

OO em Java

1 - O que significa Java ser uma linguagem interpretada e o que é a JVM?

Java Virtual Machine. É a máquina virtual, que existe entre o sistema operacional e a aplicação, uma camada extra responsável por "traduzir" - mas não apenas isso - o que sua aplicação deseja fazer para as respectivas chamadas do sistema operacional onde ela está rodando no momento. Em outras palavras, ela é responsável por gerenciar memória, threads, a pilha de execução, etc.

2 - O que é a classe Object e como ela afeta os programas em Java?

A classe Object é a superclasse de todas outras classes. Todos objetos, incluindo arrays, implementam os métodos dessa classe.

3 - O que é um Float em Java. Qual a diferença do Float para um float?

Float é uma classe que possibilita instânciação de objetos para serem utilizados com a lógica de ponto flutuante. Um Float representa uma instância de uma classe, que possui atributos e métodos, enquanto um float é um tipo primitivo.

```
Float número;        // um objeto da classe float
float numero2;        // um tipo primitivo float
```

Associação

1 - O que é e quais são os tipos de associação entre classes em OO?

Em OO existem duas associações: Agregação e Composição.

- Agregação: Relação entre classes de forma que uma faz parte da outra, mas cada objeto possui seu próprio ciclo de vida, ex:

```
public class Foo {
    private Bar bar;
    Foo(Bar bar) {
        this.bar = bar;
    }
}
```

O objeto bar pode continuar a existir depois de ser utilizado pela classe Foo.

- Composição: Relação entre classes de forma que uma depende da outra para existir, ex:

```
public class Foo {
    private Bar bar = new Bar();
}
```

O objeto bar é instanciado dentro da classe Foo e quando Foo terminar, Bar também será finalizado.

2 - Exemplifique cada tipo de associação.

Interfaces

1 - O que é e para que serve uma Interface?

Interface é uma estrutura semelhante a classe, porém que possui apenas métodos que serão implementados por classes que virão a implementar essa interface. As interfaces são usadas quando queremos abstrair um tipo de comportamento. Uma interface serve para esclarecer quais classes possuem determinado comportamento, ou seja, quais classes possuem tal “contrato”, possibilitando a reutilização de código.

2 - Como se cria e se utiliza uma interface em Java?

Uma interface deve ser criada com métodos públicos que serão sobre-escrevidos pelas subclasses, além disso, a classe que for implementar a interface deve possuir a assinatura *implements*.

Threads

1 - O que é programação concorrente e o que é uma thread?

Programação concorrente ocorre quando o programa precisa executar duas ou mais tarefas “ao mesmo tempo”, quando isso acontece, o processador precisa lidar com as tarefas de forma simultânea, fazendo as tarefas concorrer pelos recursos do processador. Em um único programa, caso seja necessário efetuar as tarefas simultaneamente é necessário utilizar Threads.

2 - Quais são as duas maneiras de criar uma thread em Java?

A thread pode ser feita pela implementação da interface Runnable com a sobrescrita do método run(), ou pode ser feita quando a classe herda da classe Thread diretamente, também tendo que implementar o método run().

3 - O que é o escalonador e qual o seu papel em relação às threads?

O escalonador é responsável por escolher qual a próxima thread a ser executada e fazer a **troca de contexto** (context switch), é ele quem irá alternar a execução das threads.

4 - O que é a sincronia entre threads e como é implementada?

Sincronia entre threads é a forma de lidar com a região crítica do código, esta região é a parte do código que não pode ser acessada por mais de uma thread ao mesmo tempo. Para manter essa sincronia, é utilizada uma ferramenta que funciona como uma chave, ou seja, outra thread só poderá utilizar o método após a primeira thread parar de utilizá-lo, assim devolvendo a chave. A palavra chave **synchronized** dá essa característica a um bloco de código e recebe qual é o objeto que será usado como chave, exemplo:

```
public class Conta {  
    private double saldo;  
    // outros métodos e atributos...
```

```

        public void atualiza(double taxa) {
            synchronized (this) {
                double saldoAtualizado = this.saldo * (1 + taxa);
                this.saldo = saldoAtualizado;
            }
        }
        public void deposita(double valor) {
            synchronized (this) {
                double novoSaldo = this.saldo + valor;
                this.saldo = novoSaldo;
            }
        }
    }
}

```

UML

1 - Defina uma classe abstrata com três classes filhas e desenhe o diagrama de Classes apontando atributos e métodos de cada um.

2 - Faça um diagrama de Caso de Uso baseado no "sistema" que você modelou as classes, no item anterior?

Swing

1 - O que é e a para que serve a biblioteca Swing do Java?

Swing é uma API que possibilita a utilização de interfaces gráficas no Java.

2 - Explique, do ponto de vista do código, com se constrói uma Interface Gráfica (por exemplo, você citará o JFrame).

```

public class Menu {
    JFrame frame = new JFrame();
    JButton iniciar = new JButton();
}

```

3 - Como funciona o tratamento de eventos na biblioteca Swing?