Teoria:

1. O que é uma **Classe**?

R: Classes: Estrutura de dados que combina ações, fornece definições para as instancias e dão suporte a herança e polimorfismo

1. O que são métodos de encapsulamento? Explique para que serve o método **set** e o método **get.**

R: São métodos públicos que permitem acessar indiretamente os atributos de uma classe.   
Set – Esses métodos são usados para modificar o valor de um atributo privado de uma classe.   
Get – Esses métodos são usados para recuperar o valor de um atributo de uma classe

1. Para que serve a palavra reservada **new**?  
   R: Serve para criar uma instancia ou objeto de uma classe
2. Explique o que é **desenvolvimento em camadas**.

R: é uma abordagem de design de software que divide um sistema em diferentes componentes ou camadas, cada uma com responsabilidades específicas

1. O que são **objetos / variáveis de instância**?  
   R: São os atributos de uma classe
2. Para que serve a palavra reservada **void**?  
   R: Para indicar que o método/função não retorna valor.
3. Explique o que fazem as sentenças:
   * **Console.WriteLine();** R: Imprime uma informação no console.
   * **Console.ReadLine();** R:Serve para capturar as informações enviadas ao sistema.
4. Para que serve a classe **StreamWriter**?  
   R: É uma Classe que serve para escrever a saída das informações em arquivos de texto/xml
5. Explique o que são classes de **entidade** e qual o seu objetivo.  
   R: Servem para criar uma abstração do mundo real.
6. Explique o que são classes de **repositório** de dados e qual o seu objetivo.  
   R: São responsáveis por lidar com a persistência e recuperação de objetos de domínio (ou entidades) em um sistema de software.
7. Explique o que são classes de **controle** de dados e qual o seu objetivo.  
   R: O objetivo principal dessas classes é fornecer uma camada intermediária entre a camada de apresentação (interface do usuário) e a camada de acesso a dados (classes de repositório, por exemplo), gerenciando a lógica de negócios associada ao processamento de dados.
8. O que é **encapsulamento** em C#?

a) É um recurso que permite que classes sejam estendidas para adicionar mais funcionalidades.

b) É um mecanismo que permite que dados sensíveis sejam ocultados do acesso externo e protegidos por meio de métodos getters e setters.

c) É um recurso que permite que diferentes partes do sistema sejam desenvolvidas e testadas separadamente.

1. Qual a finalidade dos métodos **getters** e **setters** em C#?
2. Os getters e setters permitem que os dados sensíveis sejam ocultados do acesso externo e protegidos pelo encapsulamento.
3. Os getters e setters permitem que as classes sejam estendidas para adicionar mais funcionalidades.

c) Os getters e setters permitem que diferentes partes do sistema sejam desenvolvidas e testadas separadamente.

1. O que é **herança** em C#?
2. É um mecanismo que permite que dados sensíveis sejam ocultados do acesso externo e protegidos por meio de métodos getters e setters.
3. É um mecanismo que permite que uma classe filha herde propriedades e comportamentos de uma classe pai.

c) É um recurso que permite que diferentes partes do sistema sejam desenvolvidas e testadas separadamente.

1. Qual a finalidade da **camada de repositório** de dados em uma

arquitetura em camadas?

1. É responsável pela apresentação dos dados na interface do usuário.
2. É responsável pelo processamento dos dados de negócio da aplicação.
3. É responsável pelo acesso e manipulação dos dados armazenados no banco de dados.

Prática:

**Tarefa 1**

Crie um projeto do C#:

* Desenvolva as classes de entidades abaixo utilizando os relacionamentos de composição:

Gráfico, Gráfico de caixa estreita

Descrição gerada automaticamente

* Crie camadas no projeto de controle, repositórios e principal para executar um fluxo de cadastro de produto no sistema.
* A camada de controle deverá capturar os dados preenchidos pelo usuário e armazená-los com o auxílio das entidades
* A camada de repositório deverá gravar os dados do produto em um arquivo de extensão TXT

Diagrama

Descrição gerada automaticamente

* Ao final, o TXT deverá ser gerado com o conteúdo conforme abaixo:

**DADOS DO PRODUTO:**

**ID........: 8ce082a3-8209-44cb-9631-95fe9636cafb**

**NOME......: Notebook Dell**

**DESCRICAO.: Notebook Core 17 SSD**

**PRECO.....: 6000.0**

**QUANTIDADE: 10**

**CODIGO....: INFO001**

**DESCRICAO.: Produtos de Informática**

**Tarefa 2:**

Crie um projeto C#.

* Faça um programa que leia dados fornecidos pelo usuário e ao final exporte o seguinte arquivo XML deverá ser gerado pelo sistema:

**<?xml version="1.0" encoding="UTF-8" ?>**

**<Turma>**

**<IdTurma>ceb9bb56-b575-4cd6-ac01-f81202e2ad26</IdTurma>**

**<Nome>Treinamento em C# WebDeveloper</Nome>**

**<DataInicio>18/06/2022</DataInicio>**

**<Professor>**

**<IdProfessor>d7a60dee-f21d-4db1-b648-715c12d34767</IdProfessor>**

**<Nome>Sergio da Silva Mendes</Nome>**

**<Telefone>(21) 96957 5900</Telefone>**

**<Email>sergio.coti@gmail.com</Email>**

**</Professor>**

**<Curso>**

**<IdCurso>90fa8469-0853-4771-810a-611ebb285d91</IdCurso>**

**<Titulo>C# WebDeveloper</Titulo>**

**<CargaHoraria>96</CargaHoraria>**

**</Curso>**

**</Turma>**

**Tarefa 3:**

Um hospital deseja informatizar o processo de gerenciamento de exames médicos. Nesse sistema, é necessário registrar informações sobre médicos, pacientes e os exames realizados. Cada exame é associado a um único médico e a um único paciente. O sistema deve permitir a exportação desses dados para um arquivo XML para fins de armazenamento e compartilhamento.

Entidades e Atributos:

**Médico:**

Atributos:

Nome: String - o nome do médico.

Especialidade: String - a especialidade médica do profissional.

CRM: String - o número do Conselho Regional de Medicina do médico.

ID: int - identificador único do médico.

**Paciente:**

Atributos:

Nome: String - o nome do paciente.

Idade: int - a idade do paciente.

Sexo: char - o sexo do paciente (M - Masculino, F - Feminino).

Endereço: String - o endereço do paciente.

ID: int - identificador único do paciente.

**Exame:**

Atributos:

Tipo: String - o tipo de exame realizado.

Data: Date - a data em que o exame foi realizado.

Resultado: String - o resultado do exame.

Médico: Médico - o médico responsável pelo exame.

Paciente: Paciente - o paciente que realizou o exame.

ID: int - identificador único do exame.

O sistema deve ser capaz de criar instâncias dessas entidades, associar corretamente os médicos, pacientes e exames, e exportar esses dados para um arquivo XML para posterior análise e armazenamento.