UDESC – CCT
Universidade do Estado de Santa Catarina
Disciplina: Programação Orientada a Objetos – POO TADS
Professor: Valmor Adami Jr
Monitor: Vinícius Takeo Friedrich Kuwaki – vtkwki@gmail.com

Lista de Exercícios – Prova 1

(1) Uma empresa está interessada em desenvolver um sistema de uma "Casa Inteligente", modele um diagrama UML conforme as especificações abaixo:

A casa possui cômodos dos quais este possui: janela(s), porta(s) e lâmpada(s). As janelas possuem seu modo automático e seu modo manual, quando a janela está no modo automático, ela utiliza seu sensor para controlar a sua abertura, sempre que estiver chovendo a janela será fechada. Em seu modo manual o usuário pode abrir e fechar a janela sempre que desejar;

As portas possuem uma fechadura, esta por sua vez tranca ou destranca a porta quando o usuário desejar, além do mais a porta também tem dois estados: aberta ou fechada, determinada pelo usuário também;

As lâmpadas estão conectadas a um sensor externo a casa, logo, as lâmpadas assim como a janela, possuem seu modo automático e seu modo manual, no primeiro ela é controlada por esse sensor, caso a luminosidade no sensor seja baixa, o sensor ativa a lâmpada, caso contrário a lâmpada permanece desligada. No modo manual o usuário pode acender ou apagar a luz.

(2) Observe o código das classes **Pessoa** e **Médico** abaixo. Determine o erro nessas classes e apresente uma solução.

```
public class Pessoa{

private String nome;
private int idade;

public Pessoa( String nome , int idade ){
this.nome = nome;
this.idade = idade;
}

}
```

```
public class Medico extends Pessoa{

private String especializacao;

Public Medico ( String especializacao ){
super();
this.especializacao = especializacao;
}
```

UDESC – CCT
Universidade do Estado de Santa Catarina
Disciplina: Programação Orientada a Objetos – POO TADS
Professor: Valmor Adami Jr
Monitor: Vinícius Takeo Friedrich Kuwaki – vtkwki@gmail.com

Lista de Exercícios – Prova 1

(3) Identifique e explique o(s) erro(s) nas classes abaixo:

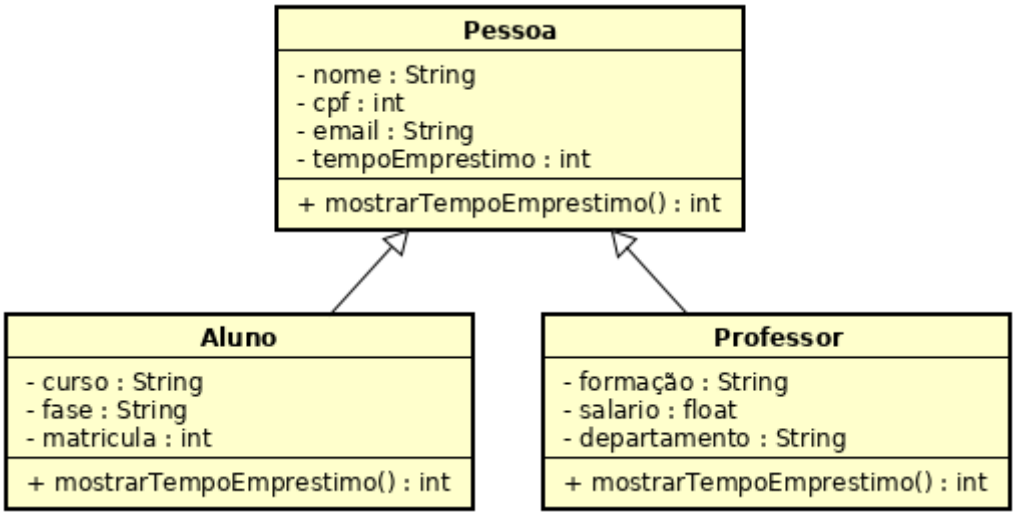
```
class Ponto2D{

private double x,y;

Ponto 2D (double _x, double _y){
x=_x; y=_y;
}

}
```

```
class Ponto3D extends Ponto2D{

private double z;

Ponto3D (double x, double y, double z){
x=_x;
y=_y;
z=_z;
}

}
```

