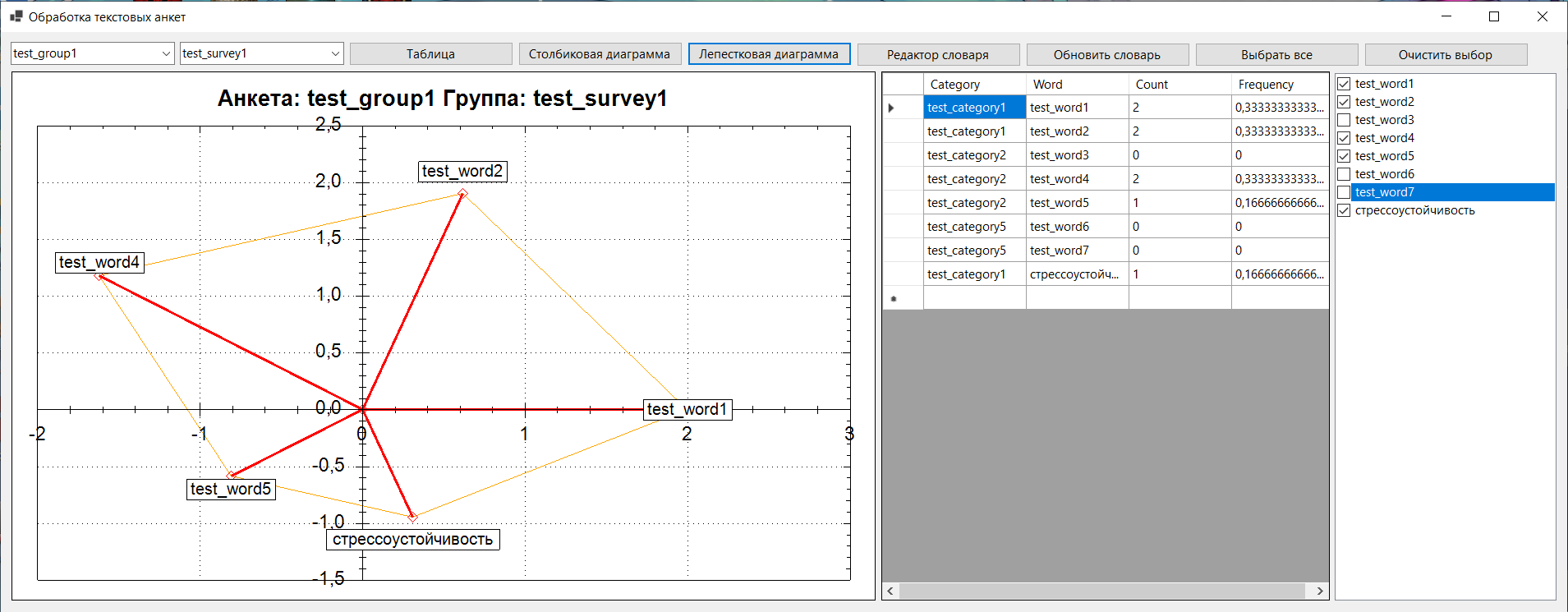


Рисунок 8 – Окно работы с текстовыми анкетами

Цель текстовой анкеты – зафиксировать мнение опрашиваемых студентов о наборе качеств, компетенций, навыков, необходимых для командной работы.

Входными данными для модуля являются: название группы, название анкеты, сохраненный список характеристик, выбранные характеристики, ответы студентов выбранной группы на обозначенную анкету.

Результат анализа текстовых анкет выводится на экран в виде списка характеристик с указанием количества отметок по каждому элементу и частоты выбора элемента, а именно, сколько студентов этой группы выбрало характеристику.

Окно содержит три области, дополняющие друг друга. Графическая область визуализирует результат: отображает столбиковую или лепестковую диаграмму по выбору пользователя. Причем, число элементов лепестковой диаграммы задается пользователем. В средней области размещено табличное представление списка. Справа расположен чек-бокс для включения характеристики в перечень анализируемых.

В верхней части окна размещены кнопки. Данные обновляются при нажатии на кнопку «Обновить словарь». Кнопки «Выбрать все» и «Очистить выбор» соответственно выставляют отметки или снимают их во всех элементах чек-бокса.

При клике на кнопку «Редактор словаря» открывается окно для правки списка, вид которого отражен на рисунке 9.

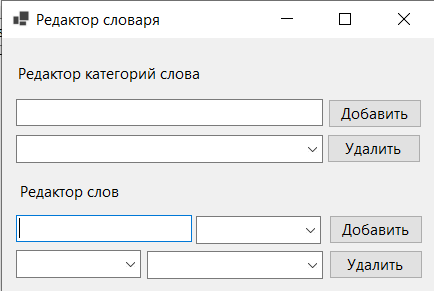


Рисунок 9 – Окно редактора справочника характеристик

Форма рис.9 позволяет создавать и удалять категорию характеристики, вносить в базу данных слово с выбранной категорией, а также выбрать категорию, слово из нее и удалить слово.

Кнопка «Обработка балльных анкет» главного окна открывает форму для отображения статистических данных по балльным анкетам с учетом фильтров, заданных пользователем, вид ее приведен на рисунке 10.

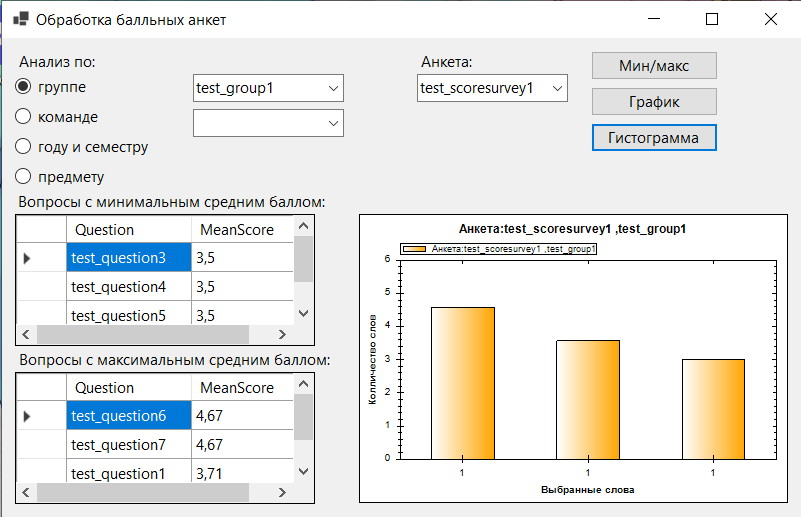


Рисунок 10 – Пример фильтров и результата анализа по балльным анкетам

Шаблоны бальных анкет создаются преподавателем в другом модуле.

Входными данными для работы модуля являются множество, по которому проводится анализ (по группе, команде, году и семестру, предмету), данные об этом множестве из выпадающих списков, выбранная анкета, ответы студентов на вопросы выбранной анкеты из БД.

Кнопка «Мин/макс» выводит в соответствующие поля по три вопроса выбранной анкеты с минимальным и максимальным средним баллом в этом множестве.

Кнопки «График» и «Гистограмма» выводят в области визуализации соответствующий вид графической интерпретации результата.

Кнопка «Оценки» главного окна открывает форму с данными успеваемости студентов из группы, выбранной пользователем, ее вид приведен на рисунке 11.

