TT10 –

Tipos Abstratos de Dados Flexíveis:

David de Sá Vieira de Faria - 699415**Exercício Resolvido – 1:**

Seja nossa Pilha, faça um método que soma o conteúdo dos elementos contidos na mesma.

|  |
| --- |
| public int getSoma(){          int soma = 0;          for (Celula i = topo; i != null; i = i.prox) {              soma += i.elemento;          } // end for          return soma;  } // end getSoma() |

**Exercício (1):**

Seja nossa Pilha, faça um método RECURSIVO que soma o conteúdo dos elementos contidos na mesma.

|  |
| --- |
| public int getSoma\_Recursiva(){          return getSoma\_Recursiva(topo);      } // end getSoma\_Recursiva()      public int getSoma\_Recursiva(Celula i ){          int soma = 0;          if(i != null){              soma += i.elemento;              soma += getSoma\_Recursiva(i.prox);          } // end if          return soma;      } // end getSoma\_Recursiva() |

**Exercício (2):**

Seja nossa Pilha, faça um método que retorna o maior elemento contido na pilha.

|  |
| --- |
| public int getMaior(){          int maior = 0;          for (Celula i = topo; (i != null) && (maior < i.elemento); i = i.prox) {              maior = i.elemento;          } // end for          return maior;      } // end getSoma() |

**Exercício (3):**

Seja nossa Pilha, faça um método RECURSIVO que retorna o maior elemento contido na pilha.

|  |
| --- |
| public int getMaior\_Recursiva(){          return getMaior\_Recursiva(topo, 0);      } // end getMaior\_Recursiva()      public int getMaior\_Recursiva(Celula i, int maior){          if(i != null){              if(maior > i.elemento)                  maior = getMaior\_Recursiva(i.prox, maior);              else                  maior = getMaior\_Recursiva(i.prox, i.elemento);          } // end if          return maior;      } // end getMaior\_Recursiva() |

**Exercício (4):**

Seja nossa Pilha, faça um método RECURSIVO para mostrar os elementos da pilha na ordem em que os mesmos serão removidos.

|  |
| --- |
| public void showPilha\_remove\_Recursiva(){      System.out.print("[ ");      showPilha\_remove\_Recursiva(topo);      System.out.print(" ]\n");  } // end showPilha\_Recursiva()  public void showPilha\_remove\_Recursiva(Celula topo){      if(topo != null){          System.out.print(topo.elemento + " ");          showPilha\_remove\_Recursiva(topo.prox);      } // end if  } // end showPilha\_Recursiva() |

**Exercício (5):**

Seja nossa Pilha, faça um método RECURSIVO para mostrar os elementos da pilha na ordem em que os mesmos foram inseridos.

|  |
| --- |
| public void showPilha\_inserir\_Recursiva(){      System.out.print("[ ");      showPilha\_inserir\_Recursiva(topo);      System.out.print(" ]\n");  } // end showPilha\_Recursiva()  public void showPilha\_inserir\_Recursiva(Celula topo){      if(topo != null){          showPilha\_inserir\_Recursiva(topo.prox);          System.out.print(topo.elemento + " ");      } // end if  } // end showPilha\_Recursiva() |

**Exercício (6):**

Seja nossa Pilha, faça um método ITERATIVO para mostrar os elementos da pilha na ordem em que os mesmos foram inseridos.

|  |
| --- |
| public void showPilha\_inserir() {      int n = 0, j = 0;        for(Celula i = topo; i != null; i = i.prox)          n++;      int []tmp = new int[n];        for(Celula i = topo; i != null; i = i.prox, j++){          tmp[j] = i.elemento;      } // end for        System.out.print("[ ");      for(int i = n-1; i >= 0; i--){          System.out.print(" " + tmp[i]);      } // end for      System.out.print(" ] \n");  } // end showPilha\_inserir() |

**Exercício (7):**

As ilustrações abaixo mostram a execução dos métodos construtor e do inserir de uma pilha, apresente o código dessa classe e desses métodos.

|  |
| --- |
| class Pilha{      private Celula topo;      public Pilha(){          topo = null;      } // end Pilha()  } // end class Pilha() |

|  |
| --- |
| public void inserir(int value){      Celula tmp = new Celula(value);      tmp.prox = topo;      topo = tmp;      tmp = null;  } // end inserir()  public static void main(String []arg) {  pilha.inserir(3);  } // end main() |

|  |
| --- |
| public void inserir(int value){      Celula tmp = new Celula(value);      tmp.prox = topo;      topo = tmp;      tmp = null;  } // end inserir()  public static void main(String []arg) {  pilha.inserir(3);  pilha.inserir(5);  } // end main() |