



Objetivo

Desenvolver os conceitos estudados na disciplina em uma aplicação do mundo real.

Organização

O trabalho será desenvolvido em grupos de 3 pessoas

A primeira etapa (primeira unidade) representa 30% do trabalho, mas a nota será entregue apenas após a apresentação da segunda etapa.

Disposição

Primeira etapa:

- Desenvolvimento do modelo Orientado a Objetos para a aplicação em linguagem UML;
- Desenvolvimento em Java das classes necessárias para resolver o problema proposto (estrutura básica das classes – nome, atributos, construtor, gets e sets).

Segunda etapa:

- Finalizar a construção das classes de acordo com os novos conceitos estudados, incluindo os métodos específicos;
- Construir a interface do sistema (desktop)
- Finalizar a aplicação

Apresentação:

A apresentação deverá ser realizada pela equipe

Cada aluno será avaliado individualmente

A nota é individual para cada aluno a depender da sua avaliação oral

Importante

Você deve usar o máximo de conceitos de Orientação a Objetos que estudamos na sala de aula.

Considere o seguinte cenário para uma clínica veterinária:

Todo animal atendido na clínica deve ser cadastrado como nome, raça, data de nascimento e tutor. O tutor é a pessoa responsável pelo animal e também deve ter um cadastro com nome, cpf, email, telefone de contato e endereço. Desta forma, um tutor pode ter vários animais sob sua responsabilidade.

A clínica tem diversos veterinários que devem ser cadastrados no sistema indicando nome, cpf, email, telefone, especialidade (ex. cardiologista, ortopedista, anestesta, clinico geral) e o número do CFMV (conselho federal de medicina veterinária). Também os funcionários da clínica, ex. atendentes e cuidadores, devem ser cadastrados com nome, cpf, email e telefone e turno de trabalho.

Para ser atendido o animal pode ou não marcar uma consulta. Em caso de marcação a clínica tem uma agenda que atende um animal a cada 20 minutos (de 8:00 as 12:00 e de 14:00 as 18:00). A marcação é feita para o animal em uma especialidade e em uma data e hora específica. Não é indicado o médico, qualquer médico disponível faz o atendimento.

Toda consulta realizada para um animal fica registrada no sistema indicando a data, veterinário que realizou a consulta, problema, diagnostico e medicamentos prescritos para o animal.

O animal também pode ir a clinica para tomar vacina. Sempre que isso acontece a vacina tomada fica registrada para que seja possível emitir o cartão de vacina do animal. O cartão de vacina contem todas as vacinas, isto é, nome da vacina, data em que foi ministrada e validade. Desta forma, é possível saber quando o animal precisará tomar novamente a vacina.

O preço da consulta varia a depender da especialidade do veterinário. O preço das vacinas também são tabelados, ou seja, cada vacina tem um preço específico. O sistema deve calcular o valor a ser pago pelo tutor sempre que o animal é atendido na clínica.

Para o cenário acima, construir uma aplicação em Java que permita :

- Efetuar todos os cadastros necessários (inclusão, alteração, exclusão e consulta);
- Realizar agendamento (agendar e cancelar agendamento)
- Realizar consulta (incluir consulta e consultar consulta)
- Realizar vacina (incluir e consultar)
- Emitir cartão de vacina (imprimir na tela)
- Emitir cobrança para o atendimento de um animal (seja para consulta ou vacina) (calcular o valor e apresentar na tela)
- Consultar as vacinas que estão para vencer no mês para um determinado animal. (listar o nome da vacina e a data de vencimento)
- Emitir prontuário do animal (com os dados de todas as consultas realizadas) (imprimir na tela a data da consulta, medico, problema, diagnostico e medicamentos. Apresentar do mais recente para o mais antigo)

