МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное  
учреждение высшего образования  
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

(ФГАОУ ВО «ЮФУ»)

Институт компьютерных технологий и информационной безопасности

Кафедра вычислительной техники

К защите допустить:

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_ А.Н. Самойлов

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по специальности 09.05.01 Применение и эксплуатация автоматизированных систем специального назначения (специализация «Автоматизированные системы обработки информации и управления»)

на тему:

«Автоматизированная система для многокритериального ранжирования программируемых логических контроллеров»

Руководитель ВКР:  
доцент кафедры СиПУ,  
к. т. н., доцент А. А. Кузьменко

*(подпись, дата)*

Нормоконтроль:  
доцент кафедры ВТ,  
к. т. н., доцент Д. А. Беспалов

*(подпись, дата)*

Выполнил:  
студент группы КТсо5-2 В. П. Иванов

*(подпись, дата)*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № строки | Формат | Обозначение | Наименование | Кол.  листов | № экз. | Приме-  чание |
| 1 |  |  | Документация общая |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  | Вновь разработанная |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 | А4 | ЦТРК.09.05.01.150061.СА1-20 | Блок – схема генерации отчета | 1 |  |  |
| 6 |  |  | системой для онлайн голосования |  |  |  |
| 7 | А1 | УДК 004.009;342.8ПЛ1-20 | Плакат «Безопасность человеко – |  |  |  |
| 8 |  |  | машинного взаимодействия» | 1 |  |  |
| 9 | А1 | УДК 004.009;342.8ПЛ1-20 | Плакат «Технико-экономическое |  |  |  |
| 10 |  |  | обоснование» | 1 |  |  |
| 11 | \*) | ЦТРК.09.05.01.150061.ПЗ-20 | Пояснительная записка | 93 |  | А4, А1 |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |
| 17 |  |  |  |  |  |  |
| 18 |  |  |  |  |  |  |
| 19 |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |
| 21 |  |  |  |  |  |  |
| 22 |  |  |  |  |  |  |
| 23 |  |  |  |  |  |  |
| 24 |  |  |  |  |  |  |
| 25 |  |  |  |  |  |  |
| 26 |  |  |  |  |  |  |
| 27 |  |  |  |  |  |  |
| 28 |  |  |  |  |  |  |

### У11

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У11

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У11

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У11

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### У1

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

### *У 1*

## *ИКТИБ, гр. КТсо5-2*

## *ИКТИБ, гр. КТсо5-2*

## *ИКТИБ, гр. КТсо5-2*

## *ИКТИБ, гр. КТсо5-2*

###### ЦТРК.09.05.01.150061.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЮФУ, гр. КТсо5-1ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150061.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЮФУ, гр. КТсо5-1ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150061.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЮФУ, гр. КТсо5-1ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150061.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЮФУ, гр. КТсо5-1ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

###### ЦТРК.09.05.01.150074.ВР-20

## *Автоматизированная система для многокритериального ранжирования программируемых логических контроллеров*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

## *Изм.Лист № докум.* *Подп.* *ДатаАвтоматизированная система для онлайн голосования и определения победителя с использованием процедур рейтингового голосования*

## *Ведомость выпускной работы*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

## *Изм.Лист № докум.* *Подп.* *ДатаАвтоматизированная система для онлайн голосования и определения победителя с использованием процедур рейтингового голосования*

## *Ведомость выпускной работы*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

## *Изм.Лист № докум.* *Подп.* *ДатаАвтоматизированная система для онлайн голосования и определения победителя с использованием процедур рейтингового голосования*

## *Ведомость выпускной работы*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

### *Изм.Лист № докум.* *Подп.* *Дата*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

*Утв.*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

### *Лит. Лист Листов*

*Дата*

*Н.контр. Беспалов Д.А.*

*Н.контр. Беспалов Д.А.*

*Н.контр. Беспалов Д.А.*

*Н.контр. Беспалов Д.А.*

*Пров. Кузьменко А.А.*

*Пров. Кузьменко А.А.*

*Пров. Кузьменко А.А.*

*Пров. Кузьменко А.А.*

*Разраб. Иванов В.П.*

*Разраб. Шамраева А.Н.*

*Разраб. Шамраева А.Н.*

*Разраб. Шамраева А.Н.*

ТЕХНИЧЕСКОЕ ЗАДАНИЕ   
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

по специальности 09.05.01 Применение и эксплуатация автоматизированных систем специального назначения (специализация «Автоматизированные системы обработки информации и управления»)

студенту группы КТсо5-2

Иванову Виктору Павловичу

1. **Тема выпускной квалификационной работы:**

«Автоматизированная система для многокритериального ранжирования программируемых логических контроллеров» утверждена приказом ЮФУ № -К от . .2024 г.

1. **Требования и исходные данные к работе:**

Исходные данные: задача многокритериального ранжирования программируемых логических контроллеров.

Постановка задачи: разработать автоматизированную информационную систему, реализующую метод многокритериального ранжирования PROMETHEE II, в виде веб-приложения, использующего стек информационных технологий: HTML, CSS, JS, Python.

Предметная область: автоматизированные информационные системы.

Цель работы: проектирование автоматизированной информационной системы поддержки принятия решений.

Требования:

* формирование математической модели задачи многокритериального ранжирования программируемых логических контроллеров;
* разработка программного модуля, осуществляющего вычисления по методу PROMETHEE II;
* разработка графического интерфейса веб-приложения;
* проектирование автоматизированной информационной системы, обеспечивающей ввод новых или корректировку существующих входных данных (матрицы решений, весов важности критериев), выбор функций предпочтения, осуществляющей вычисления методом PROMETHEE II, вывод итогового полного или краткого отчёта с промежуточными результатами и итоговым ранжированием.

1. **Перечень подлежащих разработке вопросов (содержание работы):**
2. введение;
3. описание предметной области и анализ существующих решений;
4. формирование математической модели задачи многокритериального ранжирования программируемых логических контроллеров;
5. разработка программного модуля для вычислений методом PROMETHEE II;
6. разработка графического интерфейса веб-приложения;
7. проектирование и тестирование автоматизированной информационной системы;
8. разработка руководства пользователя;
9. заключение;
10. список использованных источников.
11. **Перечень графического материала:**

Организационная структура лист

Сравнительный анализ лист

Диаграммы листа

Архитектура лист

Логическая модель лист

Экранные формы листов

1. **Срок сдачи студентом законченной ВКР руководителю:**

**« » июня 2024 г.**

1. **Дата выдачи задания: «2» марта 2024 г.**

Руководитель ВКР:  
доцент кафедры СиПУ,  
к. т. н., доцент А. А. Кузьменко

*(подпись, дата)*

Исполнитель:

студент группы КТсо5-2 В. П. Иванов

*(подпись)*

УДК 004.09;342.8

Иванов Виктор Павлович, группа КТсо5-2

Выпускная квалификационная работа

на тему «Автоматизированная система

для многокритериального ранжирования программируемых логических контроллеров»

Южный федеральный университет,

Инженерная технологическая академия в г. Таганроге,

Институт компьютерных технологий и

информационной безопасности,

2024 г.

АННОТАЦИЯ

Выпускная квалификационная работа направлена на создание информационной системы для многокритериального ранжирования программируемых логических контроллеров. Рассмотрены преимущества метода PROMETHEE II. Спроектирована и построена автоматизированная система в виде веб-приложения, позволяющая на основе предпочтений лица, принимающего решение, произвести ранжирование предложенных альтернатив, вывести итоговый или краткий отчёт по результатам.

UDC 004.09;342.8

Ivanov Victor, KTso5-2 Group

Graduation qualifier work

on the topic "Automated system

for multi-criteria ranking of programmable logic controllers "

Southern Federal University,

Engineering Technology Academy in Taganrog,

Institute of Computer Technology and

Information security,

2024

ABSTRACT

The final qualification work is aimed at creating an information system for multi-criteria ranking of programmable logic controllers. The advantages of the PROMETHEE II method are considered. An automated system in the form of a web application has been designed and built, which allows, based on the preferences of the decision-maker, to rank the proposed alternatives, to output a final or summary report on the results.

УДК 004.09;342.8

РЕФЕРАТ

Выпускная квалификационная работа содержит 93 стр., 23 рис., 14 табл., список использованных источников из 39 наименований, прил.

СИСТЕМА ОНЛАЙН ГОЛОСОВАНИЯ, РАНЖИРУЮЩАЯ ПРОЦЕДУРА, АВТОМАТИЗИРОВАННАЯ СИСТЕМА, ПРОЦЕДУРА БОРДА, ПРОЦЕДУРА КОУПЛЕНДА, ПРОЦЕДУРА КУМБСА, ПРОЦЕДУРА ФИШБЕРНА, ПРОЦЕДУРА КОНДОРСЕ, ПРОЦЕДУРА НАНСОНА, ПРОЦЕДРА ДОДЖСОНА, ПРОЦЕДУРА СИМПСОНА

Предметом исследования в приведенной работе является автоматизированная система многокритериального ранжирования, использующая метод PROMETHEE II.

Цель работы представляет собой создание автоматизированной системы для многокритериального ранжирования программируемых логических контроллеров.

Для решения поставленной цели был изучен метод многокритериального ранжирования PROMETHEE II, создана серверная часть, реализующая этот метод на языке программирования Python. Разработан интерфейс веб – приложения с использованием следующего стека технологий: HTML, CSS, JavaScrypt.

Основным результатом работы является: создание автоматизированной системы многокритериального ранжирования программируемых логических контроллеров в виде веб-приложения.

Область применения – все производства, у которых есть задача выбора программируемых логических контроллеров.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 12](#_Toc43326501)

[1 Описание предметной области 14](#_Toc43326502)

[1.1 Понятие и общая идеология голосования 14](#_Toc43326503)

[1.2 Области применения системы онлайн голосования 17](#_Toc43326504)

[1.3 Обзор аналогов 20](#_Toc43326505)

[1.3.1 ГАС «Выборы» 20](#_Toc43326506)

[1.3.2 Polys 21](#_Toc43326507)

[1.3.3 Система «ДОМСКАНЕР» 21](#_Toc43326508)

[1.3.4 ИС «Кворум» 22](#_Toc43326509)

[1.3.5 I-VOTE 22](#_Toc43326510)

[1.3.6 Electionrunner 22](#_Toc43326511)

[1.3.7 Сравнительная характеристика 23](#_Toc43326512)

[1.1 Выявление проблемы 25](#_Toc43326513)

[1.2 Описание процедур рейтингового голосования 27](#_Toc43326514)

[1.1.1 Процедура Борда 27](#_Toc43326515)

[1.1.2 Модифицированная процедура Борда 28](#_Toc43326516)

[1.1.3 Процедура Кондорсе 29](#_Toc43326517)

[1.1.4 Процедура Симпсона 29](#_Toc43326518)

[1.1.5 Процедура Доджсона 30](#_Toc43326519)

[1.1.6 Процедура Нансона 31](#_Toc43326520)

[1.1.7 Процедура Кумбса 32](#_Toc43326521)

[1.1.8 Процедура Коупленда 33](#_Toc43326522)

[1.1.9 Процедура Фишберна 34](#_Toc43326523)

[2 Проектирование системы для онлайн голосования, использующая рейтинговые процедуры голосования 35](#_Toc43326524)

[2.1 База данных системы онлайн голосования 35](#_Toc43326525)

[2.2 Общая архитектура системы онлайн голосования 38](#_Toc43326526)

[2.3 Функциональная структура системы онлайн голосования 40](#_Toc43326527)

[3 Разработка системы для онлайн голосования 45](#_Toc43326528)

[3.1 Обоснование выбора IDE 45](#_Toc43326529)

[3.2 Описание процесса реализации системы 46](#_Toc43326530)

[3.3 Интерфейс системы онлайн голосования 49](#_Toc43326531)

[3.4 Описание работы системы 53](#_Toc43326532)

[3.5 «Ручное» вычисление результатов голосования 56](#_Toc43326533)

[3.5.1 Процедура Борда 56](#_Toc43326534)

[3.5.2 Модифицированная процедура Борда 57](#_Toc43326535)

[3.5.3 Процедура Кондорсе 58](#_Toc43326536)

[3.5.4 Процедура Симпсона 58](#_Toc43326537)

[3.5.5 Процедура Доджсона 58](#_Toc43326538)

[3.5.6 Процедура Нансона 59](#_Toc43326539)

[3.5.7 Модифицированная процедура Нансона 60](#_Toc43326540)

[3.5.8 Процедура Кумбса 61](#_Toc43326541)

[3.5.9 Процедура Коупленда 62](#_Toc43326542)

[3.5.10 Процедура Фишберна 62](#_Toc43326543)

[3.6 Сравнение «ручных» вычислений с результатами работы системы 63](#_Toc43326544)

[4 Технико-экономическое обоснование 65](#_Toc43326545)

[5 Безопасность человеко-машинного взаимодействия 74](#_Toc43326546)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 81](#_Toc43326547)

[ПРИЛОЖЕНИЕ А 85](#_Toc43326548)

[ПРИЛОЖЕНИЕ Б 86](#_Toc43326549)

[ПРИЛОЖЕНИЕ В 87](#_Toc43326550)

[ПРИЛОЖЕНИЕ Г 88](#_Toc43326551)

[ПРИЛОЖЕНИЕ Д 89](#_Toc43326552)

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В данной работе приняты следующие сокращения:

АС – автоматизированная система

БД ­– база данных

БЧМВ – безопасность человеко – машинного взаимодействия

ГАС – государственная автоматизированная система

ЕВС – Европейский вещательный союз

ИС – информационная система

ЛПР – лицо, принимающее решение

ОЗУ – оперативное запоминающее устройство

ПЗУ – постоянное запоминающее устройство

ПО – программное обеспечение

ПЛК – программируемый логический контроллер

СЗИ – средство защиты информации

ТЭО – технико – экономическое обоснование

ФЗ – федеральный закон

CURD – Create Update Read Delete

JWT – JSON web token

ВВЕДЕНИЕ

Сегодня в промышленной автоматизации для обеспечения высокой надежности и эффективности производственных процессов критически важно подбирать математически обоснованные технические решения. ПЛК является одним из ключевых элементов в АС. Основная задача ПЛК состоит в управлении различными механизмами и процессами на производстве. Выбор нужного ПЛК из множества альтернатив на рынке может быть сложной задачей из-за разнообразия технических характеристик и функциональных особенностей. Поэтому разработка АС для многокритериального ранжирования ПЛК является актуальной и значимой задачей.

