

Tipos abstratos de dados - Prática

Marco A L Barbosa

malbarbo.pro.br

O código inicial dos exercícios está disponíveis na página <https://malbarbo.pro.br/ensino/2023/6884/>.

- 1) Seu amigo disse que não é possível implementar a classe `Robo` de forma diferente da que vimos em sala. Mostre que ele está errado fazendo uma implementação que usa apenas um campo do tipo string que armazena a info (nome seguido da posição entre parênteses) do robô. Crie uma cópia do arquivo `robo_tad.py` chamada `robo_info.py` e escreva a implementação nesse novo arquivo. Compare as duas implementações (a da sala e essa) e discuta qual é a mais adequada.
- 2) Faça uma implementação da classe `Dias` usando 7 campos booleanos, cada um indicando se um dia está no conjunto de dias ou não. Crie uma cópia do arquivo `dias_tad.py` chamada `dias_bools.py` e escreva a implementação nesse novo arquivo.
- 3) Faça uma implementação alternativa (diferente do exercício anterior) da classe `Dias`. Crie uma cópia do arquivo `dias_tad.py` chamada `dias_alt.py` e escreva a implementação nesse novo arquivo. Compare as duas implementações e discuta qual é a mais adequada.
- 4) Faça uma implementação da classe `Selecao` usando três campos: a célula inicial, o início e o fim do intervalo. Crie uma cópia do arquivo `selecao_tad.py` chamada `selecao_inicio_fim.py` e escreva a implementação nesse novo arquivo.
- 5) Faça uma implementação alternativa (diferente do exercício anterior) da classe `Selecao`. Crie uma cópia do arquivo `selecao_tad.py` chamada `selecao_alt.py` e escreva a implementação nesse novo arquivo.. Compare as duas implementações e discuta qual é a mais adequada.
- 6) Em alguns empresas é comum o uso de um banco de horas, que é um esquema de compensação de horas extras. Quando um funcionário trabalha além da sua jornada normal, as horas e minutos extras trabalhados são “depositados” em um banco de horas. Posteriormente o funcionário pode usar o saldo no banco de horas para se ausentar do trabalho. Projete um TAD `BancoHoras` com operações para depósito de horas e minutos, “saque” de horas e minutos (o saldo deve ser suficiente) e consulta de saldo (que deve gerar uma string no formato HH:MM). Em qualquer operação a quantidade de minutos não podem ser maior que 60.
- 7) Faça uma implementação da classe `BancoHoras` usando um campo para armazenar o saldo em minutos. Faça a sua implementação no arquivo `bancohoras_min.py`.
- 8) Faça uma implementação alternativa (diferente do exercício anterior) da classe `BancoHoras` usando dois campos, uma para horas e outros para minutos. Faça a sua implementação no arquivo `bancohoras_hm.py`. Compare as duas implementações e discuta qual é a mais adequada.