



LINGUAGEM DE PROGRAMAÇÃO I

MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNIESP

O que vamos estudar?

- Paradigma Orientada a Objetos
- Estrutura da linguagem
- Classes e objetos
- Encapsulamento
- Abstração
- Herança
- Polimorfismo
- Interfaces
- Coleções

Avaliações

05/10 – 1ª avaliação

70% prova escrita/prática
30% exercícios

30/11 – 2ª avaliação

50% prova escrita
50% projeto

09/12 – Reposição

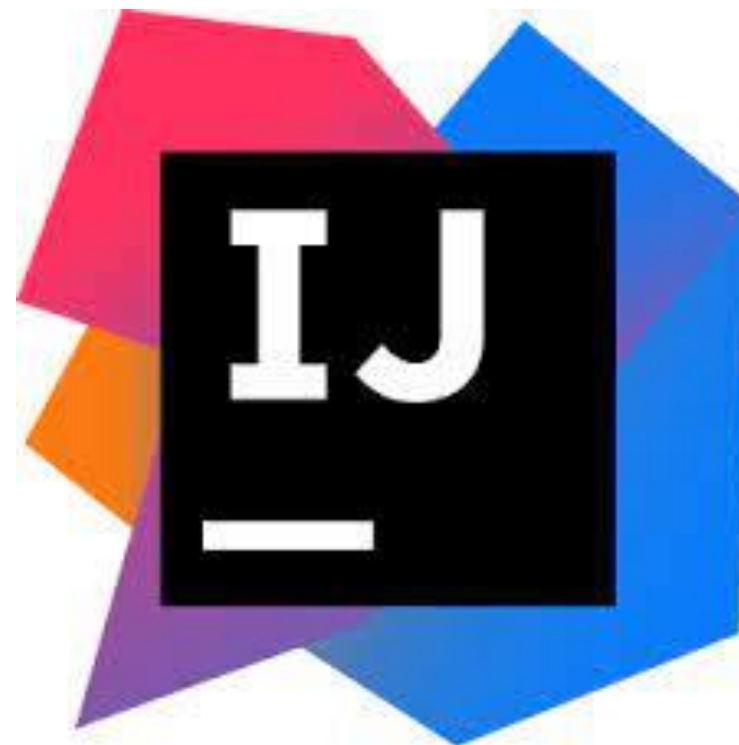
Fez exercícios e projeto, mas não fez a prova?
A reposição vale 100%

16/12 – Final

Todo o conteúdo escrito

INTRODUÇÃO

Introdução



Introdução

Java é uma linguagem de paradigma Orientado a Objetos



Classe Principal

Estrutura para representar um determinado modelo

```
public class Main {  
    public static void main(String[] args) {  
  
        System.out.println("Hello world!");  
    }  
}
```

Atributo

São as propriedades da classe

Tipo	Tamanho	Uso/Valores
boolean	1 bit	true / false
byte	1 byte	Inteiros de -127 a 128
short	2 bytes	Inteiros de -32767 a 32768
char	2 bytes	Códigos de caracteres (numéricos) – 0 a 65535
int	4 bytes	Inteiros (aprox. -2.000.000 a +2.000.000)
float	4 bytes	Ponto flutuante de simples precisão
long	8 bytes	Inteiros muito grandes
double	8 bytes	Ponto flutuante de dupla precisão

Exemplo: `int` idade; `double` media; `String` nome; `boolean` resp;

Operadores Aritméticos

Operador	Função	Exemplo
+	Adição	4+2
-	Subtração	18 - 21
*	Multiplicação	3 * 4
/	Divisão	18 / 18
%	Resto da Divisão	7 % 3
++	Incremento	i++
--	Decremento	j--

Exemplo: `int x = 10; int y = 5; int total = x+y;`

Operadores Relacionais

Operador	Função	Exemplo
>	Maior que	<code>a > b</code>
>=	Maior ou igual	<code>c >= j</code>
<	Menor que	<code>d < 85</code>
<=	Menor ou Igual	<code>u <= p</code>
==	Igual	<code>x == y</code>
!=	Diferente	<code>a != 12</code>

Exemplo: `int x = 10; int y = 5; boolean resp = x >= y;`

Operadores Lógicos

Operador	Função	Exemplo
&&	and E	a>b and b<c
	or OU	a>b b<c
!	not NÃO	!(a>b)

Exemplo:

```
int x = 10;  
int y = 5;  
boolean resp = (x > y) && (y == 0);
```

Exemplo na classe:

```
public class Main {  
    public static void main(String[] args) {  
  
        int num1 = 20;  
        int num2 = 40;  
        int total;  
  
        total = num1+num2;  
  
        System.out.println(total);  
    }  
}
```

Estrutura de Decisão Simples

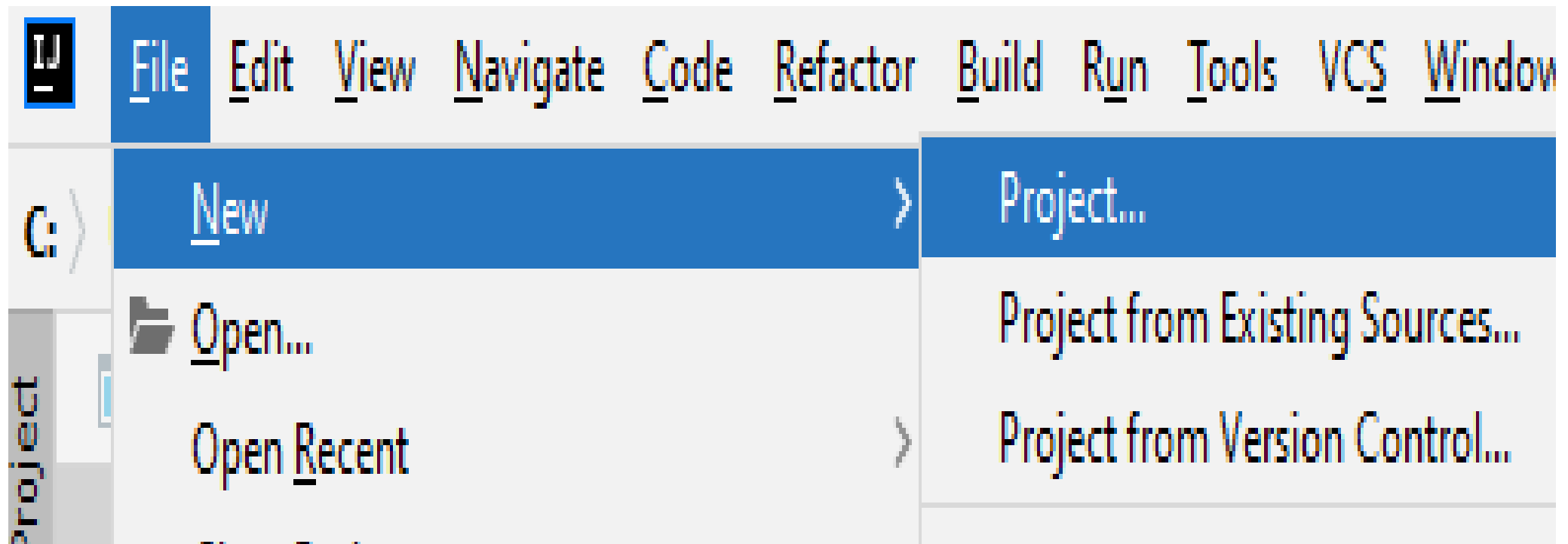
```
if (num1 > num2) {  
    System.out.println("é maior ");  
} else {  
    System.out.println("é menor");  
}
```

