

<b>Primeiro Trabalho</b>	
MC302EF - Programação Orientada a Objetos Instituto de Computação Universidade Estadual de Campinas	Adivinhando animais André Santanchè 2015.1

## Instruções

---

O trabalho consiste na implementação de parte de um sistema em Java capaz de simular o comportamento do jogo Animals (<http://www.animalgame.com/>). O objetivo central do trabalho é implementar um entrevistador que deverá adivinhar qual animal, dentre um grupo de animais pré-definidos, o entrevistado está “pensando”.

Os trabalhos deverão ser submetidos utilizando o sistema de versionamento Mercurial ao projeto fluid2learn (<http://code.google.com/p/fluid2learn/>) até o dia 09/03/2015.

## Especificações

---

Parte do sistema em Java a ser desenvolvido já está implementado e disponível para download na página do projeto Fluid2Learn (<https://github.com/santanche/fluid2learn>).

### Estrutura base do sistema

Dentro do pacote `pt.c01interfaces.s01knowledge.s01base` é possível encontrar os três pacotes relevantes ao desenvolvimento do projeto apresentados a seguir.

#### Pacote `inter`

Este pacote contém a definição de cinco interfaces utilizadas para a manipulação dos arquivos textos presentes no pacote anterior. A interface `IBaseConhecimento` oferece o método `listaNomes()`, que lista o nome de todos os animais cujos arquivos correspondentes encontram-se no pacote `db`; e o método `recuperaObjeto(String nome)`, responsável por recuperar o arquivo texto de um determinado animal e carregá-lo em um objeto que implementa a interface `IObjetoConhecimento`. Esta última interface, por sua vez, é composta de um conjunto de objetos que implementa a interface `IDeclaracao`, que fornece métodos para leitura de propriedades e valores de cada uma das declarações presentes nos arquivos textos que descrevem os animais.

#### Pacote `impl`

Contém as classes que implementam as três interfaces apresentadas no pacote anterior.

### Entrevistado / entrevistador

Para que o sistema cumpra com seu objetivo, é necessário criar um entrevistado (que será o jogador) e um entrevistador (que será o adivinhador). Para estes dois elementos, também foram definidas duas interfaces disponíveis no pacote `pt.c01interfaces.s01knowledge.s01base`. A interface `IResponder` especifica dois métodos que podem ser acessados publicamente:

1) `public String ask(String question)` → Este método recebe como parâmetro uma string correspondente a alguma propriedade que pode ou não estar presente no arquivo que descreve o animal escolhido pelo entrevistado, e retorna uma string contendo um de três valores possíveis: “sim”, “nao” ou “nao sei”. As duas primeiras respostas são definidas no caso em que a propriedade encontra-se no arquivo do animal em questão. A terceira resposta é utilizada caso contrário.

2) `public boolean finalAnswer(String answer)`; → Este método pode ser invocado apenas uma vez, recebendo como parâmetro uma string identificando o animal que o entrevistador julga ser o animal

escolhido pelo entrevistado. O método retorna um valor booleano indicando o acerto ou erro por parte do entrevistador. Este método só poderá ser acionado uma única vez ao final da entrevista.

A segunda interface disponível é o IEnquirer. O único método disponível nesta interface é:

1) public void connect(IResponder responder); → Este método recebe como parâmetro um objeto que implemente a interface IResponder, e tem como único objetivo conectar o entrevistador ao entrevistado, iniciando a sequência de perguntas que deve levar o entrevistador a identificar qual o animal o entrevistado escolheu. Ao final, este método deve utilizar-se do método finalAnswer disponível em IResponder para verificar o acerto ou o erro por parte do entrevistador.

Há uma implementação de referência do entrevistador/entrevistado em `pt.c01interfaces.s01knowledge.s02app`

O pacote `pt.c01interfaces.s01knowledge.s02app.bd` é o local onde serão armazenados os arquivos em formato texto que contém as informações que descrevem os animais participantes de jogo. Cada arquivo texto contém declarações (especificadas uma por linha) formadas por uma propriedade identificando alguma característica do animal seguida de um valor binário (no nosso caso: sim ou não). Existe um arquivo texto para cada animal participante do jogo. O nome do arquivo texto é formado pelo nome do animal com a extensão `.txt`. Estes arquivos não devem ser lidos diretamente pelo programa a ser desenvolvido, mas indiretamente pela interface `IBaseConhecimento` detalhada a seguir.

No pacote `pt.c01interfaces.s01knowledge.s02app.app` está disponível uma implementação ilustrativa para a instanciação da base de conhecimento. Já no pacote `pt.c01interfaces.s01knowledge.s02app.actors` é possível encontrar duas implementações: uma para o entrevistador e outra para o entrevistado. Estas implementações são ilustrativas, e estão aí para auxiliá-lo na compreensão do trabalho.

## Atividades do trabalho

Para este trabalho, o professor juntamente com o PED da disciplina irão implementar “secretamente” o IResponder. Espera-se que você implemente o IEnquirer:

O Enquirer disponível é uma primeira implementação, onde estamos interessados apenas em obter a resposta correta, não importando o método utilizado.

Este trabalho prevê uma segunda implementação, onde esperamos que você faça uso de estrutura de dados em Java para melhor estruturar o projeto com desempenho satisfatório.

Lembre-se que o IEnquirer pode (e deve) fazer uso das implementações já existentes para manipular os arquivos textos conforme julgar necessário. Pode ser utilizado qualquer abordagem para a escolha das perguntas a serem feitas ao IResponder. A única restrição existente é que não é permitido fazer uma mesma pergunta duas vezes. Vale lembrar que o IResponder não trapaceia: ou seja, ele é sempre sincero ao responder as perguntas recebidas.

A base de dados contendo os animais que serão utilizados na avaliação do trabalho só será revelada depois da data de entrega do trabalho (espera-se que a solução proposta seja genérica o suficiente para funcionar corretamente com qualquer base de dados que obedeça às especificações apresentadas anteriormente). Para facilitar os testes, sugerimos que os alunos enviem suas bases de exemplo para o repositório git online, assim é possível compartilhar a base e melhorar a qualidade dos testes.