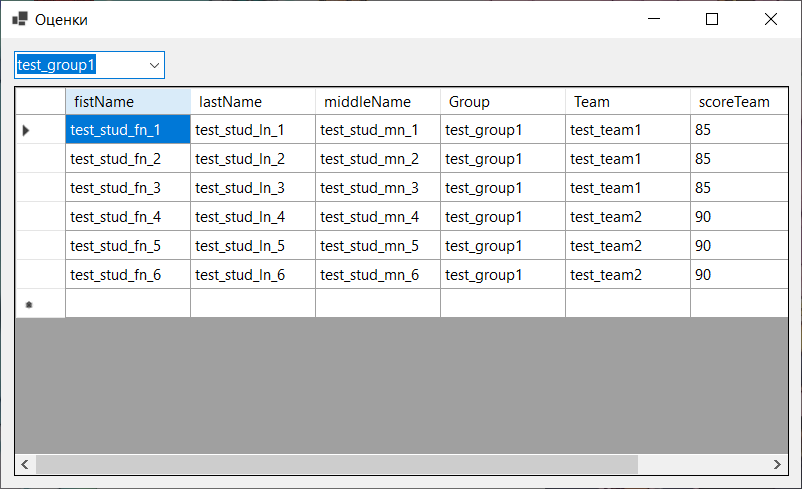


Рисунок 11 –Окно «Оценки»

При выборе группы из выпадающего списка, выводится таблица, содержащая фамилию, имя, отчество студентов обозначенной группы, команду, итоговую оценку команды, среднюю оценку за этапы у команды. Поскольку по требованиям заказчика преподаватель решает выставлять ли ему оценку за этап, в случае отсутствия оценок за все этапы у группы, таблица будет содержать значения «-1» в столбце средней оценки.

Нажатие на кнопку «Результаты теста Р. Белбина» позволяет вывести на экран баллы членов группы по выбору пользователя для каждой роли из теста Р. Белбина, а также фамилию, имя, отчество и команду этого студента. Пример формы приведен на рисунке 12.

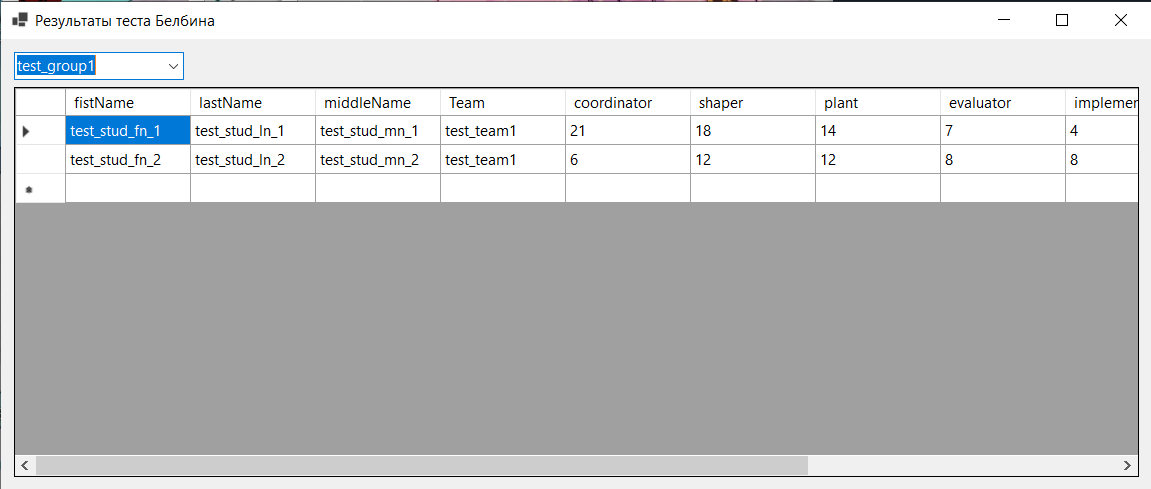


Рисунок 12 – Окно «Результаты теста Белбина»

Щелчок на кнопке «Самооценка команд» открывает окно, отображающее таблицу самооценок. Вид окна приведен на рисунке 13. Пользователь при этом выбирает команду и этап, для которого зафиксирована самооценка. Выводятся две таблицы, в первой отображены оценки студента своих сокомандников, во второй – средняя оценка сокомандниками студента.

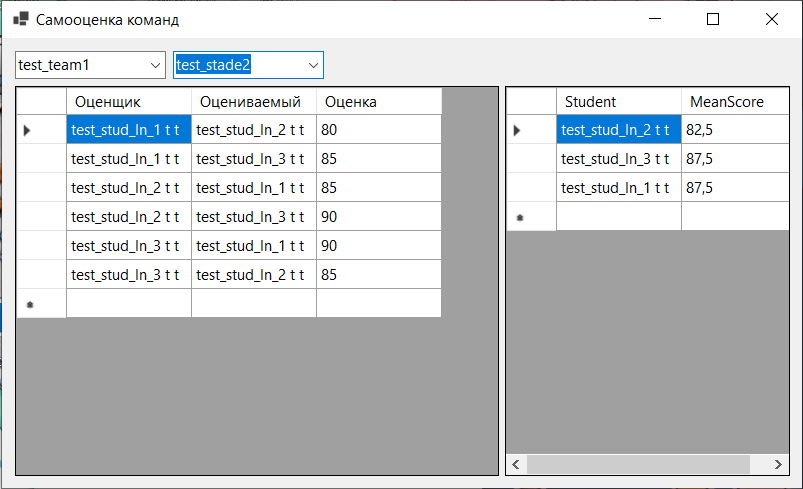


Рисунок 13 – Вид окна «Самооценка команд»

Нажатие на кнопку «Конфликты» главного окна позволяет пользователю вывести статистику распавшихся команд. Пример приведен на рисунке 14.

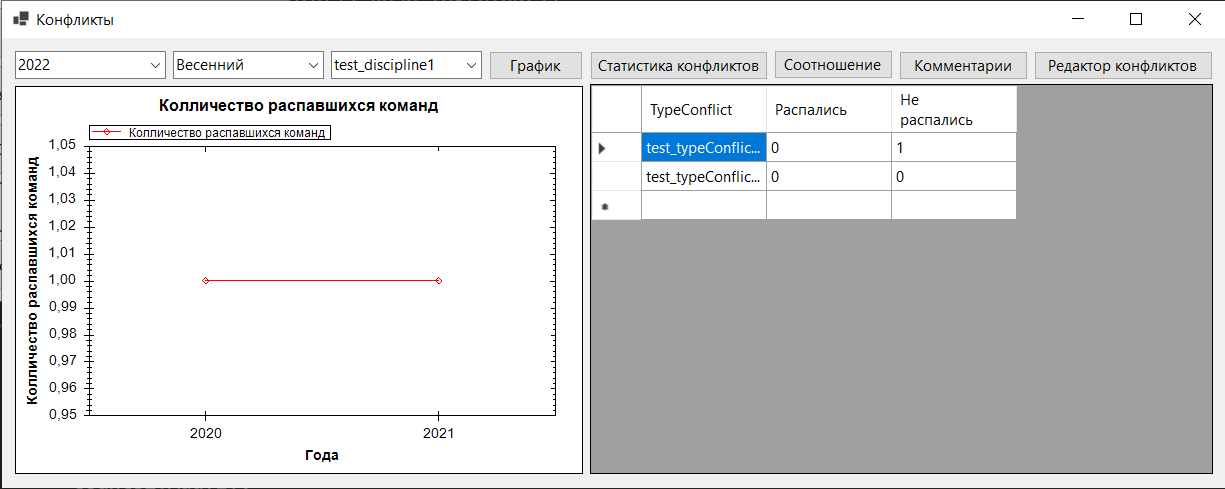


Рисунок 14 – Вид окна «Конфликты»

Пользователю следует выбрать год, семестр, название предмета.

Кнопка «График» окна «Конфликты» строит график зависимости количества распавшихся команд по годам.

Кнопка «Статистика конфликтов» выводит таблицу, где отображены тип конфликта, количество конфликтов этого типа, которые привели и не привели к распаду команд.

Кнопка «Соотношение» выводит количество распавшихся, нераспавшихся команд и общее число команд.

Кнопка «Комментарии» выводит таблицу «Команда - комментарий преподавателя».

При нажатии на кнопку «Редактор конфликтов» открывается окно, содержащее поля для редактирования информации о конфликтах в БД. Входными параметры формы являются: причина конфликта, команда, этап, на котором зафиксирован конфликт, и результат, привел конфликт к распаду команды или нет. Пример окна приведен на рисунке 15.

