**PROGRAMAÇÃO ORIENTADA A OBJETO**

**EXERCÍCIO I**

1. DESCREVA O QUE VOCÊ ENTENDEU SOBRE ORIENTAÇÃO A OBJETO.

Tecnologia que enxerga os sistemas como sendo coleção de objetos integrantes. Consiste em considerar os sistemas computacionais não como uma coleção estruturada de processos, mas sim como uma coleção de objetos que interagem entre si.usa abstração para criar modelos baseado no mundo real.

2. DIFERENCIE CLASS DE OBJETO.

As classes definem os tipos dos objetos, por isso, objetos também são chamados de instâncias da classe que os define e usam o nome da classe como seu tipo. Um objeto é uma é entidade do mundo real que pode ser representado, de acordo com sua relevância para solução de um determinado problema, no ambiente estudado.

Classes definem um conjunto com características semelhantes. Os objetos são as instancias de cada classe.

\*Então o objeto é uma instância da classe. Então na classe você pode dizer que aquele objeto tem um cor, no objeto você diz qual é a cor. Só pode dizer isso porque foi definido na classe que essa informação deve estar no objeto.

3. O QUE SÃO MÉTODOS?

Métodos são as operações que podem atuar sobre os dados e que expressam as “mensagens” às quais os objetos respondem, e estes consistem em construtores, subprogramas e funções.\* Eles definem o comportamento dos objetos daquela classe, ou seja, são ações que o objeto pode realizar.

\*\*Os métodos são ações ou procedimentos, onde podem interagir e se comunicarem com outros objetos. A execução dessas ações se dá através de mensagens, tendo como função o envio de uma solicitação ao objeto para que seja efetuada a rotina desejada.

4. O QUE SÃO ATRIBUTOS?

Atributos são as características que descrevem o objeto, ou seja, são variáveis ou campos que armazenam os diferentes valores que as características dos objetos podem conter.

5. QUAIS SÃO AS VANTAGENS DA ORIENTAÇÃO A OBJETOS?

Em relação a outros paradigmas de programação, o grande diferencial está no conceito de herança, mecanismo através do qual definições existentes podem ser facilmente estendidas.

Facilidade de manutenção; escalabilidade; reutilização de código; portabilidade (usando Java, pois proporciona a instalação em vários maquinas).

6. O QUE É JAVA?

É uma linguagem de programação Orientada a Objetos muito conveniente para o desenvolvimento de software que funcione em conjunto com a internet.

Fortemente tipada; originária do C, linguagem interpretada;

\*As aplicações em Java normalmente podem ser executadas em qualquer plataforma que possua a Java Virtual Machine (JVM) instalada, independente da arquitetura do computador.

7. PESQUISE EXEMPLOS DE CÓDIGOS EM JAVA E EXECUTE NA LINHA DE COMANDO.

**Exemplo 1**

public class AloMundo {

static public void main(String[]args) {

System.out.print(“Alo Mundo”);

}

}

**Exemplo 2**

public class MDC {

private int X=0;

private int Y=0;

public int Calcula (int X, int Y) {

if (Y == 0)

return X;

else if (X == 0)

return Y;

else

return Calcula(Y,X%Y);

}

Public static void main (String[] args) {

MDC a1 = new MDC();

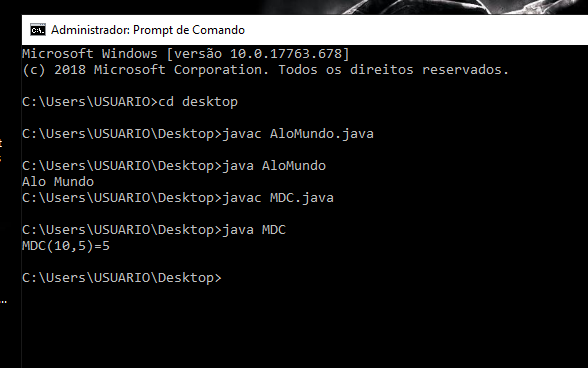
Int R1 =a1.Calcula(10,5);

System.out.print(MDC(“10,5) = “+ R1 +’\n’);

System.exit(0);

}

}



7. APRESENTE A SINTAXE PARA AS SEGUINTES DECLARAÇÕES:

1. Variável inteira

private int numero;

1. Constante booleana

private final numero = false;

1. Variável real que começa com o valor 10

private float numero = 10;

1. Instanciar objeto de classe Funcionario

Funcionario dados = new funcionario();

8. QUANTAS VEZES O CÓDIGO ABAIXO SERÁ EXECUTADO E QUAL SERÁ O VALOR DO CONTADOR QUANDO O LAÇO TERMINAR?

Int contador = 10

do {

...

