

Prova JAVA

Nome Completo: Rômulo Souza Fernandes

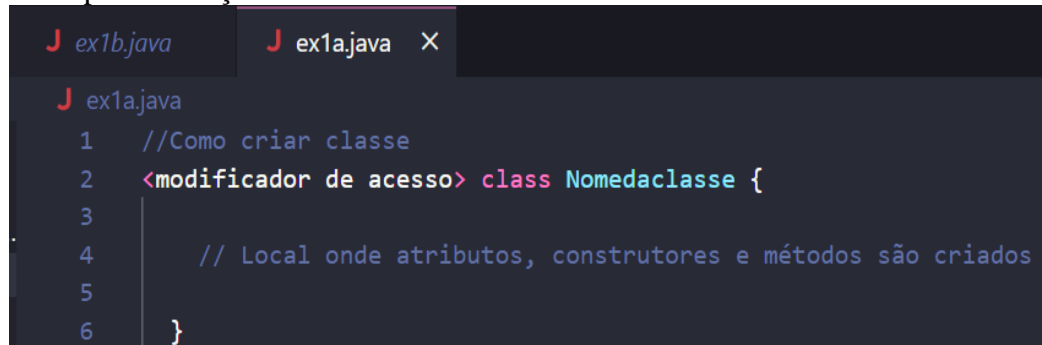
Data: 22 de outubro de 2022

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

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

Na programação orientada a objetos uma classe possui determinados atributos, que são características. Esses atributos podem possuir um valor inicial, que é determinado por um construtor, na sua criação, o atributo então tem um valor padrão que é definido, além das ações realizadas sobre as classes ou pelas classes, chamamos isso de procedimentos. Contudo, entendemos que uma classe é um molde que informa os atributos que são as características e/ou os procedimentos que são as ações que a classe pode realizar.

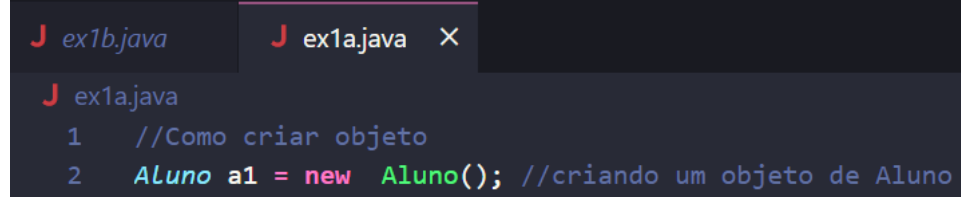
Exemplo da criação de uma classe:



```
J ex1b.java J ex1a.java X
J ex1a.java
1 //Como criar classe
2 <modificador de acesso> class Nomedaclass {
3
4     // Local onde atributos, construtores e métodos são criados
5
6 }
```

Na programação orientada a objetos um objeto é definido por uma classe e também é baseado em uma. Assim, quando um objeto é criado com base em uma certa classe, ele assume que possuirá os mesmos métodos e atributos que foram definidos pela classe. Podendo criar diversos objetos diferentes entre si que possuem o mesmo molde.

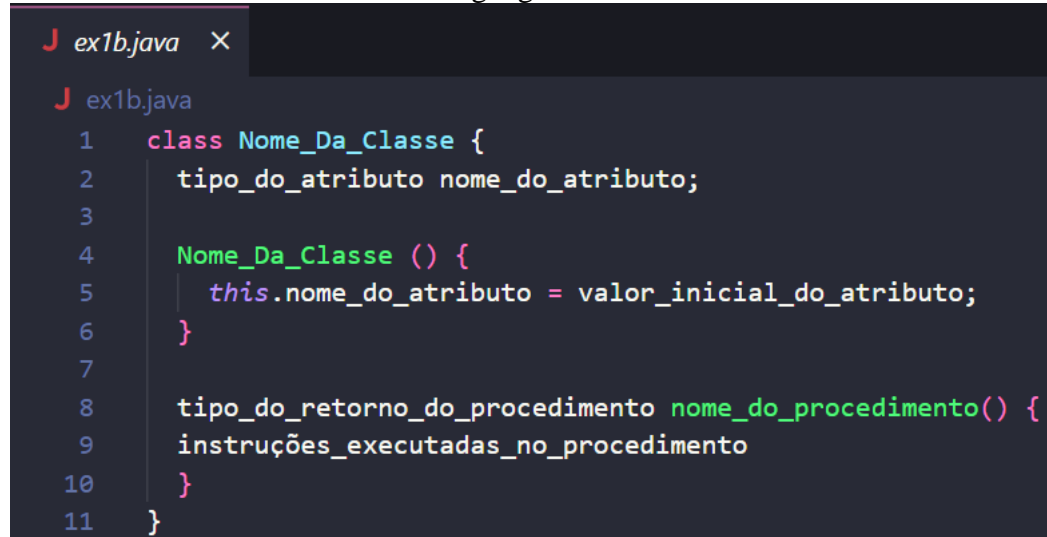
Exemplo da criação de um objeto:



```
J ex1b.java J ex1a.java X
J ex1a.java
1 //Como criar objeto
2 Aluno a1 = new Aluno(); //criando um objeto de Aluno
```

