



Trabalho em Grupo sobre Python

Sistema Para Clínicas Médicas

Contexto

Durante a disciplina de programação orientada a dados, vocês estão aprendendo a programar na linguagem de programação em **Python**, que é multi-paradigma. Neste trabalho, o objetivo é avaliá-lo relativo a parte de Programação Orientada a Objetos (POO), Módulos e Manipulação de Arquivos. O trabalho também será uma forma de exercitar e aplicar os conceitos de forma prática.

O Hospital São Lucas da PUCRS (HSL) decidiu implementar um novo sistema de teste para controlar pacientes em pós operatório. Eles precisam de uma solução que forneça um mapeamento dos funcionários disponíveis para atendimento (médicos (as) e enfermeiros (as)), fazer o controle da entrada e saída de pacientes e gerenciar receituários médicos. Aproveitando que este semestre está ocorrendo a disciplina de Programação Orientada a Dados, convidaram o professor para criar uma lista de requisitos para um protótipo do novo sistema de gerenciamento da divisão de pós operatório do HSL, com o objetivo de avaliar o seu conhecimento e o quão apto você está para atuar no desenvolvimento deste projeto.

Restrições

- Está permitido o uso de todos os métodos da classe **str** (string) e estrutura de dados composta (listas, dicionários, tuplas, conjuntos e arranjos).
- Apenas os módulos **abc** e **sys** podem ser importados para implementar o sistema ou a aplicação bancária.
- Outros métodos como **len()** que são nativos e não dependem de importação de módulos estão autorizados ([lista de métodos](#)). **Métodos que não foram apresentados em aula, devem ser justificados e explicados através de comentário no código.**

Requisitos do Código

O sistema da divisão de pós operatório deve ser devolvido para que possa ser usado como um pacote do Python. Além disso, deve ser desenvolvido uma aplicação em Python que utilize este pacote para realizar uma sequência de operações que estão pré-definidas em um arquivo.

Especificações e requisitos do sistema de clínica de pós operatório:

- 1) Todo o sistema proposto ao HSL deve ser implementado como um pacote do Python chamado **Clinica**, composto de um módulo chamado **clinicaSys**.
- 2) Esse módulo **clinicaSys** precisa implementar pelo menos cinco classes principais: **Secretaria**, **Medico**, **Enfermeiro**, **Paciente** e **Pessoa**.

3) **Classe Pessoa:**

- a) A classe **Pessoa** deve ser uma **classe abstrata** que possua métodos abstratos para **cadastrarDados** e **obterDados**.
- b) Os atributos da classe **Pessoa** são:
 - i) Nome Completo
 - ii) CPF
 - iii) Data de Nascimento
 - iv) Estado Civil

4) **Classe Medico:**

- a) A classe **Medico** deve ser uma subclasse de **Pessoa** que implementa os métodos para **cadastrarDados** e **obterDados**. Tais métodos, são responsáveis por manipular dados no arquivo **funcionarios.dat**, que se comporta como uma base de dados.
- b) Esta classe deve possuir o atributo **CRM**.
- c) Este implementa também métodos próprios para **internar**, **liberar** e **diagnosticar** pacientes.
 - i) O método **diagnosticar** deve armazenar a condição do paciente no arquivo **pacientes.dat**.
- d) Os métodos **internar**, **liberar** e **diagnosticar** deverão manipular os dados no arquivo **pacientes.dat**, que se comporta como uma base de dados.
 - i) Toda vez que um paciente é internado, ele terá decréscimo de 500 nos créditos do seu convênio.
 - ii) Toda vez que um paciente é diagnosticado, ele terá decréscimo de 150 nos créditos do seu convênio.
 - iii) Se o paciente tiver crédito negativo, o mesmo não será atendido.
- e) Você deve garantir que um mesmo médico (mesmo CRM) não possa ser cadastrado duas vezes.

5) **Classe Enfermeira:**

- a) A classe **Enfermeira** deve ser uma subclasse de **Pessoa** que implementa os métodos para cadastrar e obter dados.
- b) Esta classe deve possuir o atributo **COREN**.
- c) Esses métodos de cadastrar e obter dados deverão ser os responsáveis por manipular os dados no arquivo **funcionarios.dat**, que se comporta como na uma base de dados.
- d) Esta deve possuir um método próprio de **cadastrarPacientes**.
- e) Você deve garantir que uma mesma **Enfermeira** (mesmo COREN) não possa ser cadastrada duas vezes.
- f) Essa classe deve possuir um método **gerarRelatorio**, o qual recebe como argumento uma instância da classe médico e outra instância da classe paciente. Este método deve criar um novo arquivo com nome **rel_<cpf_paciente>.txt** e escrever dentro do mesmo um simples relatório. Um exemplo deste arquivo vocês encontrarão no template do trabalho.

