(1) Em orientação a objetos, é melhor:

(a) Usar uma interface ou uma classe abstrata? Por que? Cite exemplos.

Resposta: Usar interface é o mais indicado. Uma classe pode assinar várias interfaces facilitando o desacoplamento da aplicação. Interface facilita a implementação de testes unitários, uma vez que torna possível a criação de objetos que imitam o comportamento de objetos reais (Mock objects).

(b) Usar herança ou delegação a outros objetos? Por que? Cite exemplos.

Resposta: Usar herança é o mais adequado. A herança melhora a manutenibilidade do código ao concentrar definições que estariam distribuídas em vários pontos diferentes do projeto em apenas um local específico.

(2) Um hipermercado muito tradicional descobriu uma fórmula mágica para calcular o preço a ser cobrado por um determinado item, fazendo com que o lucro seja maximizado.

Resposta: https://github.com/lucianolimacl/test_hiplatform/blob/master/CalculadoraPrecoProduto.cs

(3) Um candidato a prefeito quer saber quais ruas ele deve visitar para impactar o maior número de eleitores. Abaixo seguem objetos já existentes que representam casas e ruas:

Resposta: https://github.com/lucianolimacl/test_hiplatform/blob/master/RelatorioEleitoresPorRua.cs

(4) Esta questão aborda o tratamento de erros orientado a objetos.

(a) E boa prática definir um tipo específico de exceção que estende da classe Exception? Se sim, em quais casos?

Resposta: Caso a aplicação esteja trabalhando com um fluxo de negócio muito complexo onde seja esperado o estouro de exceções é aconselhável o uso de Exceptions customizadas para melhorar o tratamento desses cenários.

(b) Quando você capturaria uma exceção através de clausulas try e catch? Por que?

Resposta: Capturaria em caso de acesso a recursos externos, por exemplo banco de dados, disco de arquivos, api etc. Caso o recurso esteja indisponível a exceção ajuda na criação de fluxo alternativos no código ou auxilia na resolução de problemas posteriormente através dos logs capturados.

(c) Em quais situações você lançaria uma exceção? Cite exemplo

Resposta: Lançaria exceções apenas em cenários que não fossem possíveis tratar no fluxo do código. Por exemplo em caso de acesso à sistema de arquivos que esteja indisponível no momento.

(5) Considere um web service responsável por crédito e débito em uma conta corrente, que implementa os seguintes métodos:

Analise a solução, considerando concorrência entre chamadas e escopo de transações.

Resposta: As operações Debitar e Creditar precisam estar dentro de uma transação para manter a integridade dos dados. Reescrito fica dessa forma:
https://github.com/lucianolimacl/test_hiplatform/blob/master/GerenciadorConta.cs

```
public void Debitar(long idConta, double valor)
{
    using (TransactionScope tran = new TransactionScope())
    {
        Conta conta = contaDao.BuscaConta(idConta);
        if (conta.PodeDebitar(valor))
        {
            conta.Debite(valor);
            contaDao.Atualiza(conta);
            tran.Complete();
        }
        else
        {
            tran.Dispose();
            throw new SaldoInsuficienteException();
        }
    }
}
```

```
0 references
public void creditar(long idConta, double valor)
{
    using (TransactionScope tran = new TransactionScope())
    {
        Conta conta = contaDao.BuscaConta(idConta);
        conta.Credite(valor);
        contaDao.Atualiza(conta);
        tran.Complete();
    }
}
```

(6) Uma rede de supermercados que vende alimentos e produtos de limpeza teve acesso a pesquisas de mercado, feitas por diversos institutos de pesquisa, a pedido dos fabricantes. Os dados foram modelados em um banco de dados, conforme mostra a figura 1...

Faça uma consulta SQL que devolve linhas que representam possíveis kits ordenados pelo lucro, isto é, preço do item menos o custo.

Resposta: https://github.com/lucianolimacl/test_hiplatform/blob/master/ConsultaKitsProdutos.sql