**Exercícios**

Nos exercícios abaixo, onde aparecer "Java", você poderá também usar C++/STL.

1. Defina o que é feito na etapa de *Análise* e o que é feito na etapa de *Projeto* ao desenvolver software.

Na Análise é a etapa onde o desenvolvedor especifica e faz um estudo baseado dos dados colhidos nos levantamentos de requisitos com a finalidade de encontrar algo mais próximo da ideia que o cliente deseja.

No projeto é a etapa onde será pensado como o software funcionará internamente, qual tecnologia será utilizada, como será dividida a equipe e o que cada um vai fazer.

1. Enumere as vantagens da abordagem Orientada a Objetos para o desenvolvimento de software.

Devido a facilidade de maior reaproveitamento de código, componentes bem modularizados com funções bem definidas.

1. Qual é o motivo de levantar Requisitos Funcionais para desenvolver software e o que faz parte de uma descrição de Requisitos Funcionais?

Para sabermos identificar os objetivos ou restrições estabelecidas por clientes e usuários do sistema. Os requisitos Funcionais são a descrição das diversas funções que clientes e usuários querem ou precisam que o software ofereça.

1. Mostre como instanciar um objeto da classe ContaBancária em Java fornecendo o CPF (um string) do titular como argumento. Com o objeto resultante, faça um depósito de R$100,00 e imprima o saldo. Você pode escolher nomes apropriados para os métodos. FIZ
2. Explique o que é um Iterator em Java. Qual é sua principal vantagem?
3. Mostre a implementação de uma classe ContaBancária. Invente atributos e métodos.

FIZ

1. Explique a diferença de funcionamento entre um "return" e um "throw". Seja específico.
2. Mostre, usando Java, como especializar uma classe ContaBancária para criar uma ContaCorrente e uma ContaPoupança.
3. Explique as vantagens e desvantagens do polimorfismo. Dê exemplos.

Polimorfismo nada mais é do que a habilidade de criar objetos que realizam as mesmas funções de maneiras diferentes.

1. Explique a afirmação: "Em Java, o conceito de interfaces permite obter mais polimorfismo do que seria possível com classes abstratas".
2. Qual é a diferença entre "herança de tipo" e "herança de implementação"?

Na herança temos o conceito de uma Classe Pai, Classe Base ou Superclasse (todas essas nomenclaturas denotam a mesma coisa) que é a classe que foi herdada pelas subclasses. A Classe Filha ou Subclasse é a classe que herda da classe Pai.

Herança de tipo é capacidade de uma classe herdar uma interface (interface - nesse contexto - não significa necessariamente algo que faz uso da keyword interface, mas sim tudo aquilo que é tornado público pela classe") e, assim, ser referenciada por esse outro tipo.

Na Herança de Implementação a sub-classe herda a implementação da super-classe. É um mecanismo para compartilhar código e representação.

1. Quais são as vantagens e desvantagens de acoplamento forte entre objetos?

Vantagens:

Desvantagens:

O código é mais difícil de entender, pois entender cada classe requer entender outras;

O reuso é comprometido, pois usar uma classe impacta em saber também usar outras;

Mais propenso a introdução de bugs;

1. Ao falar de boa programação, fala-se: "A decomposição deve esconder algo." O que poderia ser escondido, por exemplo?
2. O que é uma "responsabilidade de uma classe"? Por que queremos minimizar o número de responsabilidades? "Mais" não seria melhor?
3. Por que modelar papeis (roles) através de herança é inferior a modelá-los através de composição?