b. Explique como se define uma classe em Java

Para se definir uma classe na linguagem Java é necessário:



```
J ex1b.java X
J ex1b.java
1 class Nome_Da_Classe {
2     tipo_do_atributo nome_do_atributo;
3
4     Nome_Da_Classe () {
5         this.nome_do_atributo = valor_inicial_do_atributo;
6     }
7
8     tipo_do_retorno_do_procedimento nome_do_procedimento() {
9         instruções_executadas_no_procedimento
10    }
11 }
```

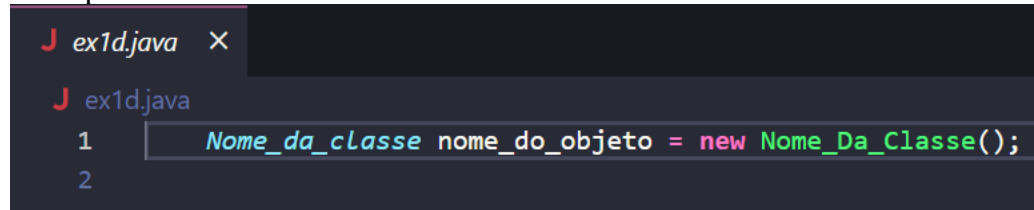
c. Por que uma classe é uma abstração lógica?

Para que uma classe seja criada é necessário compreender o seu funcionamento essencial e assim é possível definir amplamente, essa definição ampla é definido como abstração do objeto, objeto esse que temos como objetivo definir as ações e as características. A abstração necessita ser lógica e ter coerência, com isso uma classe é uma abstração lógica.

d. Como os objetos são criados?

Na POO os objetos são criados através da instancia de uma classe que anteriormente foi definida e abstraída, assim é possível criar objetos derivados da classe e cada um desses objetos criados através dessa classe é denominado como instância da classe.

Exemplo:

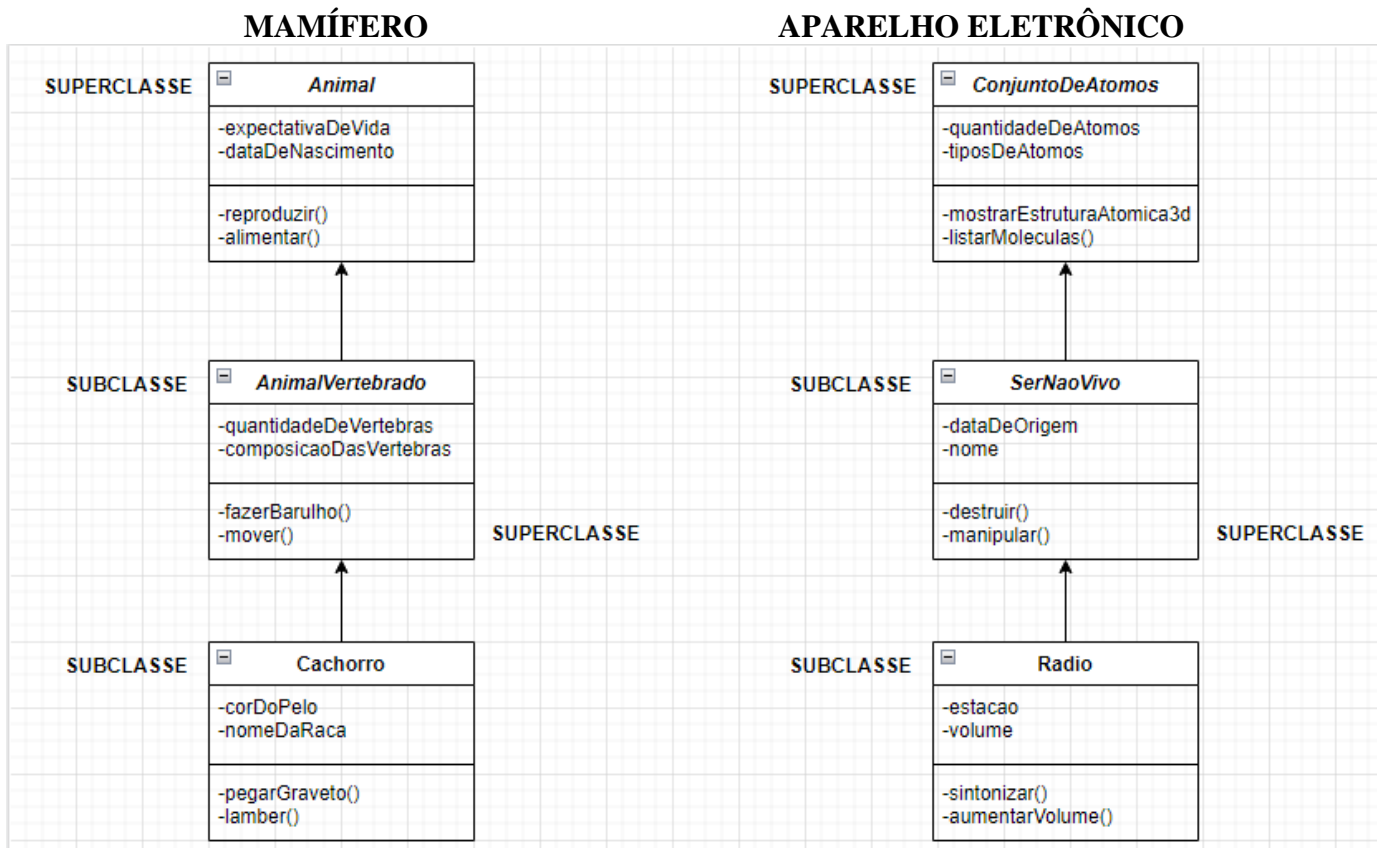


```
J ex1d.java X
J ex1d.java
1 Nome_da_classe nome_do_objeto = new Nome_Da_Classe();
2
```