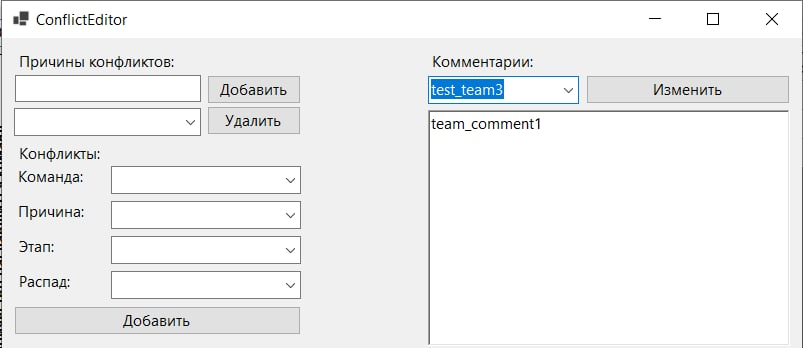


Рисунок 15 –Форма «Внесение данных о конфликтах»

В этом же окне можно добавлять причины конфликтов и изменять комментарии для выбранной команды.

Далее приведен контрольный пример на функции построения лепестковой диаграммы. В ходе анализа текстовых анкет нужно открыть окно «Обработка текстовых анкет» путем нажатия на кнопку «Обработка текстовых анкет» главного окна, выбрать группу и анкету из выпадающего списка, отметить три слова для анализа из столбца в правой части окна, нажать на кнопку «Лепестковая диаграмма». Результат работы продемонстрирован на рисунке 16.

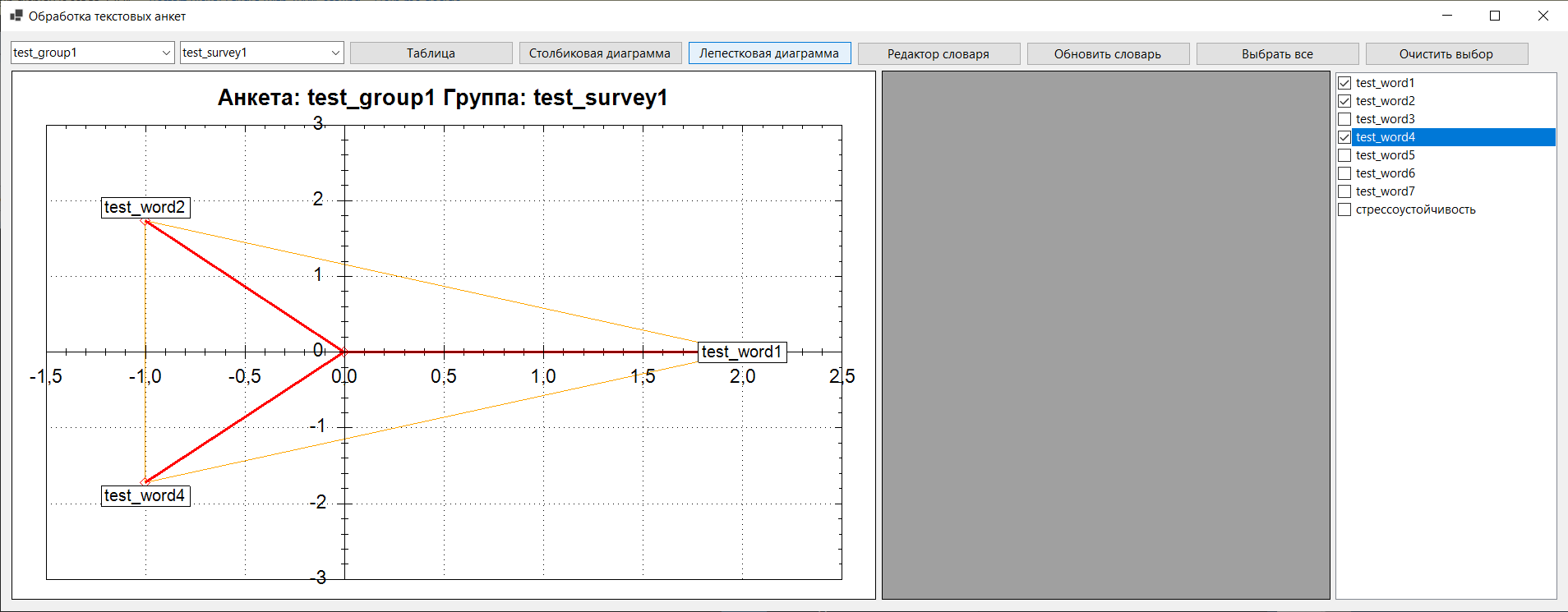


Рисунок 16 –Лепестковая диаграммы для трех слов

Для просмотра финальной и средней оценок студентов за этапы по группе нужно открыть окно «Оценки» нажатием на кнопку «Оценки» главного меню. После чего выбрать название группы из выпадающего списка. Результат представлен на рисунке 17.

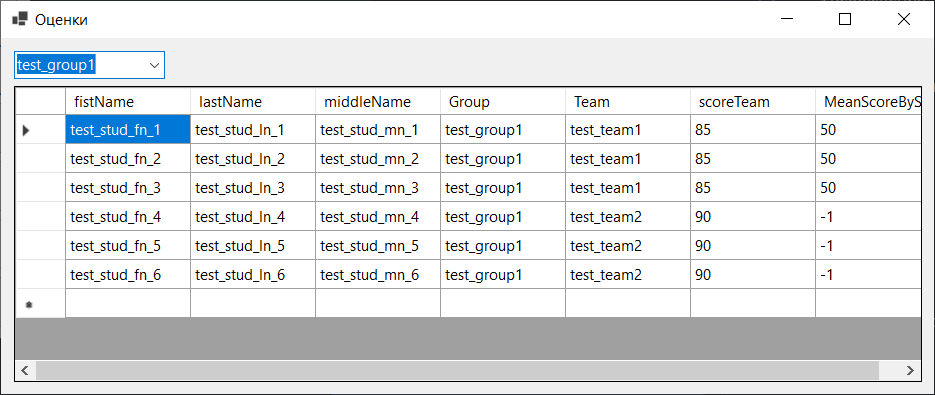


Рисунок 17 –Просмотр окна «Оценки»

8 Оценка пользовательского интерфейса

Проведём оценку пользовательского интерфейса для созданного продукта. Результат принятия или непринятия заказчиком нового ПО и время на его внедрение напрямую зависит от удовлетворенности пользователя дизайном, понятности ему интерфейса.

К характеристикам пользовательского интерфейса относят: доступность, отзывчивость, эффективность, привлекательность. Доступность поставили на первое место, поскольку интуитивно не понятный интерфейс даже при наличии подробного руководства пользователя вызывает отторжение у работающего с программой. Нагромождение управляющих элементов, высокая вложенность форм не экономят время пользователя. Отзывчивость означает быструю реакцию на действия в системе. Эффективность реализует возможность работающему с наименьшими усилиями выполнить интересующее его действие. Привлекательность – подбор цветовых решений, размер и расположение окон, наглядность графического представления данных.

На этапе предпроектного собеседования с заказчиком, изучения предметной области мною выявлены потребности заказчика.

Пользователи системы – преподавательский состав кафедры АСУ.

Для объективной оценки качества реализованного пользовательского интерфейса применим два метода: анкетирование на выявление уровня удовлетворенности пользователей и метод контрольного списка.

8.1 Измерение удовлетворенности посредством анкетирования

Оценка пользовательского интерфейса проводилась посредством двух анкет, шаблоны которых взяты из пособия [25]. Первая анкета позволяет быстро оценить удовлетворенность пользователей.

Каждый участник отмечает прилагательные из перечня, которые подходят к системе [25]. Результаты анкетирования представлены в таблице 27.

Таблица 29– Результаты анкетирования

|  |  |
| --- | --- |
| Прилагательное | Количество повторений |
| Устаревший | 2 |
| Эффективный | 3 |
| Ясный | 1 |
| Непоследовательный | 1 |
| Безопасный | 2 |
| Яркий | 1 |
| Стандартный | 2 |
| Управляемый | 4 |
| Интуитивный | 3 |
| Скучный | 1 |
| Последовательный | 2 |
| Запутанный | 1 |
| Простой | 3 |
| Удобный | 2 |
| Понятный | 2 |
| Гибкий | 2 |
| Непривлекательный | 1 |

Из таблицы 29 следует, что топ-3 лидеров прилагательных – управляемый, эффективный, интуитивный, простой. Положительные оценки превалируют.