Вышеописанная система позволит инженерам и техническим специалистам эффективно сравнивать и анализировать различные модели ПЛК, учитывая комплекс определенных критериев и технических параметров. Автоматизация процесса ранжирования не только ускоряет выбор наиболее подходящей альтернативы, но и способствует повышению эффективности инженерных и производственных процессов, оптимизации использования ресурсов и снижению затрат.

В этой работе показан процесс разработки АС для многокритериального ранжирования ПЛК, анализируются ключевые аспекты выбора критериев, рассмотрен метод многокритериального ранжирования PROMETHEE II.

# Описание предметной области

## Введение в ПЛК

ПЛК является основой для АСУ в различных отраслях промышленности, включая энергетику, медицину, транспорт и многое другое. ПЛК созданы для автоматизации процессов, управления технологическими установками и машинами с использованием ПО, которое выполняет функцию контроля состояния входов и вырабатывает определенные последовательности заданных действий, отражающихся в изменении значений выходов.

Все ПЛК содержат следующие ключевые элементы:

1. Процессор: обрабатывает информацию и выдает управляющие команды.
2. Память: предназначена для хранения программ управления и временных данных.
3. Модули ввода и вывода: нужны для подключения ПЛК к оборудованию.
4. Интерфейсные модули: интеграция с другими системами.

## Критерии для ранжирования ПЛК

Чтобы произвести ранжирование ПЛК, нужно учесть множество факторов, которые можно разделить на следующие группы:

1. Технические характеристики: объем памяти, частота процессора, количество входов и выходов.
2. Физические особенности: размер и цвет ПЛК.
3. Функциональность: поддержка различных протоколов связи, удобная среда программирования, возможность модификации и расширения системы.
4. Экономические факторы: стоимость ПЛК, наличие гарантии и её срок.
5. Обратная связь с производителем: наличие технической поддержки и обучающих материалов.

## Выявление проблематики темы ВКР

Многокритериальное ранжирование ПЛК является сложной задачей из–за противоречивости критериев между собой и большого количества альтернатив.

АСУ для многокритериального ранжирования ПЛК могут значительно упростить процесс выбора подходящего контроллера, произведя анализ больших объёмов данных и предоставив пользователям удобные и интуитивно понятные инструменты для ранжирования ПЛК. Эти системы помогут оптимизировать процесс принятия решений, повысить эффективность проектирования систем и снизить риски, которые связаны с выбором неоптимального оборудования.

## Метод многокритериального ранжирования PROMETHEE II

### Процедура Борда

Процедура Борда (1770 г.) была исторически первой системой голосования, где использовалась ранжирующая процедура учета мнений голосующих.

Процедура состоит из следующих шагов:

1. каждый участник ранжирует все варианты A1, ..., Am по предпочтительности;
2. в каждом s-м индивидуальном строгом упорядочении первый вариант получает (m-1) баллов, второй вариант — (m-2) балла, последний вариант — 0 баллов;
3. для каждого варианта Аi вычисляется значение функции Борда:

, (1.1)

где ­­– сумма баллов Борда, присвоенных варианту Аi во всех индивидуальных упорядочениях;

1. Упорядочение вариантов строится по убыванию значения функции Борда. Лучший вариант A\* соответствует максимальному значению функции Борда [27]:

(1.2)

### Модифицированная процедура Борда

1. каждый участник ранжирует все варианты А1, ..., Аm по предпочтительности.
2. подсчитывается количество индивидуальных строгих упорядочений одного и того же вида.
3. для каждого варианта Аi подсчитывается общее число g (Аi, Аj) участников, предпочитающих данный вариант Аi всем остальным вариантам Аj при парных сравнениях во всех полученных упорядочениях.
4. для каждого варианта Аi вычисляется значение модифицированной функции Борда:

, (1.3)

1. упорядочение вариантов строится по убыванию значения модифицированной функции Борда. лучший вариант А\* соответствует максимальному значению модифицированной функции Борда [27]:

. (1.4)

### Процедура Кондорсе

Принцип Кондорсе (1785 г.) гласит: победителем на выборах объявляется кандидат, превосходящий при попарном сравнении всех остальных кандидатов.

Процедура Кондорсе состоит из следующих шагов:

1. каждый участник ранжирует все варианты А1, ..., Аm по предпочтительности;
2. подсчитывается количество индивидуальных строгих упорядочений одного и того же вида;
3. для каждого варианта Аi подсчитывается общее число g (Аi, Аj) участников, предпочитающих данный вариант Аi всем остальным вариантам Аj при парных сравнениях во всех полученных упорядочениях;
4. лучший вариант при парном сравнении вариантов определяется построением общего упорядочения [28].

### Процедура Симпсона

Процедура Симпсона (1965 г.) позволяет избежать возникновения парадокса несогласованности Кондорсе и состоит в следующем:

1. каждый участник ранжирует все варианты А1, ..., Аm по предпочтительности;
2. подсчитывается количество индивидуальных строгих упорядочений одного и того же вида;
3. для каждого варианта Аi подсчитывается общее число g (Аi, Аj) участников, предпочитающих данный вариант Аi всем остальным вариантам Аj при парных сравнениях во всех полученных упорядочениях;
4. для каждого варианта Аi определяется значение функции Симпсона:

, (1.5)

равное минимальному числу участников, предпочитающих данный вариант Аi всем остальным вариантам Aj (j ≠ i) при всех парных сравнениях;

1. упорядочение вариантов строится по убыванию значения функции Симпсона. лучший вариант А\* соответствует максимальному значению функции Симпсона [29]:

. (1.6)

### Процедура Доджсона

Основная идея процедуры Доджсона (1877 г.) заключается в следующем: все варианты упорядочиваются по числу голосов, которых им не хватает для того, чтобы превосходить все остальные варианты по простому большинству (более половины от общего числа) голосов.

1. каждый участник ранжирует все варианты А1, ..., Аm по предпочтительности;
2. подсчитывается количество индивидуальных строгих упорядочений одного и того же вида;
3. для каждого варианта Аi подсчитывается общее число g (Ai, Aj) участников, предпочитающих данный вариант Аi всем остальным вариантам Aj при парных сравнениях во всех полученных упорядочениях;
4. строится вспомогательная матрица H = (hij)m\*m, где hij = g (Ai, Aj)/ g (Aj, Ai) — групповая матрица парных сравнений вариантов по числу поданных голосов. Для каждого варианта Аi по строке матрицы Н находятся варианты Аj для которых hij <1. Это как раз и будут те варианты, в сравнении с которыми вариант Аi получил меньше половины от общего числа голосов. Обозначим совокупность таких вариантов через A1/2;
5. для каждого варианта Аi вычисляется значение функции Доджсона:

. (1.7)

где , если число t избирателей четно, и при нечетном t.

Лучший вариант A\* соответствует минимальному значению функции Доджсона [29].

### Процедура Нансона

Процедура Нансона (1882 г.) объединяет подход Борда и принцип Кондорсе. Основная идея процедуры состоит в последовательном сокращении множества вариантов путем исключения получивших минимальное число голосов до тех пор, пока не останутся не исключаемые (недоминируемые) варианты. Эти варианты и будут считаться результатом коллективного выбора.

1. каждый участник ранжирует все варианты А1, ..., Аm по предпочтительности;
2. подсчитывается количество индивидуальных строгих упорядочений одного и того же вида;
3. для каждого варианта Аi подсчитывается общее число g (Ai, Aj) участников, предпочитающих данный вариант Аi всем остальным вариантам Aj при парных сравнениях во всех полученных упорядочениях;
4. определяется вариант , соответствующий минимальному значению функции Борда:

. (1.8)

Этот вариант исключается из рассмотрения.

1. повторяются шаги 1-4, и на суженном множестве A1 = A\{} вариантов находится следующий вариант , имеющий минимальное число голосов, который аналогично исключается из рассмотрения;
2. шаги 1-4 повторяются, пока не останется один или несколько вариантов, которые дальше нельзя исключить. Оставшиеся наиболее предпочтительные варианты составляют множество А\*, определяемое функцией Нансона:

(1.9)

В модифицированной процедуре Нансона из рассмотрения исключается не единственный вариант, имеющий минимальную оценку Борда, а сразу несколько вариантов, оценки которых меньше среднего по всем вариантам значения функции Борда [29]:

. (1.10)

### Процедура Кумбса

Основная идея процедуры Кумбса (1950 г.) состоит в последовательном сокращении множества вариантов путем исключения худших по большинству голосов вариантов до тех пор, пока не останутся не исключаемые варианты.

1. по результатам голосования определяются худшие варианты: для каждого варианта суммируются голоса, в предпочтениях которых соответствующий вариант занимает последнее место;
2. альтернатива, получившая наибольшее число голосов, выбывает;
3. множество альтернатив сужается до нового множества, и процедура продолжается до тех пор, пока не останутся только не исключаемые варианты [29].

### Процедура Коупленда

В основе процедуры Коупленда (1950 г.) лежит идея учета различия индивидуальных предпочтений участников, аналогичная идее модифицированной процедуры Борда. Однако здесь различие определяется не количеством голосов, поданных за более и менее предпочтительные варианты, а числом самих предпочтительных вариантов.

1. каждый участник сравнивает все варианты А1, ..., Аm попарно друг с другом независимо от сравнений других пар вариантов;
2. для каждого варианта Ai подсчитывается число n (Ai, Aj) вариантов Aj, уступающих варианту Ai, и число n (Aj, Ai) вариантов Aj, превосходящих вариант Ai по простому большинству голосов;
3. для каждого варианта Ai вычисляется значение функции Коупленда:

(1.11)

равное разности числа вариантов Aj, которые уступают варианту Ai, и числа вариантов Aj, которые превосходят вариант Ai по простому большинству голосов;

1. упорядочение вариантов строится по убыванию значения функции Коупленда. Лучший вариант А\* определяется максимальным значением функции Коупленда [29]:

(1.12)

### Процедура Фишберна

Эта процедура объединяет идеи процедур Борда, Кондорсе и Коупленда, определяя предпочтительность каждого варианта по числу вариантов, которые данный вариант превосходит по простому большинству голосов.

1. каждый участник сравнивает все варианты А1, ..., Аm попарно друг с другом независимо от сравнений других пар вариантов;
2. для каждого варианта Ai вычисляется значение функции Фишберна, равное числу вариантов Aj, которые уступают варианту Ai по простому большинству голосов:

. (1.13)

1. упорядочение вариантов строится по убыванию значения функции Фишберна. Лучший вариант А\* определяется максимальным значением функции Фишберна [29]:

. (1.14)

# Проектирование системы для онлайн голосования, использующая рейтинговые процедуры голосования

## База данных системы онлайн голосования

Прежде чем переходить к разработке программного продукта, надо разработать проект базы данных, которая будет использоваться в данной системе.

При проектировании базы данных были разработаны диаграммы БД на физическом уровне, которое представляют собой таблицы со столбцами, связанные между собой внешними ключами [30].

В процессе логического проектирования БД были выделены следующие сущности:

1. таблица пользователей, атрибутами которой являются:

* верификационный номер пользователя;
* id голосования;
* идентификационный номер пользователя;
* логин;
* соль;
* роль;
* ФИО пользователя;
* пароль пользователя;
* подтверждение регистрации;

1. таблица голосований, атрибутами которой являются:

* id голосования;
* дата создания голосования;
* окончание голосования;
* название голосования;
* описание голосования;
* статус голосования;
* ожидаемое количество пользователей;
* аватар голосования;
* победитель.

1. таблица кандидатов, атрибутами которой являются:

* id кандидата;
* описание кандидата;
* имя кандидата;
* фамилия кандидата;
* отчество кандидата;
* программа;
* дата рождения;
* должность;
* образование;
* учебные заведения;
* пароль кандидата,
* фото кандидата.

1. таблица результатов голосования, атрибутами которой являются:

* id результата;
* id голосования,
* имя процедуры голосования;
* результаты процедур голосований.

1. таблица сессий, атрибутами которой являются:

* id сессии;
* token;
* соль;
* логин;
* временная метка;
* id пользователя.

1. таблица пользовательских голосов, атрибутами которой являются:

* id пользователя;
* id голосования;
* выбранная пользователь последовательность кандидатов;
* сессия.

На рисунке 2.1 изображения физическая модель базы данных системы онлайн голосования с использования процедур рейтингового голосования, на которой видны таблицы с их атрибутами. В реляционных базах данных связь "многие ко многим" реализуется путем добавления дополнительной таблицы, и связью "один к многим" [31].

