

UNIP LIMEIRA – UNIVERSIDADE PAULISTA
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CAIO YAGO VILELA

F00JED-7

IMPLEMENTAÇÃO DO DIAGRAMA
USANDO O ECLIPSE

LIMEIRA – SÃO PAULO
2020

CAIO YAGO VILELA

F00JED-7

**IMPLEMENTAÇÃO DO DIAGRAMA
USANDO O ECLIPSE**

Relatório de Atividade Prática
Supervisionada (APS) para avaliação no 4º
Semestre letivo do curso de Ciência da
Computação apresentado à UNIP Limeira –
Universidade Paulista.

Orientador (es): Danilo Pereira

LIMEIRA – SÃO PAULO
2020

SUMÁRIO

1. OBJETIVO.....	4
2. JDK, JVM E JRE.....	5
3. CONCEITOS E DIAGRAMA.....	6
4. ARRAYLIST E HASHSET.....	7
5. CÓDIGOS.....	8
6. FICHA.....	15
7. BIBLIOGRAFIA.....	16

1. OBJETVO

Explicar a diferença entre Java Development Kit (JDK), Java Virtual Machine (JVM) e Java Runtime Environment (JRE).

Questões Sobre os conceitos de Orientação a Objetos em Java, analise as sentenças e marcar (V) para verdadeiro e (F) para falso.

Análise o diagrama de classe e fazer a implementação usando a IDE de sua preferência (Eclipse ou NetBeans).

Fazer a implementação das classes, interfaces e os relacionamentos entre elas.

Explicar as principais diferenças entre as estruturas de dados ArrayList (Lista) e HashSet (Conjunto) em Java. Fazer exemplos de código em Java.

2. JDK, JVM E JRE



Avaliação Presencial
Ciência da Computação
Unip Limeira - SP

Disciplina	Aplicações de Linguagem de Programação Orientada a Objetos - LPOO	Data	11/09/2020
Professor	Danilo Rodrigues Pereira	(0 – 5 pontos)	
Trabalho	Lista de Exercícios – 01		

Nome	Caio Yago Filipe
-------------	------------------

RA	F00760 - 4
-----------	------------

INSTRUÇÕES:

- Preencha o seu nome e registro acadêmico corretamente.
- Enviar a lista de exercícios e código-fonte zipados com a seguinte nomenclatura: Lista_Exercicio_01_NOME_RA.zip ou Lista_Exercicio_01_NOME_RA.rar
- Trabalhos caracterizados como plágio serão desconsiderados e todos terão as notas iguais a zero (0).
- Link para o envio: <https://bit.ly/32icgDe>
- Data Limite: 23/10

- 1) [1 ponto] Explique a diferença entre Java Development Kit (JDK), Java Virtual Machine (JVM) e Java Runtime Environment (JRE).

JVM é, em síntese, um programa que executa os aplicativos Java. É ele o faz convertendo os bytescodes gerados após a compilação do algoritmo escrito, em um código executável de máquina. Vimos também que a JVM é a pedra angular da plataforma. É considerada o componente mais importante.

JRE permite ao usuário final executar aplicativos Java em seu computador sendo assim essencial para execução de programas Java. Podemos entender que o JRE como sendo um plugin que inclui a JVM, bibliotecas e componentes adicionais para executar aplicações de linguagem.

JDK ~~compreende~~ compreende um pacote de "ferramentas" voltado exclusivamente para programadores, para desenvolver aplicações baseadas na linguagem Java. Ele é um conjunto de utilitários que permite criar sistemas de software para a plataforma Java. Ele inclui o JRE, a JVM, um conjunto de classes de API, compilador Java e arquivos necessários para criar aplicações.

- 02) [1 ponto] Sobre os conceitos de Orientação a Objetos em Java, analise as sentenças e marque

3. CONCEITOS E DIAGRAMA



Avaliação Presencial Ciência da Computação Unip Limeira - SP

(V) para verdadeiro e (F) para falso:

(V) O principal objetivo da programação orientada a objetos (POO) é facilitar a manutenção das aplicações. A POO é composta de três principais componentes: Classes, Métodos e Atributos.