Вторая анкета представляет собой таблицу с перечнем утверждений и 5-шаговой шкалой для оценки, причем «-2» означает, что утверждение совершенно не соответствует ожиданиям пользователя, «+2» совершенно соответствует. Результаты анкетирования представлены в таблице 28.

Таблица 30 – Результаты второй анкеты

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вопросы анкеты | Шкала оценок | | | | |
| -2 | -1 | 0 | +1 | +2 |
| Во время выполнения заданий я редко ошибался |  |  | 1 | 2 | 1 |
| Система работает достаточно быстро |  |  |  | 1 | 3 |
| Мне нравится внешний вид интерфейса |  | 1 | 1 | 2 |  |
| Начать работу было легко |  |  |  | 2 | 2 |
| Всякий раз, когда я ошибался, я с легкостью замечал и исправлял свою ошибку |  | 1 |  | 2 | 1 |
| Во время выполнения заданий я чувствовал себя вполне уверенно |  |  |  | 3 | 1 |
| В любой момент времени я понимал, что я должен сделать дальше |  | 1 |  | 2 | 1 |

Из таблицы 30 делаем вывод, что при тестировании системы работающие в ней редко ошибались, остались довольны скоростью отработки действий и логикой.

Итак, интерфейс в целом устраивает пользователей.

8.2 Контрольный список для оценки качества интерфейса

Проверим наличие грубых ошибок в программе методом контрольного списка.

Результаты оценки качества интерфейса с использованием контрольного списка представлены в таблице 29.

Таблица 31 – Контрольный список для оценки качества интерфейса

|  |  |  |
| --- | --- | --- |
| Элемент | Требование | Выполнение |
| Кнопки | Кнопки имеют один дизайн и выглядят одинаково | Да |
| Область действия клика на кнопку совпадает с ее видимым размером | Да |
| Текст на кнопках, запускающих действия, - инфинитивная форма глагола | Нет |
| Чекбоксы и радиокнопки | Внутри группы радиокнопок одна обязательно установлена по умолчанию. | Да |
| Чекбоксы и радиокнопки внутри своих групп расставлены по вертикали | Да |
| Если чекбоксов в группе больше 10, вводится дополнительный, выставляющий/снимающий все чекбоксы. | Да |
| Структура интерфейсных форм | В группах интерактивных элементов этих элементов не больше семи | Да |
| Если в окне есть свободное место, наиболее частотная терминационная кнопка больше остальных | Да |
| Кнопки находятся в секции, на которую они оказывают непосредственное воздействие | Да |
| Подписи к интерфейсным элементам размещены единообразно | Да |
| Текст в интерфейсе | В таблицах все столбцы с цифрами выравниваются по левому краю | Да |
| Подписи к интерфейсным элементам начинаются с прописной буквы | Да |

Оценка интерфейса методом контрольного списка показала, что грубые ошибки в программе не выявлены, в целом пользователи им удовлетворены.

9 Оценка трудозатрат на разработку проекта

Оценка трудозатрат на разработку ВКР бакалавра по направлению 09.03.01 производится по методике COCOMO II, описанной в методическом пособии [26], позволяет оценить трудоемкость и время разработки программного продукта. Качественно оценить трудозатраты на разработку, значит: избежать сдвига сроков сдачи продукта, необоснованных переработок команды, превышения бюджета, сохранить лояльность заказчика.

В модели COCOMO II используется формула регрессии с параметрами, определяемыми на основе отраслевых данных и характеристик проекта.

Оценка проекта разделяется на две стадии:

- предварительная оценка на начальной фазе;

- детальная оценка после проработки архитектуры.

9.1 Расчет параметров для оценки трудоемкости разработки

Рассчитаю предварительно трудоемкость проекта (измеряется в чел.\*мес.) по формуле:

, (1)

, (2)

где A=2,94;

SIZE — размер программного продукта, измеряется в тысячах строк исходного кода;

EMi — множители трудоемкости. Значение n=7 (для предварительной оценки) и n=17 (для детальной оценки);

B=0,91;

SFj — факторы масштаба.

Для определения параметра SIZE необходимо провести оценку размера программного продукта методом функциональных точек. В моем случае, размер программного продукта составил порядка 5 тысяч строк исходного кода.

Определено пять факторов масштаба SF в модели COCOMO II, которые представлены в таблице 30.

Обозначения факторов масштаба:

- PREC – опыт аналогичной разработки проекта;

- FLEX – гибкость процесса разработки;

- RESL – архитектура и уровень рисков;

- TEAM – сработанность команды;

- PMAT – зрелость процессов.

Таблица 32 – Значения факторов масштаба SF в модели COCOMO II

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Фактор масштаба | Оценка уровня фактора | | | | | |
| Very Low | Low | Nominal | High | Very High | Extra High |
| PREC | 6,20 | 4,96 | 3,72 | 2,48 | 1,24 | 0,00 |
| FLEX | 5,07 | 4,05 | 3,04 | 2,03 | 1,01 | 0,00 |
| RESL | 7,07 | 5,65 | 4,24 | 2,83 | 1,41 | 0,00 |
| TEAM | 5.48 | 4,38 | 3,29 | 2,19 | 1,10 | 0,00 |
| PMAT | 7,80 | 6,24 | 4,68 | 3,12 | 1,56 | 0,00 |

Выбранные для моего проекта значения факторов масштаба:

- PREC – 3,72 – есть опыт работы с платформой;

- FLEX – 3,04 – присутствуют некритические ограничения;

- RESL – 4,24 – риски проанализированы и сведены к минимуму;

- TEAM –1,10 – проект разрабатывается одним человеком;

- PMAT – 3,12 – процессы работают правильно.

Расчет коэффициента E осуществляется по формуле (2):

9.2 Расчет предварительной оценки трудоемкости

Расчет предварительной оценки трудоемкости осуществляется по формуле\_(1). Определим значения следующих множителей трудоемкости EMi для моей работы:

- PERS — уровень квалификации программистов,

- RCPX — степень сложности и надёжности проекта,

- RUSE — повторное использование продукта,

- PDIF —сложность платформы разработки,

- PREX — опыт работы программистов,

- FCIL — используемые инструменты и оборудование,

- SCED — требования к сжатию расписания.

Значения множителей трудоемкости приведены в таблице 31.

Таблица 33 – Значения множителей трудоемкости

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Фактор масштаба | Оценка множителя трудоёмкости | | | | | | |
| Extra Low | Very Low | Low | Nominal | High | Very High | Extra High |
| PERS | 2,12 | 1,62 | 1,26 | 1,00 | 0,83 | 0,63 | 0,50 |
| RCPX | 0,49 | 0,60 | 0,83 | 1,00 | 1,33 | 1,91 | 2,72 |
| RUSE | n/a | n/a | 0,95 | 1,00 | 1,07 | 1,15 | 1,24 |
| PDIF | n/a | n/a | 0,87 | 1,00 | 1,29 | 1,81 | 2,61 |
| PREX | 1,59 | 1,33 | 1,22 | 1,00 | 0,87 | 0,74 | 0,62 |
| FCIL | 1,43 | 1,30 | 1,10 | 1,00 | 0,87 | 0,73 | 0,62 |
| SCED | n/a | 1,43 | 1,14 | 1,00 | 1,00 | 1,00 | n/a |

Значения множителей трудоемкости для моего проекта:

- PERS — 1,00;

- RCPX — 1,00;

- RUSE — 1,15;

- PDIF — 1,29;

- PREX — 1,22;

- FCIL — 0,87;

- SCED — 1,00.

Расчет предварительной оценки трудоемкости:

чел\*мес

9.3 Расчет детальной оценки трудоемкости разработки

Детальную оценку трудоемкости осуществил после проработки архитектуры. Для расчета использую следующие множители трудоемкости:

- ACAP – способности аналитика;

- AEXP – опыт разработчиков в создании приложений;

- PCAP – способности программиста;

- PCON – длительность ведения разработки командой;

- PEXP – опыт работы с платформой;

- LTEX – уровень владения используемым языком программирования;

- RELY – требуемый уровень надёжности приложения;

- DATA – размер базы данных;

- CPLX – сложность программы;

- RUSE – возможность многократного использования проекта;

- DOCU – соответствие документации потребностям жизненного цикла;

- TIME – ограничения на время;

- STOR – ограничения на память;

- PVOL – степень изменяемости платформы;

- TOOL – использование инструментов и программных средств;

- SITE – удаленная разработка;

- SCED – необходимость исполнения графика.