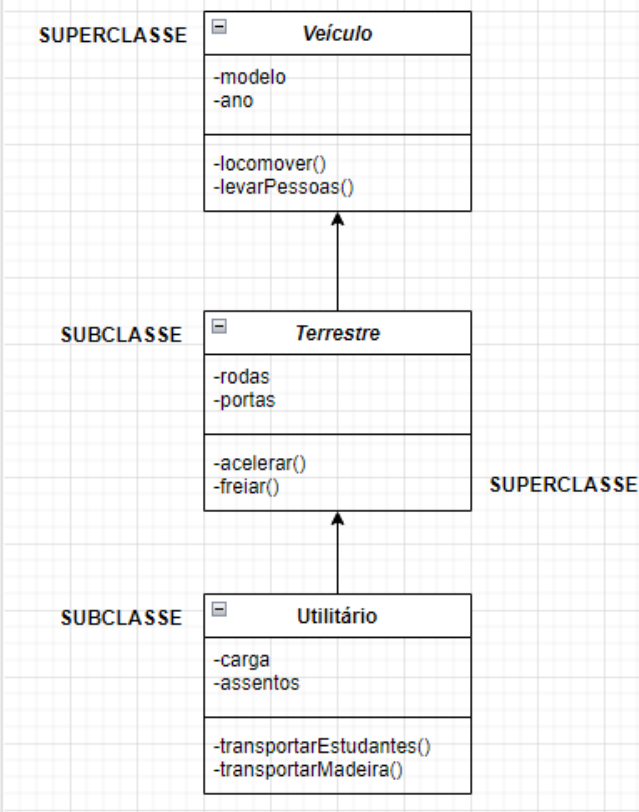
e. O que é coleta de lixo e como ela funciona?

O Garbage Collection ou coleta de lixo é usado em várias linguagens de programação, como Java e Python, para limpar a memória, atuando de forma a buscar objetos que não estão sendo usados, após achar esses objetos que são considerados lixo, eles são removidos da memória para que ela seja limpa.

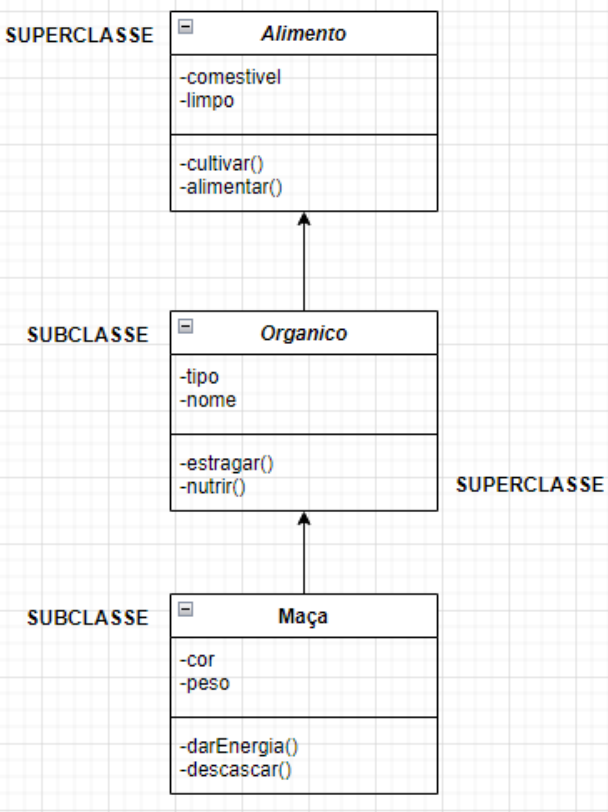
2. **(1 pontos)** Desenhar todas as classes, subclasses e superclasses associadas aos seguintes objetos, indicando em cada caso, pelo menos 2 atributos e 2 métodos, e a relação de herança entre classes
- Carro
 - Fruta
 - Mamífero
 - Computador
 - Aparelho eletrônico

