

Prova JAVA

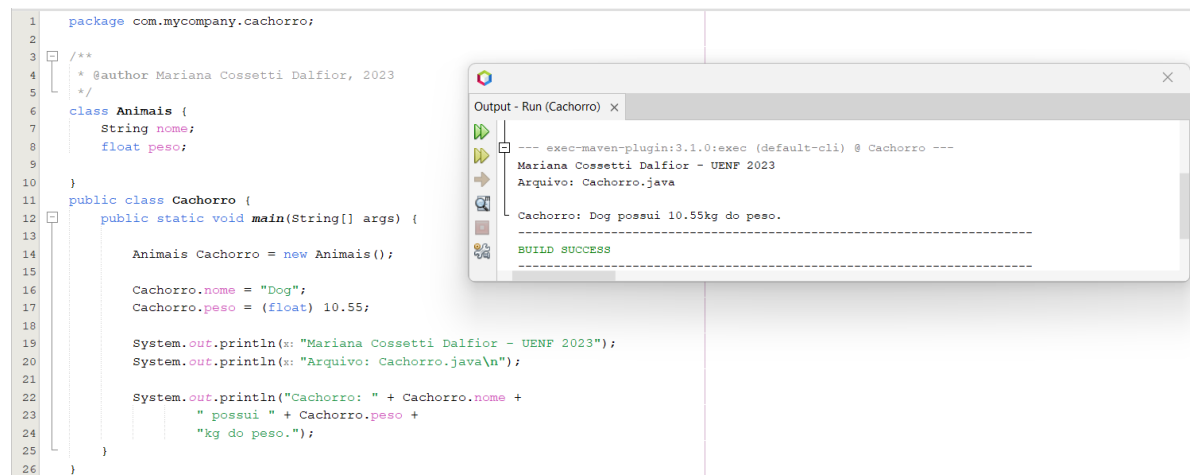
Nome Completo: Mariana Cossetti Dalfior

Data: 12 de junho de 2023

1. **(1 pontos)** Responda detalhadamente as seguintes questões (**exemplifique**):

a. Qual é a diferença entre uma classe e um objeto?

R: Classe é um conceito que encapsula abstrações de atributos e métodos que descrevem o conteúdo e/ou comportamento de uma entidade do mundo real, já o objeto é a instânciação de uma classe. Isso é, a classe é um molde para a criação de variáveis com suas propriedades e ações, que são chamadas de objetos. Um exemplo de classe seria animais e um objeto dessa classe seria cachorro:



```
1 package com.mycompany.cachorro;
2
3 /**
4  * @author Mariana Cossetti Dalfior, 2023
5  */
6 class Animais {
7     String nome;
8     float peso;
9 }
10
11 public class Cachorro {
12     public static void main(String[] args) {
13
14         Animais Cachorro = new Animais();
15
16         Cachorro.nome = "Dog";
17         Cachorro.peso = (float) 10.55;
18
19         System.out.println(x: "Mariana Cossetti Dalfior - UENF 2023");
20         System.out.println(x: "Arquivo: Cachorro.java\n");
21
22         System.out.println("Cachorro: " + Cachorro.nome +
23             " possui " + Cachorro.peso +
24             "kg do peso.");
25     }
26 }
```

Output - Run (Cachorro) x

--- exec-maven-plugin:3.1.0:exec (default-cli) @ Cachorro ---

Mariana Cossetti Dalfior - UENF 2023

Arquivo: Cachorro.java

Cachorro: Dog possui 10.55kg do peso.

BUILD SUCCESS

b. Explique como se define uma classe em Java.

R: Para a definição de uma classe em Java é utilizado a “class” seguido pelo nome que essa classe vai receber, podendo receber public, protected ou private como modificadores de acesso a essas classes para controlar a visibilidade dessas para outras classes. Uma classe geralmente define o estado e o comportamento de um objeto implementando seus atributos e seus métodos. Sintaxe de uma classe:

```
public class Classe {
    tipo atributo1;
    tipo atributo2;

    public Classe(tipo atributo1, tipo atributo2) {
```

```

        this.atributo1 = atributo1;
        this.atributo2 = atributo2;
    }

    public void método1(){
    }

    public tipo método2(){
        return valor;
    }
}

```

Exemplo:

```

1 package com.mycompany.aluno;
2
3 /**
4  * @author Mariana Cossetti Dalfior, 2023
5  */
6 class Estudante {
7     int matricula;
8     String nome;
9     String curso;
10
11     public void irAFaculdade(){
12         System.out.println("está indo a faculdade!\n");
13     }
14 }

```

c. Por que uma classe é uma abstração lógica?

R: Porque para conseguir se concentrar nos aspectos essenciais de uma entidade é necessário isolar os atributos comuns relevantes de um conjunto. Assim, podendo especializar uma classe utilizando a herança e o polimorfismo.

Exemplo:

```

1 package com.mycompany.aluno;
2
3 /**
4  * @author Mariana Cossetti Dalfior, 2023
5  */
6 class Estudante {
7     int matricula;
8     String nome;
9     String curso;
10
11     public void irAFaculdade(){
12         System.out.println("está indo a faculdade!\n");
13     }
14 }
15
16 public class Aluno {
17     public static void main(String[] args) {
18         Estudante alunoA = new Estudante();
19
20         alunoA.matricula = 2021110006;
21         alunoA.nome = "Mariana Cossetti Dalfior";
22         alunoA.curso = "C. Computação";
23
24         System.out.println("\nMariana Cossetti Dalfior - UENF 2023");
25         System.out.println("Arquivo: Aluno.java\n");
26
27         System.out.println("Aluno Nº1: " + alunoA.nome +
28             ", com a matricula: " + alunoA.matricula +
29             ", do curso: " + alunoA.curso + " ");
30
31         alunoA.irAFaculdade();
32     }
33 }

```

Output - Run (Aluno) x

```

Mariana Cossetti Dalfior - UENF 2023
Arquivo: Aluno.java

Aluno Nº1: Mariana Cossetti Dalfior, com a matricula: 2021110006, do curso: C. Computação
está indo a faculdade!

BUILD SUCCESS

```

Como podemos perceber ao criar a classe Estudante foram criados também atributos para ela, em que esses possuem características comuns e relevantes que todos os estudantes possuem.

d. Como os objetos são criados?

R: Um objeto é criado com base em sua classe. Para realizar essa instância é utilizada a palavra reservada new. No caso da criação de um método construtor é necessário a passagem dos parâmetros desse método dentro da instanciação.

Classe objeto = new Classe();

Exemplo:

```
1 package com.mycompany.aluno;
2
3 /**
4  * @author Mariana Cossetti Dalfior, 2023
5  */
6 class Estudante {
7     int matricula;
8     String nome;
9     String curso;
10
11     public void irAFaculdade() {
12         System.out.println("está indo a faculdade!\n");
13     }
14 }
15
16 public class Aluno {
17     public static void main(String[] args) {
18         Estudante alunoA = new Estudante();
19     }
20 }
```

e. O que é coleta de lixo e como ela funciona?

