

Relatório 2 - Linguagem de Programação Python (I)

Yuri Vacelh Zamulhak Zdebski

Descrição da atividade

A atividade consistiu em assistir 2 vídeos do canal Cod3rCursos, sobre a linguagem Python e desenvolver uma aplicação simples utilizando os conhecimentos adquiridos.

Vídeo 1

O primeiro vídeo aborda o equivalente em Python da biblioteca padrão (STD) do C++, explicando desde a importação de pacotes até os diferentes tipos de dados, que podem ser divididos em básicos (inteiros, strings, booleanos) e estruturas de dados (listas, dicionários). Também são apresentados os tipos de operadores e as estruturas de controle (ifs e loops). Essa introdução cobre principalmente os fundamentos da linguagem, incluindo conceitos com os quais eu já tinha certa familiaridade, por ter tido contato com Python durante o ensino técnico e na graduação.

Vídeo 2

O segundo vídeo pode ser dividido em duas partes principais, sendo a primeira parte abordando o básico de funções e do paradigma funcional e a segunda o básico de classes e orientação a objetos.

Da primeira parte, temos uma breve introdução sobre como funcionam as funções em Python (declaração e argumentos), seguido pela parte funcional do vídeo. Nela somos apresentados a conceitos como funções lambda (funções anônimas), map (função embutida que executa uma função, que recebe por parâmetro, em cada item de um objeto iterável, retornando o objeto modificado pela função), reduce (função embutida que assim como o map, executa uma função passada por parâmetro item a item, com a diferença que retorna apenas um valor) e o conceito de compreensão de listas, que lembra muito a função map, porém aplicada somente a listas.

Sobre a segunda parte, houve uma sucinta introdução a orientação em objetos em Python. Primeiro mostrando a sintaxe de declaração de classes. Por não ser um tutorial de OOP, não entra tanto na parte teórica, mas ainda assim tem algumas partes bem interessantes. Os pontos que eu achei mais importantes (que eu não conhecia) foram, o comando @property (que define um método como propriedade da classe) e o @<atributo>.setter (que cria um setter mas sem precisar chamar o método set, apenas atribuir diretamente com <objeto>.<atributo> = <valor>), a função super() (que chama a função da superclasse, que originou a herança), e a diferença entre um método de classe e estático (o primeiro tem acesso a atributos da classe, o segundo não, apenas executa uma lógica que pode ser útil para a classe).

Conclusões

Com os conhecimentos adquiridos nesse módulo, construí um exemplo simples, mas que engloba alguns dos conceitos apresentados. Um simulador de dado, com uma classe Dado que tem um atributo faces (que diz o número de faces de um dado), e dois métodos um rolar (que gera um número aleatório num tal que $1 < num < faces$) e rolar varias vezes (que chama o primeiro n vezes). Também criei uma classe que simula o dado N vezes e calcula a média das

ocorrências e a frequência que cada face aparece (para os cálculos estatísticos tentei utilizar apenas programação funcional, mas apanhei para a frequência).