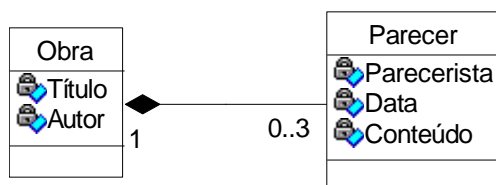


Lista de Exercícios Nro 2 – 01/11/2014

1. Crie uma classe chamada *VetorDeReais*, que internamente possui um vetor de n reais (*double*). n é passado como parâmetro na criação do objeto. Implemente métodos para:
 - a) Receber os n valores;
 - b) Retornar o valor da multiplicação (M) entre este objeto e outro objeto de *VetorDeReais* recebido como parâmetro, de tal modo que $M = (A[1] * B[n]) + (A[2] * B[n-1]) + \dots$
 - c) Retornar um novo objeto de *VetorDeReais*, criado a partir da divisão deste objeto por outro objeto recebido como parâmetro, sendo $novo[i] = A[i] / B[i]$;
 - d) Retornar a maior diferença entre dois elementos consecutivos do vetor.

2. Mesmo problema da questão anterior, mas utilizando ArrayList no lugar de vetor.

3. Uma empresa de promoções culturais recebe diferentes tipos de obras para que sejam lançadas no mercado. Antes de realizar um lançamento, cada obra é avaliada por até três pessoas especializadas (pareceristas). Para controlar as obras submetidas à avaliação, está sendo desenvolvido um sistema com a seguinte estrutura de classes:



- ✓ A classe Parecer é parte da classe Obra (agregação).
- ✓ Deve ser possível cadastrar várias obras.

Exemplos de Obras:

<u>Título</u>	<u>Autor</u>	<u>Pareceres</u>		
A escalada do terror	Fernandinho Beira-mar	Parecerista	Data	Conteúdo
		João de Abreu	10/10/2011	Livro que não pode ser publicado, pois é um manual do crime.
		Paula Prantos	01/09/2011	A violência é mostrada de forma cruel. Livro sujeito à censura prévia.
Confins do Sertão	Hector Babenco	Parecerista	Data	Conteúdo
		Lima Duarte	15/08/2012	Vídeo que demonstra a sensibilidade da alma nordestina. Vale a pena assistir.

Escreva um programa Java orientado a objetos que leia os dados disponíveis pela empresa. A qualquer momento o usuário pode consultar uma obra, através do título, imprimindo seus dados em tela (inclusive dos pareceres).

4. O sindicato dos jogadores profissionais de futebol está negociando o aumento de salários de toda a categoria, para todos os clubes, de acordo com a tabela seguinte:

Salário atual (R\$)	Aumento (%)
0 - 9.000	20
9.001 - 13.000	10
13.001 - 18.000	5
acima de 18.000	0

Escreva um programa Java orientado a objetos que leia o nome, situação (Titular ou Reserva), equipe (clube) e o salário de cada jogador e imprima o nome, salário atual e novo salário proposto.

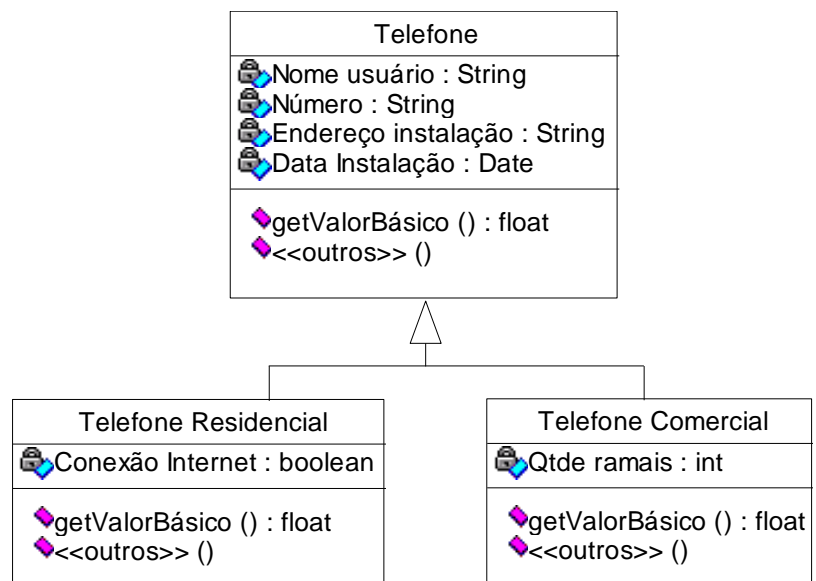
Ao final, imprima :