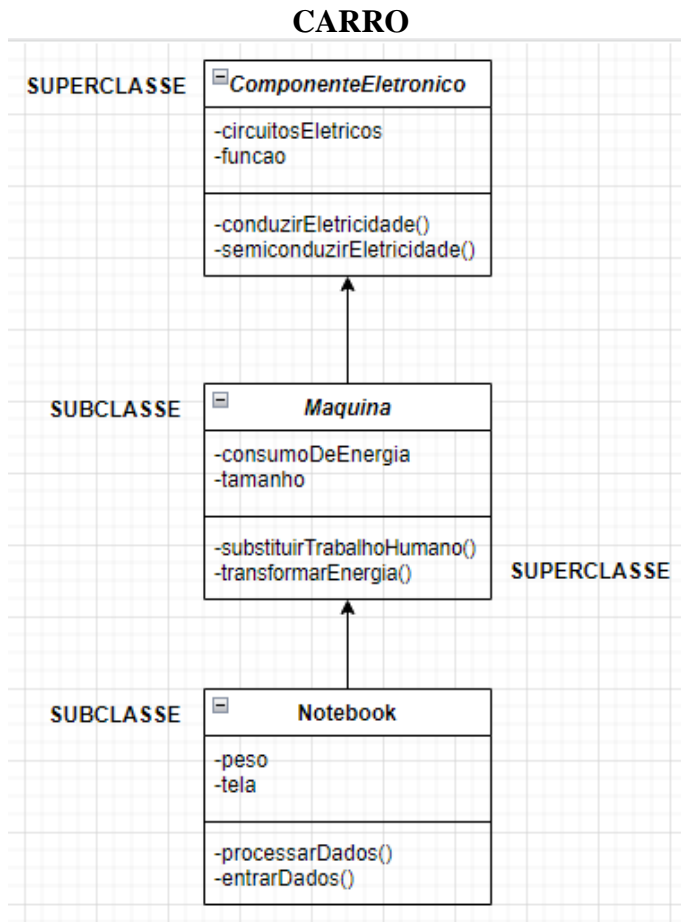


CARRO



FRUTA

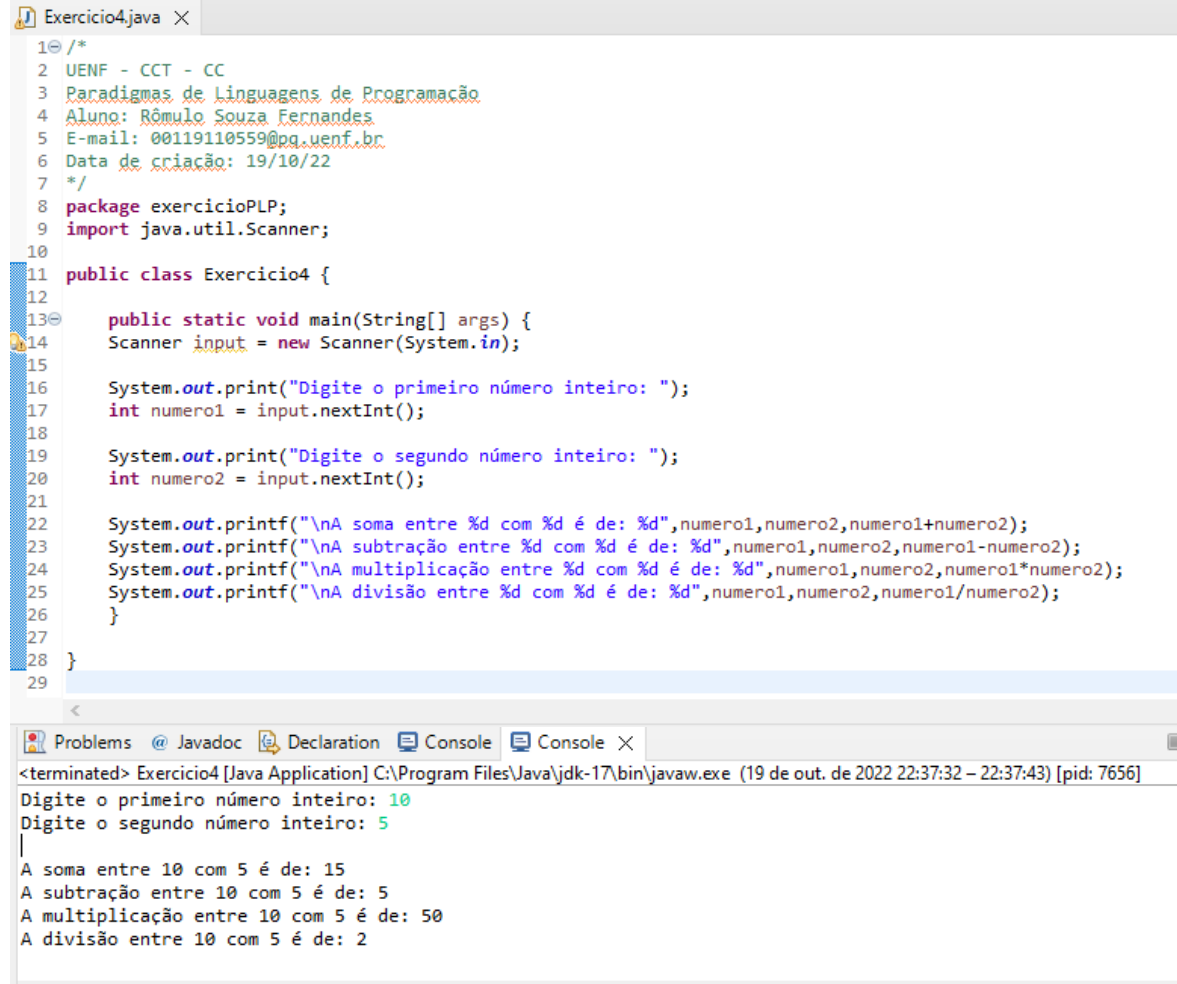




3. **(01 ponto)** Dar três exemplos de cada operação abstrata em orientação a objetos

	Conceito	Exemplo 1	Exemplo 2	Exemplo 3
Instanciação	Classe → Objeto	Universidade → UENF	Matéria → PLP	Aluno → Pedro
Classificação	Objeto → Classe	Arroz → Comida	Gato → Animal	Orquídea → Planta
Especialização	Classe geral → Classe específica	Cachorro → Cachorro Feroz	Professor → Professor de Universidade	Carro → Carro potente
Generalização	Classe específica → Classe geral	Comida quente → Comida	Gato do mato → Gato	Teclado mecânico → Teclado
Decomposição	Objeto complexo → Objeto particionado	TV → Tela de LED	Sofá → Estofamento	Mecânico → Ferramenta
Agregação	Objeto particionado → Objeto complexo	HD → Computador	Pneu → Carro	Madeira → Barco

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



```
Exercicio4.java X
1 /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Aluno: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9  import java.util.Scanner;
10
11  public class Exercicio4 {
12
13      public static void main(String[] args) {
14          Scanner input = new Scanner(System.in);
15
16          System.out.print("Digite o primeiro número inteiro: ");
17          int numero1 = input.nextInt();
18
19          System.out.print("Digite o segundo número inteiro: ");
20          int numero2 = input.nextInt();
21
22          System.out.printf("\nA soma entre %d com %d é de: %d", numero1, numero2, numero1+numero2);
23          System.out.printf("\nA subtração entre %d com %d é de: %d", numero1, numero2, numero1-numero2);
24          System.out.printf("\nA multiplicação entre %d com %d é de: %d", numero1, numero2, numero1*numero2);