Estrutura de Decisão Composta

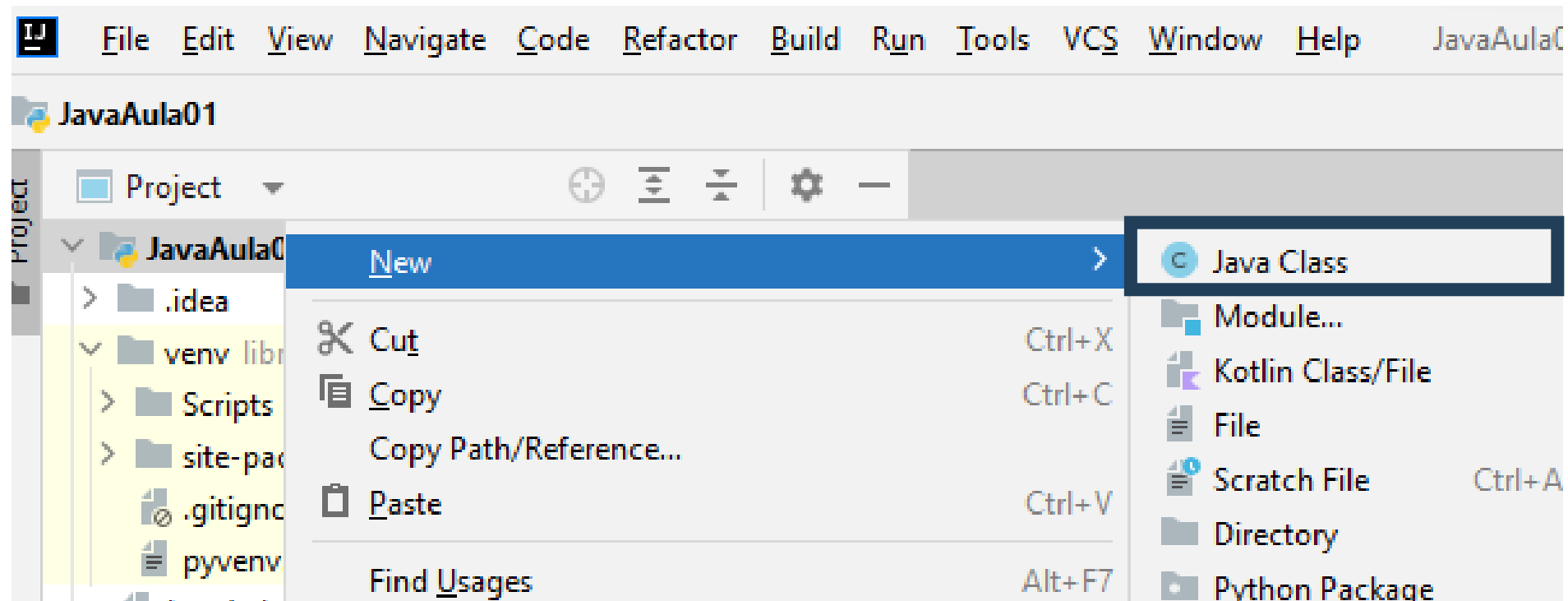
```
if (num1 > num2) {  
    System.out.println("é maior ");  
} else if (num1 == num2){  
    System.out.println("é igual");  
}else{  
    System.out.println("é menor");  
}
```

Praticando...

Praticando...



Praticando...



Elabore um programa que some dois números

```
public class Main {  
    public static void main(String[] args) {  
  
        int num1 = 20;  
        int num2 = 40;  
        int total;  
  
        total = num1+num2;  
  
        System.out.println(total);  
    }  
}
```

EXERCÍCIO

- 1) Faça um programa que calcule o salário de um funcionário com base no valor da hora trabalhada de R\$ 15.00 e na quantidade de 40 horas trabalhadas e exiba o salário.
- 2) Elabore um programa que verifique se uma variável é maior, menor ou igual a outra e exiba esta informação na tela e exiba.
- 3) Elabore um programa que receba 3 notas (informadas pela pessoa desenvolvedora). Em seguida, calcule a média e verifique se o aluno está aprovado, reprovado ou na final.

Observação: Aprovado ≥ 70 , Reprovado < 40 .

Desafio: Como saber se está na final?

Exercício

4) Faça um programa para o cálculo de uma folha de pagamento. Para isso, o sistema deve saber o valor do salário. Com base na descrição abaixo, **exiba** os descontos do IR:

- Salário até R\$900,00 (inclusive) – Isento;
- Salário de R\$901,00 até R\$ 1500, 00 (inclusive) – desconto de 5%;
- Salario de1501,00 até R\$ 2500,00 (Inclusive) – desconto de 10%;
- Salário acima de 2500 – Desconto de 20%.

DESAFIO!!!

Para a questão 4, **calcule** os descontos e exiba-os!
Em seguida, informe o salário líquido



Respondendo

```
public class Main {  
    public static void main(String[] args) {  
  
        double salarioHora = 15.0;  
        int horasTrabalhadas = 40;  
  
        double salario = salarioHora * horasTrabalhadas;  
  
        System.out.println("Salário: R$" + salario);  
  
    }  
}
```

Questão 1

Elabore um programa que verifique se uma variável é maior, menor ou igual a outra e exiba esta informação na tela.

Questão 2

Elabore um programa que verifique se uma variável é maior, menor ou igual a outra e exiba esta informação na tela.

```
public class Main {  
    public static void main(String[] args) {  
  
        int num1 = 20;  
        int num2 = 40;  
  
        if (num1 > num2 ) {  
            System.out.println("é maior ");  
        } else {  
            System.out.println("é menor");  
        }  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        double nota1 = 20;  
        double nota2 = 70;  
        double nota3 = 50;  
  
        double media = (nota1+nota2+nota3)/3;  
        System.out.println(media);  
  
        if (media >= 70 ) {  
            System.out.println("Aprovado ");  
        } else if (media < 40) {  
            System.out.println("Reprovado");  
        } else {  
            System.out.println("Final");  
        }  
    }  
}
```

Questão 3

Elabore um programa que verifique se o aluno está aprovado, reprovado ou na final. Para isso, o sistema deve saber as 3 notas, calcular a média e exibir a situação. Aprovado > 70, Reprovado < 40.

OBSERVAÇÃO

Questão 3

```
System.out.println(Math.round(media));
```

```
System.out.println("A média é " + String.format("%.2f", media));
```

```
System.out.printf ("A média é %.2f ", media);
```

```
public class Main {  
    public static void main(String[] args) {
```

```
        double salario = 3500;  
        double ir;  
        if (salario <= 900) {  
            System.out.println("Isento");  
        }  
        else if (salario > 900 && salario <= 1500) {  
            System.out.println("Imposto de 5%");  
        }  
        else if (salario > 1500 && salario <= 2500) {  
            System.out.println("Imposto de 10%");  
        }  
        else {  
            System.out.println("Imposto de 20%");  
        }  
    }  
}
```

Questão 4

Faça um programa para o cálculo de uma folha de pagamento. Para isso, o sistema deve saber o valor do salário.