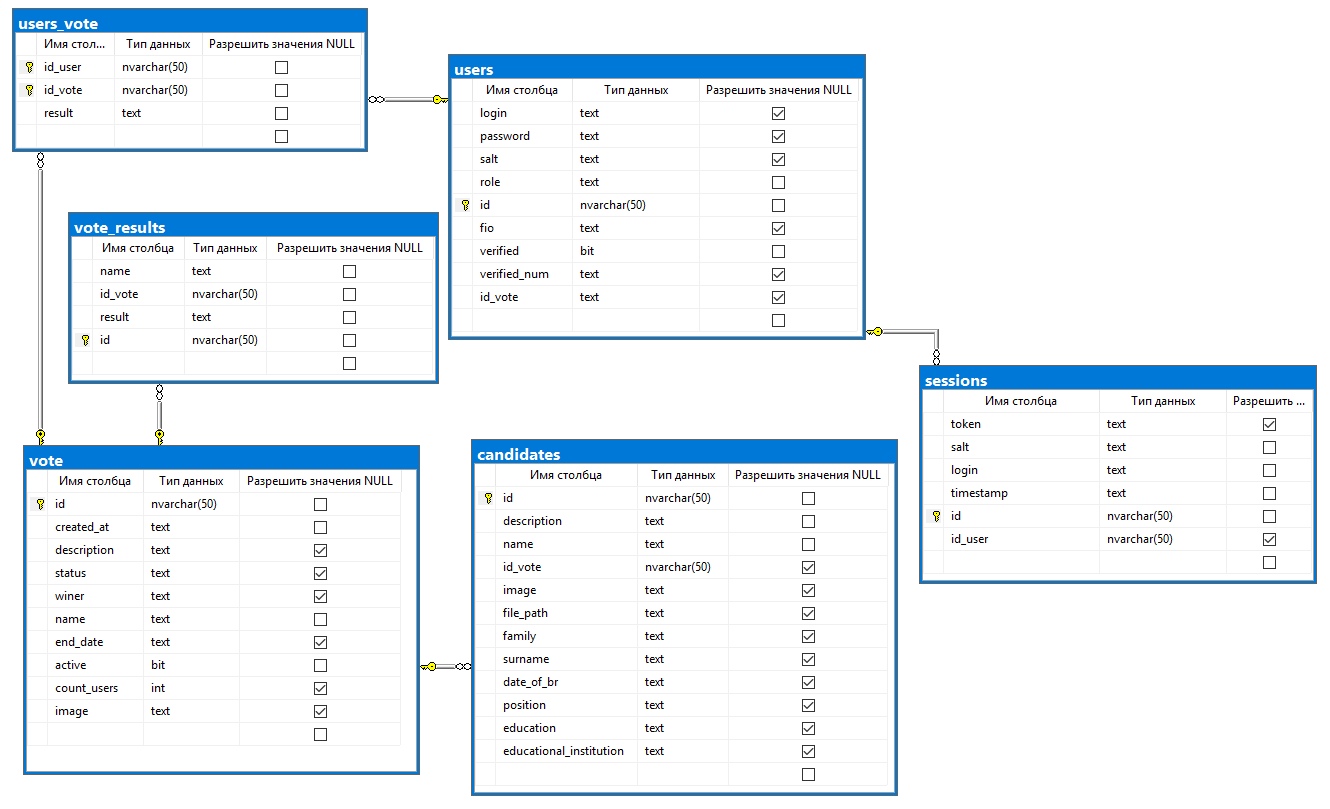


Рисунок 2.1 – Физическая модель базы данных

На рисунке 2.1 изображена физическая модель базы данных, на которой видно, что:

* таблица голосований связана по связи «одни ко многим» с таблицей кандидатов, так как у одного голосования может быть много кандидатов;
* таблица голосований связана по связи «один ко многим» с таблицей результатов голосований, так как результатам голосований является несколько рейтинговых процедур голосования;
* таблица пользовательских голосов связана по связи «один ко многим» с таблицами голосования и пользователи (таблица пользовательских голосов имеет составной первичный ключ), так как пользователь может голосовать несколько раз, но в одном голосовании он может голосовать только один раз;
* таблица пользователей связана по связи «один ко многим» с таблицей сессии, так как одни пользователь может иметь много сессий.

## Общая архитектура системы онлайн голосования

На диаграмме (рисунок 2.2) показана статистическая структура данных [32].



Рисунок 2.2 – Взаимодействие компонентов системы

Созданы следующие типы объектов системы, которые представлены в виде компонентов:

* база данных;
* модуль работы с БД, ответственен за операции CURD;
* приложение;
* модуль подсчета результатов, ответственен за подсчет голосов и их ранжировании с использованием процедур рейтингового голосования, а также формирования итогового отчет голосования с определением победителя;
* модуль работы с файлами, который включает в себя сохранение файлов, генерацию файлов отчетов;
* модуль cron, который занимается поиском открытых голосований, если дата окончания больше или равна текущей, то закрывает голосование и вызывает подсчет результатов;
* модуль авторизации, который включает в себя регистрацию и авторизацию в системе.

Представленная диаграмма (рисунок. 2.2) описывает взаимодействие компонентов в системе:

* приложение взаимодействует с модулем работы БД для выборки, обновления, создания и удаления данных;
* при регистрации либо авторизации пользователя приложение взаимодействует с модулем авторизации;
* модуль авторизации взаимодействует с модулем работы с БД для выборки данных в случае авторизации и добавлении данных нового пользователя в случае регистрации;
* модуль подсчета результата взаимодействует с приложением при закрытии голосования. закрытие голосования служит триггером для начала работы модуля подсчета результатов;
* модуль подсчет результата взаимодействует с модулем работы с базой данных для выборки входных данных о количестве голосов и их обработки с помощью процедур рейтингового голосования, а также для записи в БД итогов голосования;
* модуль cron взаимодействует с БД для получения открытых голосований и закрытия голосования, при нахождении таких служит триггером для модуля подсчета результатов;
* модуль работы с файлами взаимодействует с БД для взятия пути к файлам;
* Приложение взаимодействует с модулем работы с файлами, когда на сервер приходит запрос на сохранение или получение файла;
* база данных – вся работа с базой, осуществляется только через модуль работы с базой данных. это сделано по двум причинам – упрощает работу разработчикам, посредством создания абстракции над языком взаимодействия с базой данных и в случае принятия решения о смене базы данных, переписать нужно будет только этот модуль, в противном случает нужно было бы искать все методы работы с базой данных во всей бизнес-логике информационной системы.

## Функциональная структура системы онлайн голосования

На рисунке 2.3 изображена диаграмма действий системы онлайн голосования с использованием процедур голосования [33], на которой изображена основная последовательность действий администратора и пользователя системы.



Рисунок 2.3 – Диаграмма действий

Начинает работу администратор с создания онлайн голосования и определения времени его завершения. Далее он формирует ссылку на голосование (либо формирует QR– код) и распространяет ее. Пользователи переходят по созданной ссылке, регистрируются, либо же проходят авторизацию, если ранее был создан аккаунт в системе. Далее голосуют за одного из кандидатов, предложенных системой онлайн голосования. По истечению срока голосования система закрывает его, обрабатывает данные и формирует итоговые результаты по всем процедурам рейтингового голосования с определением победителя. Затем система генерирует файл – отчет по результатам голосования, администратор исходя из выбора заказчика принимает решение. После чего система отображает победителя на стартовой странице голосования.

На рисунке 2.4 изображена диаграмма вариантов использования системы онлайн голосования с использованием процедур рейтингового голосования [34], на которой показаны варианты работы системы. В роли акторов выступают «Администратор», «Кандидат», «Система» и «Пользователь». Создание голосования подразумевает под собой время начала и окончания голосования, создание профилей кандидатов и формирование ссылки на голосование, поэтому между данными прецедентами стоит отношение «include». Также остановка даты окончания голосования включается в открытие голосования, поэтому между данными прецедентами стоит отношение «include».



Рисунок 2.4 – Диаграмма вариантов использования

На диаграмме (рисунок 2.5) показана статическая структура данных. Созданы типы объектов системы, которые представлены в виде классов. Определены атрибуты и операции классов, а также отношения между ними. В качестве отношений используются ассоциации, композиции и зависимость.



Рисунок 2.5 – Диаграмма классов

На рисунке 2.6 изображена диаграмма последовательности, которая отображает работу системы онлайн голосования.



Рисунок 2.6 – Диаграмма последовательности

Для этого кандидату необходимо предоставить администратору данные о себе, дождаться проверки правильности заполнения данных. Также всю информацию о голосовании администратор получает от заказчика (название, описание, ожидаемое количество голосующих и их список). После получения списка пользователей администратор загружает их в система, там генерируется идентификационный номер, затем отправляется заказчику для распространения уникальных номеров. Далее создает голосование и добавляет данные о кандидатах, открывает и формирует ссылку, с дальнейшим ее распространением. Система после запроса на формирование ссылки, возвращает ее администратору. Пользователь регистрируется (проходит авторизацию) по полученной ссылке, вводит уникальный идентификационный номер, система проверяет совпадения. Пользователь голосует. По окончанию установленного времени система закрывает голосование. Система производит подсчет результатов, определяет победителя и формирует отчет. Затем администратор готовый отчет отправляет заказчику голосования, он принимает решение, после чего администратор вносит в системы победителя, в свою очередь, система отображает результаты голосования на стартовой странице голосования.

# Разработка системы для онлайн голосования

## Обоснование выбора IDE

Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows.

WebStorm – Интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы IntelliJ IDEA. WebStorm обеспечивает, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями.

Visual Studio Code — это редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. он включает в себя поддержку отладки, встроенного управления Git и GitHub, подсветку синтаксиса, интеллектуальное завершение кода, фрагменты и рефакторинг кода. Он легко настраивается, позволяя пользователям изменять тему, сочетания клавиш, настройки и устанавливать расширения, которые добавляют дополнительную функциональность. Исходный код является свободным и открытым исходным кодом и выпущен под разрешительной лицензией MIT. скомпилированные двоичные файлы являются бесплатными и бесплатными для частного или коммерческого использования.

Код Visual Studio основан на Electron, фреймворке, который используется для разработки Node.приложения js для рабочего стола, работающие на движке Blink layout engine. Хотя программа использует платформу Electron, она не использует Atom, а использует тот же компонент редактора (под кодовым названием Monaco), который используется в Azure DevOps (ранее называвшихся Visual Studio Online и Visual Studio Team Services).

Произведем сравнение существующих сред разработки методом попарного сравнения (таблица 3.1).

Таблица 3.1– Сравнительный анализ IDE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Visual Studio | WebStorm | VS Code | Сумма баллов |
| Visual Studio | 1 | 0.7 | 0.8 | 2,5 |
| WebStorm | 1.2 | 1 | 0.9 | 3,1 |
| VS Code | 1.5 | 1 | 1 | 3,5 |
| Сумма | | | | 9,1 |

На основании значимости веса факторов (таблица 3.1) ранжируемость рассматриваемых факторов принимает вид:

1. VS Code;
2. WebStorm;
3. Visual Studio.

## Описание процесса реализации системы

Создана база данных в PostgreSQL (рисунок 3.1).

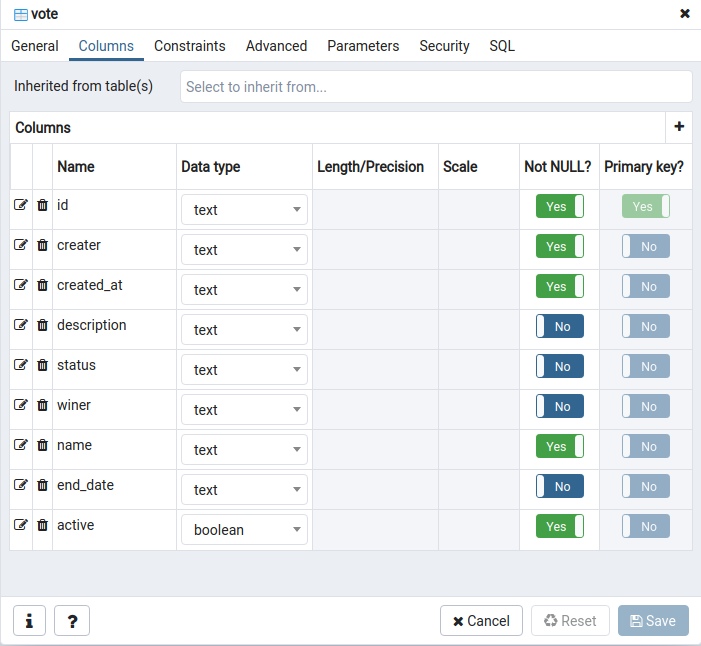


Рисунок 3.1– БД PostgreSQL

Создан модуль для обработки голосования процедурами рейтингового голосования (рисунок 3.2). В приложении Д приведен полный листинг программного модуля.

Рисунок 3.2 – Программный модуль обработки голосования

Процесс работы системы начинается с открытия администратором голосования и установления им даты закрытия. Дальше пользователи регистрируется, либо же авторизуются (если ранее была создана учетная запись), голосуют. После закрытия голосования система формирует результаты по каждой процедуре рейтингового голосования. Затем администратор формирует отчет и отправляет его заказчикам или другим ответственным лицам. После принятия решения о том, кто является победителем, администратор выставляет результаты на основную страницу голосования, для ее просмотра пользователями. Система переводит голосование в состояние «закрытое» с возможностью просмотра администратором (при необходимости и другими уполномоченными лицами) данных голосования, а именно его названия, победителя и количестве проголосовавших (количество кто должен был проголосовать и количестве, кто проголосовал). Администратор во время голосования и после него может создавать и редактировать другие голосование. Одновременно может находиться в статусе «открыто» только одно голосование.

## Интерфейс системы онлайн голосования

В процессе разработки получился готовый прототип web-приложения, представленный на рисунках 3.3 – 3.14 [35].

