

	DISCIPLINA: PROGRAMAÇÃO ORIENTADA A OBJETOS
	RESPONSÁVEL: RODRIGO DA CRUZ FUJIOKA
	PERÍODO: 2020.2
	TURMA:Sistemas de Informação(Noite)
	Lista de Exercício 01
	Versão: 1.2
GITHUB: <u>https://github.com/Spl3F/Prog-Orientada-Objeto-PUC</u>	Nota

Os códigos e respostas devem estar versionados no GITHUB

Histórico de Alterações

Versão	Data	Responsável	Alterações
1.0	05-03-2016	Rodrigo da Cruz Fujioka	Criação do Documento
1.1	09-08-2016	Rodrigo da Cruz Fujioka	Ordenação das questões teorias e práticas.
1.2	13-08-2021	Rodrigo da Cruz Fujioka	Atualização de material.

Histórico de Revisão

Versão	Data	Revisor

1. Defina Abstração, qual a sua utilidade?

Abstração é a representação de um objeto real em uma classe em java com seus atributos, características e ações.

2. Qual a relação entre Encapsulamento e Abstração?

O encapsulamento serve para restringir o acesso que o usuário da classe tem de seus atributos e métodos, gerados a partir da abstração, e isso acontece com getters, setters e conteúdos deixados como private.

3. Quais as vantagens de se utilizar Herança implemente um exemplo em forma de código?

A herança serve para reutilizar um código já escrito, e utilizá-lo em outros contextos.

Ex.: public class Carro extends Veiculo

Nesse contexto a classe carro herda os métodos e atributos que foram criados na classe Veiculo contanto que sejam public ou protected.

4. Private, Public, Static, diga para que serve cada modificador deste?

O modificador private significa que o atributo ou o método só pode ser acessado de dentro da própria classe em que ele está presente, e o oposto ocorre com public, onde qualquer um que tem acesso a classe, pode acessar os atributos e métodos que estão public. E static é quando alguma coisa com esse modificador é único para todas as instâncias da classe, ou seja todos os objetos.

5. Descreva todos os passos da criação até a execução de um programa escrito em Java.

Para um programa executar em java, é necessário a criação de uma classe, que tem o mesmo nome que o arquivo e geralmente começa com a primeira letra maiúscula, e dentro dessa classe é necessário a criação do “main” que é escrito como “public static void main(String args[])” além de precisar baixar a jdk e um editor de texto.

6. JVM, JDK e JRE para que servem?

JDK significa “Java Development Kit” é o que é necessário para a criação de um programa em java contendo o compilador e depurador sendo ele quem compila o código para bytecode, além de vir com a JRE que significa “Java Runtime Environment” que é o ambiente responsável pela execução do código e também contém a JVM que significa “Java Virtual Machine” e é o que executa o bytecode independente do sistema operacional.

7. Informe o que você entende com relação a Classes e Objetos? O que é que define o comportamento e o estado dos mesmos?

As classes são basicamente o que definem o objeto, como se fosse uma espécie de molde, contendo seus atributos e métodos, podendo ser criadas diversos objetos de uma classe. E o que define os comportamentos e estados deles são os modificadores, métodos e atributos da classe.

8. Como é realizada a declaração de uma Classe Java? Existe alguma regra com relação a nomenclatura?

O nome de uma classe em java deve ser o mesmo do nome do arquivo .java e a primeira letra do nome deve ser maiúscula por convenção, e para a classe ser criada é necessário a palavra-chave “class” antes do nome, e depois do nome fica o corpo da classe, cercado de chaves “{}”.

9. O que é Javadoc? Para que serve? Mostre um Exemplo.

O Javadoc é um gerador de documentação para programas em java no formato HTML e serve para facilitar a compreensão do programa, e segue um formato específico de comentários e tags.