Com base na descrição abaixo, imprima os descontos:

- Salário ate R\$900,00 (inclusive) – Isento;
- Salário de R\$901,00 a R\$ 1500, 00 (inclusive) – desconto de 5%;
- Salario de 1501,00 até R\$ 2500,00 (Inclusive) – desconto de 10%;
- Salário acima de 2500 – Desconto de 20%.

DESAFIO!!!

Para a questão 4, calcule os descontos e exiba-os!
Em seguida, informe o salário líquido



```
public class Main3 {  
    public static void main(String[] args) {
```

```
        double salario = 3500;  
        double ir;  
        if (salario <= 900) {  
            System.out.println("Isento");  
        }  
        else if (salario > 900 && salario <= 1500) {  
            ir = (salario * 5) / 100;  
            salario = salario - ir;  
            System.out.println("O salario líquido é:" + salario + " o imposto é" + ir + " IR de 5%");  
        }  
        else if (salario > 1500 && salario <= 2500) {  
            ir = (salario * 10) / 100;  
            salario = salario - ir;  
            System.out.println("O salario líquido é:" + salario + " o imposto é" + ir + " IR de 10%");  
        }  
        else {  
            ir = (salario * 20) / 100;  
            salario = salario - ir;  
            System.out.println("O salario líquido é:" + salário + " o imposto é" + ir + " IR de 20%");  
        }  
    }  
}
```

- Salário ate R\$900,00 (inclusive) - Isento;
- Salário de R\$901,00 a R\$ 1500, 00 - desconto de 5%;
- Salário de 1501,00 até R\$ 2500,00 - desconto de 10%;
- Salário acima de 2500 - desconto de 20%.

Dúvidas?



LINGUAGEM DE PROGRAMAÇÃO I

MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNESP

Escolha Caso

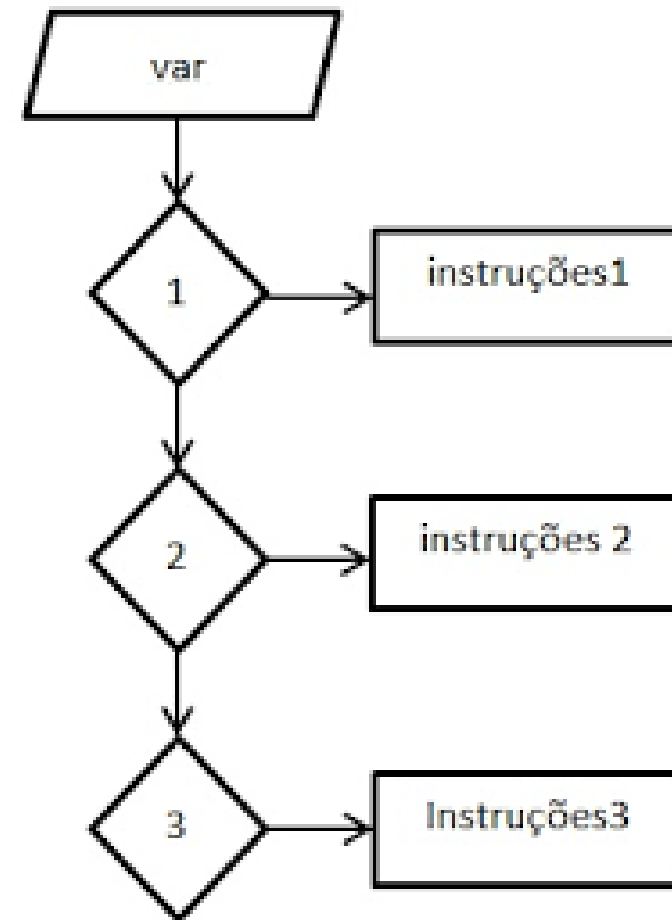
Switch() Case:

```
switch (opção) { case <valor>: }
```

Exemplo:

```
int x= 1;  
switch (x){  
    case 0:  
        System.out.println("Opção 0 escolhida");  
        break;  
}
```

```
public class Main {  
    public static void main(String[] args) {  
  
        int x= 1;  
        switch (x){  
            case 0:  
                System.out.println("Opção 0 escolhida");  
                break;  
            case 1:  
                System.out.println("Opção 1 escolhida");  
                break;  
            case 2:  
                System.out.println("Opção 2 escolhida");  
                break;  
            default:  
                System.out.println("Opção 2 escolhida");  
        }  
    }  
}
```



Repetições

FOR

```
for (ini; fim; incremento) {  
    <instruções> }  
}
```

Exemplo:

```
int i = 0;
```

```
for (i=0; i<5;i++) {  
    System.out.println(i);  
}
```

//exibe de 0 a 4

```
int i = 5;
```

```
for (i=5; i>0;i--) {  
    System.out.println(i);  
}
```

//exibe de 5 a 1

WHILE

```
while (condição) { <instruções> }
```

Exemplo:

```
int i = 0;
```

```
while(i <= 10) {  
    System.out.println(i++);  
}
```

//exibe de 0 a 10

```
int i = 0;
```

```
while(i <= 10) {  
    System.out.println(++i);  
}
```

//exibe de 1 a 11

Exemplo

```
public class Main {  
    public static void main(String args[]) {  
  
        int contador = 0;  
        while (contador < 50) {  
            System.out.println("Repetição nº: " + contador);  
            contador++;  
        }  
    }  
}
```

DO WHILE:

```
do{<instrução>}while (condição);
```

Exemplo:

```
int i = 0;
```

```
do{  
    System.out.println(i++);  
} while(i <= 10) ;
```

//exibe de 0 a 10

```
int i = 0;
```

```
do{  
    System.out.println(++i);  
} while(i <= 10)
```

//exibe de 1 a 11

Exemplo

```
public class Main {  
    public static void main(String args[]) {  
  
        int contador = 0;  
        do{  
            System.out.println("Contador " + contador);  
            contador++;  
        } while (contador < 5) ;  
    }  
}
```

```
//Contador 0  
//Contador 1  
//Contador 2  
//Contador 3  
//Contador 4
```

Praticando...

Questão 1)

Elabore um programa que simule ser um atendente virtual de uma loja (Petshop, Loja de roupas, clinica médica).

O Assistente virtual deverá informar um texto com as opções a serem escolhidas para que o cliente seja direcionado ao serviço desejado.

Para isso, exiba boas mensagens para entendimento do usuário. Exemplo:

Caso escolha 1: Compra item

Caso escolha 2: Solicitar valor do produto ou serviço

Caso escolha 3: Falar com um atendente

Praticando...