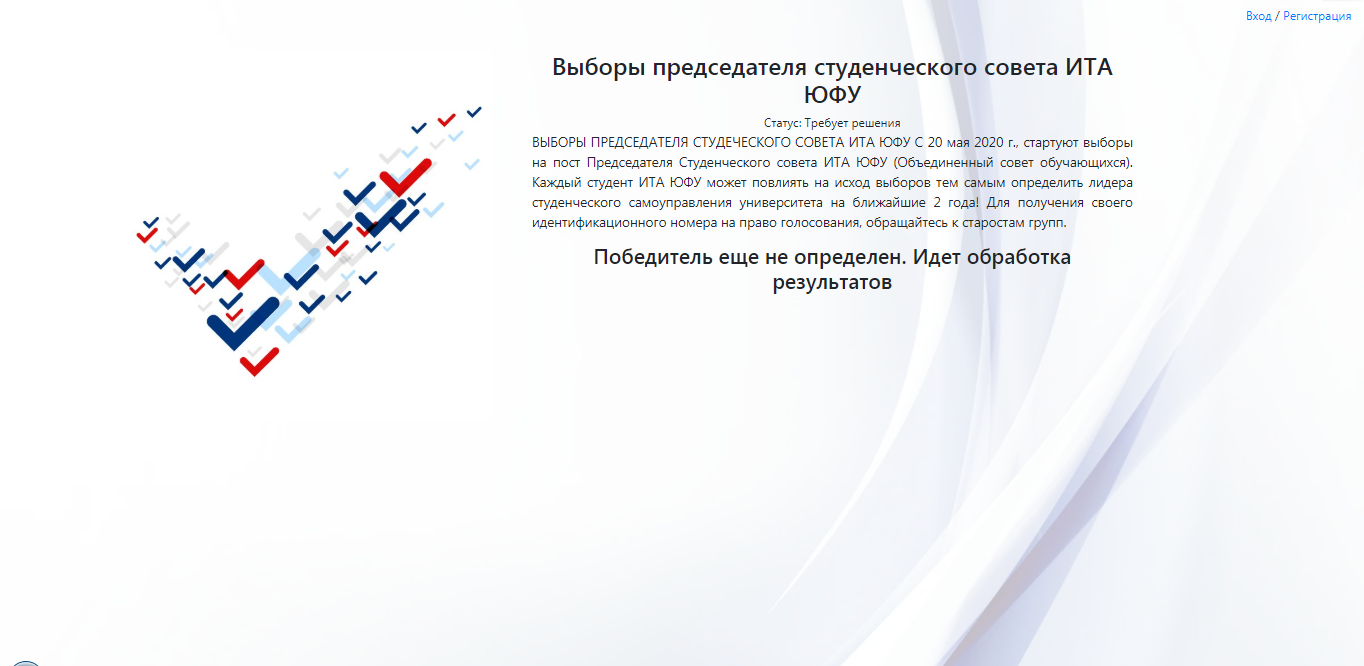


Рисунок 3.3 – Страница голосования

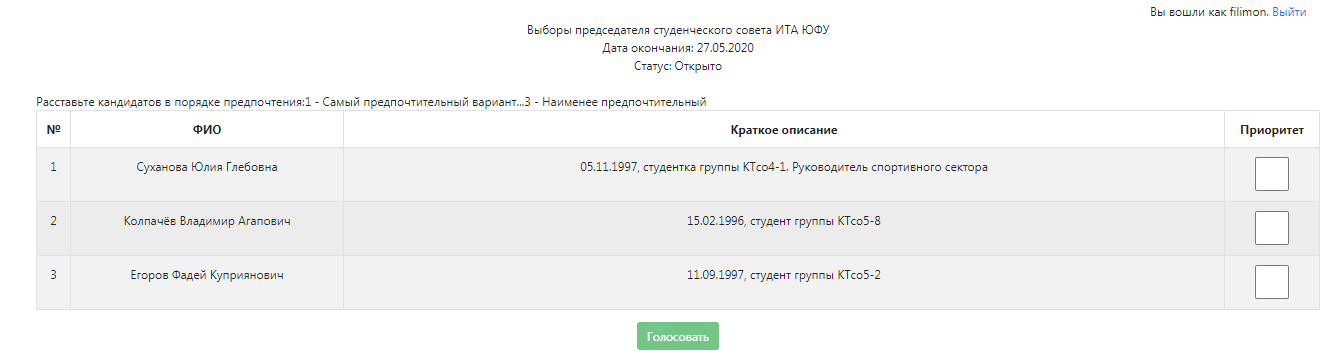
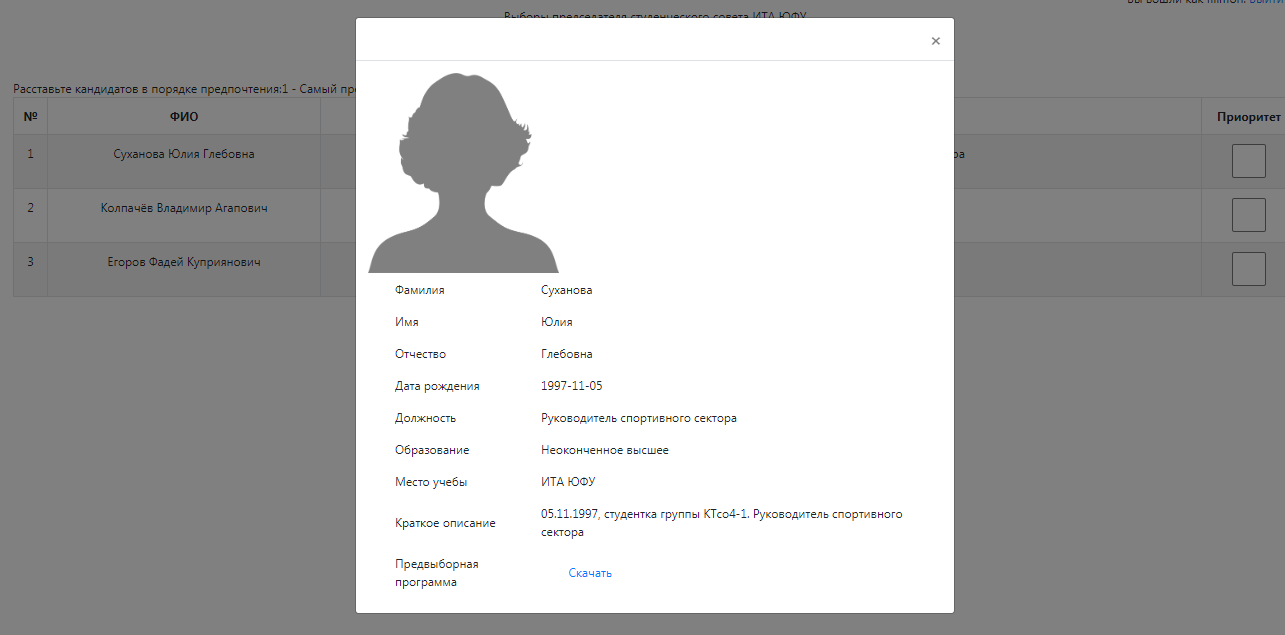
Рисунок 3.4 – Страница голосования авторизованного пользователя

Рисунок 3.5 – Просмотр профиля кандидата

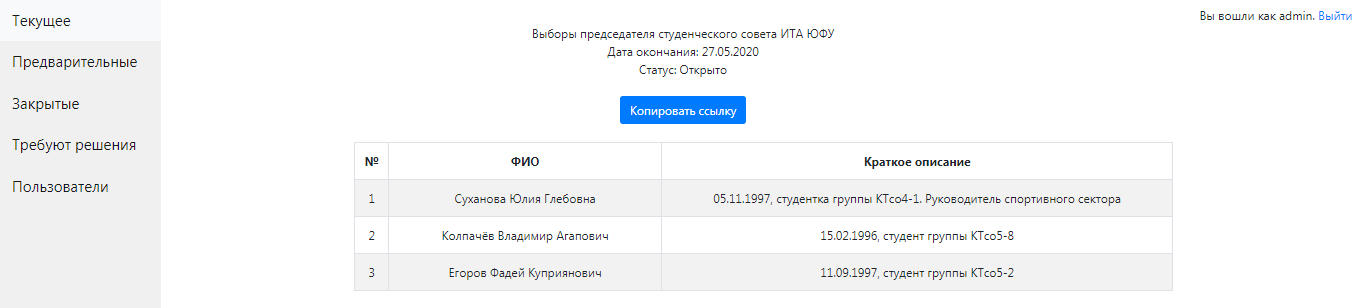


Рисунок 3.6 – Страница администратора. Текущее голосование

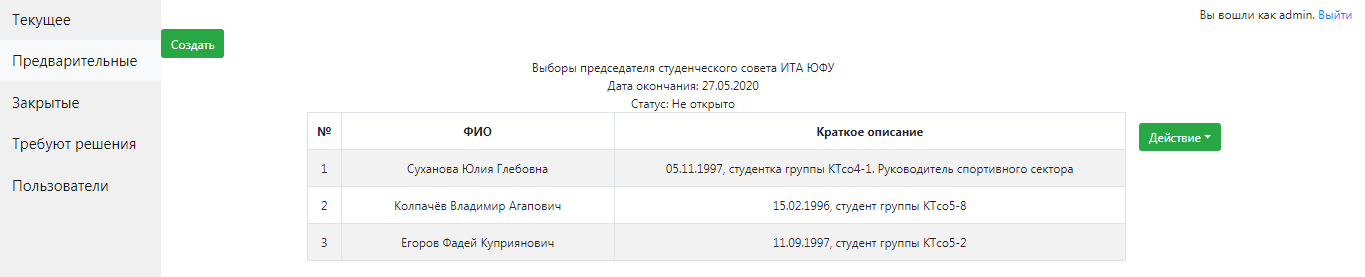
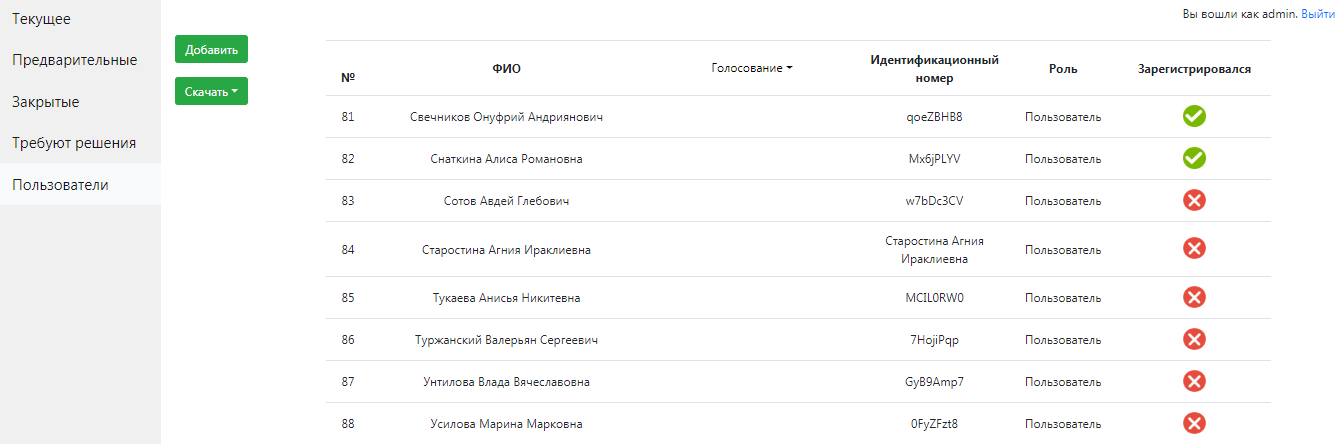
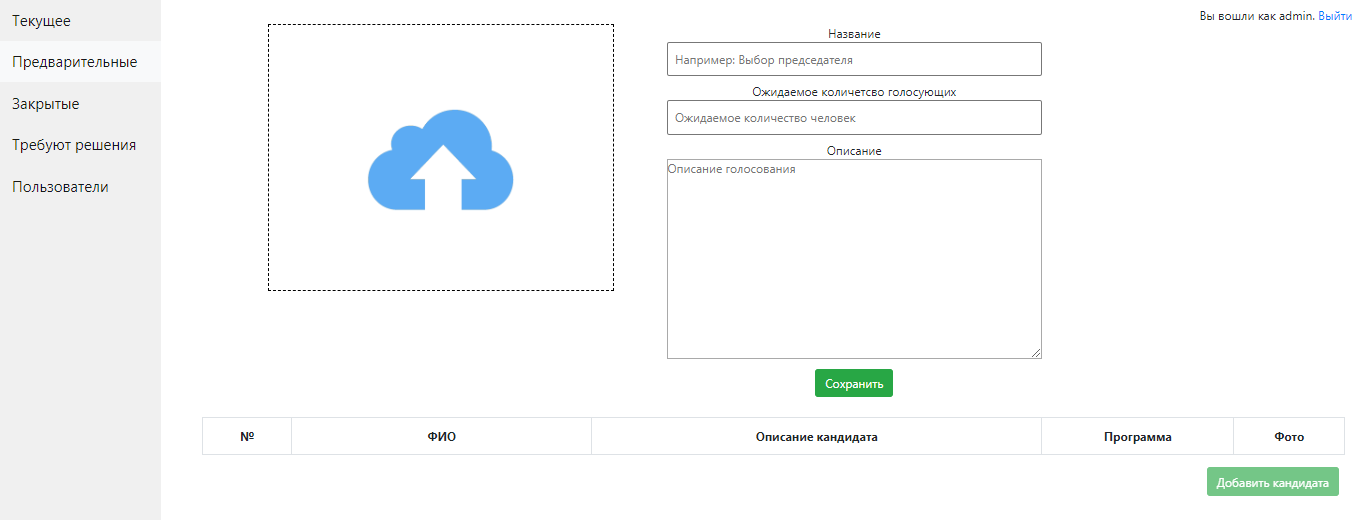


