

Gerenciador_de_Fauna

Version

Tabela de Conteúdo

Contents:

source	4
• animal module	4
• avl_tree module	6
• manager module	8
• ui module	10

Welcome to Gerenciador_de_Fauna's documentation!

source

animal module

class animal. Animal (*id* , *apelido* , *inicio_monitoramento* , *especie* , *sexo* , *data_nascimento* , *historico*=<animal.Historico object>) [código fonte]

Base: `object`

Representa um animal e seu histórico de avaliações.

Parâmetros :

- *id* (*int*) – Identificação do animal.
- *apelido* (*str*) – Apelido do animal.
- *inicio_monitoramento* (*str*) – Data de início do monitoramento.
- *especie* (*str*) – Espécie do animal.
- *sexo* (*str*) – Sexo do animal.
- *data_nascimento* (*str*) – Data de nascimento do animal.
- *historico* (*Historico*) – Histórico de avaliações do animal.

add_log (*log*) [código fonte]

Adiciona um novo registro de avaliação ao histórico do animal.

Parâmetros :

log (*Registro*) – Registro a ser adicionado.

convert_to_dictionary () [código fonte]

Converte os dados do animal e seu histórico em um dicionário.

Retorna :

Dicionário com os dados do animal e seu histórico.

Tipo de retorno :

dict

`print_historico ()` [código fonte]

Exibe o histórico de avaliações do animal em formato de tabela.

`print_info ()` [código fonte]

Exibe as informações do animal e seu histórico (se houver).

`class animal. Historico (logs = None)` [código fonte]

Base: `object`

Representa o histórico de avaliações de um animal.

Parâmetros :

`logs (list)` – Lista de registros de avaliação. Por padrão, inicia como uma lista vazia.

`add_log (log)` [código fonte]

Adiciona um novo registro ao histórico.

Parâmetros :

`log (Registro)` – Registro a ser adicionado.

`convert_to_dictionary ()` [código fonte]

Converte o histórico em uma lista de dicionários.

Retorna :

Lista de dicionários com os dados dos registros.

Tipo de retorno :

list

`get_info ()` [código fonte]

Retorna a lista de registros no histórico.

Retorna :

Lista de registros.

Tipo de retorno :

list

class animal. Registro (*data_avaliacao* , *temperatura* , *peso* , *altura* , *amostra* , *exame* , *problema_de_saude*) [\[código fonte\]](#)

Base: `object`

Representa um registro de avaliação de um animal.

Parâmetros :

- *data_avaliacao* (*str*) – Data da avaliação.
- *temperatura* (*float*) – Temperatura medida durante a avaliação.
- *peso* (*float*) – Peso do animal.
- *altura* (*float*) – Altura do animal.
- *amostra* (*bool*) – Indicador se foi coletada uma amostra.
- *exame* (*bool*) – Indicador se o exame foi aprovado.
- *problema_de_saude* (*str*) – Descrição do problema de saúde (se houver).

convert_to_dictionary () [\[código fonte\]](#)

Converte o registro em um dicionário.

Retorna :

Dicionário com os dados do registro.

Tipo de retorno :

dict

avl_tree module

class avl_tree. ArvoreAVL [\[código fonte\]](#)

Base: `object`

Implementa uma árvore AVL para armazenar e gerenciar objetos *Animal* .

A árvore AVL é uma árvore de busca binária balanceada, que garante que as operações de inserção, remoção e consulta sejam feitas em tempo $O(\log n)$.

atualizar (*chave_antiga* , *animal_atualizado*) [código fonte]

Atualiza um nó na árvore AVL substituindo o nó com a chave antiga por um nó com o novo animal.

Essa função primeiro remove o nó com a chave antiga e depois insere o nó atualizado.

Parâmetros :

- *chave_antiga* (*int*) – A chave (ID do animal) que deve ser substituída.
- *animal_atualizado* (*Animal*) – O novo animal que será inserido no lugar do antigo.

consultar (*chave*) [código fonte]

Consulta um nó na árvore AVL com base na chave.

Parâmetros :

chave (*int*) – A chave (ID do animal) a ser consultada.

Retorna :

O nó correspondente ao animal ou None se não encontrado.

Tipo de retorno :

Node or None

inorder_traversal (*raiz*) [código fonte]

Percorre a árvore AVL em ordem (in-order) e imprime as informações dos nós.

Parâmetros :

raiz (*Node*) – O nó raiz da subárvore a ser percorrida.

inserir (*animal*) [código fonte]

Insere um novo animal na árvore AVL.

Parâmetros :

animal (*Animal*) – O animal a ser inserido.

remove (*chave*) [código fonte]

Remove um nó da árvore AVL com base na chave do animal.

Parâmetros :

chave (*int*) – A chave (ID do animal) a ser removida.

class *avl_tree*. **Node** (*animal*) [código fonte]

Base: *object*

Representa um nó de uma árvore AVL.