R: Coletor de lixo ou Garbage collector é um processo utilizado para gerenciar memória na JVM, assim garantindo melhor desempenho e escalabilidade. Com ele é possível recuperar uma área de memória inutilizada pelo programa e evitar problemas de esgotamento de memória. Nesse processo é examinada a memória, identificando quais objetos estão em uso (objetos referenciados) e quais estão inutilizados (objetos não referenciados), assim, excluindo os objetos não utilizados e liberando memória de forma automática.

2. **(1 pontos)** Desenhar todas as classes, subclasses e superclasses associadas aos seguintes objetos, indicando em cada caso, pelo menos 2 atributos e 2 métodos, e a relação de herança entre classes

a. Docente



Classe: Pessoa (superclasse)
Atributos: nome, idade;
Métodos: andar(), comer()

Classe: Docente (subclasse de pessoa e superclasse de professor efetivo)
Atributos: disciplina, cargahoraria;
Métodos: ensinar(), aplicarProva();

Classe: Professor efetivo (subclasse)
Atributos: cadastro, especializacao;
Métodos: orientarAluno(), pesquisar();

Relações de herança:
A classe "Professorefetivo" herda da classe "docente" que herda da classe "Pessoa", logo Pessoa é super classe, Docente subclasse de "Pessoa", e Professorefetivo também é uma subclasse.

Pessoa (superclasse) < - Docente (subclasse)
Docente (superclasse) < - Professorefetivo (subclasse)

b. Verdura



Classe: Planta

Atributos: nome, familia;

Métodos: crescer(), respirar()

Classe: Verdura

Atributos: sabor, especie;

Métodos: preparar(), cozinhar();

Classe: Folha

Atributos: cor, textura;

Métodos: lavar(), cortar() ;

Relações de herança:

A classe "Folha" herda da classe "Verdura" que herda da classe "Planta", logo Planta é super classe, Verdura é uma subclasse de "Planta", e Folha também é uma subclasse.

Planta (superclasse) < - Verdura (subclasse)

Verdura (superclasse) < - Folha (subclasse)

c. Ave



Classe: Animais

Atributos: nome, porte;

Métodos: comer(), respirar()

Classe: Ave

Atributos: tamanho_asa, especie;

Métodos: voar(), emitir_som();

Classe: Aveaquatica

Atributos: bico, habitat;

Métodos: nadar(), comer_peixe() ;

Relações de herança:

A classe "Aveaquatica" herda da classe " Ave " que herda da classe " Animais ", logo Animais é super classe, Ave é uma subclasse de " Animais ", e Aveaquatica também é uma subclasse.

Animais (superclasse) < - Ave (subclasse)

Ave (superclasse) < - Aveaquatica (subclasse)

d. Computador



Classe: Eletronicos

Atributos: modelo, preco;
Métodos: ligar(), desligar()

Classe: Computador

Atributos: marca, processador;
Métodos: rodar_programa(), inicializar();

Classe: Notebook

Atributos: bateria, memoria;
Métodos: carregar(), abrir();

Relações de herança:

A classe "Notebook" herda da classe "Computador" que herda da classe "Eletronicos", logo Eletronicos é super classe, Computador é uma subclasse de "Eletronicos", e Notebook também é uma subclasse.

Eletronicos (superclasse) < - Computador (subclasse)

Computador (superclasse) < - Notebook (subclasse)

e. Rede



Classe: Rede

Atributos: ip, velocidade;
Métodos: conectar(), desconectar()

Classe: Roteador

Atributos: portas, preco;
Métodos: configurar(), monitorar();

Classe: Servidor

Atributos: armazenamento, funcoes;
Métodos: processar(), fornecer();

Relações de herança:

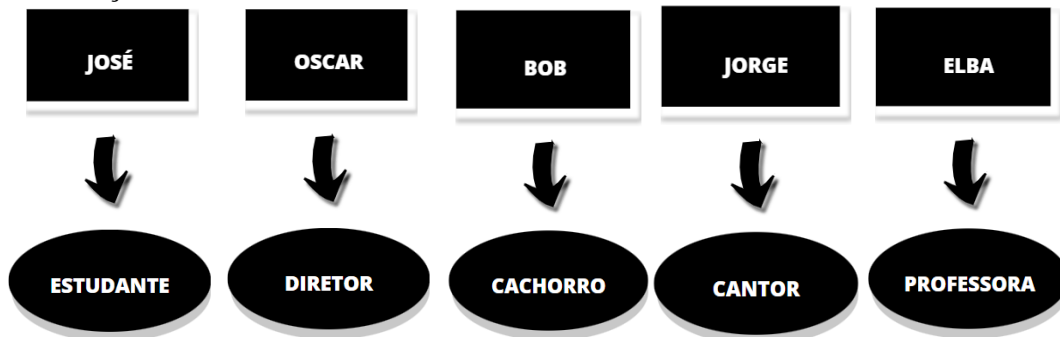
A classe "Servidor" herda da classe "Roteador" que herda da classe "Rede", logo Rede é super classe, Roteador é uma subclasse de "Rede", e Servidor também é uma subclasse.

Rede (superclasse) < - Roteador (subclasse)

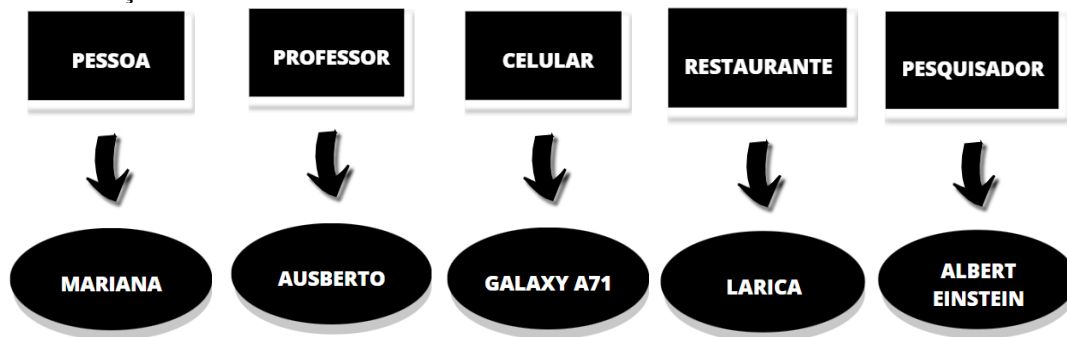
Roteador (superclasse) < - Servidor (subclasse)

3. **(01 ponto)** Dar cinco exemplos de cada operação abstrata em orientação a objetos. Ilustre com imagens ou diagramas.