Рисунок 3.7 – Страница администратора. Предварительное голосование

Рисунок 3.8 – Пользователи системы

Рисунок 3.9 – Создание нового голосования

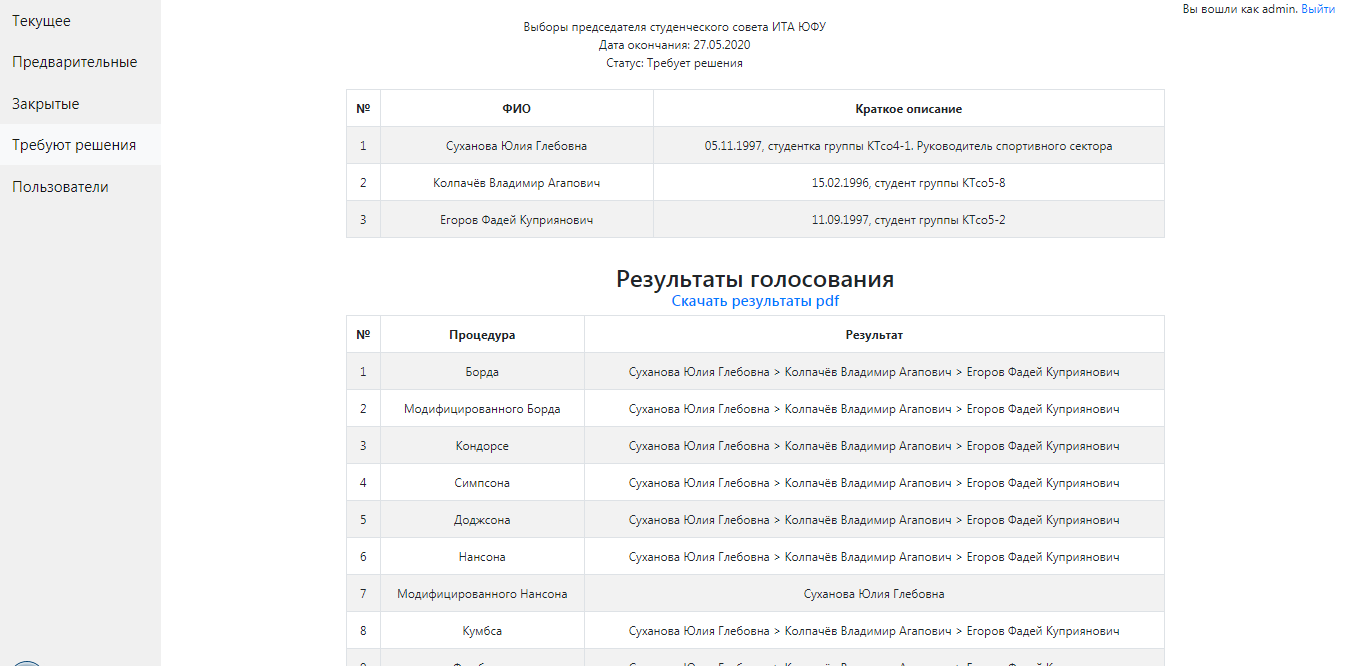
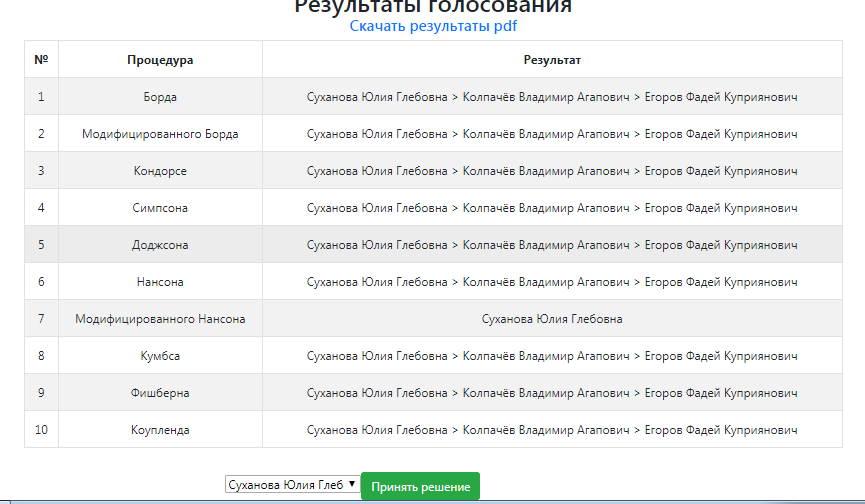
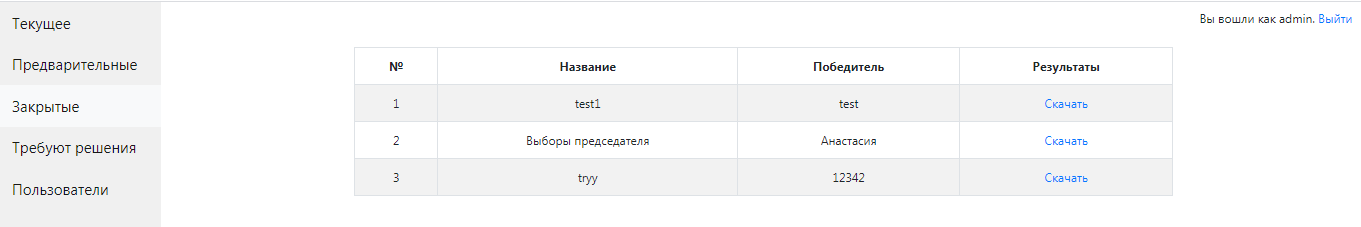


Рисунок 3.10 – Страница администратора. Результаты голосования

Рисунок 3.11 – Страница администратора. Принятие решения

Рисунок 3.12 – Страница администратора. Закрытые голосования

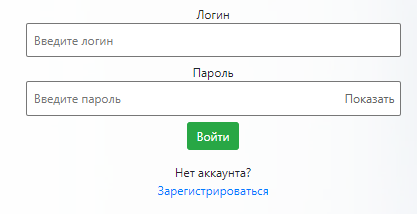


Рисунок 3.13 – Страница авторизации

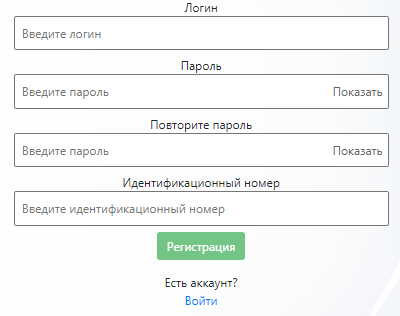


Рисунок 3.14 – Страница регистрации

## Описание работы системы

Данная система для онлайн голосования предназначена для определения победителя путем использования ранжирующих процедур голосования.

На странице администратора есть пять вкладок. Это текущее голосование, предварительные голосования, закрытые голосования, вкладка требует решения и пользователи.

Во вкладке текущее голосование (рисунок 3.6) отображается какое голосование открыто в данный момент (определяется статусом), также отображается дата окончания голосования, по истечении которого, система самостоятельно закрывает голосование и проводит вычисления по каждой процедуре. Также на этой странице отображается кнопка «скопировать ссылку», в ней содержится актуальная ссылка на голосования для распространения. Здесь также содержится название голосования и собственно само голосование.

Во вкладке предварительное голосование (рисунок 3.7) отображается созданное голосование. Имеется возможность создания нового голосования (рисунок 3.9). Здесь необходимо заполнить такие поля, как название голосования, краткое описание, ожидаемое количество голосующих (предоставляется заказчиком голосования), а также добавить кандидатов, в которых заполняются такие данные, как ФИО, дата рождения, должность, образование, учебные заведения, краткое описание, возможность добавления предвыборной кампании и фото. При нажатии кнопки «действия» имеется возможность открыть голосование, после нажатия которой, необходимо выбрать дату окончания голосования. Также при нажатии кнопки можно редактировать голосование и удалить его.

Во вкладке закрытые голосования (рисунок 3.12) отображаются все прошедшие голосования. Там отображаются названия голосований, победитель, а также имеется возможность скачать результаты. Пример файла результатов приведен в Приложении А.

Во вкладке требует решения (рисунок 3.10) отображается голосование, время проведения которого уже закончилось, но решение еще не принято. Также в данной вкладке отображается таблица с результатами голосования по каждой ранжирующей процедуре определения победителя. Система после завершения голосования генерирует таблицу с результатами, а также файл – отчет о результатах голосования, в котором помимо победителя указывается процент проголосовавших и легитимность голосования. Пример файла-отчета приведен в Приложении А. Файл-отчет доступен после нажатия кнопки «скачать результаты. pdf». После чего администратор направляет файл – отчет заказчику, он в свою очередь определяет победителя, исходя из результатов голосования. Администратор после того, как заказчик сообщит ему свое решение, выбирает победителя, путем нажатия кнопки «принять решение» (рисунок 3.11), после чего результаты выводятся на «домашнюю» страницу (рисунок 3.3). Алгоритм работы модуля отчета приведен в Приложении Б.

Во вкладке пользователи (рисунок 3.8) администратор добавляет пользователей в систему, список которых предоставил ему заказчик голосования. Администратор загружает данные пользователей в систему, она генерирует идентификационные номер для каждого пользователя, затем он скачивает файл и отправляет заказчику. Для доступа к голосованию каждому пользователю необходимости ввести уникальный идентификационный номер при регистрации, чтобы система проверила подлинность голосующего. После проверки пользователь авторизуется и уже имеет возможность голосовать. После регистрации пользователей в системе на этой вкладке будет отображаться, кто из пользователей зарегистрировался, а кто нет.

В системе реализована защита от повторной регистрации и повторного голосования одним пользователем. В системе обработаны случаи ошибок при введении некорректных данных в процессе голосования (например, при расстановке приоритетов нельзя использовать один и тот же приоритет или поставить число, превышающее количество голосующих), а также при регистрации, если введен некорректный пароль, идентификационный номер. Также при входе в систему обработчик ошибок сообщит о неверно введенных логине и/или пароле. При успешной регистрации система автоматически авторизует пользователя и перебросит на начальную страницу голосования (рисунок 3.3). В данной системе реализовано шифрование паролей и данных пользователей. Метод шифрования SHA -1. Тип авторизации Bearer – token. Тип token – JWT. Он представляет собой строку, полученную на основе формата JSON, и используется в качестве более безопасной и простой альтернативы для авторизации. Приложение использует JWT для проверки аутентификации пользователя следующим образом [36]:

1. Пользователь заходит на сервер аутентификации с помощью логина и пароля.
2. Затем сервер аутентификации создает *JWT* и отправляет его пользователю.
3. Когда пользователь делает запрос к API приложения, он добавляет к нему полученный ранее *JWT*.
4. Когда пользователь делает API запрос, приложение может проверить по переданному с запросом *JWT* является ли пользователь тем, за кого себя выдает.

Также, неавторизированные пользователи не могут голосовать, что тоже является защитой информации голосования.

Пользователь может просматривать начальную страницу голосования (рисунок 3.3), там у него имеется возможность регистрации (рисунок 3.14), либо же авторизации (рисунок 3.13), если был ранее зарегистрирован. На странице регистрации пользователю необходимо заполнить логин, пароль, повторный пароль (система автоматически проверит их совпадение, если его не будет – выведет ошибку), а также уникальный идентификационный номер. После этого пользователю остается только проголосовать. При успешном голосовании система выведет соответствующее сообщение. У пользователя имеется возможность просматривать профиль кандидата (рисунок 3.5). Перед голосованием пользователь может зайти в профиль каждого кандидата, ознакомиться с личными данными, а также скачать предвыборную кампанию кандидатов.

## «Ручное» вычисление результатов голосования

### Процедура Борда

Предположим, что N=81 избирателей должны выбрать одного из трех кандидатов А, В и С. Пусть при использовании ранжирующей процедуры учета мнения избирателей их индивидуальные предпочтения распределились следующим образом: A > B > C – 21

A > C > B – 11

B > A > C – 17

B > C > A – 8

C > A > B – 9

C > B > A – 15

Вычислим значение функции Борда для каждого из кандидатов, полагая, что в индивидуальном упорядочении первому кандидату присваивается 2 балла, второму — 1 балл, третьему — 0 баллов. Тогда получаем результаты, вычисленные по формуле (1.1):

Упорядочение кандидатов в соответствии со значениями функции Борда имеет вид: A > B > C, поскольку , согласно формуле (1.2).

Лучшим является кандидат A.

### Модифицированная процедура Борда

Решение процедура рассмотрим на исходных данных п. 3.4.1:

A > B > C – 21

A > C > B – 11

B > A > C – 17

B > C > A – 8

C > A > B – 9

C > B > A – 15

Вычислим количество голосов, предпочитающих кандидата А кандидату В и наоборот:

g(A,B) = 21+11+9 = 41, g(B,A) = 17+8+15 = 40;

Аналогично:

g(A,C) = 21+11+17 = 49, g(C,A) = 8+9+15 = 32;

g(B,C) = 21+17+8 = 46, g(C,B) = 11+9+15 = 35.

Вычислим значение модифицированной функции Борда, согласно формуле (1.3):

Упорядочение кандидатов в соответствии со значениями модифицированной процедуры Борда имеет вид: A > B > C, поскольку , согласно формуле (1.4).

Лучшим является кандидат А.

### Процедура Кондорсе

В пункте 3.4.2 вычислили:

g(A,B) = 21+11+9 = 41, g(B,A) = 17+8+15 = 40;

g(A,C) = 21+11+17 = 49, g(C,A) = 8+9+15 = 32;

g(B,C) = 21+17+8 = 46, g(C,B) = 11+9+15 = 35.

g(A,B) > g(B,A), следовательно имеет место предпочтение A > B. Аналогично, получаем A > C, B > C. Объединяя, получается согласованное коллективное предпочтение A > B >C.

### Процедура Симпсона

В пункте 3.4.2 вычислили:

g(A,B) = 21+11+9 = 41, g(B,A) = 17+8+15 = 40;

g(A,C) = 21+11+17 = 49, g(C,A) = 8+9+15 = 32;

g(B,C) = 21+17+8 = 46, g(C,B) = 11+9+15 = 35.

Вычисляем функции Симпсона для альтернатив, согласно формуле (1.5):

По формуле (1.6), вычисляем:

A > B >C. Лучшим является кандидат А.

### Процедура Доджсона

Имеем результаты попарного сравнения (пункт 3.4.2):

g(A,B) = 21+11+9 = 41, g(B,A) = 17+8+15 = 40;

g(A,C) = 21+11+17 = 49, g(C,A) = 8+9+15 = 32;

g(B,C) = 21+17+8 = 46, g(C,B) = 11+9+15 = 35.

Строим вспомогательную матрицу H (таблица 3.2) согласно hij = g(Ai , Aj)/ g(Aj , Ai)

Таблица 3.2 – Вспомогательная матрица H

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| A | 1 | 41/40 | 49/32 |
| B | 40/41 | 1 | 46/35 |
| C | 32/49 | 35/46 | 1 |

Вычисляем процедуру Доджсона для альтернатив при g1/2=(t+1)/2=41:

В матрице Доджсона в первой строке нет элементов, принадлежащих множеству А1/2, поэтому кандидат А является безусловным победителем – он при всех попарных сравнениях.

По формуле (1.7), вычисляем:

A > B >C. Лучшим является кандидат А.

