

1. Crie uma interface chamada `Desenhavel` com um método `desenhar()` que não retorna nada. Em seguida, implemente essa interface em três classes diferentes: `Circulo`, `Retangulo` e `Triangulo`. Cada classe deve implementar o método `desenhar()` para exibir o nome da forma geométrica correspondente.
2. Crie uma interface chamada `Ordenavel` com um método `ordenar()` que recebe um array de inteiros e o ordena em ordem crescente. Em seguida, implemente essa interface em uma classe chamada `BubbleSort` que implementa o algoritmo de ordenação Bubble Sort.
3. Crie uma interface chamada `Armazenavel` com dois métodos: `salvar()` e `carregar()`. Em seguida, implemente essa interface em uma classe chamada `ArquivoCache` que salva e carrega dados de um arquivo no sistema de arquivos.
4. Crie uma interface chamada `Redimensionavel` com um método `redimensionar(double fator)` que redimensiona um objeto geométrico. Em seguida, implemente essa interface em uma classe chamada `Circulo` para redimensionar o raio do círculo pelo fator fornecido.
5. Crie uma interface chamada `Registravel` com um método `registrar(String mensagem)` que registra uma mensagem de log. Em seguida, implemente essa interface em uma classe chamada `RegistroConsole` que exibe a mensagem de log no console.
6. Crie uma interface chamada `Criptografavel` com dois métodos: `criptografar(String dados)` e `descriptografar(String dadosCriptografados)`. Em seguida, implemente essa interface em uma classe chamada `AESCriptografia` que utiliza o algoritmo de criptografia AES para criptografar e descriptografar dados.
7. Crie uma interface chamada `Pesquisavel` com um método `pesquisar(String palavraChave)` que realiza uma pesquisa em um determinado recurso (por exemplo, um banco de dados). Em seguida, implemente essa interface em uma classe chamada `PesquisaBancoDados` que realiza uma pesquisa em um banco de dados.
8. Crie uma interface chamada `Renderizavel` com um método `renderizar()` que renderiza um objeto gráfico na tela. Em seguida, implemente essa interface em uma classe chamada `RenderizadorOpenGL` que utiliza a biblioteca OpenGL para renderização gráfica.
9. Crie uma interface chamada `Validavel` com um método `validar()` que verifica se um objeto é válido. Em seguida, implemente essa interface em uma classe chamada `ValidadorEmail` que verifica se uma string é um endereço de e-mail válido.
10. Crie uma interface chamada `Executavel` com um método `executar()` que realiza uma determinada ação. Em seguida, implemente essa interface em uma classe chamada `ExecutorComandos` que executa comandos do sistema operacional fornecidos como strings.