Eduardo Santiago Rosalin - 802539

PARTE A:

Exercício 3.27 Na versão do Snake descrita no Desafio 2, no decorrer do jogo a Cobra irá comer um quadradinho colorido e você precisará saber se esse quadradinho colorido é da cor da cabeça da Cobra ou não. E para isso, precisará saber a cor da cabeça da Cobra. Você tem duas alternativas para implementar uma operação que consulta o valor do primeiro elemento de uma Fila: pelo botão do volume (acionando as Operações Primitivas da Fila), ou abrindo a TV (solução dependente da implementação). Qual destas duas alternativas você considera mais interessante para seu projeto, e por que?

- R: A alternativa de consultar o valor acionando as Operações Primitivas é mais interessante pois reutiliza código e abstrai os detalhes da implementação. Assim, podemos nos concentrar na lógica do jogo sem se preocupar com o funcionamento interno da estrutura.

Exercício 3.13 As Filas São Iguais?
- R:

```
Boolean Iguais (parâmetros por referência F1, F2 do tipo Fila)
{
   se (F1.estaVazia() && F2.estaVazia())
      retorna TRUE;

   int topo1, topo2;
   F1.desenfileirar (topo1);
   F2.desenfileirar (topo2);

   bool iguais = TRUE;

   Enquanto (!F1.estaVazia() || !F2.estaVazia())
```

```
    {
        F1.desenfileirar(topo1);
        F2.desenfileirar(topo2);

        se (topo1 != topo2)
        {
            iguais = FALSE;
        }

    }

    return iguais;
}
```

PARTE B:

- **FilaEstática.hpp**

```
#ifndef FILA_ESTATICA_HPP
#define FILA_ESTATICA_HPP

#define MAX_TAM 3

class FilaEstatica
{
private:
    int vetor[MAX_TAM];
    int comeco, fim, numElementos;
```

```
public:
    FilaEstatica();
    bool enfileirar(int novoElemento);
    bool desenfileirar(int& out);
    bool estaVazia();
    bool estaCheia();

    void mostrarFila();
    bool verificarFilaCheiaSemVariavel();
    void reiniciarFila();

    bool compararFilas(FilaEstatica& f2);

};


#endif
```

- **FilaEstática.cpp**

```
#include "FilaEstatica.hpp"
#include <iostream>

#define MAX_TAM 3
//// Criar Fila
//// Queue
//// Dequeue
//// estaVazia
//// estaCheia
//// mostrarFila
//// verificarFilaCheiaSemVariavel


FilaEstatica::FilaEstatica()
```

```
{
    // vetor[MAX_TAM];
    numElementos = 0;
    comeco = 0;
    fim = 0;
}

bool FilaEstatica::enfileirar(int novo)
{
    if (estaCheia()) {
        std::cout << "Fila esta cheia :(" << std::endl;
        return false;
    }

    vetor[fim] = novo;
    fim = (fim + 1) % MAX_TAM;
    numElementos++;

    return true;

}

bool FilaEstatica::desenfileirar (int& out)
{
    if (estaVazia()) {
        std::cout << "Fila está vazia :(" << std::endl;
        return false;
    }

    out = vetor[comeco];
    comeco = (comeco + 1) % MAX_TAM;
    numElementos--;

    return true;
}
```

```
void FilaEstatica::mostrarFila()
{
    if (estaVazia()) {
        std::cout << "Fila esta Vazia" << std::endl;

        return;
    }
    for (int i = 0; i < numElementos ; i++) {
        int index = (comeco + i) % MAX_TAM;
        std::cout << vetor[index] << " ";
    }

    std::cout << std::endl;
    std::cout << "FIM DA FILA" << std::endl;

}


bool FilaEstatica::estaVazia()
{
    return numElementos == 0;
}

bool FilaEstatica::estaCheia()
{
    return numElementos == MAX_TAM;
}


bool FilaEstatica::verificarFilaCheiaSemVariavel()
{
    if (comeco == fim) {
          std::cout << "Fila esta cheia (verificacao sem
variavel)" << std::endl;
```

```
        return true;
    }
      std::cout << "Fila NAO esta cheia (verificacao sem
variavel)" << std::endl;
    return false;
}

void FilaEstatica::reiniciarFila()
{
    std::cout << "FILA REINICIADA" << std::endl;

    numElementos = 0;
    comeco = 0;
    fim = 0;
}

bool FilaEstatica::compararFilas(FilaEstatica& f2)
{
    if (estaVazia() && f2.estaVazia())
        return true;

    if (numElementos != f2.numElementos)
        return false;

    int topo1 = comeco;
    int topo2 = f2.comeco;

    for (int i = 0; i < numElementos ; i++) {
        if (vetor[topo1] != f2.vetor[topo2]) {
            return false;
        }

        topo1 = (topo1 + 1) % MAX_TAM;
        topo2 = (topo2 + 1) % MAX_TAM;
    }
```

```
    return true;

}
```

- **main.cpp**

```
#include "FilaEstatica.hpp"
#include <iostream>

int main (void)
{
    FilaEstatica f1;
    f1.enfileirar(1);
    f1.enfileirar(2);
    f1.enfileirar(3);
    f1.mostrarFila();
    f1.enfileirar(4);

    FilaEstatica f2;
    f2.enfileirar(1);
    f2.enfileirar(4);
    f2.enfileirar(3);

    bool ok = f1.compararFilas(f2);

    if (ok)
        std::cout << "Filas iguais" << std::endl;
    else
        std::cout << "Filas Diferentes" << std::endl;

    f1.reiniciarFila();
    f1.mostrarFila();

    return 0;
```

```
}
```

- **TERMINAL**

```
1 2 3
FIM DA FILA
Fila esta cheia :(
Filas Diferentes
FILA REINICIADA
Fila esta Vazia
 *  Terminal will be reused by tasks, press any key to close it.
```