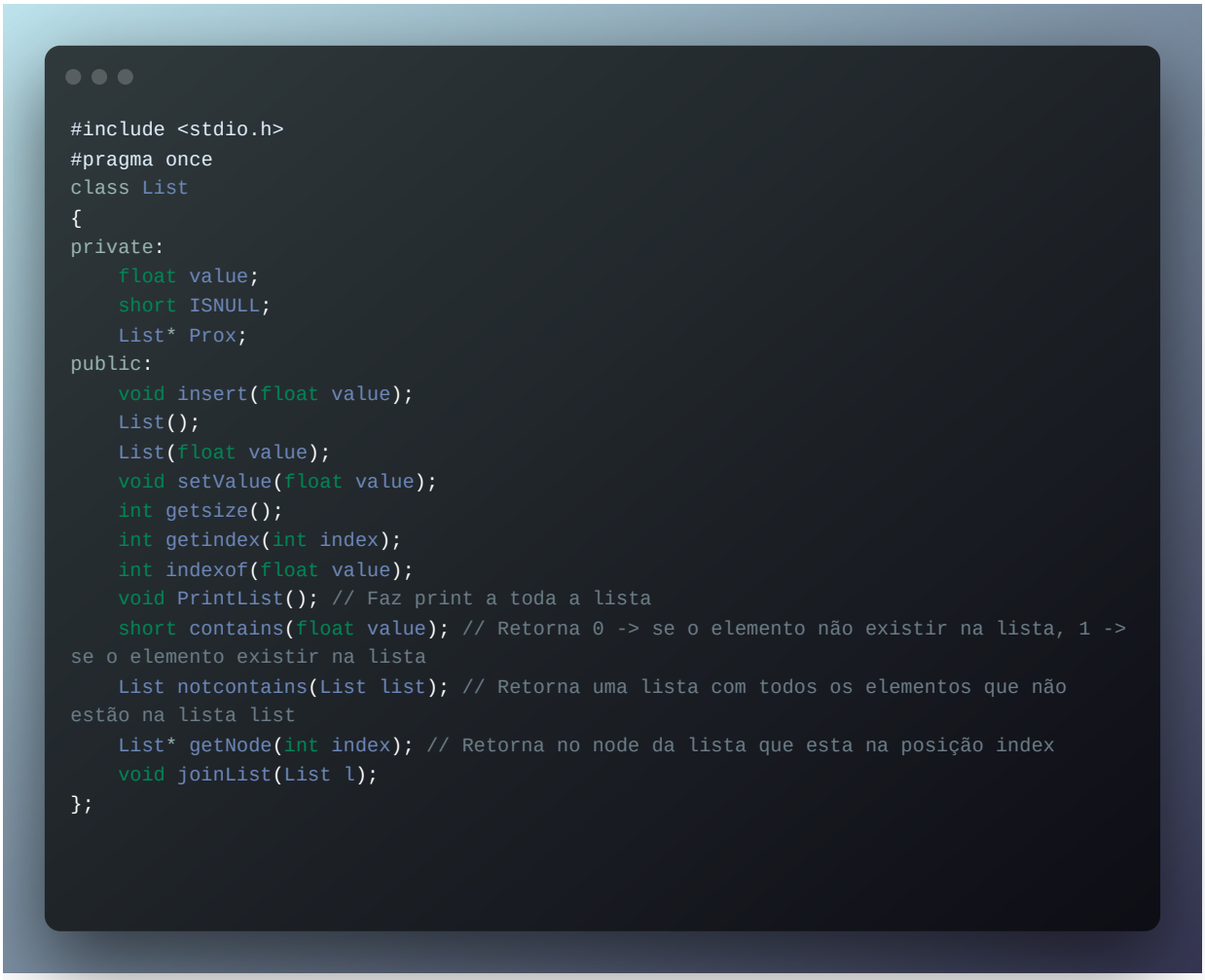
List:

Introdução:

Para uma maior facilidade de gestão de variáveis do tipo *float* criamos uma classe List. Esta classe opera de maneira semelhante a outras classe list que podem ser encontradas de maneira nativa em linguagens como: Java, Python, Ocaml, C#, JavScript, etc…

Desenvolvimento e implementação:

A implementação desta classe foi feita de maneira simples.



```
#include <stdio.h>
#pragma once
class List
{
private:
    float value;
    short ISNULL;
    List* Prox;
public:
    void insert(float value);
    List();
    List(float value);
    void setValue(float value);
    int getsize();
    int getindex(int index);
    int indexof(float value);
    void PrintList(); // Faz print a toda a lista
    short contains(float value); // Retorna 0 -> se o elemento não existir na lista, 1 ->
se o elemento existir na lista
    List notcontains(List list); // Retorna uma lista com todos os elementos que não
estão na lista list
    List* getNode(int index); // Retorna no node da lista que esta na posição index
    void joinList(List l);
};
```

Imagem da classe List

A classe tem como atributos:

- value → um float que representa o valor daquele node da lista;
- ISNULL → um short que informa que o valor daquele node é NULL ou não. Este atributo existe pois em C++ NULL = 0. Então se o valor de um node da lista fosse igual a 0 o C++ assumiria que aquele node seria NULL ou chega vazio e a lista não continuaria;
- Prox → apontador para o próximo node da list;

O métodos da classe são:

- List() e List(float value) → construtores da classe. Um sendo um construtor vazio e outro que recebe o valor do node;
- setValue(float value) → alterar o valor do node selecionado;
- getsize() → devolve o tamanho da lista, ou seja, o número total de elementos contidos na lista;
- indexof(float value) → devolve o index do node com o valor igual a value. Se existir mais que um node com o mesmo valor ele retorna o index do primeiro node.
- PrintList → faz a impressão no ecrã da lista, no formato:
  - [x,y,z,…]
- contains(float value) → retorna o valor de 0 se não existir nenhum node na lista com valor igual ao value. Retorna 1 caso contrário;
- notcontains(List List) → retorna uma lista com todos os elementos na lista que não estão na lista passada como argumento;
- getNode(int index) → retorna o node da lista que esta na posição index. Se o index for maior que o tamanho da lista ou for um número negativo retorna um apontador para NULL;
- joinList(List i) → faz a junção dos elemento da lista atual com os elementos presentes na lista passada como argumento. Se o mesmo valor estiver na duas lista ele é adicionado na mesma. Resultando numa lista com valores repetidos.