Значения множителей приведены в таблице 32.

Таблица 34 – Значения множителей трудоемкости

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Фактор масштаба | Оценка уровня множителя трудоёмкости | | | | | |
| Very Low | Low | Nominal | High | Very High | Extra High |
| ACAP | 1,42 | 1,29 | 1,00 | 0,85 | 0,71 | n/a |
| AEXP | 1,22 | 1,10 | 1,00 | 0,88 | 0,81 | n/a |
| PCAP | 1,34 | 1,15 | 1,00 | 0,88 | 0,76 | n/a |
| PCON | 1,29 | 1,12 | 1,00 | 0,90 | 0,81 | n/a |
| PEXP | 1,19 | 1,09 | 1,00 | 0,91 | 0,85 | n/a |
| LTEX | 1,20 | 1,09 | 1,00 | 0,91 | 0,84 | n/a |
| RELY | 0,84 | 1,09 | 1,00 | 0,91 | 0,84 | n/a |
| DATA | n/a | 0,23 | 1,00 | 1,14 | 1,28 | n/a |
| CPLX | 0,73 | 0,87 | 1,00 | 1,17 | 1,34 | 1,74 |
| RUSE | n/a | 0,95 | 1,00 | 1,07 | 1,15 | 1,24 |
| DOCU | 0,81 | 0,91 | 1,00 | 1,11 | 1,23 | n/a |
| TIME | n/a | n/a | 1,00 | 1,11 | 1,29 | 1,63 |
| STOR | n/a | n/a | 1,00 | 1,05 | 1,17 | 1,46 |
| PVOL | n/a | 0,87 | 1,00 | 1,15 | 1,30 | n/a |
| TOOL | 1,17 | 1,09 | 1,00 | 0,90 | 0,78 | n/a |
| SITE | 1,22 | 1,09 | 1,00 | 0,93 | 0,86 | 0,80 |
| SCED | 1,43 | 1,14 | 1,00 | 1,00 | 1,00 | n/a |

Значения множителей трудоемкости для разрабатываемого проекта:

- ACAP = 0,85;

- AEXP = 1,00;

- PCAP = 1,00;

- PCON = 0,90;

- PEXP = 0,91;

- LTEX = 1,00;

- RELY = 0,91;

- DATA = 1,14;

- CPLX = 1,17;

- RUSE = 1,15;

- DOCU = 1,23;

- TIME = 1,29;

- STOR = n/a;

- PVOL = 0,87;

- TOOL = 0,90;

- SITE = 0,86;

- SCED = 1,00.

Рассчитаем детальную оценку трудоемкости по формуле (1):

9.4 Расчет длительности работы

Для расчета длительности разработки воспользуемся формулой:

(3)

где C = 3,67;

D = 0,28;

PMNS – трудоёмкость проекта без учета множителя SCED.

мес

Полученное значение длительности разработки укладывается в установленные временные рамки проекта. Стоит также отметить, что детальная оценка трудоемкости оказалась в несколько раз меньше предварительной.

ЗАКЛЮЧЕНИЕ

В соответствии с темой выпускной квалификационной работы бакалавра была разработана информационная система для организации командной работы студентов старших курсов, позволяющая аккумулировать, анализировать и отражать данные по результатам проектной деятельности.

Произведено исследование командного подхода в обучении в ходе проектной деятельности студентов высших учебных заведений.

По итогам изучения требований заказчика сформулированы задачи проекта, рассмотрены возможные существующие аналоги, принято решение о проектировании информационной системы.

Для проектирования информационной системы были выбраны следующие программные средства: клиент-серверная СУБД – PostgreSQL, веб-приложение pgAdmin4, язык программирования С#, среда разработки – Microsoft Visual Studio 2019, библиотеки Npgsql, ZedGraph, менеджер пакетов NuGet.

Определён перечень необходимой входной и выходной документации в виде таблиц.

В поставленные сроки произведено описание структуры таблиц базы данных, реализованы и отлажены разработанные алгоритмы.

Было разработано руководство пользователя и приведен контрольный пример.

Произведены оценка времени и трудозатрат, которые необходимы на реализацию проекта, и оценка пользовательского интерфейса с помощью анкетирования и контрольного списка.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Компетенция командность в индивидуальном плане развития: шпаргалка для HR и не только / Event Center, 2020. URL: <https://hr-solution.org/kompetencija-komandnost-v-individualnom-plane-razvitija-shpargalka-dlja-hr-i-ne-tolko/> (дата обращения: 08.02.2022).

2. Работа в команде (Координация и кооперация) / Россия - страна возможностей, 2018. URL: <https://rsv.ru/other-skills/8/> (дата обращения: 10.02.2022).

3. Личностные компетенции молодого специалиста важнее наличия опыта работы / Исследовательский центр портала Superjob.ru, 2020. URL: <https://www.superjob.ru/research/articles/112422/lichnostnye-kompetencii-molodogo-specialista-vazhnee-nalichiya-opyta-raboty/> (дата обращения: 10.02.2022).

4. Зорина М.А. Командная работа на занятии: от теории к практике / М.А. Зорина // Экспериментальные и теоретические исследования в современной науке: сб. ст. по матер. XXXIV-XXXV междунар. науч.-практ. конф. № 4-5(32). – Новосибирск: СибАК, 2019. – С. 42-46. URL: https://sibac.info/conf/modernscience/xxxiv/134005 (дата обращения: 15.02.2022).

5. Командные компетенции. Диагностика и развитие команды / System Approach, 2021. URL: <http://system-approach.ru/team_competences/> (дата обращения: 15.02.2022).

6. Карякин А.М. Командная работа. Основы теории и практики: учебное пособие / А.М. Карякин, В.В.Великороссов // – М.: Русайнс, 2021. – 188 с.

7. Демушкина О.С. Личностные характеристики сотрудников команды проекта как фактор эффективности проектной деятельности в сфере информационных технологий / О.С. Демущкина // Азимут научных исследований: педагогика и психология. – 2018. –Т.7. – № 1(2). – С. 269-272.