Resposta Questão1)

```
public class Main {  
    public static void main(String[] args) {  
        int x = 1;  
        System.out.println(  
            "Caso escolha 1: Compra Blusa \n" +  
            "Caso escolha 2: Solicitar valor do produto ou serviço\n" +  
            "Caso escolha 3: Falar com um atendente\n");  
        switch (x) {  
            case 1:  
                System.out.println("Ótima escolha!");  
                break;  
            case 2:  
                System.out.println("O valor da blusa é R$ 2,00");  
                break;  
            case 3:  
                System.out.println("Falar com atendente");  
                break;  
        }  
    }  
}
```

DESAFIO

Praticando...

Questão 1)

Elabore um programa que simule ser um atendente virtual de uma loja (Petshop, Loja de roupas, clinica médica). O Assistente virtual deverá informar um texto com as opções a serem escolhidas para que o cliente seja direcionado ao serviço desejado. Para isso, exiba boas mensagens antes de elaborar os comandos de escolha.

Caso escolha 1: Compra item

Caso escolha 2: Solicitar valor do produto ou serviço

Caso escolha 3: Falar com um atendente

Caso escolha 0: Retornar ao menu principal

Usando o teclado

Scanner()

```
import java.util.Scanner;
```

```
import java.util.Scanner;
```

```
public class Main {  
    public static void main(String args[]) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Informe um número");
```

```
        int x = sc.nextInt();
```

```
        System.out.println( x );
```

```
    }
```

```
}
```

String	next();
Inteiro	nextInt();
Double	nextDouble();


```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        int r ;
        Scanner sc = new Scanner(System.in);
        do {
            System.out.println("Caso escolha 1: Compra Blusa, Caso escolha 2: Solicitar valor do produto ou serviço");
            int x = sc.nextInt();
            switch (x) {
                case 1:
                    System.out.println("Ótima escolha!");
                    break;
                case 2:
                    System.out.println("O valor da blusa é R$ 2,00");
                    break;
            }
            System.out.println("Para ver novamente, digite 0");
            r = sc.nextInt();
        } while (r == 0);
    }
}
```

DESAFIO

Por limitação de espaço, aqui tem apenas o case 1 e 2

Dúvidas?



LINGUAGEM DE PROGRAMAÇÃO I

MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNIESP

RELEMBRANDO...

Elabore um programa de uma indústria realiza a folha mensal de pagamentos de seus empregados baseando-se nas seguintes regras:

- a. Inicialmente, ler os dados de cada funcionário (matrícula, nome e salário bruto);
- b. Depois, o programa deve processar o desconto do INSS, que é 15% do salário bruto, e o salário líquido é a diferença entre o salário bruto e a dedução do INSS.
- c. Por fim, exibir na tela o seu contracheque, cujo formato é dado a seguir
 - **Matrícula:**
 - **Nome completo:**
 - **Salário Bruto:**
 - **Dedução INSS:**
 - **Salário Líquido:**

```
import java.util.Scanner;
```

RESPOSTA



```
public class Main2 {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        int sair;  
        do {  
            System.out.println("Dados do funcionário ":");  
  
            System.out.print("Matrícula: ");  
            int matricula = scanner.nextInt();  
  
            scanner.nextLine(); // Limpar o buffer  
  
            System.out.print("Nome completo: ");  
            String nome = scanner.nextLine();  
  
            System.out.print("Salário bruto: ");  
            double salarioBruto = scanner.nextDouble();  
  
            ...  
        }  
    }  
}
```

```
double descontoINSS = salarioBruto * 0.15;  
double salarioLiquido = salarioBruto - descontoINSS;
```

```
System.out.println(" Matrícula: " + matricula);  
System.out.println(" Nome completo: " + nome);  
System.out.println(" Salário Bruto: R$" + salarioBruto);  
System.out.println(" Dedução INSS: R$" + descontoINSS);  
System.out.println(" Salário Líquido: R$" + salarioLiquido);
```

```
System.out.print("Digite 0 para repetir: ");  
sair = scanner.nextInt();  
} while (sair == 0);  
System.out.println("Programa encerrado");
```

```
}
```

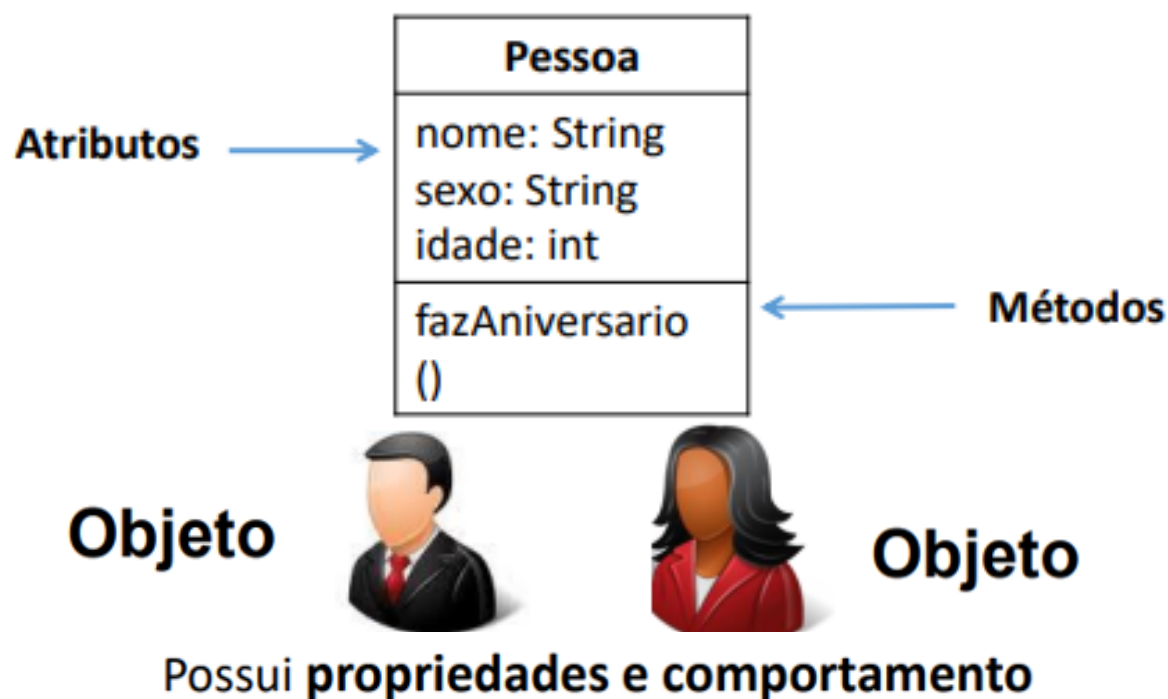
```
}
```

Orientação a objetos

Paradigma baseado na composição e integração entre unidades de software chamadas de **objeto**

Classe em Java

- Define a estrutura e o comportamento de qualquer objeto da classe
- Atua como um padrão para a construção de objetos.



