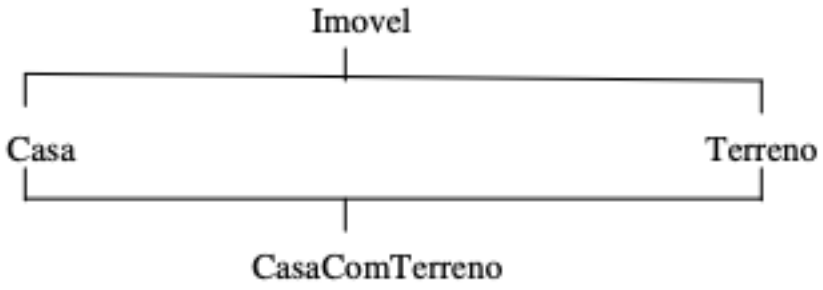
IFPB – Campus Campina Grande
Curso de Bacharelado em Engenharia de Computação
Disciplina: **Programação Orientada a Objetos (Teoria e prática) – Semestre 2021.1**
Professor: Leandro de Almeida Melo (leandro.melo@ifpb.edu.br)

Prova – II

1. (30 pontos) Uma Imobiliária de CG te contratou para desenvolver um protótipo de um sistema de informação para cadastrar os imóveis de sua carteira de ofertas. Considere que durante a especificação do sistema foram identificadas as seguintes categorias de imóveis classificadas na hierarquia abaixo. Considere também que:

- Todo Imóvel possui um preço (um número ponto flutuante) e um identificador (um número inteiro).
- Todo Casa possui um número de quartos (um número inteiro).
- Toda Terreno possui uma área (um número ponto flutuante).
- Toda CasaComTerreno possui um valor por metro quadrado (float valorMQ) e um método chamado calculoValor() que multiplica o valor por metro quatrado x número de quartos x tamanho da área.
- Todas as classes possuem toString() retornando o valor de seus atributos em forma de texto. No entanto, no toString() de CasaComTerreno deve ser impresso **os toString() anteriores mais o texto** "Valor total da cada: " mais o calculo do valor da CasaComTerreno.
- Inclua um construtor com argumentos para inicialização dos atributos em todas as classes e faça uso deles.



Faço um programa principal/drive (main) que crie uma instância de CasaComTerreno utilizando o construtor com argumentos e imprima o toString.

2. (70 pontos) implemente a classe Turma apresentado no diagrama abaixo e faça um programa principal/drive (main), para isso leve em consideração os seguintes tópicos:

- Todas as classes possuem toString() e nas classes derivadas deve-se evitar a repetição de código, ou seja, deve-se aproveitar o toString() da classe base sempre que possível.
- Inclua um construtor com argumentos para inicialização dos atributos em todas as classes e faça uso deles. Ou seja, ao criar uma turma (no construtor da classe Turma), deve-se chamar o construtor da classe associada Professor e enviar seus atributos, por sua vez, o construtor de Professor deve-se chamar o construtor de suas superclasses e/ou classes associadas até que todos os atributos estejam preenchidos. Lembre-se de criar também um construtor sem argumentos em Endereco, Pessoa e Aluno.
- Na classe Turma, os métodos addAluno(Aluno &a) e alterarProfessor(Professor &p) fazem passagem de valor por referência, ou seja, uma instancia de aluno e professor serão criadas no programa principal/drive (main) e enviados para objeto da classe Turma.
- Note que quando você declara Aluno alunos[10], será criado um vetor de Aluno com dez posições. No construtor sem argumentos da classe Aluno, inicialize a variável matricula com o valor -1. Utilize esse valor para indicar que a posição do vetor está vazia (por exemplo, alunos[0].getMatricula()==-1 indica que a posição 0 do vetor alunos está desocupada). Note que ao criar uma instancia da classe Turma, o professor já será preenchido através do construtor e o vetor de Alunos será criado com dez posições e em cada uma delas a matrícula será -1 indicando que a posição não possui nenhum aluno. Os alunos só serão adicionados através do método addAluno(Aluno &a) da classe Turma. Você deve impedir qualquer outro valor negativo para o campo da matrícula do aluno. Caso isso ocorra, adiciona-se o valor -1 na mátrícula.
- Certifique-se que no seu programa não exista um aluno duplicado na Turma, para isso, use o valor de matricula como identificador único de cada aluno.
- Certifique-se que seu programa só adicione no máximo dez alunos e um professor em cada turma.
- No método imprimirAluno(int matricula) será impresso o toString() do aluno com a matricula informada por argumento ou uma mensagem informando que não existe nenhum aluno com essa matrícula no sistema.
- No método ImprimirAlunos(), será impresso o toString() de todos os alunos do vetor alunos.
- No método removerAluno(int matricula), deve-se remover o aluno com a matricula informada por argumento do vetor, ou seja, atribuir -1 na matricula da posição desse aluno no vetor ou imprimir uma mensagem informando que esse aluno não existe no sistema.
- No método imprimirProfessor(), deve-se imprimir o toString() do professor cadastrado na turma.
- No programa principal/drive (main) teste todos os métodos de Turma.
- Observe a visibilidade dos atributos no diagrama de classe abaixo.

- Crie métodos getters e setters conforme a necessidade.