(F) Podemos representar uma classe através de diagramas UML. O diagrama UML de uma classe é composto pelo nome da classe, pelos atributos, métodos que ela define também as regras de negócios a serem implementadas.

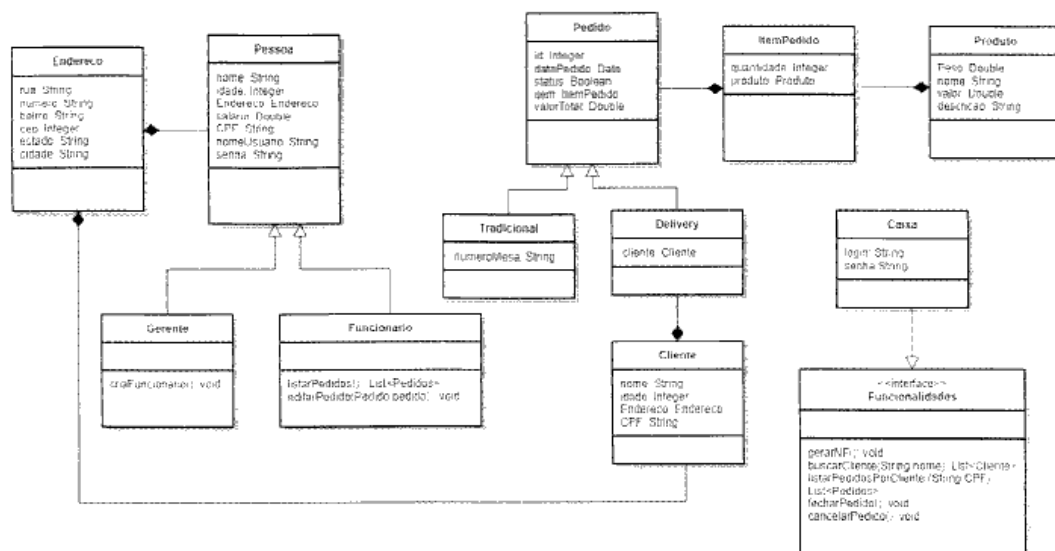
(F) A herança é uma forma eficaz de reutilização de código. Ocorre quando uma classe passa a herdar características (variáveis e métodos) definidas em outra classe. A classe genérica é denominada superclasse, classe base ou classe mãe. Já as classes específicas são denominadas subclasses, classes derivadas ou classes filhas.

(V) As classes abstratas são as que não permitem realizar qualquer tipo de instância. As classes derivadas, via de regra, deverão sobrescrever os métodos para realizar a implementação dos mesmos.

(V) Em Java, a herança só poderá ser feita em classes concretas e não é permitido herança múltipla.

(V) Em orientação a objetos, um contrato é chamado de interface. Uma interface pode definir uma série de métodos, mas nunca conter implementação deles.

03) [2 pontos] Análise o diagrama de classe a seguir e faça a implementação usando a IDE de sua preferência (Eclipse ou NetBeans).



4. ARRAYLIST E HASHSET



Avaliação Presencial
Ciência da Computação
Unip Limeira - SP

Faça a implementação das classes, interfaces e os relacionamentos entre elas.

Não é necessário implementar a regra de negócios dos métodos, apenas a assinatura dos mesmos. Exemplo:

```
public void abrirPedido(Integer idPedido) {
    // TODO
}
```

04) [1 ponto] Explique as principais diferenças entre as estruturas de dados ArrayList (Lista) e HashSet (Conjunto) em Java. Faça exemplos de código em Java.

Esta classe implementa a interface SET, apoiada por uma tabela hash (na verdade, uma instância HashMap). Não dá nenhuma garantia quanto à ordem de iteração do conjunto; em particular, não garante que a ordem permanecerá constante ao longo do tempo. Esta classe permite o elemento Nulo.

Implementação de array redimensionável da interface List. Implementa todas as operações de lista específicas e permite todos os elementos, incluindo nulo. Além de implementar a interface List, esta classe fornece métodos para manipular o tamanho da matriz que é usado internamente para armazenar a lista. (Esta classe é aproximadamente equivalente a Vector, exceto que não é sincronizada).

