PUC Minas
Instituto de Ciências Exatas e Informática

Pontifícia Universidade Católica de Minas Gerais
Bacharelado em Ciência da Computação
Algoritmos e Estruturas de Dados I
Prof.: Henrique Batista da Silva
Lista de Exercícios 12

Orientação à Objetos em C++

Orientações:

- Os exercícios devem ser feitos em **linguagem C++**.
- Cada exercício deve ser feito em um arquivo .h e .cpp (extensões dos arquivos da linguagem C++). Você deve entregar apenas um único arquivo compactado (.zip) no SGA contendo todos os exercícios resolvidos.
- Exercícios copiados receberão nota zero.
- Não deixe a lista para a última hora. Comece o quanto antes. Assim haverá tempo para esclarecer dúvidas com o professor e na monitoria.

1- Considere a classe **Vetor**, cuja assinatura está definida abaixo:

Vetor.h

```
class Vetor {

  private:

      int *v;                 // Armazena os elementos

      int numElementos;       // Informação sobre o número de elementos inseridos


  public:

      const static int TAMANHO = 10;

      Vetor();

      int obtemTamanho();

      void insereNoFinal(int novoElemento);

      int posicaoDe (int elemento);

      void alteraEm (int pos, int novoValor);

      int elementoDe (int pos);

      int elementoEm (int pos);

      void reverte();

      void imprime();

};
```

Vetor.cpp

```
#include "vetor.h"


Vetor::Vetor() { v = new int [TAMANHO]; }

int Vetor::obtemTamanho() { return 0; }

void Vetor::insereNoFinal(int novoElemento) { }

int Vetor::posicaoDe (int elemento)  { return 0; }

void Vetor::alteraEm (int pos, int novoValor) { }

int Vetor::elementoDe (int pos)  { return 0; }

int Vetor::elementoEm (int pos)  { return 0; }

void Vetor::reverte() { };

void Vetor::imprime() { };
```

Implemente todos os métodos da classe **Vetor** acima, obedecendo às seguintes descrições dos métodos:

a. Vetor(): Construtor que inicializa o atributo numElementos com zero e zera todas as posições do vetor;
b. obtemTamanho(): Retorna o número de elementos armazenados no vetor;
c. insereNoFinal(novoElemento): Insere novoElemento na primeira posição vazia do vetor;
d. posicaoDe(elemento): Retorna a posição da primeira ocorrência de elemento no vetor ou o valor -1, caso o elemento não seja encontrado.
e. alteraEm(pos, novoValor): Atribui novoValor ao elemento da posição pos do vetor, caso esta seja uma posição válida, retorna -1, caso contrário.
f. elementoEm(pos): Retorna o valor armazenado na posição pos, caso esta seja uma posição válida, retorna -1, caso contrário.
g. imprime(): Imprime os elementos do vetor, separados por espaço.
h. reverte(): Reverte os elementos do vetor. Por exemplo, o vetor inicial {1,3,4,1,2}, após a chamada desta função, o vetor ficará {2,1,4,3,1,};

Após a implementação da classe Vetor, a função main, a seguir, deverá funcionar:

```
#include "vetor.h"

using namespace std;


int main (int argc, char *argv[])

{

    Vetor *v = new Vetor();
```

```
    v->insereNoFinal(10);
    v->insereNoFinal(8);
    v->insereNoFinal(16);
    v->insereNoFinal(7);
    v->insereNoFinal(5);
    v->insereNoFinal(13);

    v->imprime();

    v->alteraEm(3,19);
    v->alteraEm(15,9);

    int i;
    for (i = 0; i < v->obtemTamanho(); i++)
        cout << "Elemento na posicao " <<i <<": " <<v->elementoEm(i);

    v->reverte();
    v->imprime();
}
```

2- Nesta aula, você deverá alterar a classe Vetor (definida anteriormente) para:

a. Permitir a realocação do vetor ao se tentar inserir um elemento e não houver posições vazias. A cada realocação você deverá dobrar o tamanho do vetor. Inicie com 10 elementos.

b. Permitir a retirada de todas as ocorrências de um elemento. Ao remover um elemento, se este não for o último elemento, deve-se deslocar todos os demais para não deixar posições vazias no vetor. Se o número de elementos após a retirada ficar menor que a metade do tamanho do vetor, então você deverá realocar o vetor com a metade do número de elementos.

c. Modifique o programa principal (função main) para testar se novo programa.