Módulo 1 - Estruturas básicas - Lista #3

Data de entrega: 13 de abril de 2017

Acompanham esta lista os arquivos **mod1_lista3.h** e **mod1_lista3.cpp**, contendo as declarações iniciais. Modifique estes arquivos para implementar as questões pedidas e envie-os de volta zipados com nome no padrão <numero_matricula>.zip por email para **profs-eda@tecgraf.puc-rio.br**, com o assunto **[EDA] Lista 3**. Atenção: Crie um arquivo contendo a função main do seu programa para testar suas implementações, mas envie SOMENTE os arquivos e as classes solicitadas.

---

1. Modifique a sua classe Abb para que ela se torne uma árvore balanceada, do tipo AVL, por ora sem implementar a remoção. Cada nó da árvore é representado por uma struct que contém o valor da chave inteira, o fator de balanceamento e os ponteiros para os elementos up (pai), left (filho esquerdo) e right (filho direito).

   As funções **show** e **show_rec** já estão implementadas, não modifique-as. Utilize a função show para visualizar a árvore e testar as suas implementações.

   O seguinte trecho de código teste:

```
Avl a;
a.insert(5);
a.insert(8);
a.insert(10);
a.insert(4);
a.insert(3);
a.insert(6);
a.insert(12);
a.insert(11);

a.show("Arvore A:");
std::cout << "Altura de A: "
              << a.height() << std::endl;

Avl b(a);
a.show("Arvore B:");
```

Deve produzir a saída:

```
Arvore A:
<5<4<3..>.><8<6..><11<10..><12..>>>>
Altura de A: 3
Arvore B:
<5<4<3..>.><8<6..><11<10..><12..>>>>
```

Declaração da classe Avl:

```
#ifndef MOD1_LISTA3
#define MOD1_LISTA3

#include <string>

struct AvlNode
{
  int _key;

  //fator de balanceamento = hd-he
  int _balance_factor;

  AvlNode* _up;
  AvlNode* _left;
  AvlNode* _right;
};

class Avl
{
public:
  //Cria uma arvore vazia
  Avl();

  //Cria uma arvore com um primeiro elemento
  Avl(int key);

  //Cria uma arvore a partir de outra
  Avl(const Avl& orig);

  //Destroi a arvore
  virtual ~Avl();

  //Retorna a altura da arvore
  int height();

  //Insere um elemento na arvore
  void insert(int key);
```

```
  //Exibe a arvore no formato
  //<raiz<sub-arvore esquerda><sub-arvore direita>>
  void show(const std::string& title);

  /** Funcoes para percorrer a arvore em ordem: **/

  //Posiciona o cursor no primeiro elemento
  bool first();

  //Posiciona o cursor no ultimo elemento
  bool last();

  //Atualiza o no cursor para o proximo elemento
  bool next();

  //Atualiza o no cursor para o elemento anterior
  bool prev();

  //Exibe o valor do no cursor atual
  int value();

private:

  /* Mantenha as funcoes privadas necessarias
   * criadas para a Abb e crie as funcoes
   * (recursivas ou nao) que julgar necessarias
   * para a Avl */

  //Funcao recursiva para exibir a arvore
  void show_rec(AvlNode* node);

private:
  AvlNode* _root;
  AvlNode* _cursor;
};

#endif
```