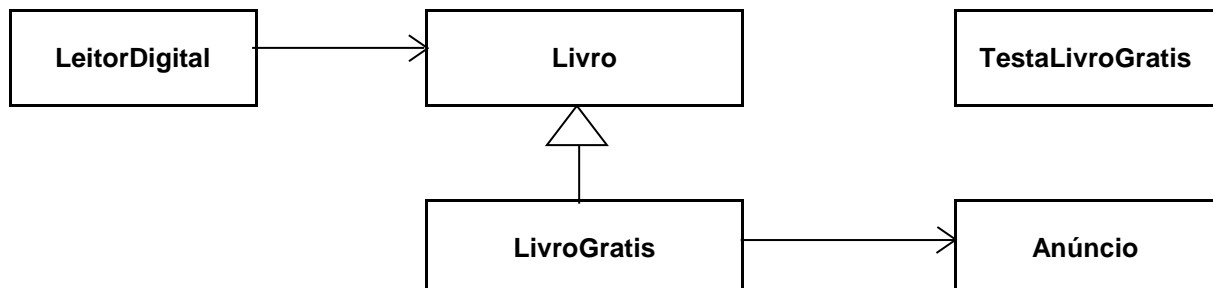


Em grupos de até 2 alunos. O trabalho é extraclasse – **NÃO DEVE SER FEITO DURANTE AS AULAS.**

Simulação (simplificada) de um *leitor de livros digitais*, os *e-books*. O modelo UML das principais classes aparece abaixo:



Programa as cinco classes do diagrama. Escolha bons nomes para os membros. Usar o seguinte comentário no início do texto fonte de cada classe:

/ Alunos : xxxxxxxxxxxxxx e xxxxxxxxxxxxxx Trabalho B Lab 1 Turma xx 2016/1 */**

Classe Livro (ou Ebook)

Atributos privados (somente estes):

- título do livro
- número de páginas – quantidade total de páginas do livro
- preço do livro
- tempo total de leitura – em minutos
- percentual já lido de todo livro
- interrupção – o número da página onde foi interrompida a leitura na última sessão de leitura que ocorreu



Construtor: recebe o título do livro e o seu número de páginas, como parâmetros. O preço do livro deve ser calculado a R\$ 0,15 por página.

Métodos:

+ *fazLeitura* – este método simula uma sessão de leitura do livro. Para simular a quantidade de páginas lidas durante a sessão, calcular um inteiro randômico menor ou igual à quantidade de páginas que ainda não foram lidas. Calcular o tempo de leitura da sessão à razão de 3 minutos por página. Ao final da sessão, atualizar os três últimos atributos listados acima e retornar a quantidade de páginas lidas nessa sessão de leitura.

+ *exibeDados* – imprime na tela os valores de todos os atributos do objeto, um em cada linha, com títulos adequados.

+ métodos *get* e *set* que o aluno achar necessários para o funcionamento da aplicação.

Classe LivroGratis

- Subclasse da classe **Livro**. Trata-se de herança. Portanto, evite repetir código já existente na superclasse.
- Um LivroGratis é um Livro gratuito, que exibe um anúncio a cada sessão de leitura.

Atributos privados (além dos herdados):

- um array de objetos do tipo **Anuncio**
- último anúncio exibido – inteiro que indica a posição do array de anúncios do último anúncio apresentado

Construtor: recebe três parâmetros: o título do livro, o número de páginas e o tamanho do array de anúncios. O preço deve ser zerado. Instanciar o array.

Métodos:

+ insere os anúncios – para cada posição do array de anúncios, instanciar um objeto do tipo **Anuncio**, lendo seu título do teclado.

+ *fazLeitura* – sobrescreve o método de mesmo nome da superclasse. Faz tudo que o método da superclasse faz, mais o que segue. Ao início da sessão de leitura, o próximo anúncio do array é exibido: simular isso chamando o método correspondente da classe **Anuncio**. Atualizar o atributo que controla o último anúncio exibido e retornar a quantidade de páginas lidas durante a sessão.

+ *exibeDadosDeAnuncios* – mostra na tela os dados dos objetos do array de anúncios.

+ *exibeDados* – Mostra na tela todos os valores dos atributos do livro grátis e chama, a seguir, o método anterior, para mostrar os anúncios armazenados no array de anúncios.

Classe **Anuncio**

Atributos privados (somente estes):

- título do anúncio – que pode ser o nome do produto ou empresa que está anunciando
- número de exibições – conta a quantidade de vezes que o anúncio foi apresentado (exibido) no leitor digital.

Construtor: recebe só o título do anúncio via parâmetro. O contador de exibições inicia em zero.

Métodos:

+ *exibe* – método que apenas simula a apresentação (exibição) do anúncio. Deve imprimir na tela a frase *Exibido o anúncio xxxxxxxxxxxxxxxxxxxxxx*, onde os x's devem ser substituídos pelo título do anúncio. Atualizar o atributo que conta as exibições.

+ *exibeDados* - método que imprime, numa linha, o título do anúncio e o número de exibições até o momento.