Exemplo

Atributos

Classe

Método

```
class Pessoa {  
    String nome;  
    int idade;  
  
    void fazAniversario() {  
        idade++;  
    }  
}
```

Exemplo prático 1

Defina uma classe Pessoa que faça aniversário e exiba essa idade (pré definida pelo programador) na tela


```
public class Pessoa {  
    int idade = 2;  
  
    public void niver () {  
        idade++;  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
  
        Pessoa p = new Pessoa();  
  
        System.out.println("antiga " + p.idade);  
        p.niver();  
  
        System.out.println("atual: " + p.idade);  
    }  
}
```

Exemplo Prático 2

Elabore um programa em Java que seja capaz de realizar a soma de dois números e retornar o total na tela

```
public class Calculadora {  
    int n1, n2, total;  
  
    public void somar() {  
        total = n1 + n2;  
        System.out.print(total );  
    }  
}
```



```
import java.util.Scanner;
```

```
public class CalculadoraMain {  
    public static void main(String[] args) {
```

```
        Calculadora c = new Calculadora();  
        Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Número 1");
```

```
        c.n1 = sc.nextInt();
```

```
        System.out.print("Número 2");
```


```
        c.n2 = sc.nextInt();
```

```
        c.somar();
```



```
    }  
}
```

```
public class Calculadora {  
    int n1, n2, total;  
  
    public void somar() {  
        total = n1 + n2;  
    }  
}
```



```
import java.util.Scanner;
```

```
public class CalculadoraMain {  
    public static void main(String[] args) {
```

```
        Calculadora c = new Calculadora();  
        Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Número 1");
```

```
        c.n1 = sc.nextInt();
```

```
        System.out.print("Número 2");
```

```
        c.n2 = sc.nextInt();
```

```
        c.somar();
```

```
        System.out.print(c.total );
```



```
    }  
}
```

```
public class Calculadora {  
    int total;  
  
    public void somar(int x, int y) {  
        total = x + y;  
        System.out.print(total );  
    }  
}
```

```
import java.util.Scanner;
```

```
public class CalculadoraMain {  
    public static void main(String[] args) {
```

```
        Calculadora c = new Calculadora();  
        Scanner sc = new Scanner(System.in);
```

```
        int n1,n2;
```

```
        System.out.print("Número 1");
```

```
        n1 = sc.nextInt();
```

```
        System.out.print("Número 2");
```

```
        n2 = sc.nextInt();
```

```
        c.somar(n1,n2);
```

```
    }  
}
```

OBSERVAÇÃO!

void x return

```
public class Calculadora {  
    int total;  
  
    public int somar(int x, int y) {  
        total = x + y;  
        return total;  
    }  
}
```

```
import java.util.Scanner;
```

```
public class CalculadoraMain {  
    public static void main(String[] args) {  
  
        Calculadora c = new Calculadora();  
        Scanner sc = new Scanner(System.in);  
        int n1,n2;  
        System.out.print("Número 1");  
        n1 = sc.nextInt();  
        System.out.print("Número 2");  
        n2 = sc.nextInt();  
        System.out.print(c.somar(n1,n2));  
    }  
}
```


LISTA DE EXERCÍCIO

Lista de LP1

Elabore um código em Java, usando o conceito de repetição e orientação a objetos, conforme a última aula, para as seguintes problemáticas:

- 1) Defina uma classe Aluno que faça o cadastro do nome e das 3 notas e a operação do cálculo da média. Ao final, exiba (na classe principal) a média. Em seguida, informe se ele está aprovado, reprovado ou na final. Se a média for ≥ 70 está aprovado, abaixo de 40 está reprovado, senão, está na final.
- 2) Defina um sistema onde o usuário informe (na classe principal, através do teclado) qual opção deseja visualizar. Caso informe 1 ele vai ver a soma de dois números. Caso informe 2, vai ver a subtração, 3 divisão e 4 multiplicação. Observação: Estas operações deverão ser realizadas em uma classe chamada Matemática.
- 3) Crie uma classe chamada "Produto" que representará um produto com os atributos nome, preço e quantidade em estoque. Implemente métodos para adicionar estoque e vender unidades do produto. Na classe principal, crie um objeto Produto, realize operações de:
 - **Adicionar estoque**
 - **Vender produtos**
 - **Exibir o estoque.**
- 4) Elabore um programa que simule um sistema bancário de um cliente. Inicialmente o saldo está zerado, mas o cliente pode escolher qualquer uma das seguintes opções do menu:
 - **Consultar saldo**
 - **Depositar**
 - **Sacar**
 - **Transferir**

Uma observação é que o cliente só pode sacar e transferir se o saldo da conta não estiver zerado.

- 5) Crie um programa que simule um sistema de biblioteca para gerenciar empréstimos de livros. Os usuários podem escolher entre as seguintes opções:
 - **Registrar um novo livro com título e autor.**
 - **Registrar um novo usuário com nome e número de identificação.**
 - **Realizar um empréstimo de um livro, associando o livro a um usuário.**
 - **Devolver um livro, liberando-o para empréstimo novamente.**
 - **Exibir a lista de livros emprestados no momento.**

As questões 3 e 4, coloque no github e disponibilize o link no Moodle.

Valendo como nota de aula 1.



LINGUAGEM DE PROGRAMAÇÃO I

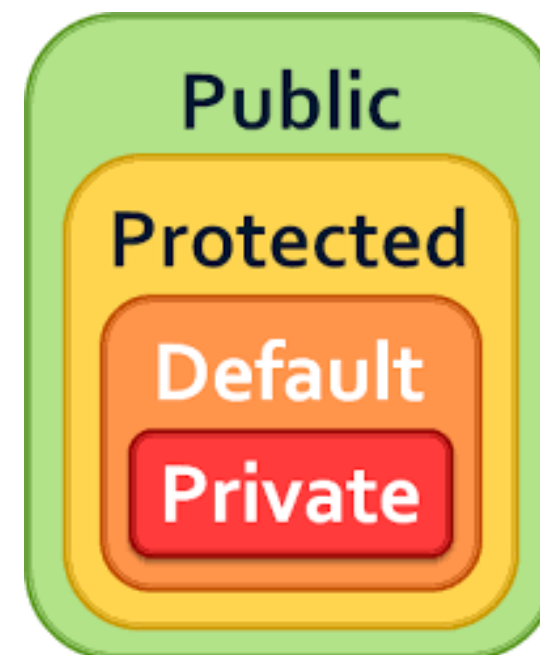
MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNESP

Métodos e Modificadores de Acesso

Modificadores de Acesso

Visibilidade	<u>public</u>	<u>protected</u>	default	<u>private</u>
A partir da mesma classe	✓	✓	✓	✓
Qualquer classe no mesmo pacote	✓	✓	✓	✗
Qualquer classe filha no mesmo pacote	✓	✓	✓	✗
Qualquer classe filha em pacote diferente	✓	✓	✗	✗
Qualquer classe em pacote diferente	✓	✗	✗	✗



```
public class Pessoa {  
    → private String nome;  
    ...  
}
```

Como a classe principal vai poder usar o atribuo se ele está privado?