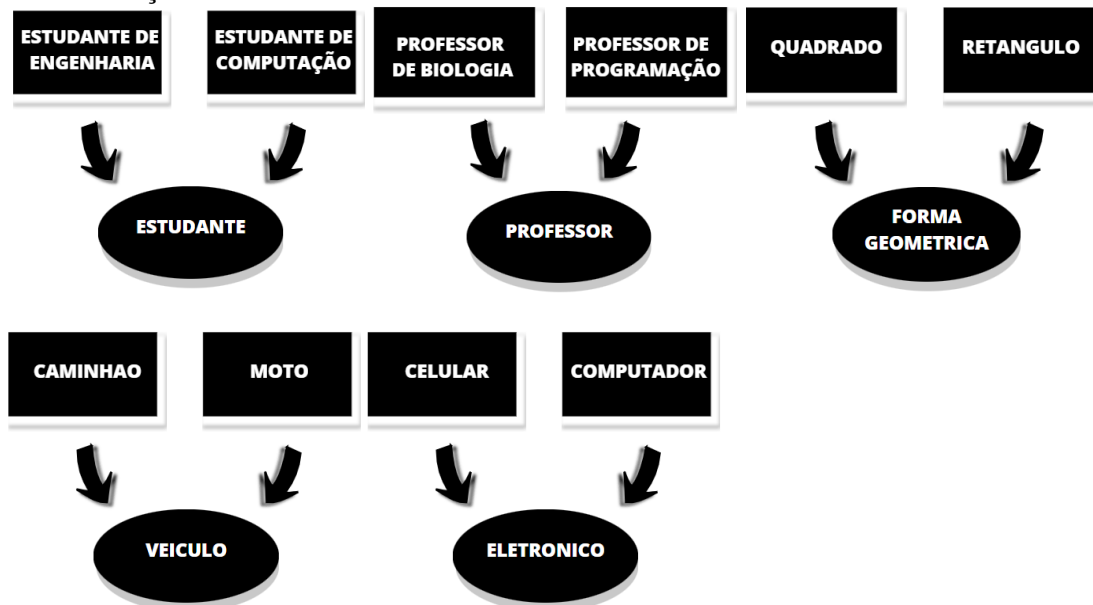
Classificação:



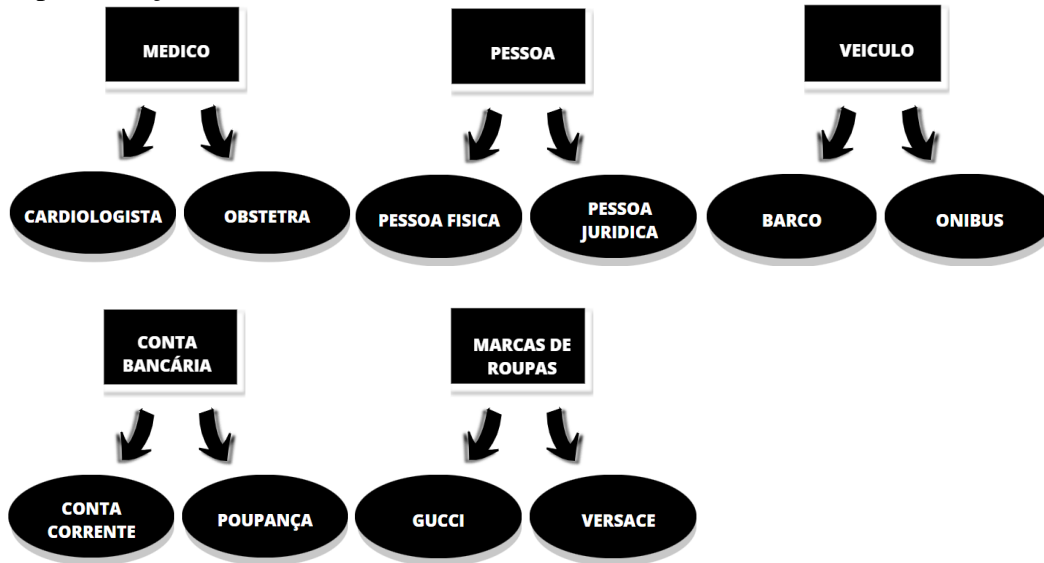
Instanciação:



Generalização:



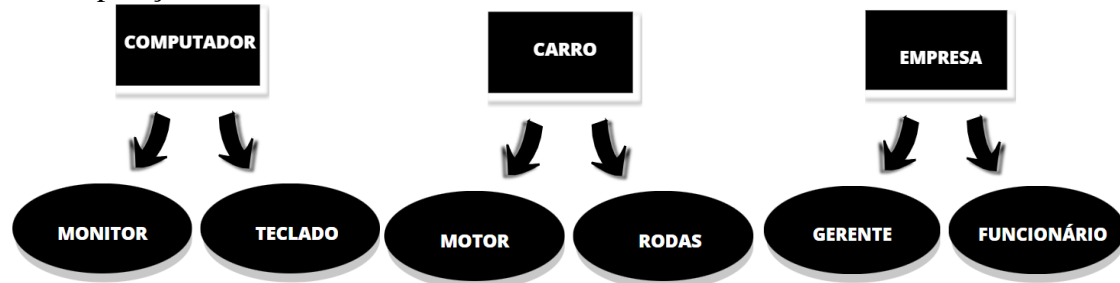
Especialização:

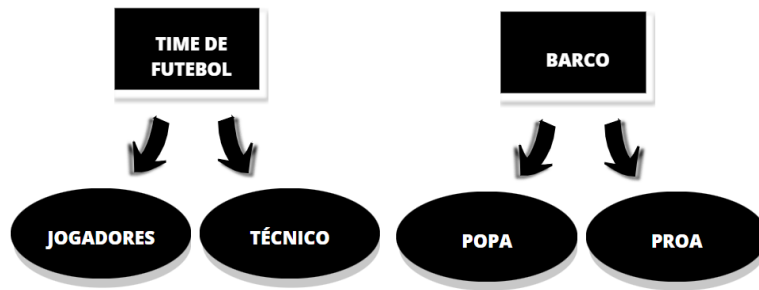


Agregação:



Decomposição:



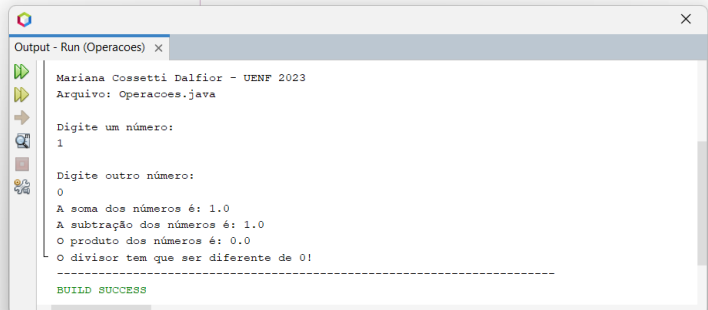


4. **(01 ponto)** Escreva um programa em Java que execute as quatro operações aritméticas de números reais.