### Процедура Нансона

Исходное множество альтернатив A = {A,B,C} найдено в пункте 3.4.1:

Исключаем кандидата С, согласно формуле (1.8). Исходная задача преобразуется в новое множество кандидатов А1, согласно формуле (1.9):

A > B – 21

A > B – 11

B > A – 17

B > A – 8

A > B – 9

B > A – 15

Находим новые значения функции Борда:

Кандидат В, имеющий наименьшее значение функции Борда, исключается. Новое множество кандидатов А2 состоит из единственного кандидата А, который и является лучшим.

Упорядочение кандидатов в соответствии с функцией Нансона имеет вид A > B > C.

### Модифицированная процедура Нансона

По формуле (1.10), вычисляем:

Соответственно на первом шаге исключаем кандидата С.

Исходная задача преобразуется:

A > B – 21

A > B – 11

B > A – 17

B > A – 8

A > B – 9

B > A – 15

Находим новые значения функции Борда:

По формуле (1.10), вычисляем:

Исключаем кандидата В.

A > B >C. Лучшим является кандидат А.

### Процедура Кумбса

При ранжирующей процедуре учета мнения избирателей их индивидуальные предпочтения распределились согласно пункту 3.4.1:

A > B > C – 21

A > C > B – 11

B > A > C – 17

B > C > A – 8

C > A > B – 9

C > B > A – 15

Определим, сколько человек считают худшими тех или иных кандидатов.

Кандидата А считают худшим: 8+15 = 23 человека,

Кандидата В считают худшим: 11+9 = 20 человек,

Кандидата С считают худшим: 21+17 = 38 человек.

Так как кандидат С набрал больше всего «худших» голосов, то его исключаем.

Исходная задача преобразуется:

A > B – 21

A > B – 11

B > A – 17

B > A – 8

A > B – 9

B > A – 15

Аналогично, определим, сколько человек считают худшими тех или иных кандидатов.

Кандидата А считают худшим: 17+8+15 = 40 человек,

Кандидата В считают худшим: 21+11+9 = 41 человек

Так как кандидат В набрал больше всего «худших» голосов, то его исключаем. В итоге остается кандидат В – победитель. Итоговое предпочтение A > B >C.

### Процедура Коупленда

Имеем результаты попарного сравнения (пункт 3.4.2):

g(A,B) = 21+11+9 = 41, g(B,A) = 17+8+15 = 40;

g(A,C) = 21+11+17 = 49, g(C,A) = 8+9+15 = 32;

g(B,C) = 21+17+8 = 46, g(C,B) = 11+9+15 = 35.

Тогда, согласно формуле (1.11), вычисляем:

Следовательно A > B >C, согласно формуле (1.12).

### Процедура Фишберна

Имеем результаты попарного сравнения (пункт 3.4.2):

g(A,B) = 21+11+9 = 41, g(B,A) = 17+8+15 = 40;

g(A,C) = 21+11+17 = 49, g(C,A) = 8+9+15 = 32;

g(B,C) = 21+17+8 = 46, g(C,B) = 11+9+15 = 35.

Тогда, согласно формуле (1.13), вычисляем:

Следовательно A > B >C, согласно формуле (1.14).

Обобщение вычисленных результатов по каждой из процедур рейтингового голосования приведены в таблице 3.3.

Таблица 3.3 – Результаты вычислений

|  |  |  |
| --- | --- | --- |
| № | Процедура | Результат |
| 1 | Борда | A > B >C |
| 2 | Доджсона | A > B >C |
| 3 | Модифицированного Борда | A > B >C |
| 4 | Кондорсе | A > B >C |
| 5 | Нансона | A > B >C |
| 6 | Симпсона | A > B >C |
| 7 | Кумбса | A > B >C |
| 8 | Фишберна | A > B >C |
| 9 | Модифицированного Нансона | A > B >C |
| 10 | Коупленда | A > B >C |

## Сравнение «ручных» вычислений с результатами работы системы

****На рисунке 3.12 приведены результаты работы системы.

Рисунок 3.12 – Результаты работы системы

Данные работы системы соответствуют следующим обозначениям при «ручном» вычислении: A – Суханова Юлия Глебовна, В – Колпачёв Владимир Агапович, С – Егоров Фадей Куприянович. Сравнение результатов, полученных при «ручном» вычислении и работе системы приведены в таблице 3.3.

Таблица 3.3 – Сравнение полученных результатов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Процедура | «Ручной» результат | Результат системы |
| 1 | Борда | A > B >C | A > B >C |
| 2 | Доджсона | A > B >C | A > B >C |
| 3 | Модифицированного Борда | A > B >C | A > B >C |
| 4 | Кондорсе | A > B >C | A > B >C |
| 5 | Нансона | A > B >C | A > B >C |
| 6 | Симпсона | A > B >C | A > B >C |
| 7 | Кумбса | A > B >C | A > B >C |
| 8 | Фишберна | A > B >C | A > B >C |
| 9 | Модифицированного Нансона | A > B >C | A > B >C |
| 10 | Коупленда | A > B >C | A > B >C |

Из таблицы 3.3 видно, что результаты работы системы и «ручные» вычисления полностью совпадают, из этого следует то, что алгоритмы подсчета результатов по каждой процедуре при работе системы были правильно составлены.

# Технико-экономическое обоснование

Данная автоматизированная система позволит решить проблемы дороговизны и трудоемкости процесса голосования. А также исключить всевозможные ошибки, которые могут возникнуть из-за человеческого фактора. Также позволит повысить «явку» избирателей, поскольку не нужно идти в избирательный участок (который может быть далеко и неудобно расположен), а можно проголосовать в удобном месте и в удобное время.

На данный момент в мире существуют подобные системы. Рассмотрим самые распространенные аналоги.

Polys – система безопасных онлайн-голосований. Эта система основа на технологии блокчейн и прозрачных криптоалгоритмов. При создании голосования с помощью данной системы можно голосовать с любого электронного устройства (ноутбук, смартфон и другие) [20].

Система I-VOTE является комплексным системно-интегрированным решением, которое широко используется при проведении международных и национальных конгрессов, школ обучения специалистов, подготовки и обучения менеджеров [23].

Electionrunner – это система онлайн голосования, позволяющая голосовать с любого электронного устройства, привлекающая своим дизайном и простой использования интерфейса. Система предоставляет каждому избирателю уникальный идентификационный номер, по которому он может проголосовать только один раз. Интересной особенностью данной системы является то, что под свое голосование вы можете скорректировать дизайн веб-страницы или мобильного приложения. Безопасность выборов обеспечивает 256 битным SLL шифрованием. Также результаты голосования система формирует в виде текстового документа, а затем отображает в виде красочных диаграмм [24].

Произведём сравнительный анализ следующих аналогов в таблице 4.1.

Таблица 4.1 – Сравнительный анализ аналогов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерии/название системы | Система онлайн голосования | I-vote | Polys | Electionrunner |
| Наличие СЗИ | Да | Да | Да | Да |
| Методы определения победителя | Ранжирующие | Неранжирующие | Неранжирующие | Неранжирующие |
| Максимальное количество кандидатов | 200 | 50 | 100 | 50 |

Основывая на данные, указанные в таблица 4.1, справедливо сделать вывод, что разрабатываемая система лучше остальных по указанным критериям.

На этапе проектирования системы по каждому этапу разработки разрабатываемой системы нужно оценить его предполагаемое время работы. Для их оценки приведены данные в таблице 4.2.

Таблица 4.2 – Продолжительность работ на этапе проектирования

|  |  |  |
| --- | --- | --- |
| Наименование работ | Длительность работ (дни) | Исполнитель |
| Разработка ТЗ | 5 | Аналитик |
| Анализ ТЗ | 5 | Аналитик |
| Поиск и изучение литературы | 7 | Аналитик |
| Анализ предметной области | 5 | Аналитик |
| Проектирование модели БД и архитектуры приложения | 14 | Аналитик  Программист |
| Составление алгоритма разработки | 10 | Программист |
| Разработка, отладка, тестирование | 20 | Программист |
| Написание документации | 5 | Программист |

Сведем продолжительность всех работ в график (рисунок. 4.1).

Рисунок 4.1. График продолжительности работ

К затратам на проектирование и разработку системы (Зс) относятся:

* Материальные затраты (Зм);
* Затраты на электроэнергию (Зэ);
* Затраты на оплату труда (Зт);
* Отчисления на социальные нужды (Зо);
* Амортизация оборудования (За);
* Прочие затраты (Зпр).

Формула для расчёта затрат на проектирование и разработку системы

Зс = Зм + Зэ + Зт + Зо + За + Зпр. (4.1)

Для создания автоматизированной системы потребуется ноутбук, например, [LENOVO S145-15IWL 81MV018FRK](https://www.citilink.ru/catalog/mobile/notebooks/1175033/).

Материальные затраты отражены в таблице 4.3.

Таблица 4.3 – Материальные затраты

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Цена за ед., руб | Количество | Стоимость, руб |
| Lenovo S145-15IWL 81MV018FRK | 35 000 | 1 | 35 000 |
| Итого: | | | 35 000 |

Затраты на оплату электроэнергии отражены в таблице 4.4. Расчет будет основан на плате за электроэнергию для физических лиц, в качестве устройства выступает: Lenovo S145-15IWL 81MV018FRK.

Таблица 4.4 – Затраты на оплату электроэнергии

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование оборудования | Потребляемая мощность, кВТ | Кол-во, шт | Время работы, часы | Цена электроэнергии за 1 кВт/ч, руб | Стоимость, руб |
| Lenovo S145-15IWL 81MV018FRK. | 0,2 | 1 | 568 | 4,98 | 565,7 |
|  |  |  |  | Итого: | 565,7 |

Основным видом затрат при разработке автоматизированной системы являются оплата труда.

Оплата труда рассчитывается по формуле:

(4.2)

Где

* – дневная заработная плата специалиста.
* – отчисления на социальные нужды (30%).
* – месячная заработная плата разработчика.
* – рабочие дни.

Рассчитаем затраты на оплату труда аналитика:

𝐿о = 60 000 руб.

𝐿д = 60 000/23 = 2 608 руб.

𝑇п = 36 дней.

𝐿п = 2 608 ∗ 36 ∗ (1 + 30/100) = 122 054 руб.

Рассчитаем затраты на оплату труда программиста:

𝐿о = 70 000 руб.

𝐿д = 70 000/23 = 3 043 руб.

𝑇п = 49 дней.

𝐿п = 3 043 ∗ 49 ∗ (1 + 30/100) = 193 839 руб.

В итоге выделенная заработанная плата на разработку автоматизированной системы равна 315 893 руб. Полученная сумма сложилась в результате работы двух человек, участвовавших в разработке (программист и аналитик). Время работы аналитика составило 36 дней, а у программиста 49 дней. Совокупность затрат на разработку ПО представлены в таблице 4.5. Данные о зарплате аналитика и программиста были взяты из источника – "hh.ru".

Таблица 4.5 – Смета затрат на разработку ПО

|  |  |
| --- | --- |
| Статьи затрат | Сумма, руб. |
| 1. Материальные затраты | 35 000 |
| 1. Затраты на электроэнергию | 565,7 |
| 1. Затраты на оплату труда, включая страховые взносы | 315 893 |
| Итого: | ≈351 459 |

Определим предполагаемый объем реализации. Автоматизированная система предназначена для оптимизации процесса голосования. А также определения победителя с помощью ранжирующих процедур голосования.

Суммарные затраты на создание 1 копии ПП (Зед) это отношение суммарных затрат на создание (Зп), ноутбук (Зноутбук) к прогнозируемому объему реализации (Q = 50):

Зед= (Зп +Зноутбук) : Q (4.3)

Оценку уровня качества разработанного программно-аппаратного комплекса находят по направлениям, указанным в таблице 4.6.

Таблица 4.6 – Бальная оценка параметров программного комплекса и аналога

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр, | Весовой | Разработка | | Аналог | |
| оценка | коэфф. аi | biн | Значимость 06-r69 | Число баллов06-r70 | Значимость 06-r71 |
| Надежность | 0.3 | 6 | 1,8 | 5 | 1,5 |
| Эффективность | 0.2 | 5 | 1,0 | 4 | 0,8 |
| Удобство использования | 0.1 | 3 | 0,3 | 6 | 0,6 |
| Простота использования | 0.2 | 6 | 1,2 | 4 | 0,8 |
| Стабильность | 0.1 | 5 | 0,5 | 3 | 0,3 |
| Итого | 06-r72 | 06-r73 | 06-r74 | 06-r75 | 06-r76 |
| Коэффициент качества | Кк=∑ai biн / ∑aibia = 1,125 | | | | |

Коэффициент качества оказался равен 1,125 (таблица 4.6) и он больше 1, это означает, что целесообразно разрабатывать данную автоматизированную систему.

Текущие затраты разработки и аналога сравнимы, поэтому исключаются из последующих расчетов. Стоимость самого дешевого прямого аналога составляет 15 000 рублей.

Коэффициент цены потребления (таблица 4.7) вычисляется как отношение интегрального стоимостного показателя нового ПП к интегральному стоимостному показателю аналога, то есть:

Кэ = Ip / Ia = 7067 / 15050 = 0,46

Где

* Кэ – коэффициент цены потребления.
* Iр – стоимостной показатель разработки.
* Iа – стоимостной показатель аналога.

Таблица 4.7 – Вычисление коэффициента цены потребителя

|  |  |  |
| --- | --- | --- |
| Наименование статьи калькуляции | Аналог (Ia)  Сумма, руб. | Разработка (Ip) Сумма, руб. |
| Единовременные затраты | 15 000 | 7017 |
| Текущие затра­ты на эксплуатацию изделия | 50 | 50 |
| Итого, интегральный стоимостный показатель (цена потребления) | 15 050 | 7067 |
| Коэффициент цены потребления, Кэ=Ip/Ia | 0,46 | |

Представим расчёт показателя сравнительной технико-экономической эффективности в таблице 4.8.

