

## Atividade prática

Um sistema de contas a pagar deve calcular os pagamentos a serem feitos. Esses pagamentos são calculados de diversas formas. Para evitar que o sistema de contas a pagar conecte-se com diversas classes de pagamentos, pode-se criar uma interface que será implementada por outras classes. Veja detalhes abaixo:

Observe para o operador ternário para verificar condições

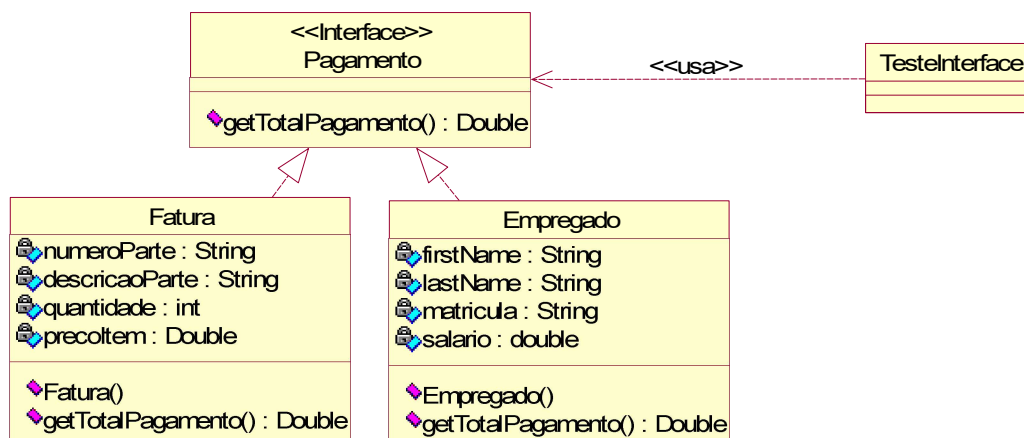
variable = (condition) ? expressionTrue : expressionFalse;

Exemplo:

```
int qtd = 11%2;  
String recebe = (qtd == 0) ? "Par" : "impar";
```

Saída: ímpar

## Diagrama de classes



## Código das classes acima

### Definição da interface pagamento

```
//Declaração da interface Pagamento  
  
public interface Pagamento {  
    public double getTotalPagamento(); // cálculo do total pagamento  
    //sem nenhuma implementação. Será implementado por alguma classe  
}  
// fim da declaração da interface
```

## Definição da classe fatura

```
//declaração da classe fatura que implementa Pagamento

public class Fatura implements Pagamento {

    private String numeroParte;
    private String descricaoParte;
    private int quantidade;
    private Double precoItem;

    // construtor
    public Fatura( String parte, String descricao, int qte, double
preco) {
        this.numeroParte = parte;
        this.descricaoParte = descricao;
        this.quantidade = qte;
        this.precoItem = preco;
    }

    public void setNumeroParte(String parte) {
        numeroParte = parte;
    }

    public String getNumeroParte() {
        return numeroParte;
    }

    public void setDescricaoParte(String descricao) {
        descricaoParte = descricao;
    }

    public String getdecricaoParte() {
        return descricaoParte;
    }

    public void setQuantidade(int qte) {
        quantidade = (qte < 0) ? 0 : qte ; // nao pode ser negativa
    }

    public int getQuantidade() {
        return quantidade;
    }

    public void setPrecoItem(double preco) {
        precoItem = ( preco < 0.0 ) ? 0.0 : preco; // valida preco
    }

    public Double getPrecoItem() {
        return precoItem;
    }

    // retorno string do objeto fatura
    public String toString() {
        return ("Fatura: \n" + "\nNumero Parte: " +
this.getNumeroParte() + "\nDescrição: " +
this.getdecricaoParte() + "\nQuantidade: " +
```

```

        this.getQuantidade() + "\nPreço por Item: " +
        this.getPrecoItem());
    }

    /**
     * metodo requerido para executar o metodo da interface Pagamento
     */
    public double getTotalPagamento() {
        return (this.getQuantidade() * this.getPrecoItem()); //calcula
preço total
    }
}

```

## Definição da classe empregado

```

//declaração da classe empregado que implementa Pagamento

public class Empregado implements Pagamento {
    private String firstName;
    private String lastName;
    private String matricula;
    private double salario;

    // construtor
    public Empregado( String first, String last, String mat, double
sal) {
        firstName = first;
        lastName = last;
        matricula = mat;
        salario = sal;
    }
    public void setFirstName(String first) {
        firstName = first;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setLastName(String last) {
        this.lastName = last;
    }
    public String getLastName() {
        return lastName;
    }
    public void setMatricula(String mat) {
        matricula = mat;
    }
    public String getMatricula() {
        return matricula;
    }
    public void setSalario(double sal) {
        this.salario = ( sal < 0.0 ) ? 0.0 : sal; // valida sal
    }
    public double getSalario() {
        return (this.salario);
    }

    // retorno string do objeto fatura
    public String toString() {

```

```

        return ("Número Matrícula: " + this.getFirstName() +
            this.getLastName() + this.getMatricula() +
            this.getSalario());
    }

    /**
     * metodo requerido para executar o metodo da interface Pagamento
     */
    public double getTotalPagamento() {
        return getSalario() * 10.0; //calcula preco total
    }
}

```

## Definição da classe testeInterface para testar uso da interface Pagamento

```

//Classe para testar a interface pagamento
public class Main {
    public static void main(String args[]) {

        Pagamento pagObjetos[] = new Pagamento[4];

        pagObjetos[0] = new Fatura("123", "bicicleta", 2, 200.00);
        pagObjetos[1] = new Fatura("222", "bola", 5, 10.00);
        pagObjetos[2] = new Empregado("Pedro", "Sousa", "222",
            350.00);
        pagObjetos[3] = new Empregado("João", "Silva", "111", 100.00);

        System.out.println( "Faturas e empregados processados
            polifomicamente pela interface:\n ");

        // processa os objetos
        for (int i = 0; i < pagObjetos.length; i++) {
            System.out.println(pagObjetos[i].toString() + " pagamento
                feito " + pagObjetos[i].getTotalPagamento());
        }
    }
}

```