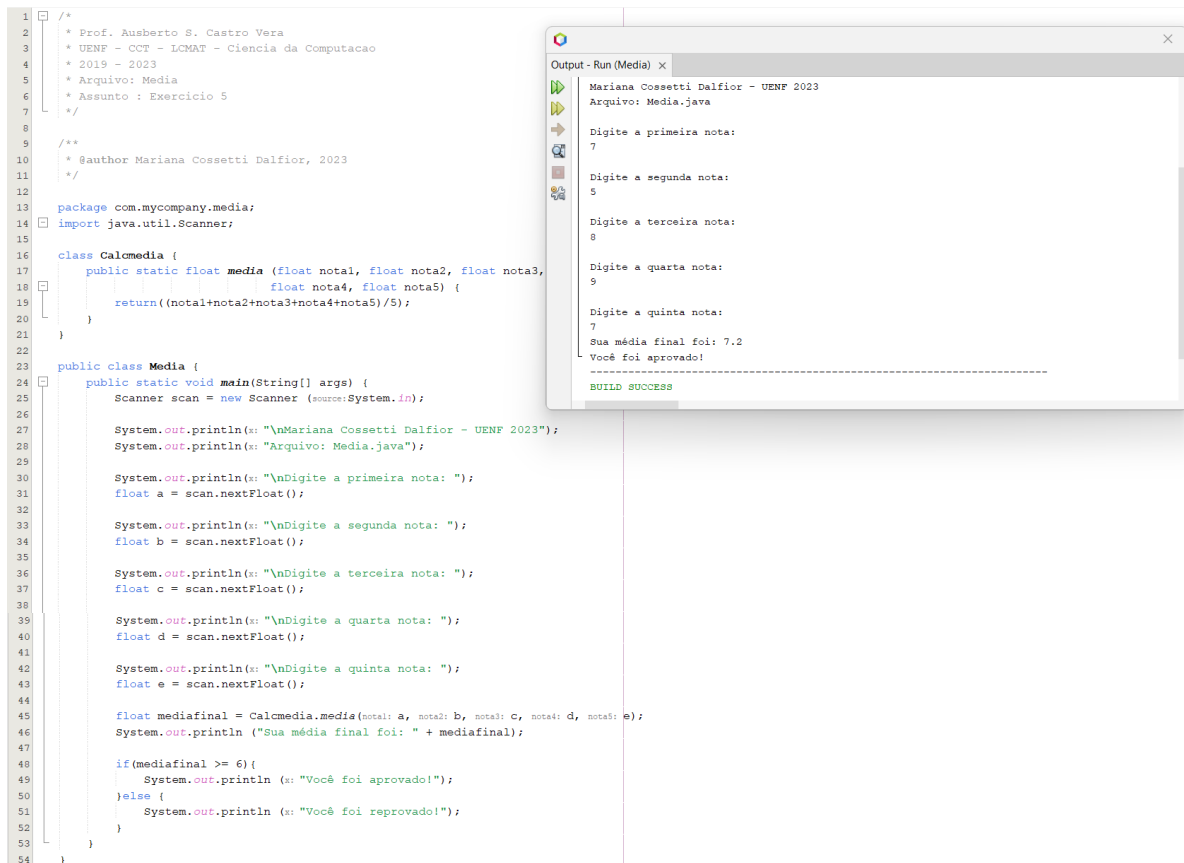
```

1  /*
2   * Prof. Ausberto S. Castro Vera
3   * UENF - CCT - LCMAT - Ciencia da Computacao
4   * 2019 - 2023
5   * Arquivo: Operacoes
6   * Assunto : Exercício 4
7   */
8
9
10 /**
11  * @author Mariana Cossetti Dalfior, 2023
12  */
13 package com.mycompany.operacoes;
14
15 import java.util.Scanner;
16
17 class matematica {
18     public static float soma(float x, float y) {
19         return x+y;
20     }
21
22     public static float subtracao(float x, float y) {
23         return x-y;
24     }
25
26     public static float produto(float x, float y) {
27         return x*y;
28     }
29
30     public static String divisao(float x, float y) {
31         if (y == 0) {
32             return "O divisor tem que ser diferente de 0!";
33         } else {
34             float divisao = (float) x/y;
35             return "O resultado da divisão é: " + divisao;
36         }
37     }
38 }
39
40 public class Operacoes {
41
42     public static void main(String[] args) {
43         Scanner scan = new Scanner (source: System.in);
44
45         System.out.println("\nMariana Cossetti Dalfior - UENF 2023");
46         System.out.println("Arquivo: Operacoes.java");
47
48         System.out.println("\nDigite um número: ");
49         float a = scan.nextFloat();
50
51         System.out.println("\nDigite outro número: ");
52         float b = scan.nextFloat();
53
54         float soma = matematica.soma(x: a, y: b);
55         float subtracao = matematica.subtracao(x: a, y: b);
56         float produto = matematica.produto(x: a, y: b);
57         String divisao = matematica.divisao(x: a, y: b);
58
59         System.out.println("A soma dos números é: " + soma);
60         System.out.println("A subtração dos números é: " + subtracao);
61         System.out.println("O produto dos números é: " + produto);
62         System.out.println(s: divisao);
63     }
64 }

```



5. **(01 ponto)** Escreva um programa JAVA que calcule a média de cinco notas. O programa deve mostrar o valor da média e indicar se foi aprovado ou reprovado.



```
1  /*
2   * Prof. Ausberto S. Castro Vera
3   * UENF - CCT - LCMAT - Ciencia da Computacao
4   * 2019 - 2023
5   * Arquivo: Media
6   * Assunto : Exercicio 5
7   */
8
9  /**
10   * @author Mariana Cossetti Dalfior, 2023
11   */
12
13  package com.mycompany.media;
14  import java.util.Scanner;
15
16  class Calcmidia {
17      public static float media (float nota1, float nota2, float nota3,
18                                float nota4, float nota5) {
19          return((nota1+nota2+nota3+nota4+nota5)/5);
20      }
21  }
22
23  public class Media {
24      public static void main(String[] args) {
25          Scanner scan = new Scanner (source: System.in);
26
27          System.out.println("Mariana Cossetti Dalfior - UENF 2023");
28          System.out.println("Arquivo: Media.java");
29
30          System.out.println("\nDigite a primeira nota: ");
31          float a = scan.nextFloat();
32
33          System.out.println("\nDigite a segunda nota: ");
34          float b = scan.nextFloat();
35
36          System.out.println("\nDigite a terceira nota: ");
37          float c = scan.nextFloat();
38
39          System.out.println("\nDigite a quarta nota: ");
40          float d = scan.nextFloat();
41
42          System.out.println("\nDigite a quinta nota: ");
43          float e = scan.nextFloat();
44
45          float mediafinal = Calcmidia.media(nota1: a, nota2: b, nota3: c, nota4: d, nota5: e);
46          System.out.println ("Sua média final foi: " + mediafinal);
47
48          if(mediafinal >= 6){
49              System.out.println ("Você foi aprovado!");
50          }else {
51              System.out.println ("Você foi reprovado!");
52          }
53      }
54  }
```

Output - Run (Media) x

Mariana Cossetti Dalfior - UENF 2023
Arquivo: Media.java

Digite a primeira nota:
7

Digite a segunda nota:
5

Digite a terceira nota:
8

Digite a quarta nota:
9

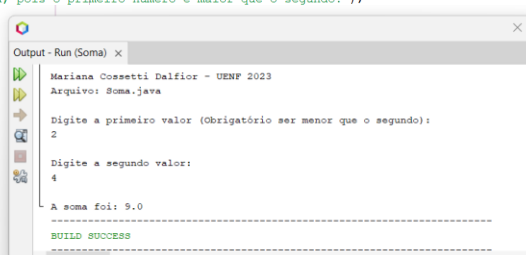
Digite a quinta nota:
7

Sua média final foi: 7.2
Você foi aprovado!