Ex.:

```
/**  
 *  
 * @author Felipe Noleto  
 */  
  
/**  
 *  
 * @param descricao do parametro do metodo  
 * @return descricao do retorno do metodo  
 * @exception descricao das excecoes geradas  
 */
```

10. Fale sobre o ciclo de vida dos Objetos.

O objeto é criado em disco, depois é carregado em memória, então procura por membros da classe estáticos já inicializados, em seguida é alocada memória para o objeto e suas variáveis, em fim é chamado o construtor da classe, e após seu uso é removido o objeto e suas variáveis da memória.

11. O que é Garbage Colector? Para que serve?

O Garbage Colector, que significa coletor de lixo, é uma forma de gestão de memória automática, que no caso procura por memória que foi alocada para o programa e não está mais sendo utilizada, e a torna disponível novamente.

12. Comente a afirmativa “Em java o programador tem que se preocupar em liberar a memória quando estiver programando”.

Essa frase é falsa graças a existência do garbage collector, que cuida desse aspecto para o programador.

13. Declare três variáveis com tipos diferentes, sem atribuir valor. Depois atribua um valor qualquer a elas com dados digitados pelo usuário.

```
import java.util.Scanner;

class Teste{
    int numero;
    String nome;
    float numeroReal;

    public static void main(String[] args) {

        Teste teste = new Teste();
        Scanner sc = new Scanner(System.in);

        teste.numero = sc.nextInt();
        teste.nome = sc.nextLine();
        teste.numeroReal = sc.nextFloat();
    }
}
```

14. Crie uma variável do tipo int, atribuindo um valor a ela. Depois crie uma variável do tipo double, atribuindo a ela o valor da primeira variável criada.

```
int variavel = 10;
double numero = variavel;
```

15. Implemente um programa que receba dados digitados pelo usuário e atribua a uma variável do tipo String e imprima a frase inserida no console.

```
import java.util.Scanner;

class Teste{
    String variavel;

    public static void main(String[] args) {

        Teste teste = new Teste();
        Scanner sc = new Scanner(System.in);

        teste.variavel = sc.nextLine();

        System.out.println(teste.variavel);
    }
}
```

16. Concatene na variável criada acima a frase “ – Exemplo String”.

```
import java.util.Scanner;

class Teste{
    String variavel;

    public static void main(String[] args) {

        Teste teste = new Teste();
        Scanner sc = new Scanner(System.in);

        teste.variavel = sc.nextLine();

        System.out.println(teste.variavel + " - Exemplo String");
    }
}
```

17. Imprima na tela (console) o número de caracteres da String.

```
import java.util.Scanner;

class Teste{
    String variavel;

    public static void main(String[] args) {

        Teste teste = new Teste();
        Scanner sc = new Scanner(System.in);

        teste.variavel = sc.nextLine();

        System.out.println(teste.variavel.length());
    }
}
```

18. Crie uma nova String, pegando o pedaço da primeira String que compreenda a frase “Exemplo String”.

19. Calcule o resto da divisão de 99 por 4.

```
System.out.println(99 % 4);
```

20. Crie uma variável inteira com um valor qualquer e verifique se o valor desta variável é menor que 15 ou maior que 100.

```
class Teste{

    int variavel = 10;
```

```
public static void main(String[] args) {  
  
    Teste teste = new Teste();  
  
    if (teste.variavel < 15){  
        System.out.println("Menor que 15");  
    }  
    else if (teste.variavel > 100){  
        System.out.println("Maior que 100");  
    }  
}  
}
```

21. Crie uma variável com valor de ponto flutuante com um valor qualquer e verifique se o valor desta variável está entre 1.99 e 5.99.

```
class Teste{  
  
    float variavel = 1.98f;  
  
    public static void main(String[] args) {  
  
        Teste teste = new Teste();  
  
        if (1.99f < teste.variavel && teste.variavel < 5.99f){  
            System.out.println("esta entre 1.99 e 5.99");  
        }  
    }  
}
```

