INE5408-03208A|INE5609-03238B (20182) - Estruturas de Dados

NAVEGAÇÃO

Painel
- Página inicial do site
- Moodle UFSC
- Curso atual
  - INE5408-03208A|INE5609-03238B (20182)
    - Participantes
    - Emblemas
    - Geral
    - Tópico 1
    - Tópico 2
      - Vídeos de Simulações de Filas e Pilhas
      - Video Aula sobre Pilhas e Filas com Vetores
      - **Implementacão de Pilha com vetor**
        - Descrição
        - Enviar
        - Editar
        - Visualizar envios
      - Testes (Pilha)
      - Implementacão de Fila com vetor
      - Testes (Fila)
      - Lecture 3
    - Tópico 3
    - Tópico 4
- Meus cursos

ADMINISTRAÇÃO
- Administração do curso

Descrição | Enviar | Editar | Visualizar envios

Nota

Revisado em sexta, 10 Ago 2018, 20:09 por Atribuição automática de nota
Nota 100 / 100
Relatório de avaliação
[+]**Summary of tests**
Enviado em sexta, 10 Ago 2018, 20:09 (Baixar)

array_stack.h

```
1  //! Copyright 2018 Matheus Henrique Schaly
2
3  #ifndef STRUCTURES_ARRAY_STACK_H
4  #define STRUCTURES_ARRAY_STACK_H
5
6  #include <cstdint>  // std::size_t
7  #include <stdexcept>  // C++ exceptions
8
9  namespace structures {
10
11 //! Static stack
12 template<typename T>
13 class ArrayStack {
14  public:
15 //! Constructor
16 ArrayStack();
17
18 //! Constructor with parameter
19 ArrayStack(std::size_t max);
20
21 //! Destructor
22 ~ArrayStack();
23
24 //! Push an object to the top of the stack
25 void push(const T& data);
26
27 //! Remove an object from the top of the stack
28 T pop();
29
30 //! Get the element from the top of the stack
31 T& top();
32
33 //! Clear the stack
34 void clear();
35
36 //! Return the size of the stack
37 std::size_t size();
38
39 //! Return the storage capacity of the stack
40 std::size_t max_size();
41
42 //! Return True if empty, false otherwise
43 bool empty();
44
45 //! Return True if full, false otherwise
46 bool full();
47
48  private:
49 static const  auto DEFAULT_SIZE = 10u;
50 T* contents;
51 int top_;
52 std::size_t max_size_;
53 };
54 }  // namespace structures
55
56 template<typename T>
57 structures::ArrayStack<T>::ArrayStack() {
58     ArrayStack(DEFAULT_SIZE);
59 }
60
61 template<typename T>
62 structures::ArrayStack<T>::ArrayStack(std::size_t max) {
63     top_ = -1;
64     max_size_ = max;
65     contents = new T[max];
66 }
67
68 template<typename T>
69 structures::ArrayStack<T>::~ArrayStack() {
70     top_ = -1;
71     max_size_ = 0;
72     delete[] contents;
73 }
74
75 template<typename T>
76 void structures::ArrayStack<T>::push(const T& data) {
77     if (full()) {
78         throw std::out_of_range("A pilha esta cheia");
79     } else {
80         top_++;
81         contents[top_] = data;
82     }
83 }
84
85 template<typename T>
86 T structures::ArrayStack<T>::pop() {
87     if (empty()) {
88         throw std::out_of_range("A pilha esta vazia");
89     } else {
90         top_--;
91         return contents[top_+1];
92     }
93 }
94
95 template<typename T>
96 T& structures::ArrayStack<T>::top() {
97     return contents[top_];
98 }
99
100 template<typename T>
101 void structures::ArrayStack<T>::clear() {
102     top_ = -1;
103 }
104
105 template<typename T>
106 std::size_t structures::ArrayStack<T>::size() {
107     return top_ + 1;
108 }
109
110 template<typename T>
111 std::size_t structures::ArrayStack<T>::max_size() {
112     return max_size_;
113 }
114
115 template<typename T>
116 bool structures::ArrayStack<T>::empty() {
```

```
117     return top_ == -1;
118 }
119
120 template<typename T>
121 bool structures::ArrayStack<T>::full() {
122     return size() == max_size();
123 }
124
125 #endif
126
```

VPL 3.1.5

Você acessou como Matheus Henrique Schaly (18200436) (Sair)
INE5408-03208A | INE5609-03238B (20182)