BUILD SUCCESS

6. **(01 ponto)** Escreva um programa JAVA que calcule a soma dos inteiros maiores o iguaes a um número inteiro A e menores ou iguaes a um inteiro B.

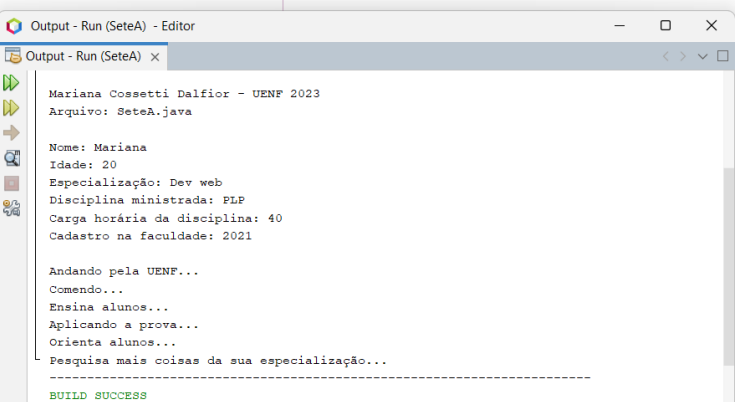
```
1  /*
2   * Prof. Ausberto S. Castro Vera
3   * UENF - CCT - LCMAT - Ciencia da Computacao
4   * 2019 - 2023
5   * Arquivo: Soma
6   * Assunto : Exercicio 6
7   */
8
9  /**
10   * @author Mariana Cossetti Dalfior, 2023
11   */
12
13  package com.mycompany.soma;
14
15  import java.util.Scanner;
16
17  class Calculasoma {
18      public static float calcSoma(float x, float y) {
19          float soma = 0;
20          if (x > y) {
21              System.out.println(":\nNão foi possível realizar a soma, pois o primeiro número é maior que o segundo!");
22          }
23          while (x <= y) {
24              soma += x;
25              x += 1;
26          }
27          return soma;
28      }
29  }
30
31  public class Soma {
32
33      public static void main(String[] args) {
34          Scanner scan = new Scanner (source: System.in);
35
36          System.out.println(":\nMariana Cossetti Dalfior - UENF 2023");
37          System.out.println("Arquivo: Soma.java");
38
39          System.out.println(":\nDigite a primeiro valor (Obrigatório ser menor que o segundo): ");
40          float a = scan.nextFloat();
41
42          System.out.println(":\nDigite a segundo valor: ");
43          float b = scan.nextFloat();
44
45          float soma = Calculasoma.calcSoma(x: a, y: b);
46          System.out.println(":\nA soma foi: " + soma);
47      }
48  }
```



7. **(2 pontos)** Para cada caso da questão 2, escrever um programa JAVA, mostrando as classes, as operações (métodos), a criação de objetos e outras características da Programação Orientada a Objetos.

a) Docente:

```
1  /*
2   * Prof. Ausberto S. Castro Vera
3   * UENF - CCT - LCMAT - Ciencia da Computacao
4   * 2019 - 2023
5   * Arquivo: SeteA
6   * Assunto : Exercicio 7 - a
7   */
8
9  /**
10   * @author Mariana Cossetti Dalfior, 2023
11   */
12
13  package com.mycompany.setea;
14
15  @
16  class Pessoa {
17      String nome;
18      int idade;
19
20      public static String andar () {
21          return "Andando pela UENF...";
22      }
23
24      public static String comer () {
25          return "Comendo...";
26      }
27  }
28
29  @
30  class Docente extends Pessoa{
31      String disciplina;
32      int cargahoraria;
33
34      public static String ensinar () {
35          return "Ensina alunos... ";
36      }
37
38      public static String aplicarProva () {
39          return "Aplicando a prova...";
40      }
41  }
```



```

39
40 class ProfessorEfetivo extends Docente{
41     String especializacao;
42     int cadastro;
43
44     public static String orientarAluno () {
45         return "Orienta alunos...";
46     }
47
48     public static String pesquisar () {
49         return "Pesquisa mais coisas da sua especialização...";
50     }
51 }
52
53 public class SeteA {
54
55     public static void main(String[] args) {
56         System.out.println(x: "\nMariana Cossetti Dalfior - UENF 2023");
57         System.out.println(x: "Arquivo: SeteA.java");
58
59         ProfessorEfetivo pessoaA = new ProfessorEfetivo();
60
61         pessoaA.nome = "Mariana";
62         pessoaA.idade = 20;
63         pessoaA.especializacao = "Dev web";
64         pessoaA.disciplina = "PLP";
65         pessoaA.cargahoraria = 40;
66         pessoaA.cadastro = 2021;
67
68         System.out.println("\nNome: " + pessoaA.nome);
69         System.out.println("Idade: " + pessoaA.idade);
70         System.out.println("Especialização: " + pessoaA.especializacao);
71         System.out.println("Disciplina ministrada: " + pessoaA.disciplina);
72         System.out.println("Carga horária da disciplina: " + pessoaA.cargahoraria);
73         System.out.println("Cadastro na faculdade: " + pessoaA.cadastro + "\n");
74         System.out.println(x: ProfessorEfetivo.andar());
75         System.out.println(x: ProfessorEfetivo.comer());
76         System.out.println(x: ProfessorEfetivo.ensinar());
77         System.out.println(x: ProfessorEfetivo.aplicarProva());
78         System.out.println(x: ProfessorEfetivo.orientarAluno());
79         System.out.println(x: ProfessorEfetivo.pesquisar());
80     }
81 }

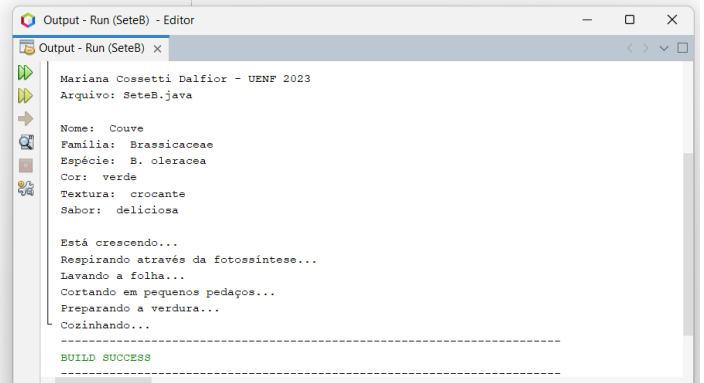
```

