

Programação Orientada a Objetos

Trabalho Prático 3

Data da entrega: 03/12/2017

Valor: 25 pts

1. Orientações Gerais

O trabalho pode ser realizado em grupos de dois alunos ou individualmente e deve ser entregue por e-mail até às 23:59 da data definida para entrega através do Moodle. É permitida a discussão de estratégias com os colegas, porém a implementação somente deve ser feita pelos integrantes do grupo. Cópias de programas (ainda que parciais) não serão aceitas. Entregas com atraso poderão ser aceitas pelo professor, porém terão descontos progressivos. Deve ser postado apenas um arquivo compactado contendo as soluções e documentação de todos os exercícios.

2. Objetivo

Desenvolva e implemente uma hierarquia de classes com comportamento polimórfico, para atender a construção de um sistema para a clínica *Smile Odontologia S/A*. Nesta hierarquia haverá uma classe base abstrata de nome **pessoa**, que modela uma pessoa física dentro do sistema. A partir dessa classe são derivadas classes para modelar os seguintes perfis e pessoas:

- Paciente
- Funcionário

Deverão ainda ser modeladas, a partir das classes **paciente** e **funcionário**, classes ligadas aos tipos de paciente (paciente de ortodontia, pediatria e clínica geral) e também tipos de funcionários (secretária, auxiliar, clínico geral, ortodontista, pediatra e ortodontista). Além dessas classes deve ser modelada uma classe **orçamento**, associada a um paciente, que deve conter uma lista de **procedimentos**, que, por sua vez, devem estar, cada um deles, associados a um **dentista**. Cada procedimento deve ter um atributo do tipo data que armazena a data de quando o mesmo foi realizado. Nesse momento utilize a classe data elaborada no TP2. A classe orçamento deve conter a data de pagamento (utilize também a classe elaborada no TP2), caso já tenha sido pago, e expor os seguintes métodos:

- `getValorTotal()`
- `getProcedimentos()`

Defina as classes derivadas, criando os atributos necessários, construtores para elas (projete estes construtores da melhor forma possível), seus destrutores e implementando os métodos especificados como puramente virtuais na classe **pessoa**.

Crie uma classe clinica que contenha atributos de uma classe "container" para armazenar ponteiros para pessoa (você pode usar um `vector<pessoa*>` como atributo desta classe), que serão utilizados para manter tanto a lista de funcionários como de pacientes da clínica. Pessoas serão adicionadas dinamicamente. Nesta classe, crie a possibilidade de fazer com que todas as pessoas do tipo paciente sejam notificadas de uma promoção. Não se esqueça de criar um destrutor para esta classe container, que desaloque as formas que estão sendo apontadas pelas pessoas*.

Exceções que possam colocar em risco a execução do programa devem ser tratadas caso seja necessário.

Crie um programa principal que teste o funcionamento de suas classes.

Bom trabalho!