5. CÓDIGOS

• CAIXA

```
package trabalho_01;

/**
 *
 * @author Caio Yago Vilela
 */

public abstract class Caixa implements Funcionalidades {

    //Atributos

    private String login;
    private String senha;

    // Gets e Sets

    public String getLogin() {
        return login;
    }
    public String getSenha() {
        return senha;
    }
    public void setLogin(String login) {
        this.login = login;
    }
    public void setSenha(String senha) {
        this.senha = senha;
    }
}
}
```

• CLIENTE

```
package trabalho_01;

/**
 *
 * @author Caio Yago Vilela
 */

public class Cliente extends Endereco {

    //Atributos

    private String nome;
    private String idade;
    private String Endereco;
    private String CPF;

    // Gets e Sets

    public String getNome() {
        .....return nome;
    }
    public String getIdade() {
        .....return idade;
    }
    public String getEndereco() {
        .....return Endereco;
    }
    public String getCPF() {
        .....return CPF;
    }
    public void setNome(String nome) {
        .....this.nome = nome;
    }
    public void setIdade(String idade) {
        .....this.idade = idade;
    }
    public void setEndereco(String Endereco) {
        .....this.Endereco = Endereco;
    }
    public void setCPF(String CPF) {
        .....this.CPF = CPF;
    }
}
}
```

• DELIVERY


```

package trabalho_01;

/**
 *
 * @author Caio Yago Vilela
 */

public class Delivery extends Cliente {

    private Cliente cliente;

    public Cliente getCLiente() {
        .....return cliente;
    }
    public void setCliente(Cliente cliente) {
        .....this.cliente = cliente;
    }

}

```

• ENDERECO

```

package trabalho_01;

/**
 *
 * @author Caio Yago Vilela
 */

public class Endereco {

    // Variaveis(Atributos)

    private String rua;
    private String numero;
    private String bairro;
    private int cep;
    private String estado;
    private String cidade;

    // Metodos

    // Gets e Sets
    public String getRua() {
        return rua;
    }

    public String getNumero() {
        return numero;
    }

    public String getBairro() {
        return bairro;
    }

    public int getCep() {
        return cep;
    }

    public String getEstado() {
        return estado;
    }

    public String getCidade() {
        return cidade;
    }

    public void setRua(String rua) {
        this.rua = rua;
    }

    public void setNumero(String numero) {
        this.numero = numero;
    }

    public void setBairro(String bairro) {
        this.bairro = bairro;
    }

    public void setCep(int cep) {
        this.cep = cep;
    }

}

```

```

        public void setEstado(String estado) {
            this.estado = estado;
        }

        public void setCidade(String cidade) {
            this.cidade = cidade;
        }
    }
}

```

• FUNCIONALIDADES

```

package trabalho_01;

import java.util.List;

/**
 *
 * @author Caio Yago Vilela
 */

public interface Funcionalidades {

    public void gerarNF();
    public List <Cliente> buscarCliente(String nome);
    public List <Pedido> listarPedidosPorCliente(String CPF);
    public void fecharPedido();
    public void cancelarPedido();
}

```

• FUNCIONARIO

```

package trabalho_01;

import java.util.List;

/**
 *
 * @author Caio Yago Vilela
 */

public class Funcionario {

    public String criaFuncionario;

    List<Pedido> listarPedidos() {
        .....return null;
    }

    void editarPedido(Pedido pedido) {
        .....
    }
}

```

• GERENTE

```

package trabalho_01;

/**
 *
 * @author Caio Yago Vilela
 */

public class Gerente {

    void criaFuncionario() {

        Funcionario novo = new Funcionario();
        novo.criaFuncionario = "CAIO YAGO";

    }
}

```

• ITEM PEDIDO

```
package trabalho_01;

/**
 *
 * @author Caio Yago Vilela
 */

public class ItemPedido extends Produto {

    //Atributos

    private Integer quantidade;
    private Produto produto;

    // Gets e Sets

    public Integer getQuantidade() {
        return quantidade;
    }
    public Produto getProduto() {
        return produto;
    }
    public void setQuantidade(Integer quantidade) {
        this.quantidade = quantidade;
    }
    public void setProduto(Produto produto) {
        this.produto = produto;
    }
}
}
```