b) Verdura:

```

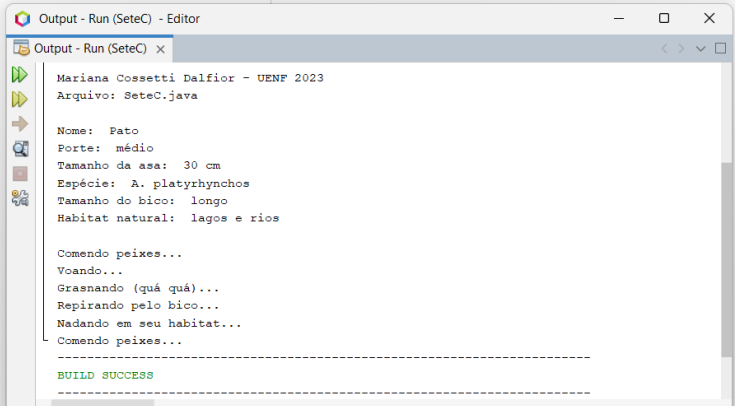
1  /*
2  * Prof. Ausberto S. Castro Vera
3  * UENF - CCT - LCMAT - Ciencia da Computacao
4  * 2019 - 2023
5  * Arquivo: SeteB
6  * Assunto : Exercício 7 - b
7  */
8
9  /**
10 * @author Mariana Cossetti Dalfior, 2023
11 */
12
13 package com.mycompany.seteb;
14
15 @
16 class Planta {
17     String nome;
18     String familia;
19
20     public static String crescer () {
21         return "Está crescendo...";
22     }
23
24     public static String respirar () {
25         return "Respirando através da fotossíntese...";
26     }
27 }
28
29 @
30 class Verdura extends Planta{
31     String sabor;
32     String especie;
33
34     public static String preparar () {
35         return "Preparando a verdura...";
36     }
37
38     public static String cozinhar () {
39         return "Cozinhando...";
40     }
41 }
42
43 class Folha extends Verdura{
44     String cor;
45     String textura;
46
47     public static String lavar () {
48         return "Lavando a folha...";
49     }
50
51     public static String cortar () {
52         return "Cortando em pequenos pedaços...";
53     }
54 }
55
56 public class SeteB {
57
58     public static void main(String[] args) {
59         System.out.println(x: "\nMariana Cossetti Dalfior - UENF 2023");
60         System.out.println(x: "Arquivo: SeteB.java");
61
62         Folha verduraA = new Folha();
63
64         verduraA.nome = "Couve";
65         verduraA.familia = "Brassicaceae";
66         verduraA.especie = "B. oleracea";
67         verduraA.cor = "verde";
68         verduraA.textura = "crocante";
69         verduraA.sabor = "deliciosa";
70
71         System.out.println("\nNome: " + verduraA.nome);
72         System.out.println("Familia: " + verduraA.familia);
73         System.out.println("Especie: " + verduraA.especie);
74         System.out.println("Cor: " + verduraA.cor);
75         System.out.println("Textura: " + verduraA.textura);
76         System.out.println("Sabor: " + verduraA.sabor + "\n");
77         System.out.println(x: Folha.crescer());
78         System.out.println(x: Folha.respirar());
79         System.out.println(x: Folha.lavar());
80         System.out.println(x: Folha.cortar());
81         System.out.println(x: Folha.preparar());
82         System.out.println(x: Folha.cozinhar());
83     }
84 }

```



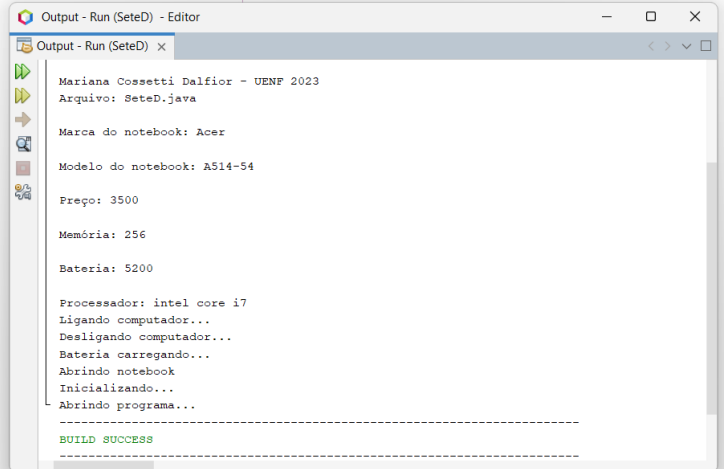
c) Ave:

```
1  /*
2  * Prof. Ausberto S. Castro Vera
3  * UENF - CCT - LCMAT - Ciencia da Computacao
4  * 2019 - 2023
5  * Arquivo: SeteC
6  * Assunto : Exercicio 7 - c
7  */
8
9
10 /**
11  * @author Mariana Cossetti Dalfior, 2023
12  */
13 package com.mycompany.setec;
14
15 @
16 class Animais {
17     String nome;
18     String porte;
19
20     public static String comer () {
21         return "Comendo peixes...";
22     }
23
24     public static String respirar () {
25         return "Respirando pelo bico...";
26     }
27 }
28
29 @
30 class Ave extends Animais{
31     String tamanho_asa;
32     String especie;
33
34     public static String voar () {
35         return "Voando...";
36     }
37
38     public static String emitir_som () {
39         return "Grasnando (quá quá)...";
40     }
41 }
42
43 class AveAquatica extends Ave{
44     String bico;
45     String habitat;
46
47     public static String nadar () {
48         return "Nadando em seu habitat...";
49     }
50
51     public static String comer_peixe () {
52         return "Comendo peixes...";
53     }
54 }
55
56 public class SeteC {
57
58     public static void main(String[] args) {
59         System.out.println("\nMariana Cossetti Dalfior - UENF 2023");
60         System.out.println("Arquivo: SeteC.java");
61
62         AveAquatica animalA = new AveAquatica();
63
64         animalA.nome = "Pato";
65         animalA.porte = "médio";
66         animalA.tamanho_asa = "30 cm";
67         animalA.especie = "A. platyrhynchos";
68         animalA.bico = "longo";
69         animalA.habitat = "lagos e rios";
70
71         System.out.println("\nNome: " + animalA.nome);
72         System.out.println("Porte: " + animalA.porte);
73         System.out.println("Tamanho da asa: " + animalA.tamanho_asa);
74         System.out.println("Espécie: " + animalA.especie);
75         System.out.println("Tamanho do bico: " + animalA.bico);
76         System.out.println("Habitat natural: " + animalA.habitat + "\n");
77         System.out.println("Comendo peixes...");
78         System.out.println("Voando...");
79         System.out.println("Grasnando (quá quá)...");
80         System.out.println("Respirando pelo bico...");
81         System.out.println("Nadando em seu habitat...");
82         System.out.println("Comendo peixes...");
83     }
84 }
```



