

OBJETIVO:

O objetivo deste trabalho é abordar a prática da programação orientada a objetos envolvendo os conceitos de classe, objetos, associações, encapsulamento, herança e polimorfismo.

Deve-se, portanto, criar um programa orientado a objetos, na linguagem Python, que apresente uma solução para o problema especificado a seguir.

DESCRIÇÃO DO PROBLEMA:

Em uma clínica trabalham médicos e existem pacientes internados.

Cada médico é identificado pelo seu CRM, possui um CPF, um RG, um nome, um telefone e recebe um salário na clínica.

Um médico tem formação em diversas especialidades (ortopedia, traumatologia, infectologia, etc), mas só exerce uma delas na clínica.

Para todo paciente internado na clínica são cadastrados os seguintes dados pessoais: nome, RG, CPF, endereço, telefone e data de nascimento.

Um paciente tem sempre um determinado médico como responsável.

Pacientes estão sempre internados em quartos individuais, que são identificados por um número e pelo andar da clínica em que estão localizados.

Diariamente, um médico passa nos quartos para visitar os pacientes, e registra as observações realizadas sobre o estado do paciente.

Essas observações são cadastradas no sistema para futuras consultas ao histórico médico do paciente.

Esse histórico deve registrar a data, o horário, a observação realizada e o nome do médico que a registrou.

TAREFA:

Desenvolver um programa orientado a objetos que atenda as especificações descritas acima.

- O programa deve especificar as classes, atributos e métodos necessários para a solução do problema.
- O programa deve ter, no mínimo, quatro classes.
- O programa deve aplicar os conceitos de encapsulamento, herança e polimorfismo.
- Além das classes, deve ser desenvolvido um programa principal, que instancia os objetos necessários e demonstra o correto funcionamento do sistema (por exemplo: vincular médico responsável ao paciente, internar paciente em um quarto, inserir anotações do médico no histórico do paciente, etc.)

ATENÇÃO:

- Insira no início do programa um comentário com os nomes dos alunos do grupo.
- Trabalhos que não utilizam o paradigma de programação orientada a objetos receberão nota zero.
- Arquivos com erros de sintaxe (ou seja, que não executam) receberão nota zero.
- Não serão aceitos programas que não sejam arquivos Python (doc, pdf, txt, links para git, etc.).
- O código fonte e demais arquivos devem ser anexados no Classroom (pode ser enviado por apenas um aluno do grupo, mas não esqueça de identificar o nome de todos os alunos no código fonte).
- Caso seja identificada cópia entre trabalhos, a nota dos grupos envolvidos será zerada (todos os trabalhos passarão por ferramentas de detecção de plágio).
- Não serão aceitos trabalhos entregues em atraso.

CRITÉRIOS DE AVALIAÇÃO:

- 3.0 pontos: Identificar, definir e implementar as classes, atributos e métodos do sistema.
- 3.0 pontos: Aplicar de forma adequada os conceitos de encapsulamento, herança e polimorfismo.
- 3.0 pontos: Criar o programa principal que cria os objetos e realiza as operações necessárias para testar as classes.
- 1.0 ponto: Organizar o código de acordo com o padrão PEP8.