Pontifícia Universidade Católica de Goiás



Felipe Fernandes Almeida Manso

**Lista de Exercícios 1**

Goiânia, 05 de setembro de 2018.

Lista de Exercicios:

1 – Objeto é um conceito, abstração ou entidade com limites bem definidos e um significado para a aplicação, encapsulam dados e operações associadas que manipulam esses dados;

2 – O termo classe refere-se à implementação de software de um tipo de objeto. Um tipo de objeto especifica uma família de objetos sem estipular como o tipo e o objeto são implementados. Os tipos de objetos são especificados durante a análise OO. Os tipos de objetos são implementados como módulos enquanto na linguagem orientada a objeto, os tipos de objetos são classificados como classes.

3 – Classe é o conceito que um objeto deve seguir, objeto é de fato o ITEM.

4 – A classe molda como serão os objetos criados seguindo suas exigências, assim todos os objetos daquela classe terão propriedades similares.

5 – Abstração é a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais. Em modelagem orientada a objetos, uma classe é uma abstração de entidades existentes no domínio do sistema de software.

6 – Uma classe abstrata é uma classe que não tem instancias diretas. Que tem como objetivo encapsular outras classes com comportamento comum, que podem sugir naturalmente na modelagem ou serem criadas para promover o reuso.

7 – Ela simplificar um código usando as mesmas variáveis e métodos da classe Herdada.

8 – É possível a herança múltipla entre Interfaces em java, mas não pode ser feito entre Classes.

Isso ocorre, pois, as classes herdadas podem possuir uma mesma assinatura em seus métodos porem com implementações diferentes neles.

9 – Polimorfismo é quando um mesmo elemento para diferentes formas, permitindo que o mesmo elemento seja utilizado de formas diferentes.

10 – O Static serve para informar que a classe não pode ser instanciada.

11 – Sim, porem os métodos e atributos estáticos só podem acessar outros métodos e atributos estáticos da mesma classe.

12 – O Final torna o atributo ou método imutável, não sendo possível fazer qualquer alteração em seu valor.

13 – Uma exceção é quando ocorre uma execução inesperada no código, seja por erro de logica ou imput incorreto de valores.

14 –

try {

  // código que inclui comandos/invocações de métodos

  // que podem gerar uma situação de exceção.

} catch (XException ex) {

  // bloco de tratamento associado à condição de

  // exceção XException ou a qualquer uma de suas

  // subclasses, identificada aqui pelo objeto

  // com referência ex

} catch (YException ey) {

  // bloco de tratamento para a situação de exceção

  // YException ou a qualquer uma de suas subclasses

} finally {

  // bloco de código que sempre será executado após

  // o bloco try, independentemente de sua conclusão

  // ter ocorrido normalmente ou ter sido interrompida

}

15 – O Throws retorna para o método que chamou o método que está entrando em exceção para que ele trate aquela exceção e não o próprio método.

16 – Cada String informada é registrada e se torna imutável, qualquer alteração realizada irá criar uma nova alocação na memória para a nova string.

17 – É recomendado o uso do StringBuffer ou StringBuilder que torna a String em algo mutável, que pode ter seu tamanho e valores alterados.

18 – É a interface absoluta na hierarquia de coleções. Dela descendem as interfaces Set, Queue (filas) e List (listas) e Map que formam a base das coleções genéricas da linguagem Java.

19 – É recomendado quando é necessário armazenar, recuperar, manipular dados com mais facilidade e agilidade.

20 – Set: Esta interface não armazena a posição em que os elementos estão armazenados, isto é, ela entende que as classes que a implementam são como conjuntos matemáticos onde a posição não é relevante. Duas implementações que posso citar são o HashSet e o TreeSet, onde o último ordena os elementos em sua ordem natural.

List: Ao contrário de Set, as classes que implementam List possui o método get(int index) entre outros, isto é, é possível repurar os elementos da coleção utilizando seu índice. Duas implementações são o ArrayList e o LinkedList. O ArrayList possui agilidade na inserção de dados enquanto que o LinkedList possui agilidade na remoção. A classe LinkedList , como seu nome indica é provavelmente uma lista ligada. A pilha também pode ser simulada com esta classe pois a mesma possui metodos getFirst e getLast.

Map: As classes que implementam Map possuem a característica de armazenar pares chaves - valor. A pesquisa nas duas primeiras interfaces se dá através do método contains(Object o) ou get(int index), já nas classes que implementam Map poderão ser pesquisadas pelo objetos chave. Exemplo: get("Teste").

21 – É possível utilizar o treeSet para a ordenação.

22 – É possível utilizar o Map para criação de estrutura chave-valor