Contador++; /\* incrementa 11,12,13,14,15,16 \*/

} while (contador <=15); /\* menor/igual 15 repete = sai do loop quando atingir 16 \*/

++contador; /\* termina em 16+1=17 \*/

9. VERIFIQUE O QUE É IMPRESSO NA TELA

For (int i=0; i<4; i++ {

Switch (i) {

Case 1:

System.out.println(“Domingo”);

break;

default:

System.out.println(“Este não é um dia válido!”);

}

}

*RESPOSTA: O i começa em 0 e vai pro default, passa a ser 1, imprime domingo, passa a ser 2 imprime default, passa a ser 3, imprime default.*

default: Este não é um dia válido!

1: Domingo

default: Este não é um dia válido!

default: Este não é um dia válido!

10. DECLARE UM ARRAY DE INTEIROS COM 5 POSIÇÕES E INSTANCIE COM OS VALORES:

6, 8, 100, 67, 8

int [] x = new int [5];

int [] x = { 6, 8, 100, 67, 8 };

11. ATRIBUA NA TERCEIRA POSIÇÃO O VALOR 0

valores [2] = 0;

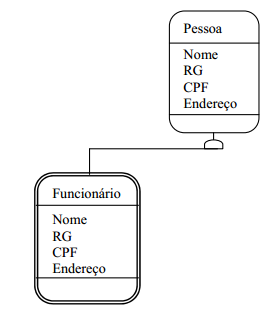
12. IDENTIFIQUE OS TIPOS DE VISIBILIDADES E DEFINA-OS

public (outras classes podem acessar esse atributo mesmo estando em pacotes diferentes), private (somente a própria classe pode acessar esses dados), protecteGed (são todas as classes do mesmo pacote mais as subclasses); package (todos podem acessar as classes daquele mesmo pacote );

13. EXPLIQUE O CONCEITO DE HERANÇA

Herança é o mecanismo pelo qual uma classe obtém as características e métodos de outra para expandi-la ou especializa-la de alguma forma, ou seja, uma classe pode herdar características, métodos, atributos de outra classes.

14. FORNEÇA ATRAVES DE DIAGRAMA DE CLASSES UMA HIERARQUIA DE HERANÇA.



\*precisa ser uma seta aponta do “filho” para o “pai”.

15. APRESENTE A SINTAX EM JAVA DA HERANÇA PRESENTE NO DIAGRAMA FORNECIDO.

public class pessoa {

Private String nome;

Private String rg;

Private String CPF;

Private String endereço;

}

public class funcionario extends pessoa {

}

16. APRESENE UM EXEMPLO EM CÓDIGO COM SUPER CLASSE ABSTRATA

public abstract class pessoa {

Private String nome;

Private String rg;

Private String CPF;

Private String endereço;

}

public class funcionario extends pessoa {

}

\*abstract não pode ser instanciada.

17. QUAL SERÁ O VALOR DAS VARIÁVEIS i, j e k?

int i=15, j, k;

j = --i; /\* j vai valer 14 \*/

i++; /\* decrementa do i = 15 \*/

k = j++; /\* k vai ser 14 e o j vai ser 15 \*/

i += 10+k /\* i recebe i (15) + 10 + k (14) \*/

RESPOSTA: i= 39, j=15, k=14

18. DEFINA UMA CLASSE COM O MÉTODO SOMA. A SOMA DEVE RETORNAR UM VALOR REAL.

public class Teste {

public float soma (float a, float b) {

return a+b;

}

}

19. DEFINA O QUE SE PEDE:

UMA CLASSE ALUNO, COM VARIÁVEIS NOME, CPF E MATRICULA.

public class Aluno {

private String nome;

private String cpf;

private String matricula;

}

UMA CLASSE AUXILIAR COM MÉTODO MAIN QUE INSTANCIE UM OBJETO DA CLASSE ALUNO E INSIRA VALORES PARA OS ATRIBUTOS. A INSERÇÃO DEVE SER FEITA ATRAVÉS DE MÉTODOS GET’S E SET’S

public class auxiliar {

public static void main(String[] args) {

Aluno var = new aluno();

var.setNome(“”);

var.setCpf(“”);

var.setMatricula(“”);

}

}