

Classe: Perfil

Atributos:

- Nome – String: Nome do perfil.
- Atividades – Lista_Atividades: Objeto contendo a lista de atividades atrelada ao perfil.

Métodos:

- Perfil(String) – Construtor:

Entrada:

Nome – String: Nome do perfil.

Processamento:

Faz uma busca na lista de perfis criados, verificando se o nome passado já foi criado. Caso sim, cria uma lista de atividades, que carregará os dados atrelados ao perfil. Caso não, cria uma lista de atividades vazia, gera um arquivo para as atividades do perfil, e adiciona o perfil no arquivo com os perfis criados.

- getNome():

Saída:

Nome – String: nome do perfil.

Processamento:

Retorna o nome do perfil.

- setNome():

Entrada:

Nome – String: nome do perfil.

Processamento:

Altera o nome do objeto.

- Remover():

Processamento:

Há a garantia de que sempre existirá dados a serem removidos (os arquivos são criados, se necessário, na instanciação da classe). Deleta o arquivo das atividades atreladas ao perfil, e remove os dados do perfil do arquivo de perfis. Libera o objeto.

- criaLista():

Processamento:

Gera um arquivo para a lista de atividades do perfil, e instancia um objeto Lista.

Classe: Lista_Atividades

(obs): Pensei em um objeto tipo Lista_Atividades para encapsular a lista pois alguns métodos vão escrever/ler dos arquivos e a classe Lista, por si só, não faz isso. Então o perfil possui um objeto Lista_Atividades, que tem a finalidade de gerenciar as atividades, que por sua vez contém a lista em si.

Atributos:

- Atividades: lista ligada.

Métodos:

- Lista_Atividades(String) – Construtor:

Entrada:

Nome – String: o nome do perfil.

Processamento:

Há a garantia que o arquivo das atividades do usuário existe, pois é criado na instanciação do perfil. Entra em um laço de instanciação de Atividades, lendo o arquivo de atividades até o final, adicionando-as à lista ligada.

- Adicionar(String):

Entrada:

Título– String: O título da atividade.

Processamento:

Faz uma busca na lista, verificando se já existe uma atividade com tal título. Caso sim, impede a criação. Caso não, instancia um objeto da classe Atividade, preenchendo-o devidamente - os parâmetros podem vir do usuário ou do arquivo. Adiciona a atividade na lista.

- Remover(String):

Entrada:

Título – String : o título da atividade.

Processamento:

Faz uma busca na lista, verificando se existe a atividade. Caso sim, remove a instância da lista e apaga as informações no arquivo de atividades.

- BuscarAtividade(String):

Entrada:

Título – String: o título da atividade.

Saída:

Atividade – Atividade: o endereço do objeto na lista, ou NULL (dá para fazer isso?).

Processamento:

Faz uma busca na lista pela atividade especificada, retornando o objeto ou NULL.

- ImprimirAtividades():

Processamento:

Imprime os títulos de todas as atividades na lista.

- deletarLista():

Processamento:

Deleta o arquivo de atividades do perfil, limpa os dados relativos às atividades no arquivo de informações dos perfis, e cria um novo objeto vazio para o perfil.

Classe: Atividade

Atributos:

- titulo – String: Título da atividade.
- descricao – String: Descrição da atividade.
- duracao – long: Tempo de duração da atividade, em segundos.
- descanso – long: Tempo de duração do descanso, em segundos.
- alarme_inicio – string: Endereço do arquivo mp3 que deve ser tocado quando a atividade é iniciada.
- alarme_final – string: Endereço do arquivo mp3 que deve ser tocado quando a atividade ou o descanso terminam.
- estatísticas – são outros atributos que serão adicionados posteriormente, caso sejam incluídos no projeto.

Métodos:

- Atividade(...) – Construtor:

Entrada:

Esse método será sobrecarregado para que ele possa se adaptar ao input do usuário. O usuário pode entrar com todos os parâmetros, ou ele pode entrar com apenas alguns parâmetros e os outros serão parâmetros padrões. O único parâmetro obrigatório é o título.

Posso especificar o construtor para cada possibilidade depois. A princípio assim fica mais sintático.

Processamento:

Não é preciso fazer verificação pois o método é instanciado apenas no contexto da Lista de Atividades, e a verificação é feita lá.