|  |
| --- |
| 8. Лыгина Н.И., Лауферман О.В. Качество учебных материалов: проектирование, использование, оценка // Мир науки. Педагогика и психология, 2020 №1, URL: [https://mir-nauki.com/PDF/64PDMN120.pdf /](https://mir-nauki.com/PDF/64PDMN120.pdf%20/) (дата обращения: 15.02.2022).  9. Лауферман О.В. Разработка программного продукта: профессиональные стандарты, жизненный цикл, командная работа: учебное пособие / О.В. Лауферман, Н.И. Лыгина. – Новосибирск: Изд-во НГТУ, 2019. – 75 с.  10. Алексеева, А. И. Командная работа студентов вуза при изучении дисциплин «Технология растениеводства» и «Основы агрономии» / А. И. Алексеева. — Текст : непосредственный // Молодой ученый. — 2017. — № 14 (148). — С. 585-587. — URL: https://moluch.ru/archive/148/41842/ (дата обращения: 15.02.2022).  11. Дьяченко, В. К. Сотрудничество в обучении. — М.: Просвещение, 2011.С. 194.  12. Джуринский, А. Н. Изменения парадигмы сравнительной педагогики / А. Н. Джуринский // Педагогика: журнал. — 2014.— № 2.— С. 107–113.  13. Атапина, И. М. Использование информационно-коммуникационных технологий в работе с одарёнными детьми / И. М. Атапина // Образование в современной школе: журнал. — 2014.— № 1.— С. 59–60.  14. Белбин Р. Типы ролей в командах менеджеров. М.: HIPPO, 2003 – 232\_с.  15. Moodle, как платформа организации eLearning и дистанционного обучения / Дьяченко А., 2012. URL: <https://habr.com/ru/post/139629/?ysclid=l1zylryfo5>/ (дата обращения: 01.03.2022).  16. Обзор 4 бесплатных систем электронного обучения: Moodle, Atutor, Ilias, Diskurs/ Системы Дистанционного Обучения, 2021. URL: <https://lmslist.ru/free-sdo/?ysclid=l1zyti73qc>/ (дата обращения: 01.03.2022).  17. Google Classroom: обзор возможностей/ iSpring, 2021. URL: <https://www.ispring.ru/elearning-insights/platforma-onlain-obucheniya/google-classroom?ysclid=l236y8yb87> /(дата обращения: 01.03.2022).  18. 52 системы управления проектами для командной работы в разных сферах/ Северная Е., 2021. URL: <https://habr.com/ru/company/yougile/blog/545614/?ysclid=l1zy9azt8e> / (дата обращения: 01.03.2022).  19. Как создать таблицу в PostgreSQL с помощью pgAdmin 4 / Заметки IT специалиста, 2020. URL: [https://info-comp.ru/create-table-in-postgresql /](https://info-comp.ru/create-table-in-postgresql%20/) (дата обращения: 17.03.2022).  20. Руководство по программированию в Windows Forms / Сайт о программировании, 2020. URL: <https://metanit.com/sharp/windowsforms/> (дата обращения: 17.03.2022).  21. Шпаргалка по ZedGraph / Софт, исходники и фото, 2019. URL: <https://jenyay.net/Programming/ZedGraph> /(дата обращения: 17.03.2022).  22. Осипов Д. Л. Технологии проектирования баз данных. – М.: ДМК Пресс, 2019. – 498 с.  23. Тарасов С. В. СУБД для программиста. Базы данных изнутри. М.: СОЛОН-Пресс, 2015 — 320 с.  24. 8 Характеристик удачного пользовательского интерфейса / Родненко В., 2009. URL: <https://habr.com/ru/post/58023/> (дата обращения: 20.04.2022).  25. Терещенко П. В. Пользовательские интерфейсы информационных систем: Методические указания к выполнению расчетно-графического задания по курсу "Диалоговые средства информационных систем" для студентов, обучающихся по направлению 230100 "Информатика и вычислительная техника" / П.В. Терещенко, Р.Г. Шахмаметов. - Новосибирск: НГТУ, 2010. - 38 с.  26. Астапчук В.А. Оценка трудозатрат на разработку информационных систем (методика COCOMO II): Методические указания к выполнению экономического раздела выпускной квалификационной работы бакалавра по направлению 09.03.01 - Информатика и вычислительная техника / В.А. Астапчук - Новосибирск: НГТУ, 2015. - 14 с. |

ПРИЛОЖЕНИЕ А

Фрагмент программного кода анализа текстовых анкет

**TextSurveyAnalysis.cs**

using Npgsql;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using ZedGraph;

namespace vkr

{

public partial class TextSurveyAnalysis : Form

{

Settings.DbSettings dbsettings = new Settings.DbSettings();

public TextSurveyAnalysis()

{

InitializeComponent();

FillCBGroup();

FillCBSurvey();

FillCheckedListBoxTSADictionary();

}

private void FillCBGroup()

{

string connString = dbsettings.GetConnString();

NpgsqlConnection nc = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm = new NpgsqlCommand();

comm.Connection = nc;

comm.CommandType = CommandType.Text;

comm.CommandText = "SELECT \"name\" FROM \"Group\" ;";

NpgsqlDataReader dr = comm.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

for (int i = 0; i < dt.Rows.Count; i++)

{

comboBoxGroup.Items.Add(dt.Rows[i]["name"]);

}

comm.Dispose();

nc.Close();

}

private void FillCBSurvey()

{

string connString = dbsettings.GetConnString();

NpgsqlConnection nc = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm = new NpgsqlCommand();

comm.Connection = nc;

comm.CommandType = CommandType.Text;

comm.CommandText = "SELECT \"name\" FROM \"Survey\" ;";

NpgsqlDataReader dr = comm.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

for (int i = 0; i < dt.Rows.Count; i++)

{

comboBoxSurvey.Items.Add(dt.Rows[i]["name"]);

}

comm.Dispose();

nc.Close();

}

private void FillCheckedListBoxTSADictionary()

{

checkedListBoxTSADictionary.Items.Clear();

WordsListEditor temp = new WordsListEditor();

DataTable dt = temp.DictionaryTable();

for (int i=0; i < dt.Rows.Count; i++)

{

checkedListBoxTSADictionary.Items.Add(dt.Rows[i][0]);

}

}

private void TSABTable\_Click(object sender, EventArgs e)

{

dataGridViewTSA.DataSource = WordsCounterRezult();

}

private DataTable WordsCounterRezult()

{

WordsCounter SelectedWordsCounter = ListWordsToFind();

SelectedWordsCounter.FindAndCount(AnswerString());

return SelectedWordsCounter.ToDataTable(TSATable().Rows.Count);

}

private WordsCounter ListWordsToFind()

{

WordsListEditor temp = new WordsListEditor();

DataTable dt = temp.DictionaryTable();

WordsCounter rezult = new WordsCounter();

for (int i = 0; i < dt.Rows.Count; i++)

{

if (checkedListBoxTSADictionary.CheckedItems.Contains(dt.Rows[i][0])){

WordCounter word = new WordCounter();

word.EditWord(dt.Rows[i][1].ToString(), dt.Rows[i][0].ToString());

rezult.AddWord(word);

}

}

return rezult;

}

private string AnswerString()

{

string rezult = "";

DataTable dt = TSATable();

for (int i = 0; i < dt.Rows.Count; i++)

{

rezult += dt.Rows[i][0];

rezult += ";";

}

return rezult.ToLower();

}

private DataTable TSATable()

{

string connString = dbsettings.GetConnString();

NpgsqlConnection nc = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm = new NpgsqlCommand();

comm.Connection = nc;

comm.CommandType = CommandType.Text;

comm.CommandText = "SELECT \"SurveyAnswer1\".\"answerText\", \"SurveyQuestion1\".\"questionText\",\"Student\".\"fistName\"," +

"\"Student\".\"lastName\" FROM \"TeamStudent\", \"Student\", \"Group\",\"SurveyAnswer1\" ,\"SurveyQuestion1\",\"Survey\" WHERE" +

" \"SurveyAnswer1\".\"teamStudentId\" = \"TeamStudent\".\"teamStudentId\" and \"TeamStudent\".\"studentId\" = \"Student\".\"studentId\"" +

" and \"Student\".\"groupId\" = \"Group\".\"groupId\" and \"SurveyAnswer1\".\"surveyQuestion1Id\" = \"SurveyQuestion1\".\"surveyQuestion1Id\"" +

" and \"SurveyQuestion1\".\"surveyId\" = \"Survey\".\"surveyId\"" +

" and \"Group\".\"name\" = '" + comboBoxGroup.SelectedItem + "' and \"Survey\".\"name\" = '" + comboBoxSurvey.SelectedItem + "';";

NpgsqlDataReader dr = comm.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

comm.Dispose();

nc.Close();

return dt;

}

private void TASBGist\_Click(object sender, EventArgs e)

{

DataTable dt = WordsCounterRezult();

zedGraphControlTSA.GraphPane.CurveList.Clear();

zedGraphControlTSA.GraphPane.GraphObjList.Clear();

zedGraphControlTSA.GraphPane.Title.Text = "";

zedGraphControlTSA.GraphPane.XAxis.Title.Text = "";

zedGraphControlTSA.GraphPane.YAxis.Title.Text = "";

zedGraphControlTSA.GraphPane.Y2Axis.Title.Text = "";

zedGraphControlTSA.GraphPane.XAxis.Type = AxisType.Linear;

zedGraphControlTSA.GraphPane.XAxis.Scale.TextLabels = null;

zedGraphControlTSA.RestoreScale(zedGraphControlTSA.GraphPane);

List<double> numbers = new List<double>();

List<string> words = new List<string>();

for (int i = 0; i < dt.Rows.Count; i++)

{

numbers.Add(Convert.ToDouble(dt.Rows[i][2]));

words.Add(dt.Rows[i][1].ToString());

}

string name="Анкета: " + comboBoxGroup.SelectedItem + " Группа: " + comboBoxSurvey.SelectedItem;

zedGraphControlTSA.GraphPane.Title.Text = name;