• PEDIDO

```
package trabalho_01;

import java.util.Date;

/**
 *
 * @author Caio Yago Vilela
 */

public class Pedido extends ItemPedido {

    //Atributos

    private Integer id;
    private Date datePedido;
    private Boolean status;
    private ItemPedido item ;
    private Double valorTotal;

    // Gets e Sets

    public Integer getId() {
        .....return id;
    }
    public Date getDatePedido() {
        .....return datePedido;
    }
    public Boolean getStatus() {
        .....return status;
    }
    public ItemPedido getItem() {
        .....return item;
    }
    public Double getValorTotal() {
        .....return valorTotal;
    }
    public void setId(Integer id) {
        .....this.id = id;
    }
    public void setDatePedido(Date datePedido) {
        .....this.datePedido = datePedido;
    }
    public void setStatus(Boolean status) {
        .....this.status = status;
    }
    public void setItem(ItemPedido item) {
        .....this.item = item;
    }
    public void setValorTotal(Double valorTotal) {
```

```

.....this.valorTotal = valorTotal;
}

}

```

• PESSOAS

```

package trabalho_01;

/**
 *
 * @author Caio Yago Vilela
 */

public class Pessoas extends Endereco {

    // Variaveis(Atributos)
    private String nome;
    private Integer idade;
    private Endereco Endereco;
    private Double salario;
    private String CPF;
    private String nomeUsuario;
    private String senha;

    // Gets e Sets
    public String getNome() {
        return nome;
    }

    public Integer getIdade() {
        return idade;
    }

    public Endereco getEndereco() {
        return Endereco;
    }

    public Double getSalario() {
        return salario;
    }

    public String getCPF() {
        return CPF;
    }

    public String getNomeUsuario() {
        return nomeUsuario;
    }

    public String getSenha() {
        return senha;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

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

    public void setEndereco(Endereco Endereco) {
        this.Endereco = Endereco;
    }

    public void setSalario(Double salario) {
        this.salario = salario;
    }

    public void setCPF(String CPF) {
        this.CPF = CPF;
    }

    public void setNomeUsuario(String nomeUsuario) {
        this.nomeUsuario = nomeUsuario;
    }

    public void setSenha(String senha) {
        this.senha = senha;
    }

}

```

- **PRINCIPAL**

```
package trabalho_01;

/**
 *
 * @author CAIO YAGO VILELA
 */

public class Principal {

    public static void main(String[] args) {

        // CLASSE PRINCIPAL

    }

}
```

- **PRODUTO**

```
package trabalho_01;

/**
 *
 * @author Caio Yago Vilela
 */

public class Produto {

    //Atributos

    private Double Peso;
    private String nome;
    private Double valor;
    private String descricao;

    // Gets e Sets

    public Double getPeso() {
        .....return Peso;
    }
    public String getNome() {
        .....return nome;
    }
    public Double getValor() {
        .....return valor;
    }
    public String getDescricao() {
        .....return descricao;
    }
    public void setPeso(Double Peso) {
        .....this.Peso = Peso;
    }
    public void setNome(String nome) {
        .....this.nome = nome;
    }
    public void setValor(Double valor) {
        .....this.valor = valor;
    }
    public void setDescricao(String descricao) {
        .....this.descricao = descricao;
    }

}
```

- **TRADICIONAL**

```
package trabalho_01;

/**
 *
 * @author Caio Yago Vilela
 */

public class Tradicional {
```

```

        private String numeroMesa;

        public String getNumeroMesa() {
            return numeroMesa;
        }
        public void setNumeroMesa(String numeroMesa) {
            this.numeroMesa = numeroMesa;
        }
    }
}

