

## ACH2002 – Introdução à Ciência da Computação II

EACH – SEGUNDO SEMESTRE DE 2008

Terceiro Exercício-Programa

Data de entrega: até 2 de dezembro de 2008.

Professor: Delano Medeiros Beder

### Introdução

Neste terceiro exercício-programa do semestre (EP3) o objetivo é implementar um *dicionário* e um *tesauro*. O dicionário é um velho conhecido e, como todos sabem, contém os diferentes significados das palavras de uma língua. Já um *tesauro* (do latim *thesaurus*) é algo menos conhecido. Trata-se de um dicionário de termos correlatos, por exemplo, sinônimos, termos contrastantes como antônimos, ou termos simplesmente relacionados semanticamente (e.g., mamífero, bovino, marsupial e primata).

Considere, por exemplo, a palavra *humano*. Se consultarmos um dicionário, veremos que esta palavra possui dois significados:

- (1) *Adj.* pertencente ou relativo ao homem: natureza *humana*; gênero *humano*;
- (2) *Adj.* bondoso, humanitário.

Porém, se consultarmos em um tesauro a palavra *humano* podemos encontrar a seguinte resposta:

- *Sinônimos*: homíneo, mortal;
- *Termos contrastantes*: animal, sobrehumano, angélico, subhumano;
- *Termos relacionados*: antropológico, etnológico.

Neste EP espera-se que os alunos implementem um programa capaz de implementar as funções de um dicionário e de um tesauro.

### Primeira Parte — Estrutura de Classes

#### Classe Dicionario

Esta classe deverá conter métodos que realizam as funções de um dicionário, a saber:

1. inserir palavras em um dicionário;
2. inserir os significados (ver classe `TermoDicionario`) das palavras no dicionário (no máximo 5);
3. realizar buscas de palavras no dicionário e imprimir o resultado encontrado, por exemplo, ao buscar a palavra *mixar* deve-se imprimir:

Mixar:

(1) V. int.: Terminar, acabar, findar.

(2) V. int.: Falhar, gorar, frustrar-se, malograr-se.

(3) V. T. dir.: Gorar, frustrar , malograr. Ex.: A chuva mixou o passeio.

4. eliminar palavras do dicionário.

## Classe Tesauro

De maneira semelhante, deve-se criar uma classe **Tesauro** que realizam as funções de um tesauro, a saber:

1. inserir palavras em um tesauro;
2. inserir os termos relacionados às palavras (ver classe **TermoTesauro**) contidas no tesauro;
3. realizar buscas de palavras no tesauro e imprimir o resultado encontrado, por exemplo, ao buscar a palavra *cancelar* deve-se imprimir:

    Cancelar (Verbo):

    (1) Sinônimos: abortar, suprimir.

    (2) Termos contrastantes: não há.

    (3) Termos relacionados: terminar, rescindir, invalidar.

4. eliminar palavras do tesauro.

Alguns detalhes importantes com relação ao tesauro:

- os sinônimos, os termos contrastantes e os termos relacionados associados a uma palavra presente no tesauro podem ser frases;
- os sinônimos, os termos contrastantes e os termos relacionados associados não precisam necessariamente estar presente no tesauro.

## Classes Termo, TermoDicionario e TermoTesauro

Você deve implementar a classe abstrata **Termo** e suas duas classes filhas concretas: **TermoDicionario** e **TermoTesauro**.

- A classe **TermoDicionario** consiste no máximo de 5 frases com significados.
- A classe **TermoTesauro** consiste em (i) 1 frase com sinônimos; (ii) 1 frase com os termos contrastantes e (iii) 1 frase com os termos relacionados.

## Interface TabelaHash

Você deve criar uma interface **TabelaHash** e realizar uma implementação de tabela hash que trata colisões utilizando encadeamento (listas ligadas). As classes **Dicionário** e **Tesauro** devem utilizar essa implementação.

A interface **TabelaHash** possui os seguintes métodos:

- void insere(String palavra, Termo termo)
- void remove(String palavra)
- Termo busca(String palavra)

## Segunda Parte — Interface e Armazenamento

### Interface

Crie, utilizando um método `main`, uma interface textual que permite ao usuário realizar as funções descritas acima para o dicionário e para o tesouro.