25          System.out.printf("\nA divisão entre %d com %d é de: %d", numero1, numero2, numero1/numero2);
26      }
27  }
28
29
```

Problems @ Javadoc Declaration Console Console X

<terminated> Exercicio4 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (19 de out. de 2022 22:37:32 – 22:37:43) [pid: 7656]

Digite o primeiro número inteiro: 10
Digite o segundo número inteiro: 5
|
A soma entre 10 com 5 é de: 15
A subtração entre 10 com 5 é de: 5
A multiplicação entre 10 com 5 é de: 50
A divisão entre 10 com 5 é de: 2

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

```
*Exercicio5.java ×
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Aluno: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 import java.util.Scanner;
11
12 public class Exercicio5 {
13
14     public static void main(String[] args) {
15         Scanner input = new Scanner(System.in);
16         double[] notas = new double[4];
17         double media = 0;
18
19         for (int i = 0; i < notas.length; i++) {
20             System.out.printf("Digite o valor para a %dº nota: ", i + 1);
21             notas[i] = input.nextDouble();
22         }
23
24         for (int i = 0; i < notas.length; i++) {
25             media += notas[i];
26         }
27
28         media = media/notas.length;
29         System.out.printf("A média foi de: %.2f\n",media);
30         if (media >= 6) {
31             System.out.println("\nAprovado.");
32         } else {
33             System.out.println("\nReprovado.");
34         }
35     }
36 }
```

Problems @ Javadoc Declaration Console Console ×

<terminated> Exercicio5 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (19 de out.)