```

• HASHSET E ARRAYLIST

```

package trabalho_02;

import java.util.Collection;

/**
 *
 * @author CAIO YAGO VILELA
 */
public class ComparaTempo {

    // no teste, é pro tempo do HashSet ser ligeiramente menor que o do ArrayList

    private long totalRegistro;
    Collection teste;

    public ComparaTempo(Collection teste, long totalRegistro){ this.totalRegistro = totalRegistro; this.teste = teste; }

    public void testaOjeto() {

        long inicio = System.currentTimeMillis();
        System.out.println("INICIO DO TESTE...");
        for(int i = 0; i < this.totalRegistro; i++){ teste.add(i); }
        long fim = System.currentTimeMillis();
        long tempoInserir = fim - inicio;
        System.out.println("TEMPO GASTO INSERÇÃO: " + tempoInserir);

        inicio = System.currentTimeMillis();
        for(int i = 0; i < totalRegistro; i++){ teste.contains(i); }
        fim = System.currentTimeMillis();
        long tempoPesquisa = fim - inicio;
        System.out.println("TEMPO GASTO PESQUISA: " + tempoPesquisa);

        .....System.out.println("TEMPO GASTO TOTAL: " + (tempoPesquisa+tempoInserir));

    }

}

```

```

/**
 *
 * @author CAIO YAGO VILELA
 */

package trabalho_02;

import java.util.ArrayList;
import java.util.Collection;
import java.util.HashSet;

public class Exemplo {

    public static void main(String[] args) {
        .....
        int total = 30000; // Variável

        // A diferença pode ser pouca por conta de 30000 ser um número relativamente pequeno pra tais comparações

        Collection testeHash = new HashSet();
        ComparaTempo c1 = new ComparaTempo(testeHash, total);

        System.out.println("Teste com HASHSET");
        c1.testaOjeto(); // Imprimindo

        Collection testeList = new ArrayList();
        ComparaTempo c2 = new ComparaTempo(testeList, total);

        System.out.println("\nTeste com ARRAYLIST");
        c1.testaOjeto(); // Imprimindo

    }
}

```

6. FICHA

FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

Atividades Práticas Supervisionadas (laboratórios, atividades em biblioteca, iniciação científica, trabalhos individuais e em grupo, práticas de ensino e outras)

NOME: Caio Yago Vilela
 RA: FOOTED-7 CURSO: Ciência da Computação
 CAMPUS: Limoeira SEMESTRE: 4º semestre TURNO: NOTURNO

			ASSINATUR A	
DATA	ATIVIDADE	TOTAL DE HORAS	ALUNO	PROFESSOR
12/09	Estudo sobre a APS	5 Horas	Caio	Yago
14/09	Revisão sobre a disciplina	2 Horas	Caio	Yago
19/09	Resumo sobre as câmeras	4 Horas	Caio	Yago
26/09	Programação e Recursos Práticos	2 Horas	Caio	Yago
23/09	Programação sobre as câmeras com uso das PDF da aula	4 Horas	Caio	Yago
03/10	Compartilhando a experiência com câmeras	5 Horas	Caio	Yago
10/10	Manuseio das câmeras	2 Horas	Caio	Yago
14/10	Implementação de câmeras	2 Horas	Caio	Yago
19/10	Resumo sobre câmeras	10 Horas	Caio	Yago
19/10	Programação sobre a APS	2 Horas	Caio	Yago
24/10	Programação de Arquivos sobre as	4 Horas	Caio	Yago
25/10	Manuseio das câmeras	4 Horas	Caio	Yago
26/10	Compartilhando a experiência e o conhecimento sobre câmeras	2 Horas	Caio	Yago

TOTAL DE HORAS: 60 Horas

7. BIBLIOGRAFIA

WIKIPÉDIA, A ENCICLOPÉDIA LIVRE. Wikipédia, a enciclopédia livre. **Wikipédia, a enciclopédia livre**. Disponível em: <https://pt.wikipedia.org/wiki/Diagrama_de_classes>. Acesso em: 18 setembro 2020.

DEVMEDIA. Declaração e utilização de classes. 2017. Disponível em: <<https://www.devmedia.com.br/java-declaracao-e-utilizacao-de-classes/38374>>. Acesso em: 20 setembro 2020.