UDESC – CCT
Universidade do Estado de Santa Catarina
Disciplina: Programação Orientada a Objetos – POO TADS
Professor: Valmor Adami Jr
Monitor: Vinícius Takeo Friedrich Kuwaki – vtkwki@gmail.com

Lista de Exercícios – Prova 1

(4) A partir das classes descritas abaixo, implemente um sistema em Java que simula um pequeno sistema de empréstimo de livros em uma biblioteca.



Neste sistema existem: Pessoas que representam qualquer pessoa da comunidade que deseja emprestar um livro nessa biblioteca e alunos e professores que também podem emprestar livros, porém, alunos possuem a vantagem de alugar um livro por 40% mais tempo que uma pessoa normal, professores também possuem a vantagem de poder alugar por mais tempo, no caso 70% a mais que uma pessoa normal. Implemente o método **mostrarTempoEmprestimo()** das classes filhas, reimplementando o método da superclasse, através de uma chamada do método original. Utilize um Array de objetos da superclasse para armazenar tanto as instâncias de objetos da superclasse como instâncias de objetos das classes filha (dica: procure a respeito do operador **instanceof**), implemente então os métodos abaixo:

- **listaProfessores()**: que lista apenas os professores presentes no array;
- **listaAlunos()**: que lista apenas os alunos presentes no array;
- **listaProfessoresSalario(float salario)**: que lista apenas os professor presentes no array que possuem o salário maior ou igual ao salário informado pelo usuário;
- **cadastraProfessor()**: que armazena um novo professor no array;
- **cadastraPessoa()**: que armazena uma nova pessoa da comunidade no array;
- **mostrarTempoEmprestimo(Pessoa pessoa)**: que recebe uma pessoa e retorna o tempo de empréstimo dela;

@considere o atributo tempoDeEmprestimo como sendo protected.

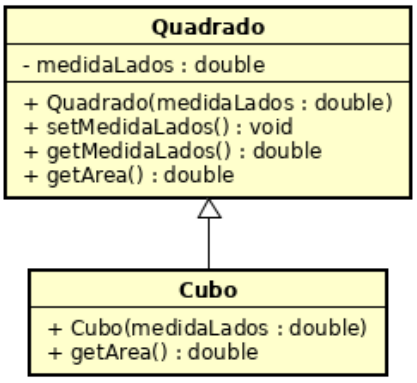
@@Considere tb que existe o metodo getNome() para acessar o nome da Pessoa.

UDESC – CCT
Universidade do Estado de Santa Catarina
Disciplina: Programação Orientada a Objetos – POO TADS
Professor: Valmor Adami Jr
Monitor: Vinícius Takeo Friedrich Kuwaki – vtkwki@gmail.com

Lista de Exercícios – Prova 1

(5) Implemente a classe Quadrado descrita abaixo, onde a área de um quadrado é dado por:

$A_{QUADRADO} = lado^2$



A partir dela implemente a classe Cubo que reescreva o método getArea(), utilizando o método da superclasse. A área do cubo é dada por:

$A_{CUBO} = 6 * A_{QUADRADO}$

(6) A partir da classe Animal, implemente três especializações com pelo menos um atributo a mais que a superclasse, e crie uma classe SistemaAnimais que permite o cadastro, remoção, inserção e busca das generalizações dessa classe. Por exemplo, suponha uma classe Pessoa e uma generalização Funcionário, ambas possuem um atributo cpf e a classe Funcionário possui um atributo a mais chamada salário, o método de busca dessas classes seriam: buscaPessoa(int cpf), onde esse método recebe o cpf de uma pessoa e se essa pessoa existe no array onde será efetuada a busca, esse método retorna a Pessoa que possui esse cpf, a mesma coisa para o Funcionário;