

”

E-fólio A | Folha de resolução para E-fólio

UNIDADE CURRICULAR: Programação por Objetos

CÓDIGO: 21093

DOCENTES: Jorge Morais, Leonel Morgado e Rúdi Gualter (tutor)

A preencher pelo estudante

NOME: Bruno Ricardo de Sá Ferreira

N.º DE ESTUDANTE: 2201529

CURSO: Engenharia Informática

DATA DE ENTREGA: 14 de Novembro de 2023

Apresentação do projeto: Aplicativo de apostas estilo casino/euromilhões.

É criada uma app do estilo casino em que o usuário pode inserir os números que quer apostar, e verificar na hora se teve a sorte de ganhar algum prêmio. O objetivo é criar uma pequena máquina de apostas fácil de usar, quer pelos usuários, quer para futuros administradores que tenham acesso a ela, que permite fazer atualizações de valores de prêmios, mudar ou acrescentar regiões, de forma segura sem precisar fazer grandes alterações ao código, ou criar novos arquivos que permitam guardar todo o historial de apostas e resultados gerado pela máquina, afim de auferir por exemplo a probabilidade de um usuário ganhar os prêmios.

Usei então o tkinter, para criar uma interface simples, que no decorrer do semestre pode ser melhorada tornando-a mais atrativa, com imagens personalizadas e cores de fundo mais atraentes para melhor experiência do usuário.

Funcionalidades da App

Interface de apostas com números sorteados e possíveis erros a ser exibidos, calcula prêmios, exibe resultados, salva todas as entradas da app num ficheiro a parte, permite configurar por região, abandonar o jogo e reiniciar o jogo, salvar aposta e conferir sorteio.

Todas as 8 classes:

Classe `ApostaInterface`, cria uma interface para que o usuário.

Métodos:

- `__init__`: Inicia todos os atributos da classe e configura a sua interface(construtor de classe em todas as classes).
- `salvar_aposta`: Validar a entrada do usuário.
- `resetar_apostas`: Limpa a entrada de texto do usuário, reativa o botão “salvar aposta”, desativa o botão conferir.

Atributos:

- `frame`, `labe`, `entry`, `button`: Widgets do Tkinter.
- `aposta_reader`: Usado para salvar as apostas.
- `data_storage`: Usado para armazenar as apostas e resultados
- `conferir_button`: Objeto do Tkinter Button, usado para verificar os resultados.
- `numeros_sorteados_interface`: Usado para exibir os números sorteados.
- `contador_apostas`: Mantém o controle do número de apostas feitas. (1 aposta).

Classe [DataStorage](#), gerencia o armazenamento de dados do jogo de loteria.

Atributos:

- self._arquivo: É uma variável privada que armazena o nome do arquivo onde os dados serão salvos.
- Setter e getter para garantir que apenas arquivos de textos sejam usados. A definir na class MegaSenaApp

Métodos:

- salvar_aposta: Salva a aposta do usuário no arquivo.
- salvar_resultado: Este método salva o resultado de um sorteio no arquivo.
- limpar_arquivo: Redefine o jogo. (Jogar novamente)
- Getter e setter: Ligados a class MegasennaApp afim de criar restrições na criação de ficheiro de armazenamento.

Classe [ApostaReader](#) gerencia as apostas feitas pelo usuário.

Atributos:

- self.data_storage: É um objeto da classe DataStorage que é usado para armazenar as apostas e os resultados.
- self.apostas: Lista que armazena todas as apostas feitas pelo usuário.

Métodos:

- salvar_aposta: Salvar a aposta.

Classe [NumeroConferente](#), verifica as apostas do usuário e calcular os prêmios.

Atributos:

- self.apostas: Lista que armazena todas as apostas feitas pelo usuário.
- self.__premio_4_acertos, self.__premio_3_acertos, self.__premio_2_acertos, self.__premio_1_acerto: São variáveis privadas que armazenam os prêmios.

