

Instruções para os trabalhos

Este documento contém todas as instruções para o desenvolvimento dos restantes trabalhos práticos (doravante TP) da disciplina. Cada um dos TPs tem seu enunciado e data de entrega específicos, siga os requisitos.

Foi combinado em sala de aula que a dinâmica desses TPs era criar um ambiente de estudo para proporcionar aos alunos aprender a utilizar a linguagem python em áreas da computação estudadas no curso. Assim, cada um dos TP vai oferecer oportunidade para resolver algum problema clássico utilizando pacotes da linguagem. O estudo e utilização de outros materiais além dos sugeridos é válido e permitido, entretanto, se houver aproveitamento de código fora de sua autoria isso deve ser informado na entrega do trabalho para não ser considerada prática de plágio. Além disso, como o objetivo dos TPs é proporcionar oportunidade para ESTUDO e APRENDIZADO, nenhum reaproveitamento de código dispensa o aluno de COMPREENDER e SABER EXPLICAR o funcionamento de todas as partes do código enviado.

Todos os TP podem ser executados em grupo de até dois alunos, desde que o MESMO GRUPO seja mantido para todos eles.

Cronograma para as entregas

TP	Entrega
3	1-out
4	8-out
5	15-out
6	22-out
7	29-out

1. TP3 : vetores e matrizes

Objetivo: O objetivo principal é estudar a utilização eficiente de matrizes na linguagem Python, muito empregada na computação matemática. Basicamente, estudar sobre o uso das bibliotecas Numpy e Scipy.

Aplicação: Implementar em python o método de Gauss-Jordan para resolução de sistemas de equações lineares. É um algoritmo clássico estudado em cálculo numérico. Primeiro, resolva o problema utilizando a biblioteca Numpy (manipulação de matrizes), depois resolva utilizando as facilidades da biblioteca Scipy (utilize o solve).

Referência: <http://www-di.inf.puc-rio.br/~tcosta/cap2.htm>

2. TP4 : threads e controle de regioes criticas

Objetivo: O objetivo principal é estudar a utilização de threads e controle de regiões críticas na linguagem Python. Basicamente, estudar sobre o uso das bibliotecas thread e threading.

Aplicação: Implementar em python um sistema produtor-consumidor que vai utilizar objetos de três classes distintas: Produtor, Consumidor e Estoque. O sistema vai empregar 3 consumidores, 3 produtores e 1 estoque.

Cada objeto da classe Produtor vai funcionar numa thread que produzirá de tempos em tempos um par (int, int) que será armazenado no estoque. O primeiro inteiro do par identifica o produtor e o segundo inteiro identifica o produto. Cada produto recebe um identificador diferente, criado incrementalmente. Para variar o tempo de produção de cada produto em cada produtor utilize um atraso aleatório entre 1 e 3 segundos para cada produto. Existe um limite máximo no tamanho do estoque, se o estoque estiver cheio o produtor deve esperar até surgir espaço no estoque.

Cada objeto da classe Consumidor vai funcionar numa thread que consumirá de tempos em tempos um produto armazenado no estoque. Para variar o tempo de requisição do consumidor utilize um atraso aleatório entre 1 e 3 segundos para cada pedido. Se o estoque estiver vazio quando o consumidor requisitar um produto, ele deve esperar até o estoque fornecer o produto.

O objeto da classe Estoque implementa uma fila (FIFO) capaz de armazenar até 10 produtos, trate essa fila como região crítica. A cada produto armazenado o objeto imprimirá a mensagem “Produtor X armazena produto Y.” onde X e Y são os identificadores do produtor e do produto respectivamente. A cada produto fornecido para um consumidor o objeto imprimirá a mensagem “Consumidor X recebe produto Y.”

Referências:

- resumo sobre multiprogramação em python
- repositório com exemplos sobre multiprogramação em python

3. TP 5 : banco de dados

Objetivo: O objetivo principal é estudar as facilidades para persistência de dados, acesso a um banco de dados, uso de SQL, etc. Existem muitas opções nessa área, então fica como sugestão estudar a biblioteca sqlite3.

Aplicação: Implementar em python algum CRUD bem simples utilizando um banco de dados. Bastam duas tabelas: nome, salário. A intenção é mostrar como é fácil a conexão e o acesso da banco de dados utilizando python.

4. TP 6 : interface grafica

Objetivo: O objetivo principal é estudar as facilidades para implementação de interfaces gráficas. Também aqui existem muitas opções de bibliotecas dependendo da plataforma gráfica que você escolher usar, por exemplo, gnome ou kde. Outra opção é utilizar alguma api portátil em python (genérica ou não) para desenvolvimento de interfaces (glade, kivy, GTK, tkinter, etc). Você pode escolher uma que achar melhor.

Aplicação: implemente uma interface gráfica para o CRUD desenvolvido no TP 5. O objetivo não é a implementação de alguma interface complexa, mas focar na didática mostrando como uma interface pode ser facilmente desenvolvida.

Referência: <https://wiki.python.org.br/ComparacaoDeGUIs>

O link aponta uma página antiga, por isso não considera GUIs mais modernas. Estou sugerindo consultar aqui, entretanto, porque oferece uma idéia da quantidade de opções disponíveis, suas vantagens e desvantagens. Considere quando for escolher a biblioteca gráfica para usar.

5. TP 7 : estruturas de dados e classes

Objetivo: O objetivo principal é aprender a empregar os conceitos de POO para desenvolver classes bem formadas.

Aplicação: Implemente a classe `Tree` que vai permitir ao usuário da classe armazenar informações como a classe `Dict` faz, com a diferença que, enquanto o dicionário implementa uma hash, a árvore implementa uma árvore balanceada (red-black). Para facilitar, considere que as chaves serão sempre strings.

Disponibilize recursos para:

- criar uma árvore: `arvore = Tree()`
- armazenar dado: `arvore['chave'] = dado`
- excluir dado: `arvore['chave'] = None`
- iterar sobre a árvore:

```
1         for (chave, dado) in arvore.preordem():
2             print(chave, '-->', dado)
3
4         for (chave, dado) in arvore.inordem():
5             print(chave, '-->', dado)
6
7         for (chave, dado) in arvore.posordem():
8             print(chave, '-->', dado)
```

Bom estudo.