Digite o valor para a 1º nota: 5
Digite o valor para a 2º nota: 9
Digite o valor para a 3º nota: 3
Digite o valor para a 4º nota: 7
A média foi de: 6,00

Aprovado.

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

```
*Exercicio6.java X
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Aluno: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class Exercicio6 {
11
12     public static void main(String[] args) {
13
14         int a = 5;
15         int b = 40;
16         int soma = 0;
17
18         for (int i = 5; i <= b; i++) {
19             soma += i;
20         }
21         System.out.printf("O valor da soma entre %d a %d foi de: %d",a,b,soma);
22     }
23
24 }

Problems @ Javadoc Declaration Console Console X
<terminated> Exercicio6 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (19 de out. de 2022 2
O valor da soma entre 5 a 40 foi de: 810
```


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.

```
*Exercicio7a.java × *Utilitario.java
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Aluno: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 import java.util.Scanner;
11
12 public class Exercicio7a {
13
14     public static void main(String[] args) {
15
16         Scanner input = new Scanner(System.in);
17         Utilitario frota1 = new Utilitario(2017);
18
19         System.out.print("Digite a marca do veículo: ");
20         frota1.marca = input.nextLine();
21
22         System.out.print("Digite o modelo do veículo: ");
23         frota1.modelo = input.nextLine();
24
25         System.out.print("Digite a quilometragem percorrida: ");
26         frota1.kilometragem = input.nextDouble();
27
28         System.out.print("Digite a quantidade de combustível abastecida: ");
29         frota1.litragemCombustivel = input.nextDouble();
30
31         frota1.consumo();
32         frota1.verificaIPVA();
33
34     }
35
36 }
37
```

```
*Exercicio7a.java  *Utilitario.java X
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Aluno: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class Utilitario extends Carro{
11
12     Utilitario(int anoDeFabricacao) {
13         super(anoDeFabricacao);
14     }
15
16     String marca;
17     String modelo;
18
19 }
```

Problems @ Javadoc Declaration Console Console X

<terminated> Exercicio7a [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (20 de out. de 2022 10:51:47 – 10:51:57) [pid: ...]

Digite a marca do veículo: Chevrolet

Digite o modelo do veículo: Spin

Digite a quilometragem percorrida: 100

Digite a quantidade de combustível abastecida: 20

Seu veículo ainda não está isento de pagar o IPVA no estado do Rio de Janeiro

```
Exercicio7b.java × Tomate.java Fruta.java
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class Exercicio7b {
11
12     public static void main(String[] args) {
13
14         Tomate cadastro = new Tomate(1);
15         cadastro.citrico = false;
16
17
18         cadastro.verificaValidade();
19     }
20
21 }
```

```
Exercicio7b.java Tomate.java × Fruta.java
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class Tomate extends Fruta {
11
12     Tomate(int lote) {
13         super(lote);
14     }
15
16
17     String nome;
18     String tipo;
19 }
20
```

```
Exercicio7b.java  Tomate.java  Fruta.java X
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9  import java.util.Scanner;
10
11 public class Fruta {
12     int lote;
13     boolean citrico;
14
15     Fruta(int lote) {
16         this.lote = lote;
17     }
18
19     int[] dataDeColheita = new int[3];
20     Scanner input = new Scanner(System.in);
21
22     public void verificaValidade() {
23
24         System.out.print("Dia da colheita: ");
25         dataDeColheita[0] = input.nextInt();
26
27
28         System.out.print("Mês da colheita: ");
29         dataDeColheita[1] = input.nextInt();
30
31
32         System.out.print("Ano da colheita: ");
33         dataDeColheita[2] = input.nextInt();
34
35
36         if(10 - dataDeColheita[1] > 1) {
37             System.out.println("Produto já fora da data de validade. ");
38         } else {
39             System.out.println("Produto ainda na validade. ");
40         }
41     }
42
43 }
44
```

<terminated> Exercicio7b [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (20 de out. de 2022 11

Dia da colheita: 12
Mês da colheita: 05
Ano da colheita: 2022
Produto já fora da data de validade.

