

Lista de exercícios nº 4 - Programação Orientada a Objetos em Java

1 – Construa uma classe Imóvel que possui como atributos: área construída, número de quartos, suíte (do tipo booleano), valor base aluguel. Essa classe possui um método chamado Calcular aluguel que recebe um valor como taxa de administração e multiplica pelo valor base aluguel (valor base aluguel x taxa administração). Depois construa mais duas classes que herdam de Imóvel. Essas classes são:

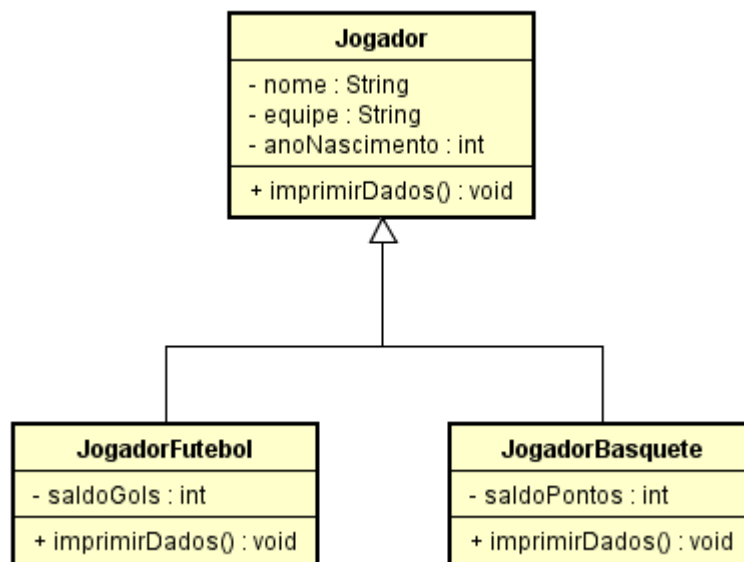
- Casa, que tem como atributos próprios: área quintal e lavanderia (booleano). A classe Casa irá sobrescrever o método calcular aluguel de Imóvel, fazendo com que a fórmula seja = valor base aluguel x taxa administração x 1.5
- Apartamento, que tem como atributo: andar e valor condomínio. A classe Apartamento também sobrescreve o método calcular aluguel e nesse caso a fórmula é = (valor base aluguel x taxa administração) + valor condomínio

Feitas as classes, escreva uma classe Aplicação, com o método main, que criará dois objetos: um do tipo casa e outro do tipo apartamento. Carregue os valores dos atributos e apresente na tela o valor de aluguel de ambos os objetos criados.

2 – Construa uma classe chamada Empregado tendo como atributos: nome, ano de nascimento, telefone e matrícula. Depois desenvolva um método **Construtor** para essa classe, ou seja, ao ser criada no programa principal será necessário informa os valores dos atributos como parâmetros.

Em seguida escreva uma classe Aplicação, com o método main, e nela crie um objeto do tipo Empregado fazendo uso de seu método construtor e apresente na tela os seus valores.

3 – Desenvolva o código para o seguinte pedaço de um diagrama UML abaixo.



Sugestão de gabarito para as questões:**Questão 1:**

```
public class Imovel {  
  
    double areaConstruida;  
    int numeroQuartos;  
    boolean suite;  
    double valorBaseAluguel;  
  
    double calcularAluguel(double taxaAdministracao) {  
  
        return valorBaseAluguel * taxaAdministracao;  
  
    }  
  
}  
-----  
  
public class Casa extends Imovel{  
  
    double areaQuintal;  
    boolean lavanderia;  
  
    @Override  
    double calcularAluguel(double taxaAdministracao) {  
  
        return valorBaseAluguel * taxaAdministracao * 1.5;  
  
    }  
  
}  
-----  
  
public class Apartamento extends Imovel {  
  
    String andar;  
    double valorCondominio;  
    @Override  
    double calcularAluguel(double taxaAdministracao) {  
  
        return (valorBaseAluguel * taxaAdministracao) +  
        valorCondominio;  
  
    }  
  
}  
-----
```



```

public class Aplicacao {

    public static void main(String[] args) {

        Casa novaCasa = new Casa();
        Apartamento novoApto = new Apartamento();

        novaCasa.areaConstruida = 200;
        novaCasa.numeroQuartos = 2;
        novaCasa.suite = false;
        novaCasa.lavanderia = true;
        novaCasa.areaQuintal = 21;
        novaCasa.valorBaseAluguel = 1500.0;

        novoApto.areaConstruida = 300;
        novoApto.numeroQuartos = 3;
        novoApto.suite = true;
        novoApto.andar = "Sexto";
        novoApto.valorBaseAluguel = 2500.0;
        novoApto.valorCondominio = 900.0;

        double taxaAdmCasa = 0.9;
        double taxaAdmApto = 0.7;

        System.out.println("#####
        ####");
        System.out.println("Valor do aluguel da casa: R$"
        +novaCasa.calcularAluguel(taxaAdmCasa));

        System.out.println("#####
        ####");
        System.out.println("Valor do aluguel do apartamento: R$"
        +novoApto.calcularAluguel(taxaAdmApto));

        System.out.println("#####
        ####");

    }
}

```

Questão 2:

```

public class Empregado {

    String nome;
    int anoNascimento;
    String telefone;
    String matricula;

    Empregado(String nome, int anoNascimento, String telefone,
    String matricula){
        this.nome = nome;
        this.anoNascimento = anoNascimento;
        this.telefone = telefone;
        this.matricula = matricula;
    }

}

```



```
public class Aplicacao {

    public static void main(String[] args) {

        Empregado emp1 = new Empregado("Marcelo Oliveira", 1971,
"3658-7789", "AN89771");

        System.out.println("Dados do empregado");
        System.out.println("-----");

        System.out.println("Nome: " +emp1.nome);
        System.out.println("Ano nascimento: " +emp1.anoNascimento);
        System.out.println("Telefone: " +emp1.telefone);
        System.out.println("Matrícula: " +emp1.matricula);
        System.out.println("-----");

    };

    }

}
```

Questão 3:

```
public class Jogador {

    String nome;
    String equipe;
    int anoNascimento;

    public void imprimirDados () {

        System.out.println("Nome do Jogador: " +nome);
        System.out.println("Equipe: " +equipe);
        System.out.println("Ano nascimento: " +anoNascimento);
        System.out.println("Tipo de jogador: " +getClass());

    }

}

-----

public class JogadorFutebol extends Jogador{

    int saldoGols;

    @Override
    public void imprimirDados () {
        super.imprimirDados();
        System.out.println("Saldo de gols: " +saldoGols);
    }

}

-----
```





```
public class JogadorBasquete extends Jogador{

    int saldoPontos;

    @Override
    public void imprimirDados() {
        super.imprimirDados();
        System.out.println("Saldo de pontos: " +saldoPontos);
    }
}
```

```
-----

public class Aplicacao {

    public static void main(String[] args) {

        JogadorFutebol jog1 = new JogadorFutebol();
        JogadorBasquete jog2 = new JogadorBasquete();

        jog1.nome = "Pedro Silva";
        jog1.equipe = "Macaé";
        jog1.anoNascimento = 1989;
        jog1.saldoGols = 35;

        jog2.nome = "Marcos Andrade";
        jog2.equipe = "Osasco";
        jog2.anoNascimento = 1995;
        jog2.saldoPontos = 346;

        jog1.imprimirDados();
        System.out.println("-----");
        jog2.imprimirDados();
    }
}
```