Таблица 4.8 – Итоговое сравнение технико-экономической эффективности разработанной системы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Параметр, | Весовой | Разработка | | | Аналог | |
| оценка | коэфф. аi | biн | Значимость 06-r69 | | Число баллов06-r70 | Значимость 06-r71 |
| Надежность | 0.3 | 6 | 1,8 | | 5 | 1,5 |
| Эффективность | 0.2 | 5 | 1,0 | | 4 | 0,8 |
| Удобство использования | 0.1 | 3 | 0,3 | | 6 | 0,6 |
| Простота сопровождения | 0.2 | 6 | 1,2 | | 4 | 0,8 |
| Стабильность | 0.1 | 5 | 0,5 | | 3 | 0,3 |
| Интегральный техн. показатель, Q |  |  |  | |  |  |
| Коэффициент качества | Кк=Qp/Qa = 1,125 | | | | | |
| Интегральный стоимостной показатель, I | = 7067 | | | = 15 050 | | |
| Коэффициент цены потребления | Кэ = Ip / Ia = 0,46 | | | | | |
| Сравнительная технико-эконом. эффективность, Эср | Эср = Кк / Кэ = 1,125 / 0,46 = 2,45 | | | | | |

В результате технико-экономического сравнения выяснилось, что разрабатываемая в дипломной работе информационная система целесообразна с точки зрения экономической эффективности. Сводный результат приведен в приложении Г.

В данной информационной системе присутствует большой функционал, в то тоже время являясь более дешевым, по сравнению с аналогами.

# Безопасность человеко-машинного взаимодействия

Для оценки влияния тех или иных факторов на организм при работе с разработанным программным продуктом, был проведен анализ напряженности труда согласно [37]. Результаты анализа представлены в таблице 5.1.

Таблица 5.1 – Оценка напряженности трудового процесса оператора

|  |  |  |  |
| --- | --- | --- | --- |
| Фактор | Фактическое значение | Нормативное значение | Класс условий труда |
| *I. Интеллектуальные нагрузки:* | | | |
| Восприятие сигналов (информации) и их оценка | Для того чтобы добавить информацию, оператор должен заполнить соответствующие поля в базе данных и сохранить изменения | Восприятие сигналов с последующей комплексной оценкой связанных па­раметров | Напряженный труд 2 степени.  3.2 |
| *II. Сенсорные нагрузки:* | | | |
| Длительность сосредоточенного наблюдения (в % от времени смены) | Т.к. информационная система не требует следить за процессом выполнения, то длительность сосредоточенного наблюдения не больше 25% от рабочей смены | 51–75 | Напряженный труд 1 степени.  3.1 |
| Наблюдение за экраном монитора (часов в смену): | Администратор следит за процессом голосования | 1–2 | Напряженный труд 2 степени.  3.2 |
| *III. Эмоциональные нагрузки:* | | | |
| Степень ответственности за собственную ошибку | Администратор несет ответственность за правильное оформление информации о голосовании | Несет ответственность за функциональное качество основной работы (задания). | Напряженный труд 1 степени.  3.1 |
| *IV. Монотонность нагрузок:* | | | |

Продолжение таблицы 5.1

|  |  |  |  |
| --- | --- | --- | --- |
| Число элементов  (приемов), необходимых для реализации простого задания или в многократно повторяющихся операциях | Для входа в систему/поиска/расчета/ добавления нужно ввести в поля для ввода данные несколько раз. Итого: 1– 5 приемов | 5-3 | Напряженный труд 1  степени  3.1 |
| *V. Режим работы:* | | | |
| Фактическая продолжительность рабочего дня | Рабочий день составляет 8 часов | 8—9ч | Напряженность труда средней степени  2 |
| Сменность работы | Односменная работа | Односменная работа (без ночной смены) | Напряженность труда средней степени  1 |
| Общая оценка условий труда | | | 3.1 |

К категории «напряженных» относятся факторы, принимающие значение 3.1 и выше. У нас, принимающих значение 3.1 и выше, не больше пяти. Следовательно, уровень напряженности труда клиента является допустимым.

Допустимые условия труда (класс 2) характеризуются такими уровнями факторов среды и трудового процесса, которые не превышают установленных гигиенических нормативов для рабочих мест. Возможные изменения функционального состояния организма восстанавливаются во время регламентированного отдыха и не должны оказывать неблагоприятного действия на состояние здоровья работающих людей. Допустимые условия труда условно относят к безопасным, поэтому обязательным является соблюдение оператором защитных и профилактических мероприятий [37].

В современном мире компьютерная техника имеет широкое применение во всех отраслях производства, поэтому необходимо организовать безопасность работы персонала. Для этого нужно следовать следующим правилам [37]:

* при расположении компьютеров по периметру линии светильников должны располагаться над каждым рабочим столом ближе к его переднему краю, обращенному к оператору. Если компьютеры в комнате расположены в ряд, то общее освещение следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест параллельно линии зрения пользователя;
* площадь на одно рабочее место с компьютером для пользователей должна составлять не менее 4,5 кв.м при работе с ЖК-монитором, а объем не менее 20,0 куб.м.;
* экран монитора должен находиться от глаз пользователя на оптимальном расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.
* рабочие кресла и столы должны быть отрегулированы по высоте в соответствии с потребностями конкретного пользователя. Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм. Стул должен надежно фиксировать положение сидящего человека, особенно спину;
* для нормального функционирования организму человека требуется чистый воздух, поэтому в рабочих помещениях нужно следить за чистотой и регулярно проветривать помещение, использовать увлажнители воздуха, если это необходимо;
* температура в помещении должна быть оптимальной. Поэтому нужно следить за ее значением и ставить при необходимости обогреватели или кондиционеры. Не менее необходимым является правильная организация режима работы и степени нагрузки персонала в соответствии с психологическими нормами:
* работа не должна быть монотонна, т.е. человек не должен выполнять только один вид работы, а наоборот, его деятельность в течение суток должна быть разнообразной, насколько это возможно.
* при работе с буквенной и графической информацией интерфейс должен быть интуитивно понятен, яркость должна быть не выше нормы, цвета не раздражающие, а также подходящий шрифт;
* продолжительность непрерывной работы с ПК не должна превышать 2 часов. При 8-часовой рабочей смене следует устанавливать перерывы через 1,5-2 часа от начала рабочей смены и после обеденного перерыва продолжительностью 15-20 минут каждый.

Все эти факторы существенно влияют на работоспособность и настроение человека, поэтому пренебрежение этими мерами могут привести к снижению работоспособности персонала, к частым ошибкам, утомлению и проблемам со здоровьем

Перед эксплуатацией помещений, в которых будет работать пользователь системы, необходимо выяснить какие существуют источники воспламенения и причины пожара или взрыва. Наиболее вероятными причинами возникновения пожара является причины электрического характера: короткое замыкание, перегрузки, искрение.

В результате короткого замыкания, а также при плохом контакте на клеммах возникают искры, которые могут привести к возгоранию оборудования и рабочих станций. Перегрузка и короткое замыкание в сети электрического тока могут вызвать возгорание изоляции сетевых кабелей.

Также причиной возгорания может стать халатность оператора при использовании зажигательных устройств в помещении. Согласно классификации по пожарной опасности [38] помещение относится к категории «D». Категория «D» включает производства, связанные с применением негорючих веществ и материалов, находящихся в холодном состоянии. По [39] помещение можно отнести к II степени огнестойкости, т. к. все конструкции (стены, перекрытия) выполнены из негорючих материалов с пределом огнестойкости от 0.25 до 2.5 часов.

В помещении, где работает пользователь, размещен огнетушитель углекислотный ОУ-5, предназначенный для тушения электрооборудования, находящегося под напряжением не свыше 1000 В. Также установлена система пожарной сигнализации с применением тепловых датчиков типа ДПС-038. В коридоре здания расположен пожарный кран в 5 метрах от офиса. Имеется план эвакуации на случай пожара. На основании вышеизложенного необходимо реализовать следующие методы противопожарной безопасности [39]:

1) для самостоятельной борьбы с пожаром в помещении использовать ручной огнетушитель ОУ-5;

2) для уменьшения вероятности возникновения пожара проводить технические мероприятия;

3) для уменьшения потерь от пожара проводить организационные мероприятия: противопожарный инструктаж, разработку плана эвакуации на случай пожара; недопущение использования дополнительных электроприборов, обогревательных приборов; запрещение курения в неразрешенных местах.

Проведение регулярной проверки работоспособности элементов системы автоматической пожарной сигнализации, а также средств тушения пожара, имеющихся в помещении.

В данном разделе выпускной квалификационной работы бы проведен анализ безопасности человеко-машинного взаимодействия для администратора системы, были выявлены правила поведения и пользования оборудованием в помещении. Были даны рекомендации по улучшению условий труда, при несоблюдении которых может понизиться работоспособность пользователя. Так как работа с системой осуществляется при помощи компьютера, то были проведены мероприятия по улучшению пожаробезопасности. Подводя итог, можно сказать, что система не наносит вреда ни экологии, ни пользователям системы. Сводный результат приведен в приложении В.

**ЗАКЛЮЧЕНИЕ**

В данной работе был проведен анализ системы онлайн голосования с использованием процедур рейтингового голосования. Описана актуальность и необходимость создания таких систем. Она будет включать в себя все функции традиционного метода голосования. Данная система будет предоставлять средства для оперативной и точной обработки данных голосования, подсчет голосов, определения победителя с помощью использования рейтинговых процедур.

В процессе проектирования информационной системы был сделан проект базы данных с указанными сущностями и атрибутами. Также был составлен проект общей архитектуры и функциональной структуры информационной системы. Разработан интерфейс веб – приложения.

В процессе разработки системы были созданы два типа пользователей: администратор и пользователь, которые имеют различный набор функций. Пользователь может регистрироваться, авторизовываться, голосовать, просматривать результаты голосования и профили кандидатов, а администратор создавать, открывать, редактировать голосования. Также администратор может добавлять и удалять пользователей. Был разработан интерфейс системы, представленный на рисунках 3.3 – 3.14.

