

1. Um software para cadastrar os alunos da ETS foi desenvolvido em python, porém feito de maneira procedural, o que deixou o código da seguinte forma:

```
nome_aluno = input("Digite o nome do aluno: ")

idade_aluno = input("Digite a idade do aluno: ")

turma_aluno = input("Digite a turma do aluno: ")

planta_aluno = input("Digite a planta do aluno: ")

aluno = {"Nome": nome_aluno, "Idade": idade_aluno, "Turma": turma_aluno,
        "Planta ": planta_aluno}

aluno_str = str(aluno)
with open("alunos.txt", "w", encoding="utf-8") as arquivo:
    arquivo.write(aluno_str)
```

Analise o código a seguir e aponte os problemas encontrados nesse código. Após a análise, refaça o código utilizando a Programação Orientada a Objetos de forma que ele fique funcional.

2. Analise o código a seguir:

```
class Livro():
    def __init__(self, titulo, autor, qtd_paginas, guardado=True):
        self.titulo = titulo
        self.autor = autor
        self.qtd_paginas = qtd_paginas
        self.guardado = guardado
```

Pensando nessa classe crie os seguintes métodos para o sistema funcionar:

- pegar\_livro: muda o atributo guardado para **"False"**, mas só pode funcionar caso o livro esteja com o guardado (guardado == True);
- devolver\_livro: muda o campo do guardado para **"True"**, mas só pode funcionar caso o livro esteja com o guardado (guardado == False);
- virar\_uma\_pagina: avança **uma** página do livro caso o atributo guardado seja **"False"** e mostra em qual página do livro está (sempre começando em 1), e não podendo ultrapassar o limite de quantidade de páginas. Tem como output o número da página do livro;
- virar\_n\_paginas: avança **uma quantidade n de** páginas do livro caso o atributo guardado seja igual a **"False"** e mostra em qual página do livro está (sempre começando em 1), e não podendo ultrapassar o limite de quantidade de páginas. Tem como output o número da página do livro;
- voltar\_uma\_pagina: volta **uma** página do livro caso ele esteja com o guardado em False e mostra em qual página do livro está (sempre começando em 1), e não podendo ultrapassar o limite de quantidade de páginas. Tem como output o número da página do livro;
- voltar\_n\_paginas: volta **uma quantidade n de** páginas do livro caso ele esteja com o guardado em False e mostra em qual página do livro está (sempre começando em 1), e não podendo ultrapassar o limite de quantidade de páginas. Tem como output o número da página do livro;
- ler\_livro: mostra um output do nome e do autor do livro.

3. Criar um software para criar um sistema bancário onde o objeto cliente deve ter os seguintes campos (Getter / Setter / Método Privado / Método Estático):

- Nome\_cliente
- Numero\_agencia
- Numero Conta
- Saldo\_cliente
- Tipo\_Conta (Corrente, Poupança)

Deve se criar todos os atributos e métodos como sendo privados, sendo somente acessados por getters e setters, além de ter um método estático para mostrar o número de contas que esse banco tem, aumentando conforme contas são criadas.

4. Construir um software utilizando Herança e Polimorfismo. O software deve conter os seguintes requisitos:

Empresa (Bosch):

Atributos:

- Nome
- CNPJ
- Nacionalidade
- Nome do Responsável (Presidente)
- Países de Atuação (lista)
- Produtos (chaves: nome do produto, preço do produto, descrição)

Métodos:

- Retornar Nome do Responsável
- Retornar os Países de Atuação
- Pesquisar por um País de Atuação
- Adicionar produto
- Retornar Relatório de Produtos
- Pesquisar por um Produto Específico

Área (PS, PT, GS, AA, PM, ED etc.):

Atributos:

- Nome
- Sigla do Setor
- Função (PT faz ferramentas, AA venda de peças...)
- Nome do Responsável (Diretor)
- Lucro
- Data de Início de Auditoria
- Data de Fim de Auditoria

Métodos:

- Retornar Nome do Responsável
- Retornar Relatório Financeiro

- Retornar Período de Auditoria (saber Data de Início e Fim de Auditoria E se está sendo auditada, ALÉM do responsável pela área)

Departamento (ETS, QMM, BISB, FCM, PUR, IDI1, MSE, CVO, OSD, MFI):

Atributos:

- Nome
- Sigla do Setor
- Função (QMM cuida da qualidade...)
- Nome do Responsável (Gestor)
- Quantidade de Funcionários

Métodos:

- Retornar Nome do Responsável
- Contratar Funcionário
- Realizar Treinamento

Nesse software Departamento deve herdar de Área, e Área deve herdar de Empresa!

Todos os objetos devem possuir um método de vistoria onde deve mostrar na tela a empresa/Setor/Subsetor que está tendo a vistoria

5. Explique brevemente o conceito de Classe – podem ser feitas analogias com itens do seu dia a dia para ajudar