BarItem graph = zedGraphControlTSA.GraphPane.AddBar(name, null, numbers.ToArray(),Color.Orange);

zedGraphControlTSA.GraphPane.XAxis.Type = AxisType.Text;

zedGraphControlTSA.GraphPane.XAxis.Scale.TextLabels = words.ToArray();

zedGraphControlTSA.GraphPane.AxisChange();

zedGraphControlTSA.GraphPane.XAxis.Title.Text = "Выбранные слова";

zedGraphControlTSA.GraphPane.YAxis.Title.Text = "Колличество слов";

zedGraphControlTSA.Invalidate();

}

private void buttonTSASelectNoth\_Click(object sender, EventArgs e)

{

for (int i=0; i < checkedListBoxTSADictionary.Items.Count; i++)

{

checkedListBoxTSADictionary.SetItemChecked(i, false);

}

}

private void buttonTSASelectAll\_Click(object sender, EventArgs e)

{

for (int i = 0; i < checkedListBoxTSADictionary.Items.Count; i++)

{

checkedListBoxTSADictionary.SetItemChecked(i, true);

}

}

private void buttonTSAUpdateDictionary\_Click(object sender, EventArgs e)

{

FillCheckedListBoxTSADictionary();

}

private void buttonTSAPolarGraph\_Click(object sender, EventArgs e)

{

DataTable dt = WordsCounterRezult();

zedGraphControlTSA.GraphPane.CurveList.Clear();

zedGraphControlTSA.GraphPane.GraphObjList.Clear();

zedGraphControlTSA.GraphPane.Title.Text = "";

zedGraphControlTSA.GraphPane.XAxis.Title.Text = "";

zedGraphControlTSA.GraphPane.YAxis.Title.Text = "";

zedGraphControlTSA.GraphPane.Y2Axis.Title.Text = "";

zedGraphControlTSA.GraphPane.XAxis.Type = AxisType.Linear;

zedGraphControlTSA.GraphPane.XAxis.Scale.TextLabels = null;

zedGraphControlTSA.RestoreScale(zedGraphControlTSA.GraphPane);

zedGraphControlTSA.GraphPane.XAxis.Cross = 0.0;

zedGraphControlTSA.GraphPane.YAxis.Cross = 0.0;

string name = "Анкета: " + comboBoxGroup.SelectedItem + " Группа: " + comboBoxSurvey.SelectedItem;

zedGraphControlTSA.GraphPane.Title.Text = name;

zedGraphControlTSA.GraphPane.XAxis.MajorGrid.IsVisible = true;

zedGraphControlTSA.GraphPane.YAxis.MajorGrid.IsVisible = true;

zedGraphControlTSA.GraphPane.YAxis.MajorGrid.IsZeroLine = false;

zedGraphControlTSA.GraphPane.XAxis.Title.IsVisible = false;

zedGraphControlTSA.GraphPane.YAxis.Title.IsVisible = false;

//угол между прямыми

double f;

if (dt.Rows.Count != 0)

{

f = 360 / dt.Rows.Count;

}

else

{

f = 360;

}

List<double> listX = new List<double>();

List<double> listY = new List<double>();

for (int i = 0; i < dt.Rows.Count; i++)

{

double x = decX(0 + i \* f,Convert.ToDouble( dt.Rows[i][2]));

double y = decY(0 + i \* f, Convert.ToDouble(dt.Rows[i][2]));

PointPairList list = new PointPairList();

list.Add(0, 0);

list.Add(x, y);

if (Convert.ToDouble(dt.Rows[i][2]) != 0)

{

listX.Add(x);

listY.Add(y);

}

LineItem myCurve = zedGraphControlTSA.GraphPane.AddCurve(/\*dt.Rows[i][1].ToString()\*/"", list, Color.Red, SymbolType.Diamond);

myCurve.Line.Width = 3;

TextObj text = new TextObj(dt.Rows[i][1].ToString(), decX(0 + i \* f, Convert.ToDouble(dt.Rows[i][2])),decY(0 + i \* f, Convert.ToDouble(dt.Rows[i][2])+0.2));

zedGraphControlTSA.GraphPane.GraphObjList.Add(text);

}

if (listX.Count != 0)

{

listX.Add(listX[0]);

listY.Add(listY[0]);

}

LineItem myCurvef = zedGraphControlTSA.GraphPane.AddCurve("", listX.ToArray(), listY.ToArray(), Color.Orange, SymbolType.None);

zedGraphControlTSA.AxisChange();

zedGraphControlTSA.Invalidate();

}

private double decX(double f, double r)

{

return r \* Math.Cos(f \* Math.PI / 180);

}

private double decY(double f, double r)

{

return r \* Math.Sin(f \* Math.PI / 180);

}

private void buttonTSAStartWLE\_Click(object sender, EventArgs e)

{

WordsListEditor WLE = new WordsListEditor();

WLE.Show();

}

}

}

ПРИЛОЖЕНИЕ Б

Фрагмент программного кода формы по конфликтам

**Conflicts.cs**

using Npgsql;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using ZedGraph;

namespace vkr

{

public partial class Conflicts : Form

{

Settings.DbSettings dbsettings = new Settings.DbSettings();

public Conflicts()

{

InitializeComponent();

FillCBYear();

FillCBSemester();

FillCBDiscipline();

}

private void FillCBYear()

{

string connString = dbsettings.GetConnString();

NpgsqlConnection nc = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm = new NpgsqlCommand();

comm.Connection = nc;

comm.CommandType = CommandType.Text;

comm.CommandText = "SELECT \"yearCreation\" FROM \"Team\" ;";

NpgsqlDataReader dr = comm.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

for (int i = 0; i < dt.Rows.Count; i++)

{

if (!comboBoxCYear.Items.Contains(dt.Rows[i]["yearCreation"]))

{

comboBoxCYear.Items.Add(dt.Rows[i]["yearCreation"]);

}

}

comm.Dispose();

nc.Close();

}

private void FillCBSemester()

{

comboBoxCSemester.Items.Add("Весенний");

comboBoxCSemester.Items.Add("Осенний");

}

private void FillCBDiscipline()

{

string connString = dbsettings.GetConnString();

NpgsqlConnection nc = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm = new NpgsqlCommand();

comm.Connection = nc;

comm.CommandType = CommandType.Text;

comm.CommandText = "SELECT \"name\" FROM \"Discipline\" ;";

NpgsqlDataReader dr = comm.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

for (int i = 0; i < dt.Rows.Count; i++)

{

comboBoxCDiscipline.Items.Add(dt.Rows[i]["name"]);

}

comm.Dispose();

nc.Close();

}

private void buttonCConflictEditor\_Click(object sender, EventArgs e)

{

ConflictEditor newForm = new ConflictEditor();

newForm.Show();

}

struct DataForGraf

{

public int year;

public int count;

public DataForGraf(int y)

{

this.year = y;

this.count = 0;

}

public void addC()

{

this.count += 1;

}

}

private void buttonCGraph\_Click(object sender, EventArgs e)

{

zedGraphControlC.GraphPane.CurveList.Clear();

zedGraphControlC.GraphPane.GraphObjList.Clear();

zedGraphControlC.GraphPane.Title.Text = "";

zedGraphControlC.GraphPane.XAxis.Title.Text = "Год";

zedGraphControlC.GraphPane.YAxis.Title.Text = "Количество распавшихся команд";

zedGraphControlC.GraphPane.Y2Axis.Title.Text = "";

zedGraphControlC.GraphPane.XAxis.Type = AxisType.Linear;

zedGraphControlC.GraphPane.XAxis.Scale.TextLabels = null;

zedGraphControlC.RestoreScale(zedGraphControlC.GraphPane);

string connString = dbsettings.GetConnString();

NpgsqlConnection nc = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm = new NpgsqlCommand();

comm.Connection = nc;

comm.CommandType = CommandType.Text;

comm.CommandText = "SELECT \"Team\".\"yearCreation\" FROM \"Team\",\"Conflict\" WHERE \"Team\".\"teamId\"=\"Conflict\".\"teamId\"" +

" and \"Conflict\".\"broke\"=true; ";

NpgsqlDataReader dr = comm.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

comm.Dispose();

nc.Close();

List<DataForGraf> Dfg = new List<DataForGraf>();

DataForGraf temp = new DataForGraf(0);

