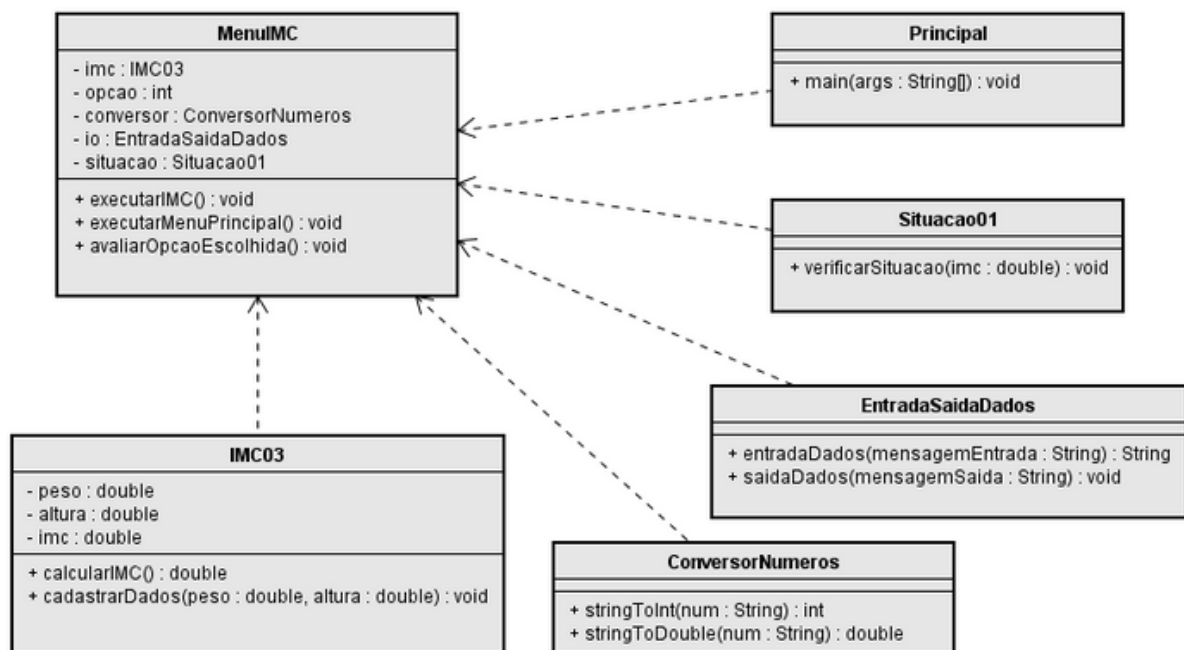


# Atividade DS

Luisa Santos Silva 2ºDS AMS - Tarde

Diagrama e Código da atividade do Projeto IMC

## Diagrama de Classe:



## Códigos de cada Classe:

### Principal:

```
8  /**
9   *
10  * @author Luisa Santos Silva
11  */
12
13  //classe principal
14  public class Principal {
15      //método main()
16      public static void main(String[] args) {
17          MenuIMC menuIMC = new MenuIMC();
18          menuIMC.executarIMC();
19      } //finalizando método main()
20  } //finalizando classe principal
21
```

## MenuIMC:

```
7  /**
8   *
9   * @author Luisa Santos Silva
10  */
11  public class MenuIMC {
12      //atributos da classe MenuIMC
13      private IMC03 imc;
14      private int opcao;
15      private ConversorNumeros conversor;
16      private EntradaSaidaDados io;
17      private Situacao01 situacao;
18
19      public MenuIMC() {
20          this.imc = new IMC03();
21          this.opcao = -1;
22          this.conversor = new ConversorNumeros();
23          this.io = new EntradaSaidaDados();
24          this.situacao = new Situacao01();
25      }
26
27      //método executarIMC()
28      public void executarIMC() {
29          do {
30              this.executarMenuPrincipal();
31              this.avaliarOpcaoEscolhida();
32          } while (this.opcao != 5);
33      }
34
35      //método executarMenuPrincipal()
36      public void executarMenuPrincipal() {
37          String mensagemMenu = "Selecione uma opção:\n"
38              + "1 - Cadastrar Peso e Altura\n"
39              + "2 - Consultar Dados Cadastrados\n"
40              + "3 - Calcular IMC\n"
41              + "4 - Verificar Situação\n"
42              + "0 - Sair\n";
43
44          String opcao = io.entradaDados(mensagemMenu);
45          this.opcao = conversor.stringToInt(opcao);
46      }
47
48      //método avaliarOpcaoEscolhida()
49      public void avaliarOpcaoEscolhida() {
50          //estrutura de decisão
51          switch (this.opcao) {
52              case 1:
53                  imc.cadastrarDados(conversor.stringToDouble(io.entradaDados("Digite seu peso:")),
54                      conversor.stringToDouble(io.entradaDados("Digite sua Altura:")));
55                  io.saidaDados("Peso e altura cadastrados com sucesso!");
56                  break;
57              case 2:
58                  io.saidaDados("Dados Cadastrados:");
59                  io.saidaDados("Peso: " + imc.getPeso() + " kg" +
60                      "\nAltura: " + imc.getAltura() + " m");
61                  break;
62              case 3:
63                  imc.calcularIMC();
64                  io.saidaDados("\nIMC Calculado: " + imc.getImc());
65                  break;
66              case 4:
67                  double imcValor = imc.getImc();
68                  situacao.verificarSituacao(imcValor);
69                  break;
70              case 0:
71                  io.saidaDados("Finalizando o Programa...");
72                  System.exit(0);
73                  break;
74              default:
75                  io.saidaDados("Opção Inválida");
76                  break;
77          }
78      }
79      } //finalizando método avaliarOpcaoEscolhida()
80  } //finalizando classe MenuIMC
81
82
```