d) Computador:

```
1  /*
2  * Prof. Ausberto S. Castro Vera
3  * UENF - CCT - LCMAT - Ciencia da Computacao
4  * 2019 - 2023
5  * Arquivo: SeteD
6  * Assunto : Exercício 7 - d
7  */
8
9  /**
10   * @author Mariana Cossetti Dalfior, 2023
11   */
12  package com.mycompany.seteD;
13
14  @
15  class Eletronicos {
16      String modelo;
17      int preco;
18
19      public static String ligar () {
20          return "Ligando computador...";
21      }
22
23      public static String desligar () {
24          return "Desligando computador...";
25      }
26  }
27
28  @
29  class Computador extends Eletronicos{
30      String marca;
31      String processador;
32
33      public static String rodar_programa() {
34          return "Abrindo programa...";
35      }
36
37      public static String inicializar() {
38          return "Inicializando...";
39      }
40  }
41
42  class Notebook extends Computador{
43      int bateria;
44      int memoria;
45
46      public static String carregar () {
47          return "Bateria carregando...";
48      }
49
50      public static String abrir() {
51          return "Abrindo notebook";
52      }
53  }
54
55  public class SeteD {
56
57      public static void main(String[] args) {
58          System.out.println("\nMariana Cossetti Dalfior - UENF 2023");
59          System.out.println("\nArquivo: SeteD.java");
60
61          Notebook noteA = new Notebook();
62
63          noteA.modelo = "A514-54";
64          noteA.preco = 3500;
65          noteA.marca = "Acer";
66          noteA.processador = "intel core i7";
67          noteA.bateria = 5200;
68          noteA.memoria = 256;
69
70          System.out.println("\nMarca do notebook: " + noteA.marca);
71          System.out.println("\nModelo do notebook: " + noteA.modelo);
72          System.out.println("\nPreço: " + noteA.preco);
73          System.out.println("\nMemória: " + noteA.memoria);
74          System.out.println("\nBateria: " + noteA.bateria);
75          System.out.println("\nProcessador: " + noteA.processador);
76          System.out.println("\nLigando computador...");
77          System.out.println("\nDesligando computador...");
78          System.out.println("\nBateria carregando...");
79          System.out.println("\nAbrindo notebook");
80          System.out.println("\nInicializando...");
81          System.out.println("\nAbrindo programa...");
82      }
83  }
```



e) Rede:

```
1  /*
2  * Prof. Ausberto S. Castro Vera
3  * UENF - CCT - LCMAT - Ciencia da Computacao
4  * 2019 - 2023
5  * Arquivo: SeteE
6  * Assunto : Exercício 7 - e
7  */
8
9
10 /**
11  * @author Mariana Cossetti Dalfior, 2023
12  */
13 package com.mycompany.setee;
14
15 @
16 class Rede {
17     String ip;
18     int velocidade;
19
20     public static String conectar () {
21         return "\nConectando a rede...";
22     }
23
24     public static String desconectar () {
25         return "\nDesconectando da rede... ";
26     }
27 }
28
29 @
30 class Roteador extends Rede{
31     int portas;
32     int preco;
33
34     public static String configurar () {
35         return "\nConfigurando o roteador...";
36     }
37
38     public static String monitorar () {
39         return "\nMonitorando tráfego no roteador...";
40     }
41 }
42
43 class Servidor extends Roteador{
44     String armazenamento;
45     String funcoes;
46
47     public static String processar () {
48         return "\nProcessando informações recebidas...";
49     }
50
51     public static String fornecer () {
52         return "\nFornecendo informações...";
53     }
54 }
55
56 public class SeteE {
57
58     public static void main(String[] args) {
59         System.out.println("\nMariana Cossetti Dalfior - UENF 2023");
60         System.out.println("\nArquivo: SeteE.java");
61
62         Servidor redeA = new Servidor();
63
64         redeA.ip = "123.21.233";
65         redeA.velocidade = 400;
66         redeA.portas = 4;
67         redeA.preco = 100;
68         redeA.armazenamento = "1 tb";
69         redeA.funcoes = "Manda informações";
70
71         System.out.println("\nIp da rede: " + redeA.ip);
72         System.out.println("\nVelocidade da rede: " + redeA.velocidade);
73         System.out.println("\nQuantidade de portas: " + redeA.portas);
74         System.out.println("\nPreço: " + redeA.preco);
75         System.out.println("\nArmazenamento do servidor: " + redeA.armazenamento);
76         System.out.println("\nFunção do servidor: " + redeA.funcoes);
77         System.out.println("\nConectando a rede...");
78         System.out.println("\nDesconectando da rede...");
79         System.out.println("\nConfigurando o roteador...");
80         System.out.println("\nMonitorando tráfego no roteador...");
81         System.out.println("\nProcessando informações recebidas...");
82         System.out.println("\nFornecendo informações...");
83
84         BUILD SUCCESS
85     }
86 }
```

8. **(02 pontos)** Resolver a Questão 9.10 e 9.11 (pág.363) do livro “Introduction to JAVA Programming”, Y. Daniel Liang, 2014. Não precisa desenhar os diagramas UML. “Implemente a classe” = “Faça um programa em Java”

a) 9.10:

```
1  /*
2  * Prof. Ausberto S. Castro Vera
3  * UENF - CCT - LCMAT - Ciencia da Computacao
4  * 2019 - 2023
5  * Arquivo: Equacao
6  * Assunto : Exercicio 8 - 9.10
7  */
8
9  /**
10   * @author Mariana Cossetti Dalfior, 2023
11   */
12  package com.mycompany.equacao;
13
14  import java.util.Scanner;
15
16  class EquacaoSeg {
17      private int a;
18      private int b;
19      private int c;
20
21      public EquacaoSeg () {
22          Scanner scan = new Scanner(System.in);
23
24          System.out.println("Valor de a: ");
25          a = scan.nextInt();
26
27          System.out.println("Valor de b: ");
28          b = scan.nextInt();
29
30          System.out.println("Valor de c: ");
31          c = scan.nextInt();
32      }
33
34      public double getA() {
35          return a;
36      }
37
38      public double getB() {
39          return b;
40      }
41
42      public double getC() {
43          return c;
44      }
45
46      public double getDiscriminant() {
47          return ((b*b) - (4 * a * c));
48      }
49
50      public double getRoot1(double delta) {
51          if (delta >= 0) {
52              return (((- b) + (Math.sqrt(a: delta))) / (2 * a));
53          } else {
54              return 0;
55          }
56      }
57
58      public double getRoot2(double delta) {
59          if (delta >= 0) {
60              return (((- b) - (Math.sqrt(a: delta))) / (2 * a));
61          } else {
62              return 0;
63          }
64      }
65  }
66
67  public class Equacao {
68
69      public static void main(String[] args) {
70          System.out.println("Mariana Cossetti Dalfior - UENF 2023");
71          System.out.println("Arquivo: Equacao.java");
72
73          EquacaoSeg equa = new EquacaoSeg();
74
75          double a = equa.getA();
76          double b = equa.getB();
77          double c = equa.getC();
78          double delta = equa.getDiscriminant();
79
80          System.out.println("Equação: " + a + "x² + " + b + "x + " + c + " = 0");
81          System.out.println("Delta = " + delta);
82
83          double res1 = equa.getRoot1(delta);
84          double res2 = equa.getRoot2(delta);
85
86          System.out.println("Resposta com delta positivo: " + res1);
87          System.out.println("Resposta com delta negativo: " + res2);
88      }
89  }
90  }
```