Métodos de Acesso

Set

Modifica o valor do atributo

Get

Pega o atributo da classe

```
public class Pessoa {
```

```
    private String nome;
```

```
    private int idade;
```

```
    public void setIdade(int idade) {  
        this.idade = idade;  
    }
```

```
    public int getIdade() {  
        return idade;  
    }
```

```
    // get e set para nome
```

```
}
```



```
import java.util.Scanner;

public class Principal {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        Pessoa p = new Pessoa();

        System.out.println("Digite sua idade");
        p.setIdade(sc.nextInt());
        System.out.println("Idade:" + p.getIdade());

    }
}
```

Nota de Aula

Enviar para prof2261@iesp.edu.br

Título do email: **[NOTAdeAULAnº_Turma]_SeuNome**

NOTA DE AULA

Defina um sistema que calcule o IMC de uma Pessoa. Para isso, o sistema deve solicitar do usuário o peso e a altura.

$$\text{IMC} = \text{peso} / (\text{altura} * \text{altura})$$

Realize esse código utilizando os conceitos de
métodos e modificadores de acesso

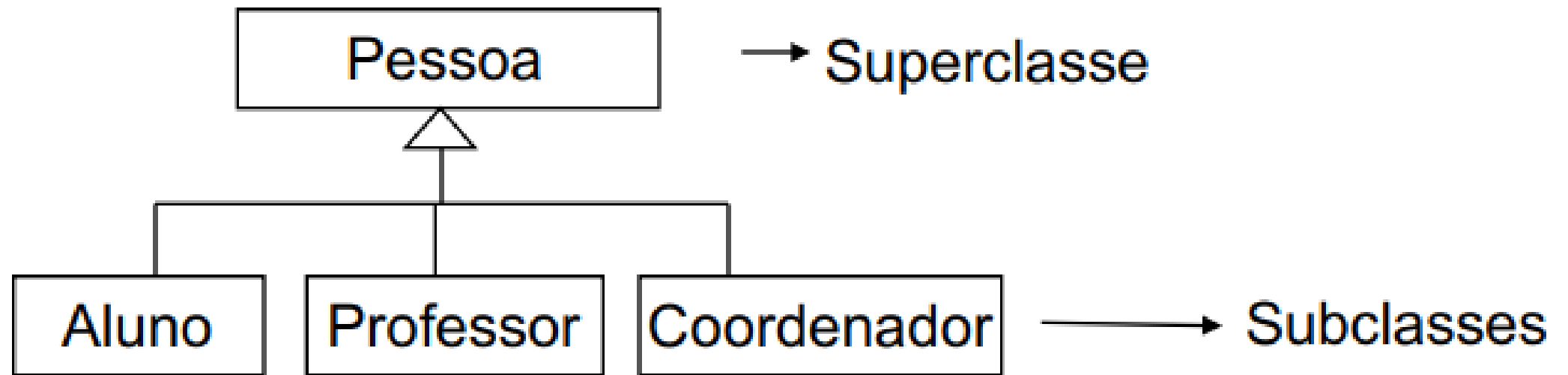


LINGUAGEM DE PROGRAMAÇÃO I

MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNESP

HERANÇA



HERANÇA

```
public class Pessoa{  
    protected String nome;  
    protected double idade;  
  
    //gets e sets  
  
    protected void aniversario(){  
        idade++;  
    }  
}
```

```
public class Aluno extends Pessoa{  
  
    private int matricula;  
  
    //getMatricula  
    //setMatricul  
  
    // métodos  
}
```

EXERCITANDO

Elabore um código de cadastro de um sistema acadêmico, onde professor possui nome, idade, especialização e salário e um aluno possui nome, idade, duas notas e uma média.

O sistema deve perguntar qual cadastro você deseja realizar:

- Caso digite 1, cadastro para professor;
- Caso digite 2, cadastro para aluno.

O sistema deve exibir os dados na tela.

RESPOSTA


```
public class Pessoa {  
    protected String nome;  
    protected int idade;  
  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
  
    public void setIdade (int idade) {  
        this.idade = idade;  
    }  
}
```

//obs.: programar a classe Professor
//public class Professor extends Pessoa{

```
public class Aluno extends Pessoa {  
    private double nota1,nota2,media;  
  
    public void setNota1(double nota1) {  
        this.nota1 = nota1;  
    }  
    public void setNota2(double nota2) {  
        this.nota2 = nota2;  
    }  
    public void setMedia(double media) {  
        this.media = media;  
    }  
    public void exibirDados() {  
        System.out.println("Dados do Aluno:");  
        System.out.println("Nome: " + nome);  
        System.out.println("Idade: " + idade);  
        System.out.println("Nota 1: " + nota1);  
        System.out.println("Nota 2: " + nota2);  
        System.out.println("Média: " + media);  
    }  
}
```

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println("Digite 1 para cadastrar um professor ou 2 para cadastrar um aluno:");
```

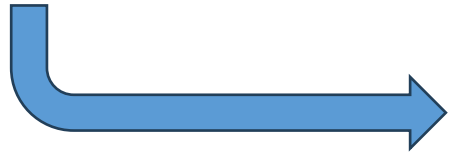
```
        int opcao = scanner.nextInt();
```

```
        scanner.nextLine(); // Limpar o buffer
```

```
        if (opcao == 1) {
```

```
            Professor professor = new Professor();
```

```
            // programação aqui
```



```
    } else if (opcao == 2) {
```

```
        Aluno aluno = new Aluno();
```

```
        System.out.print("Nome do Aluno: ");
```

```
        aluno.setNome(scanner.nextLine());
```

```
        System.out.print("Idade do Aluno: ");
```

```
        aluno.setIdade(scanner.nextInt());
```

```
        System.out.print("Nota 1 do Aluno: ");
```

```
        aluno.setNota1 (scanner.nextDouble());
```

```
        System.out.print("Nota 2 do Aluno: ");
```

```
        aluno.setNota2(scanner.nextDouble());
```

```
        aluno.exibirDados();
```

```
    } else {
```

```
        System.out.println("Opção inválida. ");
```

```
    }
```

```
}
```

```
}
```

NOTA DE AULA

QUESTÃO 1)

Elabore um programa que tenha uma super classe chamada Veiculo com os seguintes atributos: marca (String), modelo (String) e anoFabricacao (int). Este sistema deve ter duas subclasses: Carro e Moto, que herdaram da classe Veículo.

Na classe Carro, adicione um atributo adicional chamado numPortas (int)

Na classe Moto, adicione um atributo adicional chamado cilindradas (int)

Crie um método imprimirDetalhes nas classes Carro e Moto que exiba todos os atributos do veículo, incluindo os atributos herdados da classe Veículo.

Na classe Main, crie um objeto Carro e um objeto Moto, configure seus atributos e use o método imprimirDetalhes para exibir suas informações.

QUESTÃO 2)

Elabore um programa que tenha uma classe chamada Funcionario com os seguintes atributos: nome e salario e crie duas subclasses: Gerente e Diretor, que herdam da classe Funcionário.

Na classe Gerente, adicione um atributo adicional chamado departamento (String). Nesta classe, o gerente possui um método para calcularBonus de 10% do salário.

Na classe Diretor, adicione um atributo adicional chamado ações (int) que representa a quantidade de ações da empresa que o diretor possui.

Na classe Main, interaja para o cadastro e exibição dos dados.



LINGUAGEM DE PROGRAMAÇÃO I

MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNESP

CONSTRUTORES

Um construtor é um método utilizado por uma classe para criar objetos dela.

```
public class Principal{  
    public static void main(String[] args) {
```

```
        Carro c = new Carro ();
```

```
    }  
}
```

Como era antes?