Parâmetros :

animal (*Animal*) – Objeto da classe *Animal* que será armazenado no nó.

manager module

manager.adicionar_animal (*arvore*) [código fonte]

Adiciona um novo animal à árvore AVL, solicitando as informações ao usuário.

Se o ID fornecido já existir, o usuário será solicitado a escolher outro ID.

Parâmetros :

arvore (*ArvoreAVL*) – A árvore AVL onde o animal será adicionado.

manager.adicionar_registro (*arvore* , *chave*) [código fonte]

Adiciona um novo registro ao histórico de um animal na árvore AVL.

Parâmetros :

- **arvore** (*ArvoreAVL*) – A árvore AVL onde o animal está armazenado.
- **chave** (*int*) – O ID do animal ao qual o registro será adicionado.

manager.consultar_animal (*arvore* , *chave*) [código fonte]

Consulta um animal na árvore AVL pelo ID e exibe suas informações.

Parâmetros :

- **arvore** (*ArvoreAVL*) – A árvore AVL onde o animal está armazenado.
- **chave** (*int*) – O ID do animal a ser consultado.

manager.display_menu (*arvore*) [código fonte]

Exibe o menu de opções para o usuário e executa a ação correspondente.

Parâmetros :

arvore (*ArvoreAVL*) – A árvore AVL onde os animais estão armazenados.

manager.initialize (*file_path*) [código fonte]

Inicializa a aplicação carregando os dados de um arquivo JSON e exibindo o menu.

Parâmetros :

file_path (*str*) – O caminho do arquivo JSON contendo os dados dos animais.

manager.remove_animal (*arvore* , *chave*) [código fonte]

Remove um animal da árvore AVL pelo seu ID.

Parâmetros :

- **arvore** (*ArvoreAVL*) – A árvore AVL onde o animal está armazenado.
- **chave** (*int*) – O ID do animal a ser removido.

manager. sair () [código fonte]

Encerra a aplicação.

manager. salvar_alteracoes (*arvore* , *save_path*) [código fonte]

Salva as alterações da árvore AVL em um arquivo JSON.

Parâmetros :

- **arvore** (*ArvoreAVL*) – A árvore AVL cujos dados serão salvos.
- **save_path** (*str*) – O caminho do arquivo onde os dados serão salvos.

ui module

ui. solicita_altura () [código fonte]

Solicita e valida a altura do animal.

A altura deve ser um número positivo.

Retorna :

A altura do animal em centímetros.

Tipo de retorno :

str

ui. solicita_apelido () [código fonte]

Solicita e valida o apelido do animal.

O apelido deve conter apenas caracteres alfabéticos.

Retorna :

O apelido do animal.

Tipo de retorno :

str

ui. solicita_data () [código fonte]

Solicita e valida a data de início do monitoramento do animal.

A data deve estar no formato DD/MM/AAAA.

Retorna :

A data de início do monitoramento.

Tipo de retorno :

str

ui. solicita_data_avaliacao () [código fonte]

Solicita e valida a data da avaliação do animal.

A data deve estar no formato DD/MM/AAAA.

Retorna :

A data da avaliação.

Tipo de retorno :

str

ui. solicita_data_nascimento () [código fonte]

Solicita e valida a data de nascimento do animal.

A data deve estar no formato DD/MM/AAAA.

Retorna :

A data de nascimento do animal.

Tipo de retorno :

str

ui. solicita_especie () [código fonte]

Solicita e valida a espécie do animal.

A espécie deve conter apenas caracteres alfabéticos.

Retorna :

A espécie do animal.

Tipo de retorno :

str

ui. solicita_id () [código fonte]

Solicita e valida o ID do animal.

O ID deve ser um número inteiro positivo.

Retorna :

O ID do animal.

Tipo de retorno :

int

ui. solicita_peso () [código fonte]

Solicita e valida o peso do animal.

O peso deve ser um número positivo.

Retorna :

O peso do animal em quilogramas.

Tipo de retorno :

str

ui. solicita_registro () [código fonte]

Solicita os dados para um novo registro no histórico do animal, como data de avaliação, temperatura, peso, altura, coleta de amostra de sangue, exame físico e, se necessário, informações sobre problemas de saúde.

Retorna :

Um objeto de registro com os dados coletados.

Tipo de retorno :

Registro

ui. solicita_sexo () [código fonte]

Solicita e valida o sexo do animal.

O sexo deve ser "M" (masculino) ou "F" (feminino).

Retorna :

O sexo do animal.

Tipo de retorno :

str

ui. solicita_temperatura () [código fonte]

Solicita e valida a temperatura do animal.

A temperatura deve ser um número positivo.

Retorna :

A temperatura do animal em graus Celsius.

Tipo de retorno :

str

Indices and tables

- Índice
- Índice do Módulo
- Página de Busca