## IMC03:

```

7  /**
8   *
9   * @author Luisa Santos Silva
10  */
11
12  //classe IMC03
13  public class IMC03 {
14
15      // Atributos da classe IMC03
16      private double peso;
17      private double altura;
18      private double imc;
19
20      public IMC03() {
21          this(0, 0, 0);
22      }
23
24      public IMC03(double peso, double altura, double imc) {
25          this.peso = peso;
26          this.altura = altura;
27          this.imc = imc;
28      }
29
30      //métodos construtores
31      public double getPeso() {
32          return peso;
33      }
34      public void setPeso(double peso) {
35          this.peso = peso;
36      }
37      public double getAltura() {
38          return altura;
39      }
40      public void setAltura(double altura) {
41          this.altura = altura;
42      }
43      public double getImc() {
44          return imc;
45      }
46      public void setImc(double imc) {
47          this.imc = imc;
48      }
49
50      //método calcularIMC()
51      public double calcularIMC() {
52          setImc(this.getPeso() / (this.getAltura() * this.getAltura()));
53          return getImc();
54      }
55
56      // Método cadastrarDados()
57      public void cadastrarDados(double peso, double altura) {
58          setPeso(peso);
59          setAltura(altura);
60      }
61
62  } //finalizando classe IMC03
63

```

## ConversorNumeros:

```

7  /**
8   *
9   * @author Luisa Santos Silva
10  */
11
12  //classe ConversorNumeros
13  public class ConversorNumeros {
14
15      // Método que converte uma string para int
16      public int stringToInt(String num) {
17          int conversor = Integer.parseInt(num);
18          return conversor;
19      }
20
21      // Método que converte uma string para double
22      public double stringToDouble(String num) {
23          double conversor = Double.parseDouble(num);
24          return conversor;
25      }
26  } //finalizando ConversorNumeros
27

```

## EntradaSaidaDados:

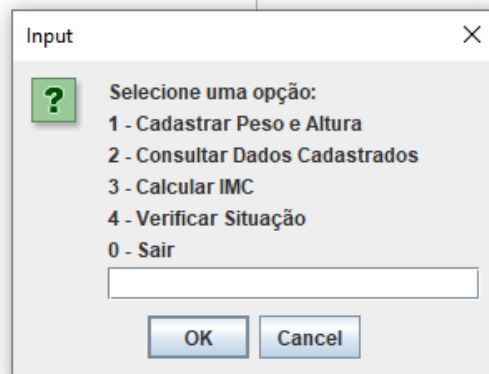
```
7  /**
8   *
9   * @author Luisa Santos Silva
10  */
11
12  //importando javax.swing.JOptionPane
13  import javax.swing.JOptionPane;
14
15  //classe EntradaSaidaDados
16  public class EntradaSaidaDados {
17
18      //metodo que realiza a entrada de dados
19      public String entradaDados(String mensagemEntrada){
20          return JOptionPane.showInputDialog(mensagemEntrada);
21      }
22
23      //metodo que realiza a saida de dados
24      public void saidaDados(String mensagemSaida){
25          JOptionPane.showMessageDialog(null, mensagemSaida);
26      }
27  } //finalizando classe EntradaSaidaDados
28
```