6) **Classe Secretaria:**

- a) A classe **Secretaria** deve ser uma subclasse de **Pessoa** que implementa os métodos para **cadastrarDados** e **obterDados**.
- b) Esses métodos de cadastrar e obter dados deverão ser os responsáveis por manipular os dados no arquivo **funcionarios.dat**, que se comporta como na uma base de dados.
- c) Nesta classe, deve ter um método **cadastrarFuncionario**, recebendo como argumento uma instância da classe **Secretária**, **Médico** ou **Enfermeira**. Isso porque a secretária da clínica é responsável pelo cadastro de funcionários, sendo ela um funcionário.

7) **Classe Paciente**

- a) A classe paciente deve ser uma subclasse de **Pessoa**.



- b) Esta classe deve possuir o atributo **Convênio**. Note que este atributo é uma instância da classe **Convênio**.
- 8) **Classe Convenio**
- a) Representar o tipo de convênio do paciente. Esta deve possuir os atributos tipo (SUS, IPE, Unimed) e o crédito que este disponibiliza para o paciente. Isto significa que o valor calculado na hora da internação do paciente precisa ser deste tipo de moeda.
 - b) As operações matemáticas necessárias precisam ser suportadas. Obs: para convênios do tipo SUS deve ser atribuído crédito zero e as consultas devem ser gratuitas.
- 9) Pelos menos uma das classes criadas (**Paciente**, **Secretaria**, **Medico** ou **Enfermeira**) deve suportar o protocolo de gerenciamento de contexto e depois você deve usá-lo na aplicação, para demonstrar o seu conhecimento no assunto. Dica do Professor: o mais simple é implementar na classe Secretaria.
- 10) É necessário também tratar erros com tipos de dados inválidos e incoerentes, como CPF, COREN, CRM, estado civil (solteiro/casado) e data de nascimento (dia/mes/ano).

Especificações e requisitos da aplicação de teste e execução das operações diárias:

- 1) A aplicação deve receber como argumento na linha de comando uma lista de arquivos texto (**1.txt, 2.txt, 3.txt, ...**), onde cada arquivo representa um dia de funcionamento da clínica, contendo uma sequência de ações e operações que devem ser feitas pela aplicação.
- 2) A aplicação não pode parar quando se deparar com operações não permitidas ou inválidas. Os erros devem ser armazenados em um arquivo de log. Por exemplo, se um cadastro foi impossibilitado de ser realizado, a aplicação deve progredir para a próxima operação ou passar para o próximo dia. Este arquivo de log serve para monitorar o que foi possível realizar ou não. Preste atenção na semântica para saber se é possível ou não executar a próxima operação, a depender do erro ocorrido e o que o mesmo pode desencadear subsequentemente.

Entrega

Vocês devem enviar ao final deste trabalho através de um pull request no repositório do GitHub:

- **Clinica**: este diretório do pacote com o módulo desenvolvido (**clinicaSys**).
- **database**: este diretório que contém os arquivos de registro do sistema.
- **operacoes**: diretório contendo os arquivos de operações.
- **logs**: diretório onde devem ser armazenados os arquivos de log (**1.log, 2.log, 3.log, ...**)
- **saida-esperada (fornecido pelo professor)**:
 - **database**
 - Exemplos dos bancos de dados funcionarios e pacientes: **funcionarios.dat, pacientes.dat**
 - **relatorios**
 - Exemplos de relatórios gerados para cada paciente.
- **app.py**: aplicação onde são realizadas as operações.
- **README.md**: arquivo descrevendo como usar a aplicação e o pacote.
- **RESUMO.md**: arquivo contendo o resumo de no máximo 400 palavras relatando as dificuldades, desafios ou outras observações que achar relevante comentar sobre a realização deste trabalho. Você também pode comentar sobre sua evolução no conhecimento. Esse resumo deve ser enviado junto dos demais arquivos.