

Introdução

A linguagem Python, por não exigir declaração explícita de variáveis, apresenta situações em que as regras de escopo das variáveis são muito sutis, principalmente para um aprendiz da linguagem, e que, na realidade, devem ser evitadas em programas desenvolvidos por profissionais. Nesta aula, vamos estruturar o projeto de nosso programa de modo que as regras de escopo sejam bem simples e intuitivas. Além disso, vamos usar arranjos para a estruturação dos dados e do programa em si. Para tanto, vamos considerar o problema de calcular estatísticas de um conjunto de dados. Suponha que os dados se referem a notas finais de INF100. As estatísticas que queremos calcular são: a média das notas, o desvio padrão das notas, a maior nota e a menor nota.

Instruções

1. Baixe do sistema de entrega de trabalhos os seguintes arquivos: `p02.py` e `notas_inf100.dat`.
2. Abra o IDLE com o arquivo fonte denominado `p02.py`. Não se esqueça de salvá-lo de tempos em tempos, porque pode ocorrer falha de energia elétrica durante a aula prática.
3. Preencha os comentários obrigatórios (nome, matrícula, data e uma breve descrição sobre o que o programa faz).
4. Estruture seu programa em cinco funções: `main()`, `media(notas)`, `desvio_pad(notas)`, `maximo(notas)` e `minimo(notas)`.
5. A função `main` já está pronta (favor não mexer nela) e faz o seguinte: lê o conjunto de dados a partir de um arquivo previamente digitado, chama as funções para calcular as estatísticas e imprime as estatísticas.
6. Implemente a função `media` com um parâmetro: o arranjo de notas. A função deve retornar a média das notas, se houver notas no arranjo. Teste o tamanho do arranjo maior que zero.
7. A função `desvio_pad` tem um parâmetro: o arranjo de notas. A função deve retornar a desvio padrão das notas, se houver mais de uma nota no arranjo. Teste o tamanho do arranjo maior que um. Para o cálculo da raiz quadrada, use a função `sqrt` fornecida pela biblioteca `math` do Python. Assim você produzirá *menor erro* de cálculo numérico do que se usar o expoente 0.5 numa potenciação.
8. A função `maximo` tem um parâmetro: o arranjo de notas. A função deve retornar a maior nota do arranjo, se houver notas. Teste o tamanho do arranjo maior que zero. NÃO se pode usar a função *builtin* `max` do Python.
9. Idem para a função `minimo`, *mutatis mutandis*. NÃO se pode usar a função *builtin* `min` do Python.
10. Teste seu programa com o arquivo de notas fornecido no *site* de entrega: `notas_inf100.dat`. Veja abaixo um exemplo de execução do programa.
11. Se seu programa entrar em *laço infinito*, digite CTRL-C na janela do *Shell*, para interromper a execução do programa. Após, conserte-o.

Exemplo de execução do programa

12 notas lidas.

Média das notas:	59.5
Desvio padrão das notas:	23.0
Maior nota:	91.0
Menor nota:	25.0

Fórmulas

$$\text{Média} = \frac{\sum_{i=0}^{n-1} x_i}{n}, \text{ se } n > 0$$

$$\text{Desvio Padrão} = \sqrt{\frac{\sum_{i=0}^{n-1} x_i^2 - \frac{(\sum_{i=0}^{n-1} x_i)^2}{n}}{n-1}}, \text{ se } n > 1$$

☞ Não se esqueça de preencher o cabeçalho do código fonte com seus dados, a data de hoje e uma breve descrição do programa.

Após certificar-se de que seu programa esteja correto, envie o arquivo do programa fonte (p02.py) através do sistema de entrega do LBI.