22. Agora compare se o valor das duas variáveis acima é igual.

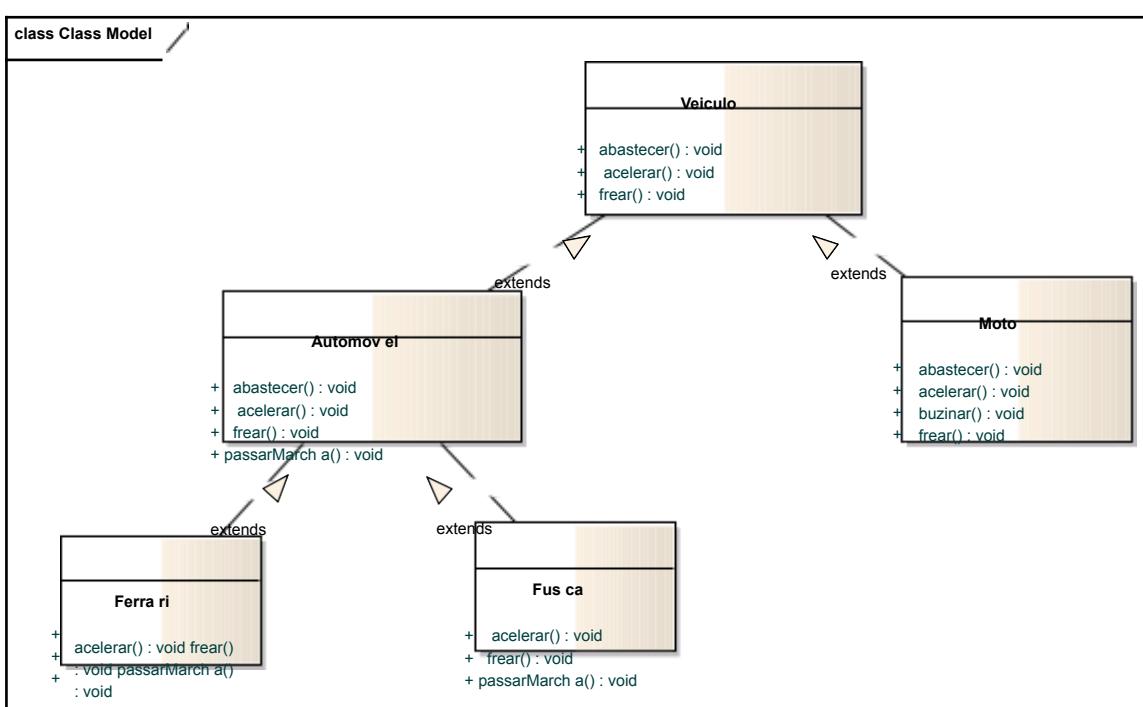
```
class Teste{  
  
    public static void main(String[] args) {  
  
        int variavel = 5;  
        double real = 5;  
  
        if (variavel == (int) real){  
            System.out.println("O valor é igual");  
        }  
    }  
}
```

23. Calcule o valor da multiplicação de um int por um double, atribuindo o valor a um int.

```
class Teste{  
  
    int variavel = 5;  
    double real = 2.21;  
    int resultado;  
  
    public static void main(String[] args) {  
  
        Teste teste = new Teste();  
  
        teste.resultado = (int) (teste.variavel * teste.real);  
    }  
}
```

```
}
```

24. Crie as classes ilustradas no diagrama abaixo. Para cada método você deve imprimir a mensagem correspondente: abastecer() deve imprimir "abastecendo" e assim por diante.



```
class Veiculo{

    void abastecer() {
        System.out.println("Abastecendo");
    }

    void acelerar() {
        System.out.println("Acelerando");
    }

    void frear() {
        System.out.println("Freando");
    }
}

class Automovel extends Veiculo{
```

```

void passarMarcha() {
    System.out.println("Passando marcha");
}

}

class Moto extends Veiculo{

    void buzinar(){
        System.out.println("Buzinando");
    }
}

class Fusca extends Automovel{

}

class Ferrari extends Automovel{
}