```
public class Principal{  
    public static void main(String[] args) {
```

```
        Carro c = new Carro ("Chevrolet", "Ônix", 2022);
```

```
    }  
}
```

Como podemos fazer agora?


```
public class Principal{  
    public static void main(String[] args) {  
  
        Carro c = new Carro ();  
  
    }  
}
```

Se um construtor não for definido pelo desenvolvedor, por padrão, ele é automaticamente incluído quando a classe é compilada.

```
public class Carro {  
    private String marca;  
    private String modelo;  
    private int ano;
```

```
    public Carro() {  
  
    }  
}
```

```
    // Métodos getters e setters  
}
```

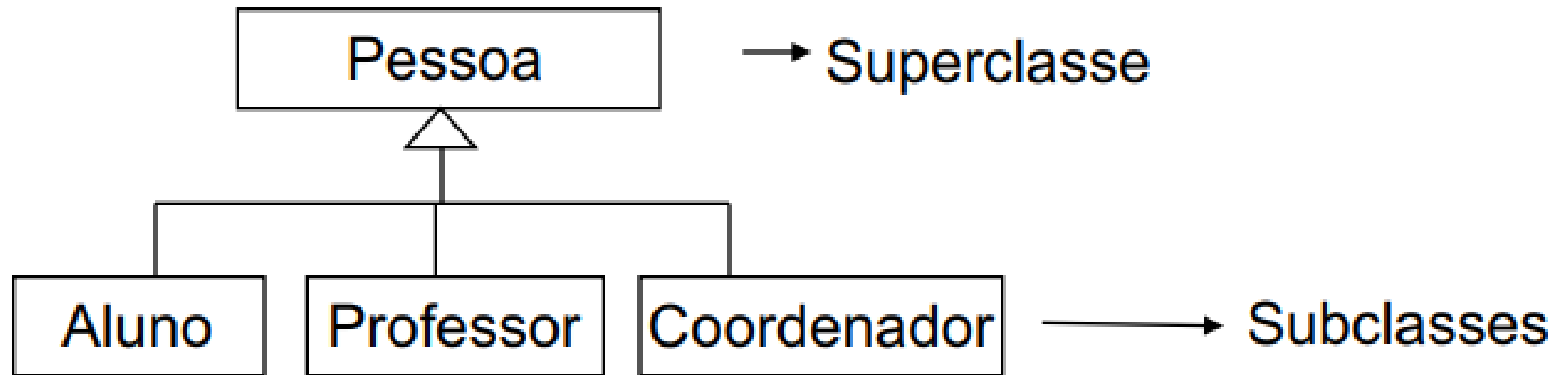
```
public class Principal{  
    public static void main(String[] args) {  
  
        Carro c = new Carro ("Chevrolet", "Ônix", 2022);  
        //Predefinindo valores  
    }  
}
```

```
public class Carro {  
    private String marca;  
    private String modelo;  
    private int ano;
```

```
    public Carro(String marca, String modelo, int ano) {  
        this.marca = marca;  
        this.modelo = modelo;  
        this.ano = ano;  
    }  
    //construtor
```

```
    // Métodos getters e setters  
}
```

OBSERVAÇÃO!



Exemplo 1

```
public class Pessoa {  
    protected String nome;
```

```
    public Pessoa(String nome) {  
        this.nome = nome;  
    }
```

```
}
```

```
public class Aluno extends Pessoa {  
    private int matricula;
```

```
    public Aluno(String nome, int mat) {  
        super(nome);  
        this.matricula = mat;  
    }
```

Exemplo 2

```
public class Pessoa{  
    protected String nome;
```

```
    public Pessoa(String nome){  
        super(); ←  
        this.nome=nome;  
    }
```

```
    // demais métodos devem vir aqui  
}
```

```
public class Aluno extends Pessoa{  
    private int mat;
```

```
    public Aluno(String nome, int mat){  
        super(nome);  
        this.mat=mat;  
    }
```

```
    // demais métodos devem vir aqui  
}
```

E a Main?

Exemplo 1

```
public class Main {  
    public static void main(String[] args) {  
  
        Aluno aluno = new Aluno("João", 12345);  
  
        System.out.println("Nome: " + a.getNome());  
        System.out.println("Matricula: " + a.getIdade());  
  
    }  
}
```

Exemplo 2

```
//import
public class Main {
    public static void main(String[] args) {
        // Scanner
        Aluno aluno = new Aluno("João", 12345);

        System.out.println("Digite o nome do aluno: ");
        String nomeAluno = sc.nextLine();

        System.out.println("Digite a matrícula: ");
        int matriculaAluno = sc.nextInt();

        Aluno aluno = new Aluno(nomeAluno, matriculaAluno);
    }
}
```


DÚVIDAS?

EXERCÍCIO

Questões 1 e 2

1) Implemente um programa em java que possua um professor com nome, idade e salário. Além disso, o programa deve ter implementado um aluno com nome, idade e matrícula . As informações devem ser feitas de forma fixa na instanciação do objeto e depois mostre todas informações na tela

2) Elabore um programa que resolva o seguinte problema:

```
Produto produto1 = new Produto("Camisa", 49.99, "Camisa polo azul");  
Produto produto2 = new Produto("Calça", 99.99, "Calça jeans preta");
```

Questão 3

Elabore um código com as classes que completem a classe principal:

```
public class Main {  
    public static void main(String[] args) {  
  
        Carro carro = new Carro("Toyota", "Corolla");  
        carro.mostrarDetalhes();  
  
    }  
}
```

Questão 4) Complete o código:

```
public class Animal {  
    // Atributos da classe Animal  
    // Construtor da classe Animal  
}  
  
public class Leao extends Animal {  
    private String pelo;  
  
    // Construtor da classe Mamifero  
  
    public void exibirInformacoes() {  
        // Método para exibir informações do mamífero  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
  
        Leao leao = new Leao("Mamífero", "Patas", "Juba");  
        Arara arara = new Arara();  
  
        System.out.println("Informações do Leao:");  
        // Exibir as informações do mamífero  
  