Output - Run (Equacao) - Editor

Output - Run (Equacao) x

Mariana Cossetti Dalfior - UENF 2023

Arquivo: Equacao.java

Valor de a:
2

Valor de b:
-7

Valor de c:
3

Equação: 2.0x² + -7.0x + 3.0 = 0

Delta = 25.0

Resposta com delta positivo: 3.0

Resposta com delta negativo: 0.5

BUILD SUCCESS

b) 9.11:


```

1  /*
2  * Prof. Ausberto S. Castro Vera
3  * UENF - CCT - LCMAT - Ciencia da Computacao
4  * 2019 - 2023
5  * Arquivo: Algebra
6  * Assunto : Exercício 8 - 9.11
7  */
8
9  /**
10   * @author Mariana Cossetti Dalfior, 2023
11   */
12   package com.mycompany.algebra;
13
14   import java.util.Scanner;
15
16   class LinearEquation {
17       private int a;
18       private int b;
19       private int c;
20       private int d;
21       private int e;
22       private int f;
23
24       public LinearEquation() {
25           Scanner scan = new Scanner(System.in);
26
27           System.out.println("Valor de a: ");
28           a = scan.nextInt();
29
30           System.out.println("Valor de b: ");
31           b = scan.nextInt();
32
33           System.out.println("Valor de c: ");
34           c = scan.nextInt();
35
36           System.out.println("Valor de d: ");
37           d = scan.nextInt();
38
39           System.out.println("Valor de e: ");
40           e = scan.nextInt();
41
42           System.out.println("Valor de f: ");
43           f = scan.nextInt();
44       }
45
46       public int getA() {
47           return a;
48       }
49
50       public int getB() {
51           return b;
52       }
53
54       public int getC() {
55           return c;
56       }
57
58       public int getD() {
59           return d;
60       }
61
62       public int getE() {
63           return e;
64       }
65
66       public int getF() {
67           return f;
68       }
69
70       public Boolean isSolvable() {
71           int m = ((a * d) - (b * c));
72           return (m != 0);
73       }
74
75       public float getX() {
76           return ((e * d) - (b * f)) / ((a * d) - (b * c));

```

Output - Run (Algebra) - Editor

Output - Run (Algebra) x

Mariana Cossetti Dalfior - UENF 2023

Arquivo: Algebra.java

Valor de a:
1

Valor de b:
2

Valor de c:
3

Valor de d:
4

Valor de e:
5

Valor de f:
6

$1x + 2y = 5$
 $3x + 4y = 6$

É solucionável!
X = -4.0
y = 4.0

BUILD SUCCESS

```

77     }
78
79     public float getY() {
80         return ((a * f) - (e * c)) / ((a * d) - (b * c));
81     }
82 }
83
84 public class Algebra {
85
86     public static void main(String[] args) {
87         System.out.println(x: "\nMariana Cossetti Dalfior - UENF 2023");
88         System.out.println(x: "Arquivo: Algebra.java");
89
90         LinearEquation eqLin = new LinearEquation();
91
92         int a = eqLin.getA();
93         int b = eqLin.getB();
94         int c = eqLin.getC();
95         int d = eqLin.getD();
96         int e = eqLin.getE();
97         int f = eqLin.getF();
98
99         System.out.println("\n" + a + "x + " + b + "y = " + e);
100         System.out.println(c + "x + " + d + "y = " + f);
101
102         boolean soluc = eqLin.isSolvable();
103         if (soluc == true) {
104             System.out.println(x: "\nÉ solucionável! ");
105
106             float X = eqLin.getX();
107             float Y = eqLin.getY();
108
109             System.out.println("X = " + X);
110             System.out.println("y = " + Y);
111         } else {
112             System.out.println(x: "Não é solucionável! ");
113         }
114     }

```

O professor

***9.10** (Algebra: quadratic equations) Design a class named `QuadraticEquation` for a quadratic equation $ax^2 + bx + c = 0$. The class contains:

- Private data fields `a`, `b`, and `c` that represent three coefficients.
- A constructor for the arguments for `a`, `b`, and `c`.
- Three getter methods for `a`, `b`, and `c`.
- A method named `getDiscriminant()` that returns the discriminant, which is $b^2 - 4ac$.
- The methods named `getRoot1()` and `getRoot2()` for returning two roots of the equation

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

These methods are useful only if the discriminant is nonnegative. Let these methods return 0 if the discriminant is negative.

Draw the UML diagram for the class and then implement the class. Write a test program that prompts the user to enter values for `a`, `b`, and `c` and displays the result based on the discriminant. If the discriminant is positive, display the two roots. If the discriminant is 0, display the one root. Otherwise, display “The equation has no roots.” See Programming Exercise 3.1 for sample runs.

***9.11** (Algebra: 2×2 linear equations) Design a class named `LinearEquation` for a 2×2 system of linear equations:

$$\begin{array}{rcl} ax + by = e & & \\ cx + dy = f & \Rightarrow & x = \frac{ed - bf}{ad - bc} \quad y = \frac{af - ec}{ad - bc} \end{array}$$

The class contains:

- Private data fields `a`, `b`, `c`, `d`, `e`, and `f`.
- A constructor with the arguments for `a`, `b`, `c`, `d`, `e`, and `f`.
- Six getter methods for `a`, `b`, `c`, `d`, `e`, and `f`.
- A method named `isSolvable()` that returns true if $ad - bc$ is not 0.
- Methods `getX()` and `getY()` that return the solution for the equation.

Draw the UML diagram for the class and then implement the class. Write a test program that prompts the user to enter `a`, `b`, `c`, `d`, `e`, and `f` and displays the result. If $ad - bc$ is 0, report that “The equation has no solution.” See Programming Exercise 3.3 for sample runs.

****9.12** (Geometry: intersecting point) Suppose two line segments intersect. The two end-