```

25. Escreva um programa que imprima na tela a soma dos números ímpares entre 0 e 30 e a multiplicação dos números pares entre 0 e 30.

```

class Teste{
    public static void main(String[] args) {
        int soma = 0;
        long multi = 1;
        for (int i = 1; i < 30; i++, i++) {
            soma = soma + i;
        }
        System.out.println(soma);
        for (int i = 2; i < 30; i++, i++) {
            multi = multi * i;
        }
        System.out.println(multi);
    }
}

```

26. Faça um programa para imprimir os números primos de 1 a 123.

```

public static void main(String[] args) {
    int i = 0;
    int y = 0;

    for (i = 1; i <= 123; i++) {
        int cont = 0;
        for(y = i; y >= 1; y--) {
            if(i % y == 0) {
                cont = cont + 1;
            }
        }
        if(cont == 2)
            System.out.println(i);
    }
}

```

```

        }
        if (cont == 2) {
            System.out.println(i);
        }
    }
}

```

27. Faça um programa para ler um número do teclado e imprimir na tela se ele é par ou ímpar. Imprima também se ele é primo.

```

import java.util.Scanner;

class Teste{

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int numero;
        numero = sc.nextInt();
        if (numero % 2 == 0){
            System.out.println("numero e par");
        }
        if (numero % 2 == 1){
            System.out.println("numero e impar");
        }

        int cont = 0;
        for(int y = numero; y >= 1; y--){
            if(numero % y == 0){
                cont = cont + 1;
            }
        }
        if (cont == 2){
            System.out.println("o numero e primo");
        }
    }
}

```

28. Determine a quantidade de homens e mulheres (separadamente) que são maiores de idade, baseado na leitura de dados de 10 pessoas.

29. O valor pago por um Hotel da Praia de Iracema para seus porteiros é de R\$ 10,25 por hora de trabalho. Faça um programa que pergunte ao usuário quantas horas ele trabalhou e imprima na tela o valor do salário a ser recebido por ele.

```

import java.util.Scanner;

class Teste{

    public static void main(String[] args) {

```

```
Scanner sc = new Scanner(System.in);
float hora = 10.25f;
int horas;

System.out.println("Quantas horas trabalhou?");
horas = sc.nextInt();
System.out.println("O valor do salario é R$" + horas * hora);
}
}
```

30. Modifique o programa anterior para que o sistema imprima uma mensagem de alerta quando o valor a ser pago ao funcionário seja inferior a R\$ 50,00.
 "Atenção, dirija- se à direção do Hotel!".

```
import java.util.Scanner;

class Teste{

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        float hora = 10.25f;
        int horas;

        System.out.println("Quantas horas trabalhou?");
        horas = sc.nextInt();
        if (horas * hora < 50) {
            System.out.println("Atenção, dirija- se à direção do Hotel!");
        } else{
            System.out.println("O valor do salario é R$" + horas * hora);
        }
    }
}
```

31. Existem 454 gramas em uma libra, e 1000 gramas em um quilo. Faça um programa que converta quilos para libras e vice-versa. (Dica: use um caractere indicando a ordem da conversão, exemplo "java q 1000" seria o comando para converter 1000 quilos para libra, e "java l 1000" seria o comando para converter 1000 libras para quilo.

```
import java.util.Scanner;

class Teste{

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("1. Converter de kilo pra libras");
        System.out.println("2. Converter de libras pra kilos");

        int opcao = sc.nextInt();

        if (opcao == 1){
            System.out.println("Quantos kilos?");
            double kilos = sc.nextDouble();
            System.out.print(kilos / 0.454);
        }
    }
}
```

```

    if (opcao == 2) {
        System.out.println("Quantas libras?");
        double libras = sc.nextDouble();
        System.out.print(libras * 0.454);
    }
}
}

```

32. Implemente um programa para calcular a área de um trapézio,
onde:
h = altura
b = base menor
B = base maior
Área = $(h \cdot (b + B)) / 2$

```

class Teste{

    public static void main(String[] args) {
        int h = 5;
        int b = 5;
        int B = 5;
        System.out.println("a area do trapezeo e = " + (h * (b + B)) / 2);
    }
}

```