

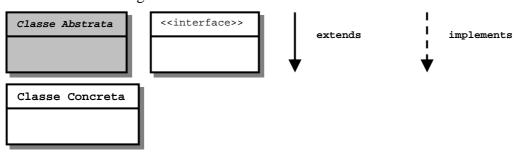
	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA – CAMPUS CAMPINA GRANDE				
ſ	CURSO:	CURSO SUPERIOR DE TECNOLOGIA EM TELEMÁTICA			
ſ	PERÍODO:	P3	TURMA: N		
ſ	DISCIPLINA:	PROGRAMAÇÃO ORIENTADA A OBJETOS			
ſ	PROFESSOR:	CÉSAR ROCHA VASCONCELOS	SEMESTRE LETIVO		

NOME:		

LISTA DE INTERFACES E POLIMORFISMO

Obs.: procure responder a todas as questões desta lista não deixando nenhuma incompleta ou em branco.

- 1) Responda as questões abaixo informando se uma alternativa é verdadeira ou não. Caso a alternativa seja falsa, forneça as devidas correções:
- a- Se você não quer que uma classe seja instanciada, marque-a com a keyword abstract
- b- Uma classe abstrata só pode ter métodos abstratos e/ou atributos
- c- Se uma classe tem pelo menos um método abstrato, ela pode ser abstrata
- d- Um método abstrato não tem corpo, e sua declaração termina apenas com ';'
- e- Ao herdar de uma classe abstrata, uma classe concreta pode implementar apenas alguns métodos (em vez de todos). Porém, terá de virar abstrata também.
- f- Uma classe só pode ter uma superclasse, mas pode implementar várias interfaces
- g- Uma interface é como se fosse uma classe abstrata pura ela define apenas métodos abstratos e não possui construtores
- h- Devemos criar uma interface com a keyword implements
- i- Devemos implementar uma interface com a keyword extends
- j- Uma classe abstrata também pode implementar várias interfaces
- k- Se uma classe deseja implementar uma interface, ela deve implementar todos os métodos desta interface, já que estes métodos são implicitamente **private** e **abstract**
- 1- Pode-se definir um tipo em Java apenas com classes
- m- Herança de tipo e herança de implementação significam a mesma coisa
- n Devemos usar classes abstratas ao fatorar código comum às subclasses
- o Uma classe abstrata não pode ter construtores, visto que Java não permite instanciá-las
- p Boa parte dos autores explica polimorfismo usando hierarquia de classes, mas posso também ter polimorfismo com interfaces (maior flexibilidade)
- q- Na ligação tardia, quem decide que método certo será chamado é o objeto receptor desta mensagem
- r- Quando temos uma referência (genérica) ligada a um objeto de uma classe mais especifica, posso chamar (via esta referência) não apenas os métodos definidos na referência, mas também métodos declarados na classe do objeto receptor.
- 2) Utilizando-se dos seguintes símbolos:

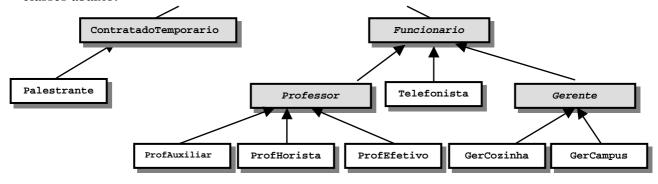


Na ferramenta UML de sua preferência, crie o **diagrama** de classes correspondente a cada um dos trechos de código mostrados abaixo:

```
I- public interface Foo {}
                                                                       4- public interface Alocavel {}
                                                                         public abstract class Sala implements
  public class Bar implements Foo{}
                                                                                           Alocavel{}
2- public abstract class Pessoa implements Cidadao {}
                                                                       5- public abstract class Gravador {}
  public class Professor extends Pessoa{}
                                                                         public class CDBurner extends Gravador
                                                                                      implements Desconto { }
  interface Cidadao{}
                                                                         public class DVDBurner extends Gravador
                                                                                      implements Desconto { }
                                                                         interface Desconto { }
3- interface Veiculo {}
  public abstract class Carro implements Veiculo { }
  public class Mercedes extends Carro{}
  public class MercedesConversivel extends
               Mercedes { }
  public class Ferrari extends Carro{}
```

Ps.: você deve usar uma ferramenta UML real como **Jude** (http://www.esm.jp/jude-web/index.html) ou **ArgoUML** (http://argouml.tigris.org/) para a confecção dos diagramas

3) Suponha uma aplicação para uma faculdade qualquer idealizada com base no diagrama de classes abaixo:



<u>Instruções:</u> nesta modelagem, temos as classes abstratas em cinza e as concretas em branco.

Seu cliente informou que precisa de um novo método no sistema chamado descontarImposto(???). Este novo método será usado para aplicar um desconto de 1% nos salários não de todos os funcionários, mas apenas dos tipos de funcionários que possuem maiores salários na faculdade, que são: GerCampus e ProfEfetivo. Sendo assim, explique, com suas palavras, os prós e os contras caso este método fosse escrito da seguinte forma:

- descontarImposto(Funcionario f) {...}
- descontarImposto(Professor p){...}
- descontarImposto(Gerente g){...}
- descontarImposto(ProfEfetivo pe){...}
- descontarImposto(GerCampus gc){...}

Qual seria a sua solução para este problema? Pense também o que seria necessário fazer para aplicar um novo desconto (agora de 2%) nos pagamentos de não-funcionários? Ex.: Palestrante visitante que recebeu um alto pagamento e que precisa ter seu salário descontado em 2%.