Dfg.Add(temp);

bool nn = false;

for (int i = 0; i < dt.Rows.Count; i++)

{

for (int j = 0; j < Dfg.Count; j++)

{

if (Convert.ToInt32(dt.Rows[i]["yearCreation"]) == Dfg[j].year)

{

Dfg[j].addC();

break;

}

nn = true;

}

if (nn == true)

{

DataForGraf newdfg = new DataForGraf(Convert.ToInt32(dt.Rows[i]["yearCreation"]));

newdfg.addC();

Dfg.Add(newdfg);

nn = false;

}

}

List<double> counts = new List<double>();

List<string> years = new List<string>();

for (int i = 0; i < Dfg.Count; i++)

{

if (Dfg[i].year != 0)

{

counts.Add(Dfg[i].count);

years.Add(Dfg[i].year.ToString());

}

}

string name = "Колличество распавшихся команд";

zedGraphControlC.GraphPane.Title.Text = name;

LineItem graph = zedGraphControlC.GraphPane.AddCurve(name, null, counts.ToArray(), Color.Red, SymbolType.Diamond);

zedGraphControlC.GraphPane.XAxis.Type = AxisType.Text;

zedGraphControlC.GraphPane.XAxis.Scale.TextLabels = years.ToArray();

zedGraphControlC.GraphPane.AxisChange();

zedGraphControlC.GraphPane.XAxis.Title.Text = "Года";

zedGraphControlC.GraphPane.YAxis.Title.Text = "Колличество распавшихся команд";

zedGraphControlC.Invalidate();

}

private bool GetSemesterByName(string name)

{

if (name == "Весенний")

{

return false;

}

else { return true; }

}

private void buttonCComment\_Click(object sender, EventArgs e)

{

string connString = dbsettings.GetConnString();

NpgsqlConnection nc = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm = new NpgsqlCommand();

comm.Connection = nc;

comm.CommandType = CommandType.Text;

comm.CommandText = "SELECT \"Team\".\"name\", \"Team\".\"comment\" FROM \"Team\", \"Project\", \"Discipline\" " +

"WHERE \"Team\".\"projectId\" = \"Project\".\"projectId\" and \"Project\".\"disciplineId\" = \"Discipline\".\"disciplineId\"" +

"and \"Discipline\".\"name\" = '" + comboBoxCDiscipline.Text + "' and \"Team\".\"yearCreation\"=" + comboBoxCYear.Text + "" +

"and \"Team\".\"typeSemester\"=" + GetSemesterByName(comboBoxCSemester.Text) + "; ";

NpgsqlDataReader dr = comm.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

dataGridViewC.DataSource = dt;

comm.Dispose();

nc.Close();

}

private void buttonCRatio\_Click(object sender, EventArgs e)

{

int AllTeam, BrokeTeam;

string connString = dbsettings.GetConnString();

NpgsqlConnection nc = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm = new NpgsqlCommand();

comm.Connection = nc;

comm.CommandType = CommandType.Text;

comm.CommandText = "SELECT \"Team\".\"name\" FROM \"Team\", \"Project\", \"Discipline\" " +

"WHERE \"Team\".\"projectId\" = \"Project\".\"projectId\" and \"Project\".\"disciplineId\" = \"Discipline\".\"disciplineId\"" +

"and \"Discipline\".\"name\" = '" + comboBoxCDiscipline.Text + "' and \"Team\".\"yearCreation\"=" + comboBoxCYear.Text + "" +

"and \"Team\".\"typeSemester\"=" + GetSemesterByName(comboBoxCSemester.Text) + "; ";

NpgsqlDataReader dr = comm.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

comm.Dispose();

nc.Close();

AllTeam = dt.Rows.Count;

NpgsqlConnection nc2 = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm2 = new NpgsqlCommand();

comm2.Connection = nc;

comm2.CommandType = CommandType.Text;

comm2.CommandText = "SELECT \"Team\".\"name\" FROM \"Team\", \"Project\", \"Discipline\",\"Conflict\" " +

"WHERE \"Team\".\"projectId\" = \"Project\".\"projectId\" and \"Project\".\"disciplineId\" = \"Discipline\".\"disciplineId\" " +

"and \"Conflict\".\"teamId\"=\"Team\".\"teamId\" and \"Conflict\".\"broke\"= true " +

"and \"Discipline\".\"name\" = '" + comboBoxCDiscipline.Text + "' and \"Team\".\"yearCreation\"=" + comboBoxCYear.Text + " " +

"and \"Team\".\"typeSemester\"=" + GetSemesterByName(comboBoxCSemester.Text) + "; ";

NpgsqlDataReader dr2 = comm2.ExecuteReader();

DataTable dt2 = new DataTable();

dt2.Load(dr);

comm2.Dispose();

nc2.Close();

BrokeTeam = dt2.Rows.Count;

DataTable rezult = new DataTable();

rezult.Columns.Add(" ");

rezult.Columns.Add("Count");

rezult.Rows.Add("Комад распалось", BrokeTeam);

rezult.Rows.Add("Комад не распалось", AllTeam - BrokeTeam);

rezult.Rows.Add("Всего команд", AllTeam);

dataGridViewC.DataSource = rezult;

}

private List<Conf> GetConfList()

{

List<Conf> rezult = new List<Conf>();

string connString = dbsettings.GetConnString();

NpgsqlConnection nc = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm = new NpgsqlCommand();

comm.Connection = nc;

comm.CommandType = CommandType.Text;

comm.CommandText = "SELECT \"name\" FROM \"TypeConflict\"; ";

NpgsqlDataReader dr = comm.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

for (int i = 0; i < dt.Rows.Count; i++)

{

Conf temp = new Conf(dt.Rows[i]["name"].ToString());

rezult.Add(temp);

}

comm.Dispose();

nc.Close();

return rezult;

}

private void buttonCStatisticConflict\_Click(object sender, EventArgs e)

{

List<Conf> LC = GetConfList();

string connString = dbsettings.GetConnString();

NpgsqlConnection nc = new NpgsqlConnection(connString);

nc.Open();

NpgsqlCommand comm = new NpgsqlCommand();

comm.Connection = nc;

comm.CommandType = CommandType.Text;

comm.CommandText = "SELECT \"Conflict\".\"broke\",\"TypeConflict\".\"name\" FROM \"Team\", \"Project\", \"Discipline\"," +

"\"Conflict\" ,\"TypeConflict\" WHERE \"Team\".\"projectId\" = \"Project\".\"projectId\" " +

"and \"Project\".\"disciplineId\" = \"Discipline\".\"disciplineId\"" +

"and \"TypeConflict\".\"typeConflictId\" = \"Conflict\".\"typeConflictId\" and \"Conflict\".\"teamId\" = \"Team\".\"teamId\" " +

"and \"Discipline\".\"name\" = '" + comboBoxCDiscipline.Text + "' and \"Team\".\"yearCreation\"=" + comboBoxCYear.Text + " " +

"and \"Team\".\"typeSemester\"=" + GetSemesterByName(comboBoxCSemester.Text) + "; ";

NpgsqlDataReader dr = comm.ExecuteReader();

DataTable dt = new DataTable();

dt.Load(dr);

comm.Dispose();

nc.Close();

for (int i = 0; i < dt.Rows.Count; i++)

{

for (int j = 0; j < LC.Count; j++)

{

if (LC[j].name == dt.Rows[i]["name"].ToString())

{

if (Convert.ToBoolean(dt.Rows[i]["broke"]))

{

LC[j].bac();

}

else

{

LC[j].nbac();

}

break;

}

}

}

DataTable rezult = new DataTable();

rezult.Columns.Add("TypeConflict");

rezult.Columns.Add("Распались");

rezult.Columns.Add("Не распались");

for (int i = 0; i < LC.Count; i++)

{

rezult.Rows.Add(LC[i].name, LC[i].bcount, LC[i].nbcount);

}

dataGridViewC.DataSource = rezult;

}

}

class Conf

{

public string name;

public int bcount;

public int nbcount;

public Conf(string n)

{

this.name = n;

this.bcount = 0;

this.nbcount = 0;

}

public void bac()

{

bcount += 1;

}

public void nbac()

{

nbcount += 1;

}

}

}