### Armazenamento

Crie, a partir da classe `SeriadorDeCatalogo.java` fornecida no EP1, as classes `SeriadorDicionario.java` e `SeriadorTesauro.java` para realizar a serialização de objetos das classes `Dicionario` e `Tesauro`. Utilize as novas classes para carregar o dicionário e o tesouro gerados em execuções anteriores e salvos em arquivo. Lembrar que na primeira execução do programa não existem os arquivos que contêm o dicionário e o tesouro.

## Terceira Parte — Visualização Gráfica (totalmente opcional)

Se você, ao invés de implementar uma interface textual, criar uma interface gráfica, o seu EP3 ganha um ponto adicional. A idéia é a mesma do EP2. Seu programa deverá abrir uma nova janela na qual o usuário poderá fazer inserções e consultas ao dicionário e ao tesouro; o resultado deve ser gerado em outra janela.

**ATENÇÃO:** Para fazer esta terceira parte, você terá que usar várias classes de Java que não são ensinadas em ACH2002. Elas não são simples e você terá que aprendê-las sozinho, por sua conta e risco!

Para criar a interface gráfica, você deverá usar um biblioteca de Java chamada *Swing*. Algumas fontes sobre o *Swing*:

1. livro do Camarão [1]: no Capítulo 11, há uma introdução ao *Swing*;
2. tutorial sobre *Swing* em <http://java.sun.com/docs/books/tutorial/uiswing/learn>.

## Quarta Parte — Dois métodos de hash (totalmente opcional)

Você pode ganhar um outro ponto adicional neste EP3 se você criar uma segunda implementação de tabela hash que trata as colisões utilizando hash duplo. A classe `Dicionário` deve utilizar uma implementação e a classe `Tesauro` a outra, porém, utilizando a mesma interface `TabelaHash`.

## Critérios importantes

### Sobre a elaboração:

- Este exercício-programa deve ser elaborado individualmente.
- Guarde uma cópia do seu programa.

### Sobre a avaliação:

- O EP3 vale até 12 pontos se as partes obrigatórias e as duas partes opcionais forem implementadas; vale até 11 se as partes obrigatórias e apenas uma das partes opcionais for realizada; e vale até 10 se partes obrigatórias forem realizadas.
- Não serão toleradas cópias! Exercícios copiados (com ou sem eventuais disfarces) receberão nota ZERO. O exercício do aluno alvo da cópia também receberá nota ZERO.
- Exercícios com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO. Exercícios sem método `main` para a execução em linha de comando receberão nota ZERO.
- É muito importante que seu programa tenha comentários e esteja bem indentado, ou seja, elaborado de maneira a ressaltar a estrutura de subordinação dos comandos do programa (conforme visto em aula). A qualidade do seu trabalho sob esse ponto de vista influenciará sua nota!
- As informações impressas pelo seu programa na tela devem aparecer da forma mais clara possível. Este aspecto também será levado em consideração no cálculo de sua nota.
- Uma regra básica é a seguinte: do ponto de vista do monitor responsável pela correção dos trabalhos, quanto mais convenientemente apresentado estiver o seu programa, melhor será a disposição dele em atribuir-lhe uma nota generosa.

### Sobre a entrega:

- O prazo de entrega: 2 de dezembro de 2008 às 24h.
- No início do arquivo, acrescente um cabeçalho bem informativo, como o seguinte:

```

/*****
/**  ACH 2002 - Introdução à Computação II                      **/
/**  EACH-USP - Segundo Semestre de 2005                      **/
/**  <turma> - <nome do professor>                             **/
/**                                                                **/
/**  Terceiro Exercício-Programa  --  Dicionário e Tesouro      **/
/**  Arquivo: <nome do arquivo>                                **/
/**                                                                **/
/**  <nome do(a) aluno(a)>                                     <número USP> **/
/**                                                                **/
/**  <data de entrega>                                         **/
*****/
```

Não é obrigatório que o cabeçalho seja idêntico a esse, apenas que contenha pelo menos estas informações.

- A entrega será feita unicamente pelo CoL. Não serão aceitos trabalhos enviados por email. Fiquem atentos, pois agora está ativada no CoL a funcionalidade "horário limite para entrega de trabalhos".
- O seu programa (todas as classes) deve estar contido em um arquivo chamado `DicionarioTesouro<Número USP>.java` para o *upload* no CoL. Por exemplo: `DicionarioTesouro56777333.java`.

## Referências

- [1] Carlos Camarão and Lucília Figueiredo. *Programação de Computadores em Java*. LTC Editora, Rio de Janeiro, 2003.