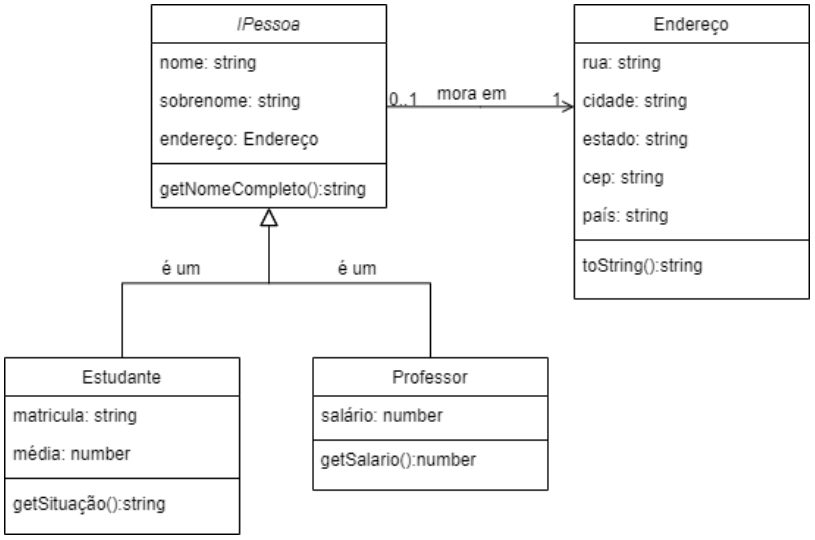
AP2 - PD - DD - 2022.1

1 - Implemente o diagrama representado pela imagem abaixo:



Leve em consideração que:

- IPessoa é uma interface;
- Estudante e Professor implementam IPessoa;
- o método getNomeCompleto deve concatenar nome e sobrenome;
- o método getSituação irá retornar APROVADO (caso média seja maior ou igual a 5.0) ou REPROVADO (caso contrário).
- O método toString deve retornar TODAS as propriedades de Endereço concatenadas em uma string.
- O método getSalario deve retornar apenas o salário do professor.

2 - Explique como é implementado o Polimorfismo em TypeScript e cite um exemplo de implementação. No seu exemplo, apresente o código das classes e interfaces que torna possível o Polimorfismo. Explique também qual a importância do Polimorfismo em linguagens orientadas a objetos.

3 (UNICAMP) - Sejam as seguintes classes:

```typescript
class DespesaMes {
    mes: number //mês da despesa
    valor: number //valor da despesa

    constructor(mes: number, valor: number) {
        this.mes = mes;
        this.valor = valor;
    }

    getMes(): number {
        return this.mes;
    }
    getValor(): number {
        return this.valor
    }
}
```

```typescript
class DespesaDia extends DespesaMes {
    dia: number // dia da despesa

    constructor(dia: number, mes: number, valor: number) {
        super(mes, valor);
        this.dia = dia;
    }

    getDia(): number {
        return this.dia
    }
}
```

- DespesaMes representa a despesa de um mês específico
- DespesaDia representa a despesa de um dia específico

Escreva uma classe que representa todas as despesas de um indivíduo. Ela mantém um vetor onde podem ser registradas tanto despesas de um dia (DespesaDia). A classe implementa os seguintes métodos:

Construtor	Recebe como parâmetro o CPF e um vetor com as despesas de um indivíduo e as guarda.
getCPF	Retorna o CPF do indivíduo.
totalizaMes	Recebe um parâmetro com um mês (number) e retorna um objeto da classe DespesaMes onde estará registrada a soma de todas as despesas que o indivíduo fez naquele mês.

4 - Seja seguinte interface:

```
1  interface IUsuario {
2      id: number
3      nome: string
4      amizades: IUsuario[]
5
6      isAmigo(usuario: IUsuario):boolean
7  }
```

Implemente uma classe concreta que herda de IUsuário (escolha um nome) e sobrescreva o método isAmigo. O método isAmigo deverá retornar TRUE se o usuário de entrada já é amigo (ou seja, está no vetor de amizades) e FALSE caso contrário.

5 (UFF-Leonardo Murta) - Identifique as classes e implemente apenas os corpos das classes/interfaces (não há necessidade de lógica):

“O supermercado vende diferentes tipos de produtos. Cada produto tem um preço e uma quantidade em estoque. Um pedido de um cliente é composto de itens, onde cada item especifica o produto que o cliente deseja e a respectiva quantidade. Esse pedido pode ser pago em dinheiro, cheque ou cartão.”

Obs.: Todas as questões valem dois pontos. A cada erro, não importa qual, será descontado 0.5 pontos da questão.