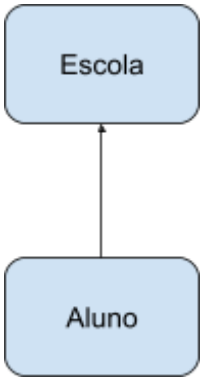
1. Exercícios de C# (Herança e Polimorfismo)

1. Elabora um programa **FazerCiculos** que instancie a subclasse Círculo do Exercício exemplo dos slides, formando um vetor de círculos. O centro de cada círculo desloca-se uma unidade para cima e para a direita relativamente ao centro do círculo precedente; o raio do círculo duplica o raio do seu precedente. O primeiro círculo tem centro e, (0,0) e um raio de 0,5 cm. O programa deve imprimir as coordenadas do centro e a área de cada círculo.

2. Define a classe **Escola** e a subclasse **Aluno** que deriva de **Escola**:



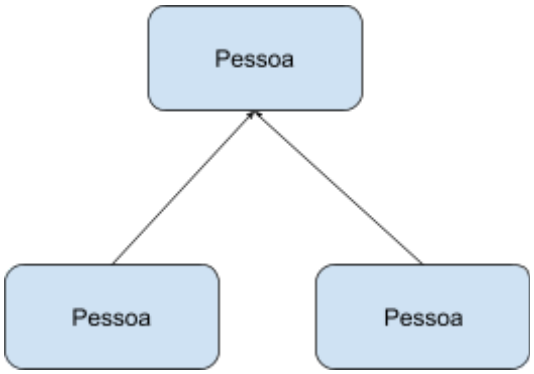
A classe **Escola** encapsula:

- Duas variáveis de instância, Nome e Morada;
- Um construtor e um destruidor;
- Propriedades para ler os valores das duas variáveis de instância.

A subclasse **Aluno** encapsula:
Duas variáveis de instância, Nome e Nota;
Um construtor e um destruidor;
Um método que se sobrepõe a ToString() para devolver os dados dos alunos.

Elabora um programa que instancie cinco alunos, imprima as suas classificações e, em seguida, destrua os respectivos objetos.

3. Defina a classe **Pessoa** e as subclasses **Amigo** e **Colega** que derivam de **Pessoa**:



A classe Pessoa encapsula:

- Duas variáveis de instância, Nome e Tel;
- Construtores;
- Assessores para ler os valores das variáveis de instância.

A subclasse Amigo encapsula:

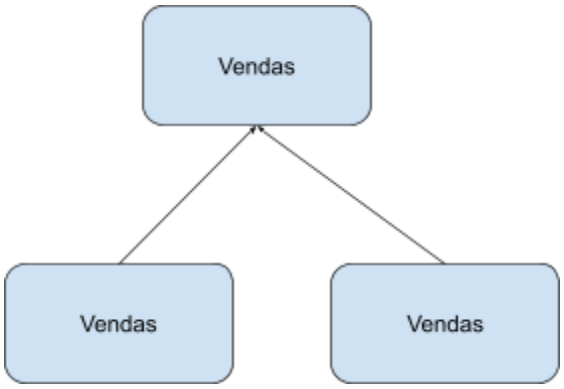
- Duas variaveis de instancia, Local e Ano de conhecimento
- Um construtor;
- Assessores para ler os valores das variáveis de instância.

A subclasse Colega encapsula:

- Duas variáveis de instância, local de trabalho e profissão;
- Um construtor;
- Assessores para ler os valores das variáveis de instância.

4. Elabora um programa que instancie as subclasses **Amigos** e **Colegas** do exercício 3, e imprima uma lista com os nomes dos amigos e os respectivos locais de conhecimento, e outra lista com os nomes e as profissões dos colegas.

5. Defina a classe **Vendas**, de que derivam as subclasses **VendasProdA** e **VendasProB**.



A classe **Vendas** encapsula:

- Uma variável estática que acumula o valor das vendas dos produtos;
- Um método estático para atualizar o valor total das vendas;
- Um método estático para devolver o valor total das vendas.