Propriedades:

premio_4_acertos, premio_3_acertos, premio_2_acertos, premio_1_acerto: Setters e getters para alterar o valor dos prêmios com integridade na [classe região](#).

Métodos:

- conferir: Método usado para verificar a aposta do usuário contra os números sorteados.
- A classe NumeroConferente trabalha em conjunto com as classes [ApostaReader](#), [DataStorage](#) e [Regiao](#).

Classe [ResultadoDisplay](#) exibe os resultados das apostas na interface gráfica do usuário.

Atributos:

- self.frame,self.label,self.text: objetos do Tkinter.

Métodos:

- mostrar_resultados: Exibe os resultados das apostas.

Classe NumerosSorteadosInterface gerencia a interface gráfica que exibe os números sorteados na loteria.

Atributos:

- self.frame,self.label,self.sorteio_entry, self.conferir_button:Objetos do Tkinter para trabalhar com a interface.
- self.numero_conferente: Objeto da classe NumeroConferente, usado para verificar as apostas.
- self.data_storage: Objeto da classe DataStorage que é usado para armazenar as apostas e os resultados.
- self.resultado_display: Objeto da classe ResultadoDisplay que é usado para exibir os resultados das apostas.
- self._aposta_feita: É uma variável booleana que indica se uma aposta foi feita.

Propriedades:

- aposta_feita: Propriedade que fornece um getter e um setter para a variável privada _aposta_feita. O setter verifica se o valor fornecido é um booleano. Apostas são apenas validadas com valores booleanos. True or False.

Métodos:

- conferir: Verifica a aposta do usuário. Ele gera os números sorteados, verifica a aposta do usuário, salva o resultado e atualiza a interface gráfica.
- gerar_sorteito: Gera os números sorteados e retorna uma lista de quatro números únicos de 1 a 60.
- aposta_realizada: Método usado para indicar que uma aposta foi feita. (True).
- resetar_interface: Método que limpa a entrada de texto, desativa o botão “Conferir” e reativa o botão “Salvar Aposta”.

Classe Regiao é usada para representar uma região específica na aplicação, torna viável os métodos get e set.

Atributos:

- self.nome: É uma string que armazena o nome da região.
- self.premio_4_acertos, self.premio_3_acertos, self.premio_2_acertos, self.premio_1_acerto: São variáveis que armazenam os prêmios para diferentes números de acertos.

Métodos:

- configurar_premios: Este método define os prêmios para diferentes números de acertos.

Classe MegaSenaApp classe principal que coordena todas as outras classes e gerencia a aplicação como um todo.

Atributos:

- self.root: É a janela principal da aplicação Tkinter.
- self.data_storage: Objeto da classe DataStorage que é usado para armazenar as apostas e os resultados.
- self.aposta_reader: Objeto da classe Apostareader que é usado para salvar as apostas.
- self.numero_conferente: Objeto da classe NumeroConferente que é usado para verificar as apostas.
- self.resultado_display: Objeto da classe ResultadoDisplay que é usado para exibir os resultados das apostas.
- self.conferir_button: Objeto do Tkinter Button que é usado para verificar os resultados.
- self.numeros_sorteados_interface: Objeto da classe NumerosSorteadosInterface que é usado para exibir os números sorteados.
- self.aposta_interface: Objeto da classe Apostainterface que é usado para aceitar as apostas do usuário.
- self.botao_jogar_novamente: Objeto do Tkinter Button que é usado para reiniciar o jogo.
- self.botao_abandonar_jogo: Objeto do Tkinter Button que é usado para encerrar o jogo.

Métodos:

- jogar_novamente: Método usado para reiniciar o jogo.
- abandonar_jogo: Método para abandonar o jogo.
- run: Método é usado para iniciar o loop principal da aplicação Tkinter.

Por fim criamos `if __name__ == "__main__":` onde é criada uma instância de MegaSenaApp, configuramos a área, e fazemos run a app.