**Руководство программиста.**

**Руководство программиста на разработку программы "Библиотечные книги".**

**1. Обзор программы**

Программа представляет собой графический интерфейс (GUI) для управления списком библиотечных книг. Она позволяет:

* Загружать данные о библиотечных книгах из файла "123.txt".
* Сохранять данные о библиотечных книгах в файл "123.txt".
* Выполнять сегментацию библиотечных книгах по учебным дисциплинам и частотой использования.
* Отображать результаты сегментации в виде диаграммы.

**2. Структура программы**

Программа состоит из трех классов:

1. Book: Представляет собой объект факультатива. Имеет следующие атрибуты:

* title: Название библиотечной книги.
* Discipline: Название дисциплины.
* Usage: Частоту использования книги

1. Library: Управляет списком книг и их сегментацией. Имеет следующие методы:

* Load\_books(): Загружает книги из CSV-файла и проверяет правильность формата данных.
* Init (): Инициализирует пустой список книг
* Segment\_by\_discipline(): Сегментирует книги по дисциплинам, возвращая словарь с количеством книг для каждой дисциплины.
* Segment\_by\_usage(): Сегментирует книги по частоте использования, возвращая словарь с количеством книг для каждой частоты использования.
* Visualize\_data(): Создает круговую диаграмму на основе переданных данных и заголовка.

с) LibraryApp: Отвечает за создание и управление графическим интерфейсом.

* Center\_window(): Располагает окно приложения в центре экрана.
* Load\_books(): Загружает книги из указанного файла.

**3. Алгоритм работы программы**

1. Инициализация:

* При запуске программы создается экземпляр класса Book.
* Book загружает данные из файла "123.txt".
* LibraryApp создает GUI программы.

2. Взаимодействие пользователя:

* Пользователь может нажать на кнопки «Сегментация по дисциплинам», «Сегментация по использованию».

3. Обработка данных:

* При нажатии на кнопки «Сегментация по дисциплинам» или «Сегментация по использованию» вызывается соответствующий метод Library (segment\_by\_discipline() или segment\_by\_usage())
* Методы segment\_by\_discipline() и segment\_by\_usage() возвращают словарь с количеством книг для каждой дисциплины и частоты их использования.
* Результаты сегментации используются для создания диаграммы.

4. Отображение результатов:

* Результаты сегментации отображаются в виде диаграммы.

**4. Описание кода**

Laba 8.py

import tkinter as tk  
from tkinter import messagebox  
import matplotlib.pyplot as plt  
import csv  
from collections import defaultdict  
  
1. Класс Book:

* Метод init(self, title, discipline, usage);
* Инициализирует объект Book с заданными параметрами.
* Метод str(self):
* Возвращает строковое представление объекта Book.

2. Класс Library:

* Метод init(self):
* Инициализирует список Book.
* Метод load\_books(self, filename):
* Загружает данные о факультативах из файла "123.txt".
* Обрабатывает возможные ошибки, например, чтобы каждая строка содержала 3 значения.
* Метод segment\_by\_discipline (self):
* Выполняет сегментацию по дисциплинам.
* Возвращает словарь с количеством дисциплин для каждого студента.
* Метод segment\_by\_usage(self):
* Выполняет сегментацию по использованию.
* Возвращает словарь с частотой использования книги для каждого преподавателя.
* Метод visualize\_data(self, data, title):
* Принимает данные в виде словаря
* Устанавливает название графика.
* Устанавливает характеристики для диаграмм.

3. Класс LibraryApp:

* Метод init(self, master):
* Создает главное окно root.
* Устанавливает заголовок окна.
* Определяет размеры окна.
* Создаются кнопки «Загрузить книги», «Сегментации данных по дисциплинам» и «Сегментации данных по использованию».
* Вызывает load\_books, segment\_by\_discipline, segment\_by\_usage
* Метод сenter\_window():
* Располагает окно приложения в центре экрана.
* Метод load\_books():
* Загружает данные о книгах из файла
* Метод def segment\_by\_discipline(self):
* Вызывает segment\_by\_discipline() из класса Library, для отображения результатов.
* Метод segment\_by\_usage(self):
* Вызывает segment\_by\_usage() из класса Library, для отображения результатов.

**5. Рекомендации по использованию**

* Программа использует файл "123.txt" для хранения данных. Файл должен быть создан вручную и содержать информацию о книгах в формате:

(Название книги, Дисциплина, Частота использования(часто, средне, редко, нкиогда))

* Для добавления новой книги необходимо отредактировать файл с информацией о дисциплине.
* Для сегментации по дисциплинам или частоте использования достаточно нажать соответствующую кнопку.

**6. Дополнительные возможности**

* Добавить возможность удаления книг.
* Добавить возможность редактирования данных о книгах.
* Добавить возможность фильтрации списка книг.
* Добавить возможность вывода результатов сегментации в файл.
* Добавить возможность использования других типов диаграмм.

**7. Тестирование**

* Тестирование программы должно включать:
* Проверку загрузки данных из файла.
* Проверку сохранения данных в файл.
* Проверку добавления новых книг.
* Проверку сегментации по дисциплинам и частоте использования.
* Проверку корректности отображения результатов сегментации.

**8. Документация**

* Данное руководство программиста является основным документом по программе.
* Дополнительную документацию можно добавить в виде комментариев к коду.
* Важно использовать стандартные соглашения по наименованию переменных и функций.
* Документация должна быть четкой, понятной и доступной для пользователя.

**9. Замечания**

* Программа использует библиотеку tkinter для создания GUI.
* Программа использует библиотеку matplotlib для создания диаграмм.
* Программа использует кодировку utf-8 для чтения и записи данных из файла.

**10. Дополнительные замечания**

* Программа может быть расширена и улучшена.
* Программа может быть использована как основа для создания более сложных приложений.
* Важно следовать принципам модульности, повторного использования кода и ясности написания кода.