        System.out.println("Informações da Arara:");  
        System.out.println(arara.getClasse());  
        System.out.println(arara.getLocomocao());  
        System.out.println(arara.getQtdeOvos());  
    }  
}
```



LINGUAGEM DE PROGRAMAÇÃO I REVISÃO

MSc. Fernanda Dias

CENTRO UNIVERSITÁRIO UNIESP

Estrutura de Decisão Simples

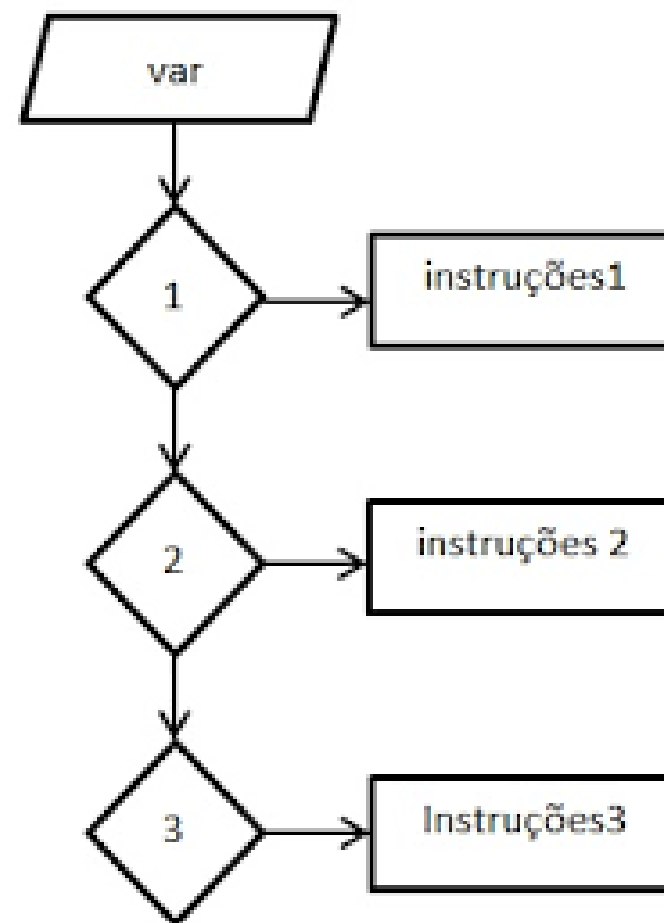
```
if (num1 > num2) {  
    System.out.println("é maior ");  
} else {  
    System.out.println("é menor");  
}
```

Estrutura de Decisão Composta

```
if (num1 > num2) {  
    System.out.println("é maior ");  
} else if (num1 == num2){  
    System.out.println("é igual");  
}else{  
    System.out.println("é menor");  
}
```

SWITCH

```
public class Main {  
    public static void main(String[] args) {  
        int x= 1;  
        switch (x){  
            case 0:  
                System.out.println("Opção 0 escolhida");  
                break;  
            case 1:  
                System.out.println("Opção 1 escolhida");  
                break;  
            case 2:  
                System.out.println("Opção 2 escolhida");  
                break;  
            default:  
                System.out.println("Opção 2 escolhida");  
        }  
    }  
}
```



FOR

```
for (ini; fim; incremento) {  
    <instruções> }  
}
```

Exemplo:

```
int i = 0;
```

```
for (i=0; i<5;i++) {  
    System.out.println(i);  
}
```

//exibe de 0 a 4

```
int i = 5;
```

```
for (i=5; i>0;i--) {  
    System.out.println(i);  
}
```

//exibe de 5 a 1

WHILE

```
while (condição) { <instruções> }
```

Exemplo:

```
int i = 0;
```

```
while(i <= 10) {  
    System.out.println(i++);  
}
```

//exibe de 0 a 10

```
int i = 0;
```

```
while(i <= 10) {  
    System.out.println(++i);  
}
```

//exibe de 1 a 11

DO WHILE:

```
do{<instrução>}while (condição);
```

Exemplo:

```
int i = 0;
```

```
do{  
    System.out.println(i++);  
} while(i <= 10) ;
```

//exibe de 0 a 10

```
int i = 0;
```

```
do{  
    System.out.println(++i);  
} while(i <= 10)
```

//exibe de 1 a 11

GET e SET

```
public class Pessoa {  
  
    private String nome;  
    private int idade;  
  
    public void setIdade(int idade) {  
        this.idade = idade;  
    }  
  
    public int getIdade() {  
        return idade;  
    }  
  
    // get e set para nome  
}
```

```
import java.util.Scanner;

public class Principal {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        Pessoa p = new Pessoa();

        System.out.println("Digite sua idade");
        p.setIdade(sc.nextInt());
        System.out.println("Idade:" + p.getIdade());

    }
}
```

HERANÇA

```
public class Pessoa{  
    protected String nome;  
    protected double idade;  
  
    //gets e sets  
  
    protected void aniversaio(){  
        idade++;  
    }  
}
```

```
public class Aluno extends Pessoa{  
  
    private int matricula;  
  
    //getMatricula  
    //setMatricul  
  
    // métodos  
}
```

CONSTRUTORES

```
public class Principal{  
    public static void main(String[] args) {  
  
        Carro c = new Carro ("Chevrolet", "Ônix", 2022);  
                                //Predefinindo valores  
    }  
}
```

```
public class Carro {  
    private String marca;  
    private String modelo;  
    private int ano;
```

```
    public Carro(String marca, String modelo, int ano) {  
        this.marca = marca;  
        this.modelo = modelo;  
        this.ano = ano;  
    }  
                                //construtor
```

```
    // Métodos getters e setters  
}
```

CONSTRUTORES e HERANÇA

```
public class Pessoa {  
    protected String nome;
```

```
    public Pessoa(String nome) {  
        this.nome = nome;  
    }
```

```
}
```

```
public class Aluno extends Pessoa {  
    private int matricula;
```

```
    public Aluno(String nome, int mat) {  
        super(nome);  
        this.matricula = mat;  
    }
```



DÚVIDAS?

EXEMPLO PARA ESTUDO

```
public class Animal {  
    // Atributos da classe Animal  
    // Construtor da classe Animal  
}  
}
```

```
public class Leao extends Animal {  
    private String pelo;  
  
    // Construtor da classe Mamifero  
  
}  
}
```

```
public class Main {  
    public static void main(String[] args) {  
  
        Leao leao = new Leao("Mamífero", "Patas", "Juba");  
        Arara arara = new Arara();  
  
        System.out.println("Informações do Leao:");  
        // Exibir as informações do mamífero  
  
        System.out.println("Informações da Arara:");  
        System.out.println(arara.getClasse());  
        System.out.println(arara.getLocomocao());  
        System.out.println(arara.getQtdeOvos());  
  
    }  
}
```

	COMPONENTE CURRICULAR: LINGUAGEM DE PROGRAMAÇÃO E BANCO DE DADOS	
	PERÍODO: P2	SEMESTRE: 2º
	NOME DO ALUNO (A):	

ATIVIDADE DE ESTUDO PARA JAVA E BANCO DE DADOS

Em um determinado sistema, um cliente pode realizar diversos pedidos que contém produtos diferentes em cada um deles. Quando o pedido é feito no sistema, apenas um vendedor tira a nota deste pedido. O sistema deve ser programado para atender que o pedido esteja relacionado unicamente a um cliente e que o vendedor possa atender diversos pedidos.

Uma informação importante é que o sistema deve solicitar o cadastro do código de identificação e nome do **cliente**, código e data do **pedido**, código e nome do **vendedor** e ser capaz de exibir tudo na tela, mas para isso, é importante uma organização para que a pessoa usuária possa selecionar quem ele vai cadastrar ou listar (switch- case).

Atenção: Neste negócio o cadastro do código e nome **produto são fixos**, porém a quantidade é algo que deve ser cadastrada manualmente. Para os produtos, temos a seguinte situação:

PRODUTO	
NÚMERO_PRODUTO	NOME_PRODUTO
111	Camisa
256	Rádio
387	Pomada
358	Garrafa
470	Tijolo

Observação: quando listar os produtos, é importante ter uma opção para vender o produto e decrementar do estoque a sua quantidade.

Com base nestas informações, realize:

- Modelo Entidade e Relacionamento;
- Dicionário de Dados;
- Modelo Relacional;
- Código Java com as classes, gets, sets, construtores e métodos.