A subclasse **VendasProdA** encapsula:

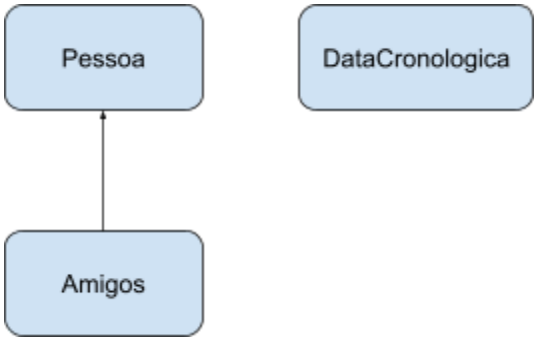
- Duas variáveis de instância **Quant** e **Pu**, que guardam, repetidamente, a quantidade e o preço unitário de cada transação do produto A;
- Um construtor que utiliza o valor total das vendas.

A subclasse **VendasProdB** deve ter:

- Uma variável de instância, **ValorTrans,** que guarda o valor de cada transação do produto B;
- Um construtor que atualiza o valor total das vendas.

Elabore um programa de teste que instancie as subclasses de **VendasProdA e VendasProdB**, registe diversas transações dos produtos A e B, e imprima o valor total dessas transações.

6. Defina a seguinte hierarquia de classes:



A subclasse **Amigos** tem uma variável de instância, Data de Nascimento, que é um objeto de **DataCronologica**.

A classe **Pessoa** encapsula:

- A variável de instância Nome;
- Um construtor;
- Um acessor para ler e atribuir valores ao atributo.

A classe **Amigos** encapsula:

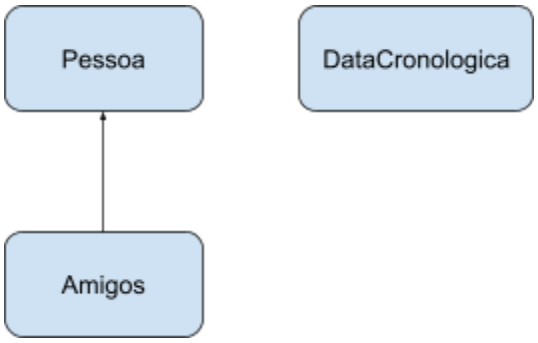
- A variável de instância DataNasc;
- Um construtor;
- Um acessor para ler e atribuir valores ao atributo.

A classe **DataCronologica** encapsula:

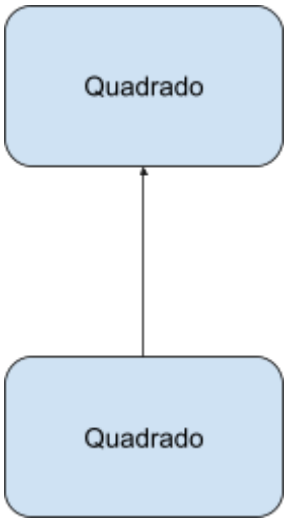
- Três variáveis de instância, **Dia**, **Mês** e **Ano**;
- Um construtor;
- Um acessor para ler e atribuir valores ao atributo.

7. Defina para a classe **DataCronologia**, dois métodos que permitam calcular a idade de uma pessoa, respetivamente, em anos e em dias. O método para calcular a idade em anos recebe o ano corrente e o método para calcular a idade em dias recebe a data de hoje.

8. Calcule a idade de um seu amigo, em anos e dias, usando as classes **Pessoa**, **Amigos** e **DataCronologica**:



9. Defina a classe **Quadrado** e a subclasse **Rectangulo** que deriva de **Quadrado**.



A classe **Quadrado** encapsula:

- A variável de instância **Comprimento**, que guarda a medida do lado do quadrado.
- Um construtor;
- Um acessor que devolve o comprimento do lado do quadrado;
- Um método **Area** que calcule a área do quadrado.

A subclasse **Rectangulo** encapsula:

- A variável de instância **Largura**, que guarda a medida da largura do retângulo;
- Um construtor;
- Um método **Dimensoes** que devolve as dimensões do retângulo;
- Um método **Area** que calcula a área do retângulo.

10. A classe **Geral** define como regime geral de avaliação a realização de um teste, sendo a classificação final de um aluno o arredondamento da nota desse teste a zero casas decimais. A subclasse **Informatica**, que deriva de **Geral**, estabelece um regime de avaliação específico, que determina que a classificação final de um aluno é a média aritmética da nota de dois testes, também arredondada a zero casas decimais: