

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра «Измерительно-вычислительные комплексы»

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на лабораторную работу № 8
по дисциплине «Алгоритмы и структуры
данных»

Тема программа «Расчет дискретных систем»

Р.02069337.№23/690-№20 ТЗ-01

Листов 5

Исполнитель:
студент гр. ИСТбд-22
Минибаева Е.Р.
«_____»_____2024 г.

2024 г.

Введение

Наименование приложения: Расчасовка дисциплин

Условное обозначение: РД-1.0

Программа «Расчасовка дисциплин» будет предоставлять следующие функциональные возможности: хранение, анализ и визуализации данных о дисциплинах в учебном заведении.

1. Основания для разработки

Учебный план включает ключевые дисциплины, связанные с информационными системами, что подтверждает необходимость разработки проектов, соответствующих современным требованиям научных и практических задач. Проект предоставит студентам возможность интегрировать теорию с практикой, применяя полученные знания в области программирования, проектирования и архитектуры систем. Разработка будет способствовать развитию у студентов ключевых профессиональных компетенций, таких как:

Способность создания (модификации) и сопровождения информационных систем (ИС), автоматизирующих задачи организационного управления и бизнес-процессы в организациях различных форм собственности с целью повышения эффективности деятельности организаций -пользователей ИС.

Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО.

2. Требования к программе или программному изделию

2.1.Функциональное назначение

Программа "Расчасовка дисциплин" предназначена для хранения, анализа и визуализации данных о дисциплинах в учебном заведении.

2.2. Требования к функциональным характеристикам

2.2.1. Требования к структуре приложения

Приложение должно быть организовано по модульному принципу для обеспечения гибкости, расширяемости и легкости в поддержке.

Каждой модуль должен иметь четкие интерфейсы для взаимодействия с другими модулями.

Модули должны быть независимыми и легко заменяемыми.

Четкая модульная структура позволит эффективно разрабатывать, тестировать и поддерживать приложение, а также упростит его обновление в дальнейшем.

2.2.2.Требования к составу функций приложения

Основные функции приложения:

1) Хранение данных:

Программа должна хранить информацию о расчасовке дисциплин, включая:

- Название дисциплины
- Вид учебной нагрузки
- Имя преподавателя дисциплины
- Данные должны сохраняться в текстовом файле "discipline.txt"

в формате: Название дисциплины, Вид учебной нагрузки, Имя преподавателя дисциплины

2) Анализ данных:

Программа должна предоставлять возможность анализа данных по следующим критериям:

Сегментация по преподавателям:

- Отображать список преподавателей и количество ведомых ими дисциплин
- Визуализировать результаты сегментации в виде круговой диаграммы

Сегментация по преподавателям:

- Отображать список учебных нагрузок и количество дисциплин относящихся к ним.
- Визуализировать результаты сегментации в виде круговой диаграммы

3) Добавление данных:

- Предоставлять возможность добавления новой информации о дисциплине с помощью удобного интерфейса.

4) Обновление данных:

- Позволять обновлять информацию о существующих дисциплинах.

5) Удаление данных:

- Обеспечивать возможность удаления данных о дисциплине.

6) Визуализация данных:

- Отображать результаты анализа в виде диаграммы с помощью библиотеки matplotlib.

2.2.3. Требования к организации информационного обеспечения, входных и выходных данных

Данные должны сохраняться в текстовом файле " discipline.txt" в формате: Название дисциплины, Вид учебной нагрузки, Имя преподавателя дисциплины.

Данные должны вводиться пользователем с клавиатуры в текстовом формате (поля для добавления новой информации о дисциплине).

В приложении должен быть реализован графический интерфейс взаимодействия с пользователем. Графические элементы интерфейса должны быть хорошо видимыми и интуитивно понятными. Сообщения об ошибках и уведомления должны быть ясными, содержать информацию о

причине и возможных действиях для пользователя. Должны отображаться в формате всплывающих окон.

2.3. Требования к надежности

Программа должна работать стабильно и не выдавать ошибок при нормальных условиях эксплуатации. Время загрузки интерфейса не должно превышать 3 секунд на стандартных устройствах, чтобы обеспечить плавный пользовательский опыт. Программа должна быть способна восстанавливаться после сбоев или ошибок, не теряя данные и сохраняя работоспособность. Программа должна обеспечивать целостность и достоверность данных, используемых и обрабатываемых ею. Программа должна быть защищена от несанкционированного доступа, модификации или уничтожения данных.

2.4. Требования к информационной и программной совместимости

Язык программирования: Python

Инструментальная среда - Любая IDE, поддерживающая Python 3.x - например, Visual Studio Code, PyCharm, IDLE, Thonny и т.д.

Операционная система - Windows, macOS, Linux - приложение должно работать на всех этих операционных системах, так как tkinter доступен во всех из них.

Библиотеки:

- tkinter - для создания графического интерфейса;
- ttk - для стилизации элементов интерфейса;
- matplotlib - для визуализации данных.

База данных:

Программа не использует базы данных, данные хранятся в текстовом файле.

Интерфейс пользователя:

- Программа должна иметь интуитивно понятный графический интерфейс с:
 - Кнопками для сегментации по преподавателям и видам учебной;

- Кнопкой для добавления новой информации о дисциплине;
- Кнопкой для обновления списка дисциплин;
- Кнопкой для удаления информации о дисциплине;
- Областью для отображения данных о дисциплине;
- Областью для отображения диаграммы.

2.5. Требования к транспортированию и хранению

2.5.1. Условия транспортирования

Требования к условиям транспортирования не предъявляются

2.5.2. Условия хранения

Обеспечение свободного доступа к проекту в репозитории до окончания срока учебы.

2.5.3. Сроки хранения

Срок хранения – до окончания срока учебы.

3. Требования к программной документации

Программа должна быть снабжена комментариями, поясняющими логику работы кода.

4. Сроки выполнения

Срок выполнения проекта: 2 недели.

5. Порядок контроля и приемки

Программа должна выполнять все описанные функциональные требования.

Программа должна иметь стабильный и эффективный код.

Программа должна быть снабжена необходимой документацией.