As durações, se forem inputs do usuário, devem vir em um formato de HH : MM : SS. Java possui uma biblioteca para gerenciamento de tempo que trabalha com este formato. Antes de armazenar a informação no objeto, esse formato é convertido para segundos.

- printInfo():

Processamento:

Imprime todas as informações da atividade.

- getTitulo():

Processamento:

Retorna o título da atividade.

- getDescricao():

Processamento:

Retorna a descricao da atividade.

- `getDuracao()`:

Processamento:

Retorna a duracao da atividade.

- `getDescanso()`:

Processamento:

Retorna o tempo de descanso da atividade.

- `setTitulo(String)`:

Entrada:

Titulo – String: Título a ser atribuído para a atividade.

Processamento:

Altera o titulo da atividade.

- `getDescricao(String)`:

Entrada:

Descrição – String: Descrição a ser atribuída para a atividade.

Processamento:

Altera a descricao da atividade.

- `setDuracao(long / HH:MM:SS)`:

Entrada:

Esse método – possivelmente – deve ser sobrecarregado, um para admitir um tipo long como entrada, e outro para admitir tempo no formato HH:MM:SS, que será convertido para long.

Processamento:

Altera a duracao da atividade.

- `setDescanso(long / HH:MM:SS)`:

Entrada:

Esse método – possivelmente – deve ser sobrecarregado, um para admitir um tipo long como entrada, e outro para admitir tempo no formato HH:MM:SS, que será convertido para long.

Processamento:

Altera o descanso da atividade.

- `setAlarmInicio(String):`

Entrada:

Endereço – String: o endereço do arquivo mp3 para o alarme.

Processamento:

Altera o endereço do arquivo mp3 do alarme.

- `setAlarmeFinal(String):`

Entrada:

Endereço – String: o endereço do arquivo mp3 para o alarme.

Processamento:

Altera o endereço do arquivo mp3 do alarme.

Classe: Arq_Auxiliar

Atributos:

- endereco_perfil – String: O endereço onde está o arquivo com as informações dos perfis.
- Endereco_atividade – String: o endereço onde estão os arquivos com as informações das atividades.

Métodos:

- listarPerfis():

Processamento:

Percorre o arquivo com as informações dos perfis, imprimindo os nomes de todos os perfis cadastrados.

- buscarPerfil(String):

Entrada:

Nome – String: O nome do perfil a ser buscado.

Processamento:

Percorre o arquivo com as informações dos perfis. Retorna true caso encontre, e false se não encontrar.

- lerPerfil(String):

Entrada:

Nome – String: O nome do perfil a ser lido:

Processamento:

Gera uma instância da classe Perfil e carrega com os dados, do perfil especificado, contidos no arquivo. Isso é feito segundo retorno positivo da Busca (Isso está pouco eficiente, melhorar depois). Retorna a instância da classe Perfil.

- gerarArqPerfil():

Processamento:

Cria o arquivo para armazenar as informações dos perfis, caso não exista.

- adicionarPerfil(Perfil):

Entrada:

Perfil – Perfil: Uma instância da classe Perfil.

Processamento:

Escreve no arquivo de perfis as informações do perfil passado. Isso será feito segundo resultado da Busca.

- `buscarAtividade(String, String):`

Entrada:

Título – String: Título da atividade a ser buscada.

Nome – String: Nome do perfil.

Processamento:

Faz uma busca no pela atividade especificada arquivo de atividades indexado pelo perfil. Retorna true caso encontre, ou false se não encontrar.

- `lerAtividade(String, String):`

Entrada:

Título – String: Título da atividade a ser buscada.

Nome – String: Nome do perfil.

Saída:

Atividade – Atividade: Uma instância da classe Atividade.

Processamento:

Sujeito ao resultado da Busca, gera um objeto Atividade, que será preenchido com as informações da atividade especificada, contidas no arquivo. Retorna o objeto.

- `gerarArqAtividade(String):`

Entrada:

Nome – String: Nome do perfil.

Processamento:

Gera um arquivo para armazenar informações das atividades, indexado pelo perfil especificado.

- `adicionarAtividade(String, Atividade):`

Entrada:

Nome – String: nome do perfil.

Atividade – Atividade: Objeto atividade, cujos dados serão escritos no arquivo.

Processamento:

Escreve no arquivo de atividades indexado pelo perfil, as informações do objeto especificado.

Para especificar... : Métodos para acessar e modificar informações específicas dentro dos arquivos. Serão úteis para os métodos acima.