1. o total dos salários atuais e dos novos salários;
2. o nome, salário atual e novo salário do jogador com menor salário atual;
3. a participação percentual dos novos salários dos titulares e dos reservas sobre o total de novos salários.
4. o nome, salário atual e novo salário dos jogadores com novo salário acima da média de novos salários.
5. dados de todos os clubes:
 - 5.1. nome do clube e cidade-sede;
 - 5.2. média salarial atual; e
 - 5.3. quantidade de jogadores.

5. Uma concessionária de telefonia precisa refazer seu sistema de controle de linhas telefônicas. Ela oferece a seus clientes (usuários) dois tipos de linhas: residenciais e comerciais. As linhas residenciais possuem um custo fixo mensal (valor básico) de R\$15,00. As linhas comerciais têm seu custo fixo mensal fixado de acordo com a data de instalação: se antes de 01/01/2012 = R\$ 25,00; a partir de 01/01/2012 = R\$37,50.

O sistema modelado possui a seguinte estrutura de classes:

- ✓ A classe Telefone é classe-base para as classes Telefone Comercial e Telefone Residencial.
- ✓ O método getValorBásico não é implementado na classe-base (abstrato), apenas nas descendentes.
- ✓ Devem ser implementados construtores e destrutores próprios para cada classe definida



A interface deve permitir o cadastramento de vários telefones. A escolha do telefone a ser criado deve ser do usuário.

A qualquer momento o usuário do sistema pode consultar os dados de um telefone (pelo seu número). Também deve ser possível verificar o potencial faturamento da empresa, baseado no valor a pagar em custo fixo de cada telefone.

Exemplo de dados:

Objeto de Telefone Residencial	Nome usuário: João da Silva Endereço Instalação: Rua Pedro I, 1234 – Blumenau / SC Data Instalação: 15/10/2014 Valor básico a pagar: R\$ 15,00	Número: (47) 3322-3232 Conexão Internet: Não
Objeto de Telefone Comercial	Nome usuário: XPTO Ltda Endereço Instalação: Rua Pedro I, 1234 – Blumenau / SC Data Instalação: 04/03/2010 Valor básico a pagar: R\$ 25,00	Número: (47) 3311-1000 Qtde ramais: 14
Objeto de Telefone Comercial	Nome usuário: Maria Antonina ME Endereço Instalação: Rua Apolônia, 4 – Blumenau / SC Data Instalação: 01/01/2012 Valor básico a pagar: R\$ 37,50	Número: (47) 3333-1212 Qtde ramais: 2

6. Uma impressora precisa de folhas para imprimir os dados recebidos. As folhas possuem um limite de caracteres por linha e de quantidade de linhas. Existem dois tipos de folhas: A4 e Carta. A folha A4 suporta 20 caracteres por 12 linhas e a folha Carta suporta 8 caracteres em 10 linhas. Quando a impressora recebe uma lista de frases para imprimir, ela passa linha a linha digitada para a folha. A folha se encarrega de imprimir os caracteres por linha suportados por ela. Os caracteres excedentes são impressos nas linhas subsequentes. Por exemplo, se a impressora passar para a folha Carta uma linha “abcdefghijklmn”, a folha irá imprimir:

