**Testes automatizados: TDD com Python**

1. **Aula 1 – Por que testar?**
   1. ALT + INSERT: Atalho para criar um novo arquivo no pycharm.
   2. Quando vamos importar uma classe presente em um arquivo separado do python, podemos fazer de duas maneiras, a primeira é somente selecionar o que o python sugere em primeiro e a segunda é seguir o caminho completo das pastas desde o root até o arquivo .py:



* + 1. Como podemos ver, as subpastas são separadas por “.” E não por “/” ou “\” como estamos acostumados.
  1. CTRL + P: Atalho para saber o que colocar como parâmetro em uma função caso não lembramos o que precisa ser passado. Semelhante ao CTRL + SPACE no VScode e no próprio Pycharm, mas para funções, atributos e afins:



* 1. @property: Já vimos antes, responsável por retornar uma propriedade privada como uma normal para visualização da mesma. Não permite alterar. A visualização é de uma segunda variável que possuí o valor da privada, permitindo que o usuário a acesse como se fosse o atributo:



* 1. Quando vamos implementar um sistema de avaliador para saber qual foi o menor e qual foi o maior lance dado em um leilão, por exemplo, podemos utilizar uma biblioteca chamada sys.
     1. Como precisamos criar duas variáveis, uma que receberá o maior e outra o menor valor, o valor de início delas precisa ser, respectivamente, o menor e o maior valor existente, pois, dessa maneira, qualquer valor que for muito alto sobrescreverá o menor valor da variável maior valor, e o contrário também é verdade para o menor valor.
     2. Dessa forma, utilizamos a biblioteca sys para que ela nos forneça esses dados a partir do sistema, sem que nós mesmos precisemos determinar isso:



* + 1. O que estamos dizendo é que queremos pegar o menor valor das informações de números flutuantes, ou seja, com “.”/”,”, existentes na biblioteca sys. O contrário é válido para o maior valor.
  1. Quando estamos criando um método, podemos passar como argumento o nome dele seguido de “:” e uma classe, ou então uma frase, indicando o que se espera receber como argumento. Chamamos esse tipo de estratégia de anotação:





* + 1. Basicamente estamos dizendo que como argumento para esse método esperamos que seja passado o objeto Leilao.
    2. Para fazer com que o maior e menor lance sejam atribuídos as suas respectivas variáveis, precisamos fazer um esquema de validação com ifs:



* + 1. Basicamente dizendo que para cada lance nos lances do lailao, deverá ser feita essa verificação: Se o valor do lance for maior que o maior lance (inicialmente determinado pelo sys, como visto acima), então o maior lance recebe esse novo valor do lance. O contrário também é verdade.
  1. ALT + ENTER: Caso esteja tentando utilizar uma classe que ainda não foi importada, ao invés de subir todo o código e digitar o import dela, pode simplesmente utilizar esse atalho e selecionar a primeira opção:



* 1. Testamos nosso código e descobrimos que existe um bug nele ao adicionar outro lance com um valor maior do que o que já havia sido definido antes. Com esse exemplo determinamos o quão importante é testar nosso código, pois se esse programa fosse de um hospital, pacientes poderiam ter recebido as doses erradas de um medicamento.