+ métodos *get*

Classe **LeitorDigital**

Atributos privados (somente estes):

- estante - um array de objetos de tipo **Livro**, que armazena os livros baixados pelo usuário para serem lidos através do aplicativo.
- preço por exibição de cada anúncio – o que vai custar ao anunciante por cada exibição de seu anúncio
- faturamento – vai acumular toda a renda do aplicativo, com a venda dos Livros baixados e com os anúncios mostrados.

Construtor: recebe o tamanho do array, isto é, a quantidade máxima de livros que poderão estar armazenados na estante simultaneamente. O construtor apenas deve instanciar o *array* e deixá-lo ainda com todas as posições *null*. Os livros serão inseridos por método próprio para isso. Inicializar o preço por exibição de anúncio com um valor constante, de R\$ 2,50.

Métodos:

+ *busca livro* – recebe texto, como parâmetro, e procura na estante um livro com este título. Se achar, retorna o objeto encontrado; se não achar, retorna *null*.

+ Continuar a programação do método abaixo iniciado, que simula a inserção de um livro ou livro grátis no array que simula a estante do leitor digital.

```
public String baixaUmLivro() { //Não alterar esta parte
    Teclado t = new Teclado();
    Livro v = null;
```

Na sequência, ler do teclado um título de livro: se já existe algum livro na estante com este título, exibir mensagem e repetir a leitura, até que o título digitado seja inédito. A seguir, ler o número de páginas do livro e perguntar ao usuário se ele quer o livro grátis ou pago. Conforme sua resposta, instanciar um objeto **Livro** ou **LivroGratis** e atribuí-lo à variável *v*. Sendo livro grátis, comandar a inserção de anúncios, chamando o método adequado. Finalmente, inserir o livro, que foi instanciado, na próxima posição vazia do array que representa a estante. Se for um livro pago, atualizar o atributo do faturamento. Se a estante estiver lotada,

nada disso pode ser feito, obviamente. O método deve retornar uma das frases seguintes, para indicar o sucesso ou não da operação:

Livro baixado com sucesso
Estante lotada

+ *exibeEstante* – Mostra na tela os dados de cada livro existente na estante, precedidos do número da posição que cada um ocupa no array, e retorna true. Se a estante estiver vazia, retornar false.

+ *sessaoDeLeitura* – Este método simula uma sessão de leitura de um livro ou livro grátis. Começa listando os livros existentes, chamando o método anterior para isso. A seguir, o usuário escolhe, via teclado, qual a obra que ele quer ler, através da digitação do respectivo número que está mostrado na tela. Validar, com repetição, esta entrada de dados. Chama o método que faz leitura da classe **Livro**. Se for um livro grátis, atualizar o atributo que acumula o faturamento, correspondente ao anúncio que é exibido na sessão. Encerrar a sessão, imprimindo a frase



Você leu xx páginas nesta sessão

(xx é substituído pela quantidade de páginas lidas durante o período) e retornar true.

Se a estante estiver vazia, a sessão é interrompida já no seu início, encerrando o método e retornando false.

+ método que imprime uma listagem de todos os anúncios existentes em todos os livros grátis que estão na estante. Tem método na classe **LivroGratis** que terá de ser chamado.

+ lista preços de livros – este método deve retornar um array contendo os preços de todos os livros não gratuitos existentes na estante. O array devolvido, portanto, não deverá ter nenhuma posição zerada. Se não houver nenhum livro não gratuito na estante, ou esta estiver vazia, retornar null.

+ Métodos *get* – só o que for necessário para o funcionamento do aplicativo

Classe TestaLivroGratis

Complete a classe de teste abaixo, programando o que pede cada comentário, e sem alterar o que já está codificado:

```
public class TestaLivroGratis() {
    public static void main(String[] args) {
        Teclado t = new Teclado();
        Livro a = new LivroGratis("O trabalho", 10+(int) (Math.random()*400),
                                   2+(int) (Math.random()*4));

        // Inserir anuncios e exibir todos os dados do livro

        // Declarar e inicializar variáveis locais necessárias

        // Simular várias sessões de leitura, parando quando o livro for todo
        // lido ou o número de páginas de uma sessão estiver no intervalo (20,27)
        do{

        }
        while ( );

        // Exibir a quantidade de sessões que foram realizadas
```

```
// Exibir a quantidade de páginas lidas na sessão de leitura mais longa

// Exibir a média de páginas por sessão

// Exibir o título do anúncio que foi apresentado menos vezes. Em
// caso de empate, qualquer um dos que empataram nessa condicao

    }
}
```

Evite cedilhas e acentuação no código Java

Entrega e apresentação nas datas que constam no plano de ensino.

Em papel, um **diagrama de objetos** com 3 livros (dos dois tipos) na estante de tamanho 5. O array de anuncios deve ter tamanho 3, com os anúncios já armazenados.

Postar os arquivos **Livro.java**, **LivroGratis.java**, **Anuncio.java**, **LeitorDigital.java** e **TestaLivroGratis.java** Obs. Não esquecer o seguinte comentário no início do texto fonte de cada classe:

/ Alunos : xxxxxxxxxxxxxx e xxxxxxxxxxxxxx Trabalho GB Turma: xx 2016/1 */**