Exercicio7c.java × Mamifero.java PeixeBoi.java

```
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class Exercicio7c {
11
12     public static void main(String[] args) {
13
14         PeixeBoi cadastro = new PeixeBoi();
15         cadastro.identificacao = 001;
16         cadastro.localDeNascimento = "Campos dos Goytacazes";
17         cadastro.peso = 12;
18
19         cadastro.verificaPeso();
20
21     }
22
23 }
```

Exercicio7c.java × Mamifero.java × PeixeBoi.java

```
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class Mamifero {
11
12     String localDeNascimento;
13
14 }
```

Exercicio7c.java Mamifero.java PeixeBoi.java X

```
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class PeixeBoi extends Mamifero {
11
12     double peso;
13     int identificacao;
14
15     public void verificaPeso() {
16         if(peso<7) {
17             System.out.println("Abaixo do peso. ");
18         }else {
19             System.out.println("Dentro do peso ideal. ");
20         }
21     }
22 }
```

Problems @ Javadoc Declaration Console Console X

<terminated> Exercicio7c [Java Application] C:\Program Files\Java\jdk-17\bin\javaw
Dentro do peso ideal.
|

```
Exercicio7d.java X Notebook.java Computador.java
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9  import java.util.Scanner;
10
11  public class Exercicio7d {
12
13      public static void main(String[] args) {
14          Scanner input = new Scanner(System.in);
15
16          Notebook ideapad = new Notebook("I3-7020u");
17
18          System.out.print("Digite o numero de série do computador: ");
19          ideapad.numeroDeSerie = input.nextInt();
20
21          System.out.print("Digite o peso do computador: ");
22          ideapad.peso = input.nextDouble();
23
24          System.out.print("Digite a quantidade de memoria ram: ");
25          ideapad.quantidadeMemoriaRam = input.nextInt();
26
27          ideapad.verificaRequisitos();
28      }
29
30  }
```

```
Exercicio7d.java Notebook.java X Computador.java
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10  public class Notebook extends Computador{
11
12      Notebook(String processador) {
13          super(processador);
14      }
15
16      double peso;
17      int numeroDeSerie;
18  }
```

```
Exercicio7d.java Notebook.java Computador.java X
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class Computador {
11
12     String processador;
13     int quantidadeMemoriaRam;
14
15     Computador(String processador){
16         this.processador =processador;
17     }
18
19     public void verificaRequisitos() {
20         if (quantidadeMemoriaRam < 8) {
21             System.out.println("O computador não possui os requisitos mínimos para executar o p
22         } else {
23             System.out.println("Seu computador possui os requisitos mínimos para executar o pro
24         }
25     }
26 }
```

```
Problems @ Javadoc Declaration Console Console X
<terminated> Exercicio7d [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (20 de out. de 2022 11:27:25 – 11:27:36) [pid
Digite o numero de série do computador: 986132198
Digite o peso do computador: 2,0
Digite a quantidade de memoria ram: 7
O computador não possui os requisitos mínimos para executar o programa.
```


Exercicio7e.java X MicroControlador.java AparelhoEletronico.java

```
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class Exercicio7e {
11
12     public static void main(String[] args) {
13
14         MicroControlador arduino = new MicroControlador();
15
16         arduino.serie = 487923;
17         arduino.modelo = "Uno";
18         arduino.memoriaROM = true;
19         arduino.voltagemNecessaria = 5;
20
21     }
22 }
23
```

Exercicio7e.java MicroControlador.java X AparelhoEletronico.java

```
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class MicroControlador extends AparelhoEletronico{
11
12     int serie;
13     String modelo;
14
15 }
```

Exercicio7e.java × MicroControlador.java AparelhoEletronico.java ×

```
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Autor: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class AparelhoEletronico {
11
12     double voltagemNecessaria;
13     boolean memoriaROM;
14
15 }
```

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”

O professor

Programming Exercises 363

***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:

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

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-

9 – A

```
*Exercicio9a.java × *QuadractEquation.java
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Aluno: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9  import java.util.Scanner;
10
11  public class Exercicio9a {
12
13  public static void main(String[] args) {
14      Scanner input = new Scanner(System.in);
15      QuadractEquation obj = new QuadractEquation();
16
17      System.out.print("Digite o valor para 'a':");
18      int a = input.nextInt();
19      obj.setA(a);
20
21      System.out.print("Digite o valor para 'b':");
22      obj.setB(input.nextInt());
23
24      System.out.print("Digite o valor para 'c':");
25      obj.setC(input.nextInt());
26
27      int result1 = obj.getDiscriminant();
28      System.out.printf("\nDiscriminante: %d\n", result1);
29
30      double result2 = obj.getRoot1();
31      System.out.printf("R1: %.2f\n", result2);
32
33      double result3 = obj.getRoot2();
34      System.out.printf("R2: %.2f\n", result3);
35  }
36
37  }
```

```
*Exercicio9a.java  *QuadractEquation.java X
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Aluno: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 import java.math.*;
11
12 public class QuadractEquation {
13
14     private int a;
15     private int b;
16     private int c;
17
18     int getDiscriminant() {
19         int disc = (b * b) - (4 * a * c);
20         return disc;
21     }
22
23     double getRoot1() {
24
25         double r1 = ((b * -1) + Math.sqrt(b * b - 4 * a * c)) / (2 * a);
26         if (getDiscriminant() < 0) {
27             return 0;
28         } else
29
30             return r1;
31     }
32
33     double getRoot2() {
34         double r2 = ((b * -1) - (Math.sqrt(b * b - 4 * a * c))) / (2 * a);
35         if (getDiscriminant() < 0) {
36             return 0;
37         } else {
```

```
*Exercicio9a.java × *QuadractEquation.java ×
38         return r2;
39     }
40 }
41
42 public QuadractEquation(int a, int b, int c) {
43     super();
44     this.a = a;
45     this.b = b;
46     this.c = c;
47 }
48
49 public QuadractEquation() {
50 }
51
52
53 public int getA() {
54     return a;
55 }
56
57 public void setA(int a) {
58     this.a = a;
59 }
60
61 public int getB() {
62     return b;
63 }
64
65 public void setB(int b) {
66     this.b = b;
67 }
68
69 public int getC() {
70     return c;
71 }
72
73 public void setC(int c) {
74     this.c = c;
```

