



Instituto Tecnológico e Profissional da Figueira da Foz

2023/2024

Curso Programador Informático

Registrador de faltas

Autor: Raquel Leal

2024

Índice

Introdução	3
Capítulo A	4
Capítulo B	4
Capítulo C	4
Conclusão	5

Introdução

O presente trabalho é sobre uma estrutura de código para um registrador em Python, mais concretamente sobre um código que mostra as funcionalidades do registrador.

Objetivo do trabalho é mostrar numa estrutura de código, onde tem alguns tópicos das coisas que demos durante as aulas, mostrando o que temos aprendido, com a linguagem de programação Python.

O trabalho está organizado em 3 partes e uma conclusão. Na primeira parte vamos falar sobre as funcionalidades do registrador, na segunda parte vamos falar sobre o código, e na terceira parte explicar a estrutura do código e as suas bibliotecas.

Como funciona o Registrador?

O registrador é simples, pois ainda não acrescentei outras funcionalidades. O registrador tem algumas funcionalidades como o menu (adicionar faltas, visualizar faltas, contar faltas, remover faltas, salvar e carregar registros), e um programa para sair. No registrador também podes adicionar mais de um aluno e os motivos pelas faltas. O registrador é útil mas ainda falta melhorar.

Estrutura do Código

O código está estruturado para criar uma interface gráfica para o utilizador , usando algumas bibliotecas como o 'tkinter'.

Está estruturado da seguinte forma:

1º com os métodos necessários (bibliotecas):

2º uma mensagem de boas vindas:

3º classes 'RegitroDeFaltas':

4º Métodos da classes 'RegistroDeFaltas':

adicionar faltas;

visualizar faltas;

contar faltas;

remover faltas;

salvar registro;

carregar registro;

5º Função para interface gráfica:

adicionar faltas;

remover faltas;

6º Função Principal 'main':

Inicializa a classe 'RegistroDefalats'.

Define o arquivo de registros.

Configura e inicia a interface gráfica.

Explicação do Código

import json

import tkinter as tk

from tkinter import messagebox, simpledialog

'json': Utilizado para salvar e carregar dados em formato JSON.

'tkinter': Biblioteca padrão do Python para criar interfaces gráficas.

'messagebox e simpledialog': Submódulos do **tkinter** que fornecem caixas de mensagem e de diálogo simples.

Classe RegistroDeFaltas

- **__init__**: Inicializa a classe RegistroDeFaltas com um dicionário vazio **self.faltas** para armazenar as faltas dos alunos.
- Métodos:
 - **adicionar_falta(nome, motivo)**: Adiciona uma falta para um aluno específico.
 - **visualizar_faltas()**: Exibe todas as faltas registradas.
 - **contar_faltas()**: Conta e exibe o número de faltas por aluno.
 - **remover_falta(nome, indice)**: Remove uma falta específica de um aluno com base no índice fornecido.
 - **salvar_registros(arquivo)**: Salva os registros de faltas em um arquivo JSON especificado.
 - **carregar_registros(arquivo)**: Carrega os registros de faltas de um arquivo JSON especificado.

Funções de Interface Gráfica

- **adicionar_falta_interface()**: Função para interagir com o usuário e adicionar uma falta através de caixas de diálogo **simpledialog**.
- **remover_falta_interface()**: Função para interagir com o usuário e remover uma falta através de caixas de diálogo **simpledialog**.

Função Principal **main()**

- **main()**: Função principal que cria a instância **registrador** da classe **RegistroDeFaltas**, inicializa a interface gráfica (**tkinter**), e define os botões para

cada funcionalidade (adicionar falta, visualizar faltas, contar faltas, remover falta, salvar registros, carregar registros e sair).

Funcionamento Geral

- A aplicação inicia com uma janela principal onde o usuário pode interagir através de botões para adicionar, visualizar, contar e remover faltas.
- Os dados são armazenados e recuperados de um arquivo JSON (`registros_de_faltas.json`) usando os métodos `salvar_registros` e `carregar_registros`.
- Mensagens informativas (messagebox) são exibidas para feedback ao usuário após cada operação.

'self' : utilizado para chamar métodos de instância.

Conclusão

Este trabalho foi muito importante para o meu conhecimento, visto que permitiu-me conhecer melhor a linguagem de programação Python, além de ter-me aperfeiçoado competências de investigação, selecção e organização. O trabalho foi um pouco complicado, pois não sabia os métodos necessários, mas tive algumas ajudas. Pretendo melhorar, pois o trabalho parece ser muito útil. Espero que o trabalho seja eficaz para Professor.