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

A etapa de análise é onde se ver se o produto proposto trará lucro, caso contrário não é recomendado investir em algo que não traz retorno. Outro fator é a coleta do que o cliente deseja no produto final, para assim depois da coleta inicializar a produção.

Na etapa do projeto inicializa a produção do sistema, com tarefas realizadas segundo os requisitos coletados pelo cliente, sendo assim a equipe viabiliza o que foi proposto. Como também é na fase da produção do projeto onde se faz entregas ao cliente para que ele veja se é realmente o que ele está almejando.

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

As principais vantagens da programação orientada a objeto é a padronização, a flexibilidade , a abstração que oo permite e as extensões.

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

O levantamento de requisitos funcionais servem para mostrar quais serão as funcionalidades que deverão ser desenvolvidas no sistema. Ou seja os requisitos funcionais definem o que o software faz em relação às suas atividades e serviços.

4- 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.

5- Explique o que é um Iterator em Java. Qual é sua principal vantagem?

A vantagem do padrão iterator é a permissão de acessar os elementos um a um os elementos de um agregado mesmo sem saber como eles estão sendo representados, assim torna-se irrelevante se a coleção de objetos está num ArrayList, HashTable ou que quer que seja.

6- Mostre a implementação de uma classe ContaBancária. Invente atributos e métodos.

7- Explique a diferença de funcionamento entre um "return" e um "throw". Seja específico.

O return irá retornar sempre que o método for chamado, já no caso do throw ele irá tratar uma exceção caso ocorra a exceção ele irá dar um retorno ao usuário.

8- Mostre, usando Java, como especializar uma classe ContaBancária para criar uma ContaCorrente e uma ContaPoupança.

9- Explique as vantagens e desvantagens do polimorfismo. Dê exemplos.

O polimorfismo permite que diversos objetos possuam o mesmo metodo, mais que atue de forma específica a necessidade daquele determinado objeto. Por exemplo em uma conta poupança e conta corrente, será diferente a mensalidade nas mesma, visto que na poupança vai render lucro ao proprietário, já a corrente terá que pagar ao banco uma taxa mensal, portanto o comportamento do método será diferente. Uma das vantagens do polimorfismo é a clareza e facilidade na manutenção do código.

10- Explique a afirmação: "Em Java, o conceito de interfaces permite obter mais polimorfismo do que seria possível com classes abstratas".

Pois na interface é como um contrato e as classes que implementa-lá terão que obedecer aquele contrato, assim há polimorfismo nisso, visto que as classes que obtiverem a interface terá que obter os mesmo métodos, podendo tratá-los para especificar a necessidade da classe. Visto isto na classe abstrata o polimorfismo é menor

11- Qual é a diferença entre "herança de tipo" e "herança de implementação"?

Na herança de tipo será implementado tudo aquilo que a classe deixou público e a classe vai implementar uma interface específica por exemplo a classe pessoa implementa uma interface estudante e uma interface bolsista. Mostrando que aquela pessoa é do tipo estudante e possui alguma bolsa. Já na herança de implementação permite que diferentes interfaces possuem os mesmo nomes de métodos, podendo assim causar conflito em quem implementar.

12- Quais são as vantagens e desvantagens de acoplamento forte entre objetos?

O acoplamento traz a dependência entre classes, uma das vantagens é que diminui a codificação, como também a organização do código. Por exemplo uma interface, animal que possuem métodos genéricos será implementado por animais como cachorro, gato, assim quem implementou a interface não precisara codificar o que é comum a todos os animais. Assim com o acoplamento a uma diminuição na codificação. Uma das desvantagens é que se por exemplo houver algum método com cálculos específicos e ele for alterado, todas as classes que implementam a interface mudaram.

13- Ao falar de boa programação, fala-se: "A decomposição deve esconder algo." O que poderia ser escondido, por exemplo?

14- O que é uma "responsabilidade de uma classe"? Por que queremos minimizar o número de responsabilidades? "Mais" não seria melhor?

A responsabilidade de classe é o que ela terá que fazer, por exemplo uma classe conta terá que ter métodos para sacar, depositar. Então não é bom que a classe conta seja responsável por cadastrar uma conta, por é melhor delegar menos responsabilidades, visto que quanto mais responsabilidade maior probabilidade de erro na classe.

15- Por que modelar papeis (roles) através de herança é inferior a modelá-los através de composição?

Pois com a composição você poderá fazer uso do codigo de maneira que irá passar informaçÞoes para outro objeto, ou seja ele irá delegar trabalho para o outro objeto.