abcdefgh

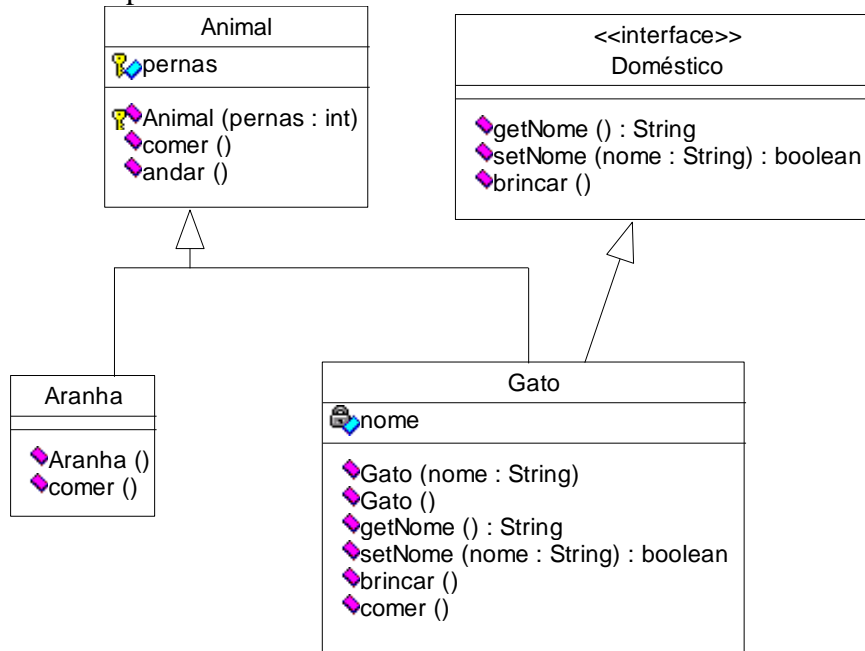
ijklmn

Ao enviar a linha para a folha, essa deve avisar à impressora (como retorno do método) se a folha chegou ao fim. No caso acima, se a última linha fosse “abcdefgh”, então ela não imprimiria “ijklmn”, e avisaria a impressora que chegou ao seu fim. A impressora troca de folha, porém, o texto “ijklmn” é perdido.

Sua aplicação deve permitir ao usuário digitar um texto a ser impresso e permitir escolher com que tipo de folhas a impressora está trabalhando. Ao mandar imprimir, uma simulação da impressão deve ocorrer em tela.

Especifique e implemente as classes necessárias para resolver esse problema.

7. Você irá criar uma hierarquia de animais, baseada na classe abstrata Animal. Algumas das classes de animais implementam uma interface chamada Doméstico.



- Crie a classe Animal, que é a superclasse abstrata de todos os animais.
 - Declare um atributo protegido chamado pernas, que guarda a quantidade de pernas.
 - Defina um construtor protegido que inicializa o atributo pernas.
 - Declare um método abstrato comer.
 - Declare um método concreto andar que imprima algo sobre como os animais caminham (incluindo o número de pernas).
- Crie a classe Aranha.
 - A classe Aranha é subclasse de Animal.
 - Defina um construtor default que chame o construtor da superclasse para especificar que as aranhas têm oito pernas.
 - Implemente o método comer.
- Crie a interface Doméstico, conforme especificado no diagrama em UML.
- Crie a classe Gato que é subclasse de Animal e implementa Doméstico.
 - Esta classe possui um atributo String para guardar o nome do animal doméstico.
 - Defina um construtor que receba um parâmetro String que especifica o nome do gato. Este construtor também deve chamar o construtor da superclasse para especificar que gatos têm quatro pernas.
 - Defina um outro construtor que não recebe parâmetros. Este deve chamar o construtor anterior (item b, utilizando a chave this) e passar uma String vazia como parâmetro.
 - Implemente os métodos da interface Doméstico.
 - Implemente o método comer.
- Crie um programa chamados TesteAnimais. O método main deve criar e manipular instâncias das classes criadas.
 - Comece com
Gato g = new Gato("Garfield");
Animal e = new Aranha();
Doméstico d = new Gato();
Experimente: a) chamar os métodos de cada objeto, b) converter (cast) os objetos, c) usar polimorfismo.