1) Proponha uma maneira de melhorar a modelagem das classes abaixo (explique e forneça a definição das novas classes).

- Livro(autores, título, ano, ISBN, número de páginas, editora, edição)

- Apostila(autores, título, número de páginas, link onde o material está disponível, data de obtenção de material, data da última atualização da apostila)

Minha resposta : Utilizando herança consigo reutilizar código em vez de ter vários atributos iguais para várias classes

- Material(autores, título,número de páginas)

- Livro extends Material(ano, isbn, editora, edição )

- Apostila extends Material(linkMaterialDisponivel , dataObtenção, dataUltimaAtualização)

2) Assinale V ou F:

a) ( V ) Um "motor de jogo" é um software altamente reusável

b) ( V ) Um "motor de jogo" é um software extensível

R: Os motores de jogo são estruturas de software projetadas para serem a base para o desenvolvimento de jogos. Eles fornecem uma ampla gama de funcionalidades e componentes reutilizáveis necessários para a criação de jogos, como renderização gráfica, física, detecção de colisão, entrada do usuário, rede, áudio e muito mais.

A extensibilidade **é uma característica fundamental** dos motores de jogo, permitindo que os desenvolvedores adicionem, modifiquem ou estendam suas funcionalidades de acordo com as necessidades específicas de seu jogo.

c) ( F ) Em repositórios de código público, segundo estudos da literatura científica, é mais comum haver mudanças relacionadas a refatoramento regular do que mudanças relacionadas a melhoria no aspecto de reusabilidade

d) ( F ) Em programas de código aberto existe pouca reutilização, sendo a maioria dos códigos programas inéditos

e) ( V ) Ainda que seja comum programadores obterem informações de sites como o StackOverflow, aplicações não são construídas com códigos advindos de sites como esse

f) ( V ) Repositórios de código público são bons locais para se explorar aspectos de software como legibilidade, reusabilidade e complexidade

g) ( V ) Alterações em métodos (mover, extração, etc) são frequentemente realizadas tendo por objetivo melhorar a reusabilidade dos códigos relacionados

h) ( v ) Alterações em nomes de métodos e atributos são frequentemente realizadas tendo por objetivo melhorar a reusabilidade dos códigos relacionados

3) Associe cada tipo de alteração abaixo com a respectiva descrição:

a) Movimentação de método

b) Pull-up de método

c) Extração de método

( b ) Mover um método de uma classe filha para uma classe pai

( a ) Mover um método para uma classe utilitária para que possa ser utilizada por outras classes

( c ) Mover um trecho de código para um método, e substituir o trecho de código pela chamada do método

4) Complete as lacunas do texto a seguir:

Módulos reusáveis devem possuir \_\_\_\_\_\_\_ (alta/baixa) complexidade, \_\_\_\_\_\_\_ (boa/nenhuma) documentação (comentários no código), \_\_\_\_\_\_ (muitas/poucas) dependências externas e confiabilidade \_\_\_\_\_\_\_\_\_\_ (comprovada/básica), por meio de testes e validações.

5) O que o paradigma de orientação a objetos aprimorou no aspecto de reusabilidade, em relação ao paradigma procedural, que contava com o uso de bibliotecas de funções e variáveis?

R: Acredito que a herança, polimorfismo e abstração.

herança no quesito reusabilidade acredito ser uma das principais, pois a classe pai pode ter um método que pode ser herdado/ reutilidade por todas as classes filho.

polimorfismo contribui com a reusabilidade, pois trata objetos diferentes de forma uniforme, utilizando o mesmo conceito do código para objetos diferente

6) Qual é o papel da engenharia reversa no contexto de reutilização de software?

R:A engenharia reversa desempenha um papel importante no contexto de reutilização de software, pois permite recuperar informações e entender o funcionamento de sistemas, componentes ou códigos existentes, facilitando sua adaptação e reutilização em novos projetos.

7) Explique como a reflexão é utilizada para estender a funcionalidade de linguagens por meio do uso de anotações.

R: A **reflexão é usada** para analisar e modificar a estrutura do código durante a execução do programa. Quando usamos em conjunto com uma anotação**, a reflexão consegue ler essa anotação e então modificar o comportamento** do programa de acordo com as informações obtidas.

Exemplo: se eu usar uma anotação de validação de dados quando o código é executado, a reflexão lê essa anotação que especifica as regras de validação. Então se a verificação falhar dependendo da anotação ela pode printar um erro ou lançar uma exceção.

8) Quando se diz que em uma linguagem "tudo é objeto", qualquer aspecto da linguagem (instância, classe, métodos, etc) pode ser refletido. Essa característica pode ser definida como:

a) auto-representação universal

b) auto-representação uniforme

R:A auto-representação uniforme significa que todos os elementos da linguagem (classes, métodos, variáveis, etc.) são tratados da mesma maneira, como objetos. Ou seja, há uma uniformidade no modo como esses elementos são representados internamente pela linguagem, permitindo manipulá-los e inspecioná-los de forma consistente, já que todos são vistos como objetos.

c) auto-representação difusa

d) coesão e) acoplamento

9) O que o comando abaixo faz?

Class<?> hm = Class.forName("java.util.HashMap");

10) Considere um código que está em um arquivo chamado "livro.pas"; esse arquivo está em um diretório chamado "modelo", que se encontra dentro de outro diretório chamado "sistema". Um terceiro programa deseja utilizar o código mencionado. Como deve ser o pseudocódigo de importação deste código pelo programa terceiro?

R: import sistema.modelo.livro;

11) Considere uma tabela chamada "livro" que possui os seguintes campos (todos string): título, ano, autores. Escolha uma linguagem e forneça o código de uma classe que mapeie os campos dessa tabela em atributos, e forneça um exemplo de código que crie um novo livro e insira na tabela.

12) Na linguagem Java, há um método chamado "toString"; esse método é equivalente ao método "\_\_str\_\_" no python. Qual é a utilidade desse método?

OBS: em c++ não existe esse método padrão para ser sobreescrito, mas há outras opções:

https://stackoverflow.com/questions/53211020/making-my-own-tostring-method-on-c-struct

R:o método toString() em Java é usado para converter um objeto em uma representação de string. Sendo útil para a impressão do objeto, auxilia na depuração e pode imprimir o objeto e identificar qual campo está dando o nullPointer.

13) Uma lista ligada e um vetor dinâmico são duas estruturas que possuem a mesma lista de operações (que especificam acesso e manipulação sequencial aos seus elementos), porém não compartilham trechos de implementação (não possuem ligação "genética"). Essas duas estruturas poderiam ser generalizadas e promover a criação de uma superclasse ou o uso de uma interface é mais apropriado nesse caso? Justifique.

R: Interface, concordo que as duas estruturas tenham as mesmas operações como adicionar, remover etc. Porém isso não significa que a implementação é a mesma, nesse caso a superclasse não daria certo, para duas implementações diferentes. Já tendo uma interface com a assinatura dos métodos soma remover, como as estrutura seriam implementadas na interface, abas as estruturas teriam um padrão;

14) Em Java, interfaces e classes abstratas são diferenciadas sintaticamente, por meio das palavras interface e extends, respectivamente. Em C++, a diferença entre interface e classe abstrata é apenas uma diferença semântica. Observe estes códigos como exemplos em C++:

https://gitlab.com/hvescovi/prog24/-/tree/main/oo/c\_plus\_plus/02-abstract-and-interface

Com base nos exemplos, explique qual é a utilidade da palavra "virtual" em C++.