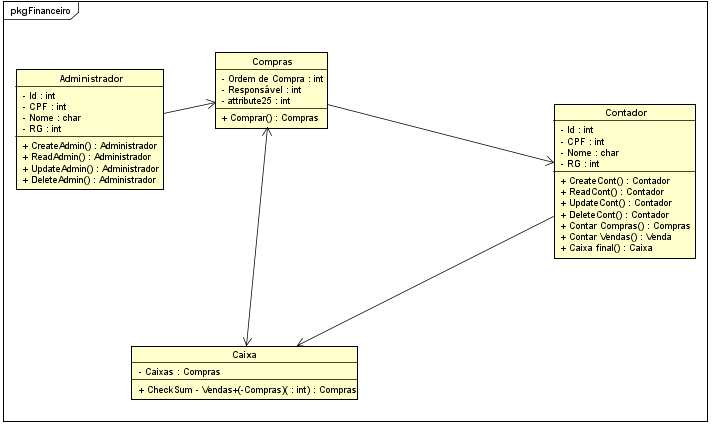
Faculdade Unievangélica

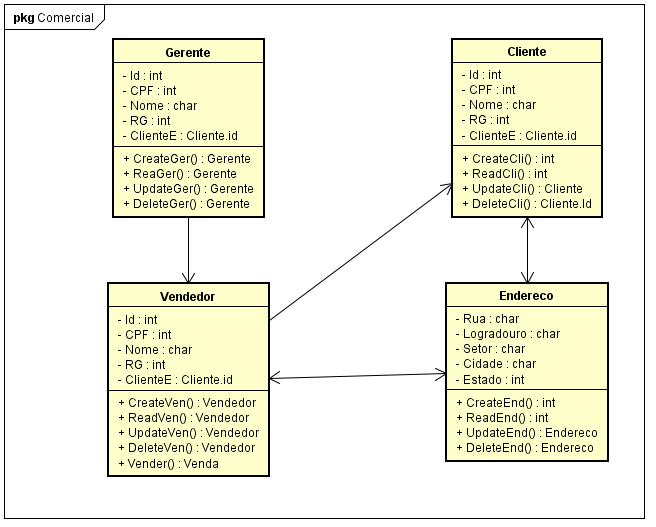
Acadêmicos: Charlley Junior Jabbar e Henrique Olimpio.

**Matéria : Programação Orientada a Objeto**

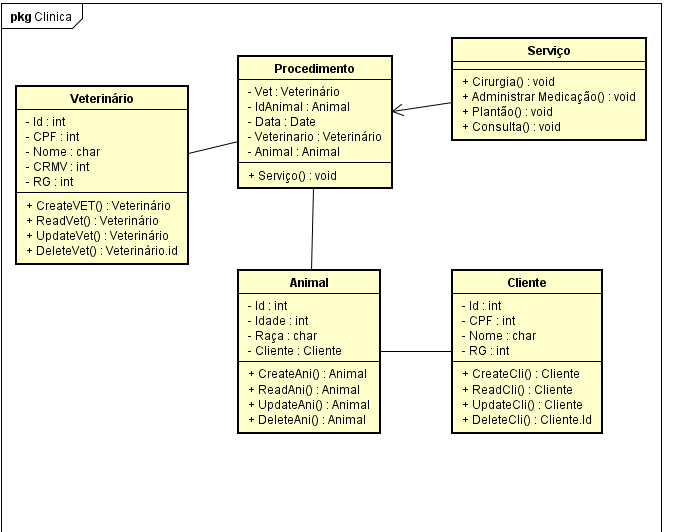
1. A)Financeiro



b) Comercial



c) Clínica



**2)a) Métodos Construtores**

Também conhecidos pelo inglês constructors, os construtores são os responsáveis por criar o objeto em memória, ou seja, instanciar a classe que foi definida. Eles são obrigatórios e são declarados conforme apresentado:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | public class Carro{    /\* CONSTRUTOR DA CLASSE Carro  \*/  public Carro(){    //Faça o que desejar na construção do objeto  }    } |

Um dos motivos é a aplicação do padrão de projeto Singleton, que controla se um objeto já foi ou não criado. Para que isso ocorra ele não pode deixar que ninguém chame diretamente o construtor da classe.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | public class Carro{    /\* CONSTRUTOR DA CLASSE Carro  \*/  public Carro(){    //Faça o que desejar na construção do objeto  }    }    public class Aplicacao {      public static void main(String[] args) {              //Chamamos o construtor sem nenhum parâmetro  Carro fiat = new Carro();  }    } |

O retorno do operador new é uma referência para o objeto recém-criado. O construtor pode receber argumentos, como qualquer método. Usando o mecanismo de sobrecarga, mais de um construtor pode ser definido para uma classe. Veja por exemplo os construtores definidos para a classe Point de Java.

b) Encapsulamento

O significado de Encapsulamento é garantir que dados "confidenciais" sejam ocultados aos usuários. Para conseguir isso, você deve:

**declarar variáveis ​​/ atributos de classe como private**

fornecer métodos públicos de obtenção e configuração para acessar e atualizar o valor de uma **private** variável;

Encapsulamento no Java. Os atributos estão todos públicos! Isso vai exatamente contra uma das nossas primeiras lições quando aprendemos Java: atributos devem ser privados e precisamos de **getters e setters** para acessá-los.

c) Métodos Gets e Sets

As variáveis private ​​só podem ser acessadas na mesma classe (uma classe externa não tem acesso a ela). No entanto, é possível acessá-los se fornecermos métodos públicos get e set .

**O get método retorna o valor da variável e o método set define o valor.**

A sintaxe para ambos é que eles começam com get ou set, seguido pelo nome da variável, com a primeira letra em maiúscula:

Os **getters e setters** são sempre usados quando queremos encapsular uma classe, ou seja, os **atributos (private**) dessa classe só poderão ser acessados por outras classes através desses métodos. Isso serve para controlar o acesso aos atributos da classe e é uma boa prática.

**d) Modificadores de acesso.**

Controlar o acesso aos seus métodos, atributos e construtores através dos modificadores private e public;

**3) O projeto**

Nas classes de cada pacote com seus métodos de construtores, com suas ações de alterar, excluir, adicionar e listar. O sistema de pet shop esta usando os métodos de get e set para encapsular outras classes no projeto.