Проведен сравнительный анализ «ручных» вычислений результатов голосования и работы системы. Они полностью совпадают, что говорит и правильной реализации алгоритмов работы системы.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Голосов, Г.В. Выборы и электоральная политика [Текст] : словарь / Г. В. Голосов, К. В. Киселев, О. Б. Подвинцев [и др.]; гл. ред. Г. В. Голосов ; ред. К. В. Киселев, О. Б. Подвинцев. — СПб.: Геликс, 2010. — 159 с.
2. Toplak, J. [Preferential Voting: Definition and Classification](https://web.archive.org/web/20110717210259/http:/rcum.uni-mb.si/~jure/preferential8_emlist.pdf) [Text] / J. Toplak // [Lex Localis](https://www.researchgate.net/journal/1581-5374_Lex_Localis). – 2017. – Vol. 15, no. 4. – P. 5–20.
3. Presidential elections 1938 – 2011 [Text] // Enviroment, Community and Local Government. – 2013. – P. 5-17, 31-32.
4. Прохоров, А.М. Рейтинговое голосование [Текст] : в 30 т. / [А. М. Прохоров](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%85%D0%BE%D1%80%D0%BE%D0%B2,_%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80_%D0%9C%D0%B8%D1%85%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D0%B8%D1%87) // [Большая советская энциклопедия](https://ru.wikipedia.org/wiki/%D0%91%D0%BE%D0%BB%D1%8C%D1%88%D0%B0%D1%8F_%D1%81%D0%BE%D0%B2%D0%B5%D1%82%D1%81%D0%BA%D0%B0%D1%8F_%D1%8D%D0%BD%D1%86%D0%B8%D0%BA%D0%BB%D0%BE%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F#%D0%A2%D1%80%D0%B5%D1%82%D1%8C%D0%B5_%D0%B8%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5). — 3-е изд. — М. : Сов. энцикл., 1969 – 1978.
5. Байчорова, М.М. Современные процедуры голосования [Текст] / М.М. Байчорова, П.В. Агеев, Н.Э. Комарова // Международный научный журнал «Молодой ученый». – 2016. – №14(18). – С.9–11.
6. Kumar, R. A Framework for Ranking Reviews Using Ranked Voting Method [Text] / R. Kumar, A. Sharan, C.S. Yadav // Proceedings of the Second International Conference on Computer and Communication Technologies. – 2015. – Vol. 1. – P. 263–272.
7. Baranwal, G. A framework for selection of best cloud service provider using ranked voting method [Text] / G. Baranwal, D.P. Vidyarthi // IEEE International Advance Computing Conference (IACC). –2014. – P. 831–837.
8. Евровидение [Электронный ресурс]. – Режим доступа : URL: <https://eurovision.tv/about/rules/> (дата обращения : 21.04.2020).
9. McLean [Australian electoral reform and two concepts of representation](http://www.nuffield.ox.ac.uk/Politics/papers/2002/w23/mclean.pdf) [Text] / McLean // [Australian Journal of Political Science](https://www.researchgate.net/journal/1363-030X_Australian_Journal_of_Political_Science). – 2002. – Vol. 31. – P. 11.
10. Harold, J. The Political Consequences of the Alternative Vote: Lessons from Western Canada [Text] / J. Harold // Canadian Journal of Political Science. – 2004. – Vol. 37, no. 3. – P. 647–669.
11. Novák, V. [The Greens have tested a non-lost electoral system that supports broad consensus](http://data.blog.ihned.cz/c1-59414520-zeleni-otestovali-volebni-system-bez-ztracenych-hlasu-ktery-podporuje-sirokou-shodu) [Электронный ресурс] / V. Novák. – Режим доступа: URL : data.blog.ihned.cz (дата обращения 21.02.2020).
12. [Functional Constituencies: The Preferential Elimination System of the 4 SFCs](http://www.eac.gov.hk/pdf/legco/2012lc_guide/en/chapter_3.pdf) [Электронный ресус] *//* **Election Briefs***. –* Режим доступа: URL : <https://www.eac.hk/en_txt/legco/fc.html> (дата обращения: 21.02.2020).
13. Президент Индии [Электронный ресурс]. – Режим доступа: URL: http://swapsushias.blogspot.com/2010/10/presidents-of-indiarashtrapati (дата обращения: 21.04.2020).
14. Городской совет Веллингтона [Электронный ресурс]. – Режим доступа: URL : <https://wellington.govt.nz/your-council/elections/electoral-systems> (дата обращения: 21.04.2020).
15. Избирательная комиссия Папуа Новая Гвинея [Электронный ресурс]. – Режим доступа: URL: <http://aceproject.org/ero-en/regions/pacific/PG/Papua_new_guinea_leaflet.pdf/view> (дата обращения: 21.04.2020).
16. [BBC News [Электронный ресурс] //Alternative vote](https://www.bbc.com/news/uk-politics-12910547). – Режим доступа: URL: <http://news.bbc.co.uk/2/hi/uk_news/politics/8506306.stm> (дата обращения: 3.05.2020).
17. Arntz, J. [Ranked-Choice Voting: A Guide for Candidates](https://web.archive.org/web/20081202040611/http:/fairvote.org/media/irv/sanfrancisco/RCVCandidateGuide04.pdf) [Text] / J. Arntz. – 2005.
18. Центральная избирательная комиссия Российской Федерации [Электронный ресурс]. – Режим доступа: URL: <http://www.cikrf.ru/reception/dlya-otdelnykh-kategoriy-izbirateley/mobilnii-izbiratel/> (дата обращения: 17.12.2019).
19. Habr [Электронный ресурс]. – Режим доступа: URL: <https://habr.com/ru/article/480152/> (дата обращения: 20.12.2019).
20. Polys [Электронный ресурс]. – Режим доступа: URL: <https://polys.me/ru> (дата обращения: 18.12.2019).
21. Домсканнер [Электронный ресурс]. – Режим доступа: URL: <http://domscanner.ru/article/18> (дата обращения: 20.12.2019).
22. ИС «Кворум» [Электронный реcурс]. – Режим доступа: URL: <https://www.rrost.ru/ru/about/kvorum/> (дата обращения: 20.12.2019).
23. ИС «I-vote» [Электронный ресурс]. – Режим доступа: URL: <https://www.i-vote.ru/> (дата обращения : 21.12.2019).
24. Electionrunner [Электронный ресурс]. – Режим доступа: URL: <https://electionrunner.com/> (дата обращения: 21.12.2019).
25. Покшиванов, Д.О. [Создание системы для онлайн-опросов и голосований](https://elibrary.ru/item.asp?id=37060143) / Д.О. Покшиванов // [Материалы XIX Всероссийской научно-практической конференции молодых ученых, аспирантов и студентов](https://elibrary.ru/item.asp?id=37060065). Секции 1-5. — Нерюнгри : Изд-во Технического института (ф) СВФУ, 2018. – С. 254-257.
26. Alhawawsha, M. [Developing of the e-government system based on java for online voting](https://elibrary.ru/item.asp?id=29424844) / M. Alhawawsha // Technological audit and production reserves. – 2017.– Vol. 3, no. 2 (35). – P. 9-13.
27. Emerson, P. The original Borda count and partial voting. Social Choice and Welfare [Text] / P. Emerson. –2011. – Vol. 40(2) – P. 353–358.
28. McLean, I. The borda and condorcet principles: Three medieval applications. Social Choice and Welfare [Text] / I. McLean. –1990. – Vol.7, no. 2. – P. 99–108.
29. Nurmi, H. Voting Procedures: A Summary Analysis [Text] / H. Nurmi //British Journal of Political Science. –1983.– Vol. 13, no. 2.– P. 181.
30. Грищенко, А.С. Исследование методов построения процедур манипулирования данными в структурно-независимых базах данных / А.С. Грищенко // Известия ЮФУ. Технические науки. – 2015. – Т. 11, вып. 172. – С. 149-157.
31. Грищенко, А.С. Формальная модель метода совершенствования процедуры вставки данных в структурно-независимых базах данных / А.С. Грищенко // Технологии разработки информационных систем: сборник статей VIII Международной научно-практической конференции ТРИС-2017. – Таганрог: Изд-во ЮФУ, 2017. – С. 175-180.
32. Леоненков, А. Самоучитель UML / А. Леоненков. – М.: Дрофа, 2006. – С. 576.
33. Иванов, Д. Моделирование на UML / Д. Иванов, Ф. Новиков. – М.: Наука и техника, 2010. – С. 640.
34. Буч, Г. UML 2.0 / Г. Буч, А. Якобсон, Дж. Рамбо. – СПб.: Питер, 2006. – С. 735.
35. Алексеев, Ю. М. Быстро и легко создаем, программируем, шлифуем и раскручиваем web-сайт:/ Ю.М. Алексеев. – М.: Лучшие Книги, 2006.-400 с.
36. Руководство Node.js [Электронный ресурс]. – Режим доступа : URL : <https://xsltdev.ru/nodejs/tutorial/jwt/> (дата обращения : 15.05.2020).
37. Р 2.2.2006-05. Гигиена труда. Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда [Текст]. – Введено взамен [Р 2.2.755-99](http://docs.cntd.ru/document/1200004531). введ. 01.11.2005.
38. СНиП 31-03-2001. Строительные нормы и правила российской федерации производственные здания строительные нормы и правила Российской Федерации. Производственные здания [Текст]. – М.: Госстрой России, ГУП ЦПП, 2001. – 8 с.
39. СНиП 2.01.02-85\*. Строительные нормы и правила  
    Противопожарные нормы [Текст]. – М: Госстрой СССР, 1991.

# ПРИЛОЖЕНИЕ А

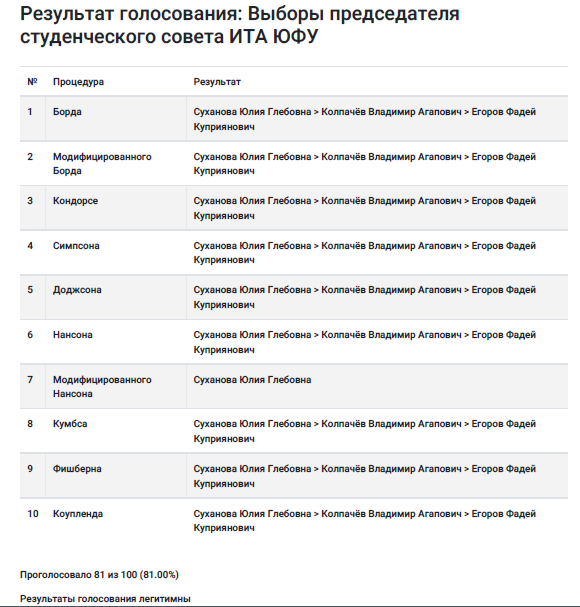
**Отчет с результатами голосования**

Рисунок Б.1 – Файл-отчет с результатами голосования

# ПРИЛОЖЕНИЕ Б

**Блок – схема генерации отчетов**



# ПРИЛОЖЕНИЕ В

**Плакат БЧМВ**



# ПРИЛОЖЕНИЕ Г

**Плакат ТЭО**



# ПРИЛОЖЕНИЕ Д

**Вычисление результатов ранжирующих процедур голосования**

Процедура Борда

const fromatResult = require("./fromatResult");

module.exports = async result => {

 const resultBoard = {};

 for (const item of result) {

   const candidates = item.result.split(";");

   for (const [index, candidat] of candidates.entries()) {

     if (!resultBoard[candidat]) resultBoard[candidat] = 0;

     resultBoard[candidat] += (candidates.length - index - 1) \* item.countvote;

   }

 }

 const formatRes = await fromatResult(resultBoard);

 return formatRes;

};

Процедура Доджсона

const formatResult = require("./fromatResult");

module.exports = async (preResultMBord, result) => {

 const tmpDod = { ...preResultMBord };

 let countPVote = 0;

 result.forEach(item => (countPVote += +item.countvote));

 const c = (countPVote % 2 === 0 ? countPVote : countPVote + 1) / 2;

 useCandidates = [];

 const resDog = {};

 for (const item in tmpDod) {

   const [candidat, b] = item.split(";");

   if (useCandidates.includes(candidat)) continue;

   const firstCandidate = [];

   Object.keys(tmpDod).forEach(itemCandidat => {

     if (itemCandidat.split(";")[0] === candidat)

       firstCandidate.push({ [itemCandidat]: tmpDod[itemCandidat] });

   });

   let reflection = 1;

   Object.keys(tmpDod).forEach(item => {

     if (item === `${b};${candidat}`) reflection = tmpDod[`${b};${candidat}`];

   });

   if (!resDog[candidat]) resDog[candidat] = 0;

   if (tmpDod[item] / reflection < 1) {

     resDog[candidat] += c - tmpDod[item];

   }

 }

 for (const item in resDog) {

   if (resDog[item] === 0) resDog[item] = Infinity;

 }

 const formatRes = await formatResult(resDog);

 return formatRes;

};

Процедура Фишберна

const formatResult = require("./fromatResult");

module.exports = async preResultMBord => {

 const result = {};

 for (const item of Object.keys(preResultMBord)) {

   const [a, b] = item.split(";");

   if (!result[a]) result[a] = 0;

   if (preResultMBord[item] > preResultMBord[`${b};${a}`]) result[a]++;

 }

 return formatResult(result);

};

Процедура Коупленда

const formatResult = require("./fromatResult");

module.exports = async preResultMBord => {

 const result = {};

 for (const item of Object.keys(preResultMBord)) {

   const [a, b] = item.split(";");

   if (!result[a]) result[a] = 0;

   if (preResultMBord[item] > preResultMBord[`${b};${a}`]) result[a]++;

   else result[a]--;

 }

 const max = Math.max(...Object.values(result));

 let countMax = 0;

 for (const item of Object.values(result)) {

   if (item === max) countMax++;

 }

 if (countMax > 1) return "Предпочтение не выявлено";

 return formatResult(result);

};

Процедура Кумбса

const formatResult = require("./fromatResult");

module.exports = async result => {

 let preResult = {};

 let resultK = {};

 let newList = result.map(item => {

   const copy = JSON.stringify(item);

   return JSON.parse(copy);

 });

 let i = -1;

 do {

   i++;

   preResult = {};

   for (const item of newList) {

     const sp = item.result.split(";");

     const lastC = sp[sp.length - 1];

     if (!preResult[lastC]) preResult[lastC] = 0;

     preResult[lastC] += +item.countvote;

   }

   const maxArr = [];

   const max = Math.max(...Object.values(preResult));

   for (const item of Object.keys(preResult)) {

     if (preResult[item] === max) {

       maxArr.push(item);

       resultK[item] = i;

     }

   }

   newList = newList.map(item => {

     const oldKey = item.result.split(";");

     const key = oldKey.filter(key => !maxArr.includes(key)).join(";");

     return { ...item, result: key };

   });

 } while (Object.keys(preResult).length > 1);

 return formatResult(resultK);

};

Модифицированная процедура Борда

const formatResult = require("./fromatResult");

module.exports = async preResultMBoard => {

 const resultMBord = {};

 for (const item in preResultMBoard) {

   const [resItem, b] = item.split(";");

   if (!resultMBord[resItem]) resultMBord[resItem] = 0;

   const reflection = preResultMBoard[`${b};${resItem}`] || 0;

   resultMBord[resItem] += +preResultMBoard[item] - reflection;

 }

 const formatRes = await formatResult(resultMBord);

 return formatRes;

};

Модифицированная процедура Нансона

const formatResult = require("./fromatResult");

module.exports = async result => {

 let newList = result.map(item => {

   const copy = JSON.stringify(item);

   return JSON.parse(copy);

 });

 let preResult = {};

 do {

   preResult = {};

   let sum = 0;

   for (const item of newList) {

     const candidates = item.result.split(";");

     for (const [index, candidat] of candidates.entries()) {

       if (!preResult[candidat]) preResult[candidat] = 0;

       preResult[candidat] +=

         (candidates.length - index - 1) \* +item.countvote;

     }

   }

   for (const item of Object.keys(preResult)) {

     sum += preResult[item];

   }

   const average = sum / Object.keys(preResult).length;

   const min = [];

   Object.keys(preResult).forEach(item => {

     if (preResult[item] < average) min.push(item);

   });

   newList = newList.map(item => {

     const oldKey = item.result.split(";");

     const key = oldKey.filter(key => !min.includes(key)).join(";");

     return { ...item, result: key };

   });

 } while (Object.keys(preResult).length > 1);

 return formatResult(preResult);

};

Процедура Нансона

const formatResult = require("./fromatResult");

module.exports = async result => {

 let newList = result.map(item => {

   const copy = JSON.stringify(item);

   return JSON.parse(copy);

 });

 const countCandidates = result[0].result.split(";").length;

 let resultNan = {};

 for (let i = 0; i < countCandidates; i++) {

   const preResult = {};

   for (const item of newList) {

     const candidates = item.result.split(";");

     for (const [index, candidat] of candidates.entries()) {

       if (!preResult[candidat]) preResult[candidat] = 0;

       preResult[candidat] +=

         (candidates.length - index - 1) \* +item.countvote;

     }

   }

   let min = { "1": Infinity };

   for (const item in preResult) {

     if (preResult[item] < Object.values(min)[0]) {

       min = { [item]: preResult[item] };

     }

   }

   min = { [Object.keys(min)]: i };

   resultNan = { ...resultNan, ...min };

   newList = newList.map(item => {

     const [remove] = Object.keys(min);

     const oldKey = item.result.split(";");

     const key = oldKey.filter(item => item !== remove).join(";");

     return { ...item, result: key };

   });

 }

 return formatResult(resultNan);

};

Процедура Симпсона

const formatResult = require("./fromatResult");

const min = array => {

 let [min] = Object.values(array[0]);

 for (const item of array) {

   const [value] = Object.values(item);

   if (value < min) min = value;

 }

 return min;

};

module.exports = async preResultMBord => {

 const tmpSimpson = { ...preResultMBord };

 let useCandidates = [];

 let resultSimpson = {};

 for (const item in tmpSimpson) {

   const [candidat] = item.split(";");

   if (useCandidates.includes(candidat)) continue;

   useCandidates.push(candidat);

   const firstCandidate = [];

   Object.keys(tmpSimpson).forEach(itemCandidat => {

     if (itemCandidat.split(";")[0] === candidat)

       firstCandidate.push({ [itemCandidat]: tmpSimpson[itemCandidat] });

   });

   let minItem = min(firstCandidate);

   resultSimpson[candidat] = minItem;

 }

 const formatRes = await formatResult(resultSimpson);

 return formatRes;

};