Luiz Filipe Silveira Bido e Gabriel Gonzaga Seabra Câmara
Trabalho avaliativo de CLP - 2024/1

Implementação com visualização gráfica e duas linguagens de programação

A aplicação escolhida para o trabalho foi o cálculo de uma função quadrática (ax^2+bx+c) feito na linguagem C e a demonstração de resultado em uma interface gráfica desenvolvida em Python, fazendo uso da biblioteca tkinter para receber os parâmetros da função e demonstrar o gráfico resultante de tais valores.

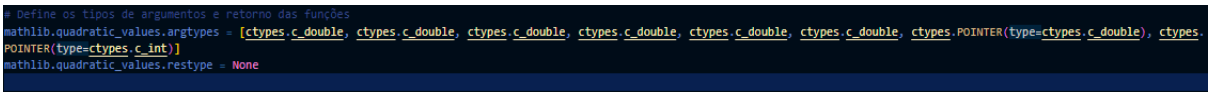
- Para utilizar as funções das bibliotecas C no programa Python, importamos a biblioteca *'ctypes'*;
- Para operações matemáticas e manipulação de arrays, importamos a biblioteca *'numpy'*;
- Para plotar o gráfico importamos a biblioteca *'matplotlib.pyplot'*;
- Para integrar o matplotlib no tkinter, tivemos que importar a biblioteca *'matplotlib.backends.backend_tkagg.FigureCanvasTkAgg'*;
- Para a interface gráfica, importamos *'tkinter'*;
- Para a manipulação do sistema operacional, que no programa é usado para obter o caminho absoluto da biblioteca C, importamos a biblioteca *'os'*.

O carregamento da biblioteca C é feito na etapa do código demonstrada abaixo:

```
# Caminho absoluto para a biblioteca .so
so_path = os.path.abspath(path='./libmathlib.so')
mathlib = ctypes.CDLL(name=so_path)
```

- *'so_path'* recebe o caminho absoluto para a biblioteca compartilhada .so;
- *'mathlib'* recebe o carregamento da biblioteca C.

Na etapa seguinte, é definido todos os tipos de argumentos e retornos das funções:



- *'argtypes'* é a lista de tipos do argumentos que a função C espera para ser executada;
- *'restype'* é o tipo de retorno da função C, que no caso do nosso programa não possui.

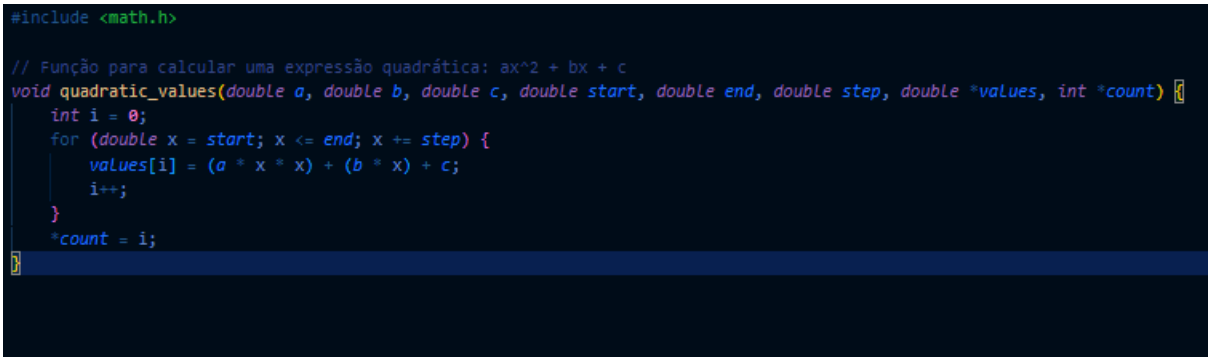
Todas as funções estarão dentro da classe '*QuadraticApp*':

- Função '*__init__*':
  - Inicializa a classe 'QuadraticApp' com o objeto 'root' do 'tkinter';
  - Define o título da janela como "Função quadrática";

- Função *'create_widgets'*:
  - Cria widgets 'Label' e 'Entry' para entrada dos parâmetros da função quadrática (a, b, c, início, fim e passo);
  - Cria um botão 'Button' para plotar o gráfico;
  - Configura a área para exibir o gráfico usando 'FigureCanvasTkAgg'

- Função *'plot_graph'*:
  - Obtém os parâmetros da função quadrática a partir das entradas do usuário;
  - Verifica se os valores são numéricos, senão exibe uma mensagem de erro;
  - Calcula o número de pontos e aloca memória para os valores usando 'ctype';
  - Chama a função C 'quadratic_values' (função do mathlib.c) com os parâmetros apropriados;
  - Converte os valores retornados para arrays;
  - Plota o gráfico da função quadrática usando 'matplotlib';
  - Atualiza o gráfico na interface;

Já no arquivo *'mathlib.c*', a única função presente é a *'quadratic_values'* que irá calcular os pontos necessários para a montagem do gráfico no arquivo '*app.py*'.



```c
#include <math.h>

// Função para calcular uma expressão quadrática: ax^2 + bx + c
void quadratic_values(double a, double b, double c, double start, double end, double step, double *values, int *count) {
    int i = 0;
    for (double x = start; x <= end; x += step) {
        values[i] = (a * x * x) + (b * x) + c;
        i++;
    }
    *count = i;
}
```