## Situacao01:

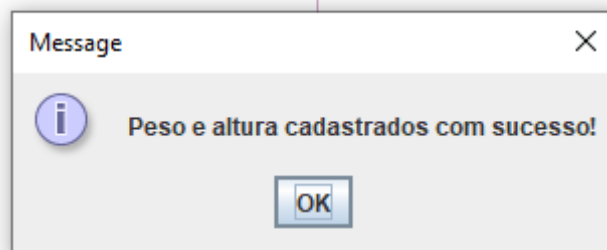
```
7  /**
8   *
9   * @author Luisa Santos Silva
10  */
11
12  //importando javax.swing.JOptionPane;
13  import javax.swing.JOptionPane;
14
15  //classe Situacao01
16  public class Situacao01 {
17
18      //método verificarSituacao()
19      public void verificarSituacao(double imc) {
20          if (imc < 18.5) {
21              JOptionPane.showMessageDialog(null, "Você está classificado com a situação de Abaixo do peso");
22          } else if (imc >= 18.5 && imc <= 24.9) {
23              JOptionPane.showMessageDialog(null, "Você está classificado com a situação de Dentro do Normal");
24          } else if (imc > 24.9 && imc <= 29.9) {
25              JOptionPane.showMessageDialog(null, "Você está classificado com a situação de Sobrepeso");
26          } else if (imc > 29.9 && imc <= 34.9) {
27              JOptionPane.showMessageDialog(null, "Você está classificado com a situação de Obesidade");
28          } else if (imc > 34.9 && imc <= 39.9) {
29              JOptionPane.showMessageDialog(null, "Você está classificado com a situação de Obesidade Severa");
30          } else if (imc > 39.9) {
31              JOptionPane.showMessageDialog(null, "Você está classificado com a situação de Obesidade Mórbida");
32          }
33      } //finalizando método verificarSituacao()
34
35  } //finalizando classe Situacao01
36
```

## Resultados:

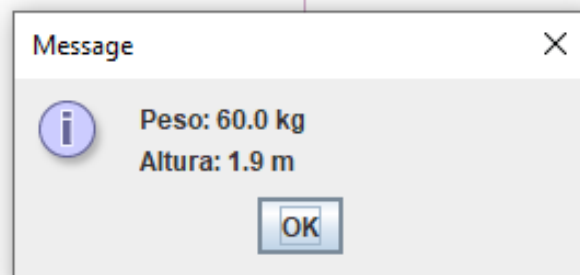
**Ao executar obtive a seguinte tela:**



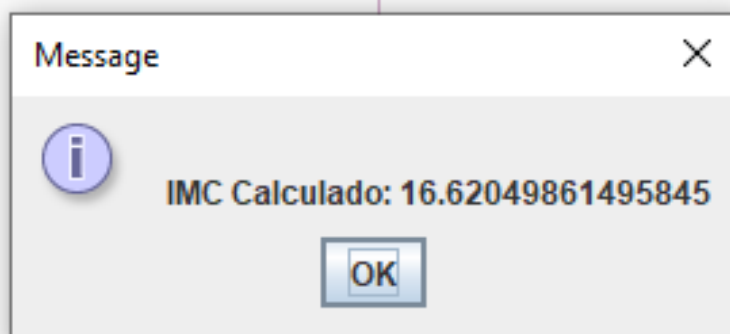
**quando selecionei a opção 1 e cadastrei o peso e altura:**



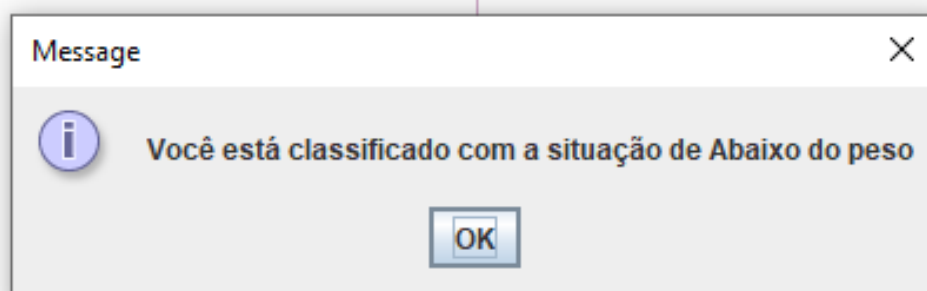
**Depois selecionei a opção 2 e e consultei o que foi cadastrado:**



**Depois selecionei a opção 3 e calculei o IMC:**



**Depois selecionei a opção 4 e obtive sua situação:**



**Depois selecionei a opção 0 e sai da execução.**

**OBS: Na atividade postei um vídeo da execução  
acontecendo também.**