Problems @ Javadoc Declaration Console Console ×

<terminated> Exercicio9a [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (19 de out. c

Digite o valor para 'a':1
Digite o valor para 'b':9
Digite o valor para 'b':2
|
Discriminante: 73
R1: -0,23
R2: -8,77

9 – B

```
*Exercicio9b.java X *LinearEquation.java
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Aluno: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 import java.util.Scanner;
11
12 public class Exercicio9b {
13
14     public static void main(String[] args) {
15         LinearEquation obj = new LinearEquation();
16         Scanner input = new Scanner(System.in);
17
18         System.out.print("Digite o valor para 'a': ");
19         obj.setA(input.nextInt());
20
21         System.out.print("Digite o valor para 'b': ");
22         obj.setB(input.nextInt());
23
24         System.out.print("Digite o valor para 'c': ");
25         obj.setC(input.nextInt());
26
27         System.out.print("Digite o valor para 'd': ");
28         obj.setD(input.nextInt());
29
30         System.out.print("Digite o valor para 'e': ");
31         obj.setE(input.nextInt());
32
33         System.out.print("Digite o valor para 'f': ");
34         obj.setF(input.nextInt());
35
36         boolean result1 = obj.isSolvable();
37         System.out.printf("\nisSolvable: %b\n", result1);
38
39         double result2 = obj.getX();
40         System.out.printf("X: %.2f\n", result2);
41
42         double result3 = obj.getY();
43         System.out.printf("Y: %.2f\n", result3);
44     }
45
46 }
47
```

```
*Exercicio9b.java  *LinearEquation.java X
1  /*
2  UENF - CCT - CC
3  Paradigmas de Linguagens de Programação
4  Aluno: Rômulo Souza Fernandes
5  E-mail: 00119110559@pq.uenf.br
6  Data de criação: 19/10/22
7  */
8  package exercicioPLP;
9
10 public class LinearEquation {
11
12     private int a;
13     private int b;
14     private int c;
15     private int d;
16     private int e;
17     private int f;
18
19     public LinearEquation(int a, int b, int c, int d, int e, int f) {
20         super();
21         this.a = a;
22         this.b = b;
23         this.c = c;
24         this.d = d;
25         this.e = e;
26         this.f = f;
27     }
28
29     public LinearEquation() {
30     }
31
32
33     public boolean isSolvable() {
34         if ((a * b - b * c) != 0) {
35             return true;
36         }
37     }
38 }
```



```
*Exercicio9b.java  *LinearEquation.java X
37         return false;
38     }
39
40     public double getX() {
41         double x = (e*d-b*f)/(a*d-b*c);
42         return x;
43     }
44
45     public double getY() {
46         double y = (a*f-e*c)/(a*d-b*c);
47         return y;
48     }
49
50     public int getA() {
51         return a;
52     }
53
54     public void setA(int a) {
55         this.a = a;
56     }
57
58     public int getB() {
59         return b;
60     }
61
62     public void setB(int b) {
63         this.b = b;
64     }
65
66     public int getC() {
67         return c;
68     }
69
70     public void setC(int c) {
71         this.c = c;
72     }
--
```

```

73
74 public int getD() {
75     return d;
76 }
77
78 public void setD(int d) {
79     this.d = d;
80 }
81
82 public int getE() {
83     return e;
84 }
85
86 public void setE(int e) {
87     this.e = e;
88 }
89
90 public int getF() {
91     return f;
92 }
93
94 public void setF(int f) {
95     this.f = f;
96 }
97
98 }
99

```

Problems @ Javadoc Declaration Console Console X

<terminated> Exercicio9b [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (19 de out. de 2022 2

Digite o valor para 'a': 8
 Digite o valor para 'b': 7
 Digite o valor para 'c': 6
 Digite o valor para 'd': 2
 Digite o valor para 'e': 4
 Digite o valor para 'f': 3
 |
 isSolvable: true
 X: 0,00
 Y: 0,00