



Orientações iniciais:

- A prova é individual e sem consulta;
- A prova vale **100 pontos**, que equivalem à nota **10**;
- Desligue todos os equipamentos eletrônicos (Smartphones, Tablets, Smartwatches, etc...)
- Consulta à Internet é proibida.
- Distribua bem o tempo gasto em cada questão. A quantidade de pontos de cada questão corresponde ao tempo que você deveria gastar em cada uma delas;
- A prova tem duração de mínima de **40 minutos** e duração máxima de **3 horas**;
- Caso você tenha alguma dúvida referente ao enunciado das questões durante a prova, pergunte em voz alta. O professor não irá atendê-lo individualmente.
- Crie uma pasta com o seu nome na sua Área de Trabalho e armazene todas as implementações da prova dentro dessa pasta. Você pode utilizar um editor de texto comum ou o Eclipse. Crie dentro dessa pasta um arquivo chamado *README.txt*, que contém um descritivo de como você organizou os exercícios (1 projeto no eclipse, vários projetos no eclipse, utilizou editor de texto comum, etc.).

BOA PROVA!



Prova 1

Linguagem de Programação Orientada a Objetos — 2017

1. [50 pontos] No ano passado as redes sociais foram tomadas por postagens e comentários envolvendo uma importante figura no cenário do empreendedorismo do Brasil: Bel Pesce.

Bel Pesce é formada pelo MIT, dona de uma empresa de cursos chamada FAZInova, conferencista e escritora. Apesar de seu currículo ser rechado de pontos pomposos, por conta de uma campanha publicitária mal sucedida de *crowdfunding*, alguns curiosos começaram a investigá-la e perceberam diversas incoerências e pontos nebulosos na sua biografia.

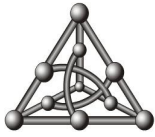
Por conta de casos com o da Bel Pesce, muitos investidores ficaram receosos em investir seu dinheiro em empreendedores com currículos pessoais duvidosos. Para ajudar os investidores a analisar o currículo de empreendedores Brasileiros em mais detalhes, você criará o SIGEP (Sistema de Gestão de Empreendedores), que armazenará informações sobre os principais empreendedores e investidores do Brasil.

O SIGEP deverá armazenar, pelo menos:

- As informações básicas de um Empreendedor: *nome* (com no máximo 30 caracteres), *cpf* (com o formato *YYY.YYY.YYY-YY*) e *endereço* (*estado, cidade, bairro, rua, número e complemento*). O sistema também deve armazenar todas as ideias de negócio que um empreendedor já teve, armazenando para cada ideia de negócio: a *data* em que o empreendedor a externou (no formato *dd/mm/aaaa*), se já foi implantada e um descritivo textual da ideia (com no máximo 1000 caracteres). Além disso, o sistema também deve armazenar uma referência para cada investidor que investiu seu dinheiro no empreendedor em questão, bem como a quantidade de dinheiro investida e a espécie do investimento (dinheiro, ouro, prata ou bronze);
- As informações básicas de um Investidor: *nome* (com no máximo 30 caracteres) e *quantidade de dinheiro* disponível para investimento. Além disso, um investidor deve guardar uma referência para cada empreendedor no qual seu dinheiro foi investido, bem como a quantidade de dinheiro investida.

O SIGEP deverá permitir:

1. Adicionar um empreendedor com suas informações básicas e uma ideia de negócio;
2. Adicionar uma ideia de negócio a um empreendedor já cadastrado;
3. Adicionar um investidor com suas informações básicas;
4. Listar, para cada empreendedor, suas informações básicas, bem como informações sobre suas ideias de negócio e os dados dos investidores que já aplicaram dinheiro nele(a) (se houver);
5. Informar que um investidor (já cadastrado) investiu seu dinheiro em um empreendedor (já cadastrado), armazenar a quantidade investida e a espécie do investimento.



Os seguintes requisitos devem ser obedecidos para a modelagem e implementação do SIGEP:

- (a) Pelo menos as seguintes classes devem ser criadas: Empreendedor, Endereco, IdeiaDeNegocio e Investidor;
- (b) Os conceitos de orientação a objetos e encapsulamento devem ser utilizados massivamente em toda a implementação;
- (c) Um empreendedor pode receber uma quantidade limitada de investimentos;
- (d) Um investidor pode investir em uma quantidade limitada de empreendedores,



2. [50 pontos] Pokemon é uma criatura baseada em animais, que possui poderes especiais e diferentes. Criadas por Satoshi Tajiri e desenhadas por Ken Sugimori, estas criaturas logo se tornaram os preferidos entre o público e a franquia já conta com bilhões de euros em lucros.

O jogo se baseia em uma teoria de que, em tempos remotos, várias criaturas diferentes surgiram no planeta. Estas criaturas desenvolveram-se e logo nasceram várias sub-espécies, com vários tipos, como Normal, Eléctrico, Lutador, etc. Para este exercício, considere o seguinte contexto:

- Todo Pokemon tem um nome, nível de vida (inteiro entre 0 e 100) e força (inteiro entre 0 e 100). Todo Pokemon tem um comportamento chamado `attack`. O método `attack` recebe como parâmetro um outro Pokemon que se deseja atacar, e nenhum outro parâmetro.
- Os Pokemons subdividem-se em Pokemons da Água, Terra, Eletricidade, Fogo e Normal. Existem outros 11 grupos, mas por simplicidade vamos adotar apenas estes cinco.

Abaixo segue uma tabela que indica o que acontece se um Pokemon da classe que está na linha atacar outro que está na coluna.

	Água	Terra	Eletricidade	Normais	Fogo
Água	-15/-10	0/-25	-25/0	0/-10	0/-25
Terra	-25/0	-15/-10	0/-25	0/-10	0/-25
Eletricidade	0/-25	-25/0	-15/10	0/-10	0/-10
Normais	-10/0	-10/0	-10/0	-15/10	-5/-10
Fogo	-25/0	-25/0	-10/0	-10/-5	-15/-10

Tabela 1: Tabela de dano ao atacar oponentes.

Os valores separados por uma `'/'` significam, respectivamente: quanto de dano (diminuição da vida) o Pokemon da linha vai receber se atacar um Pokemon da coluna, e quanto o Pokemon da coluna receberá de dano caso for atacado por um Pokemon da linha. Exemplo: Se um Pokemon da água atacar um Pokemon da Terra ele não recebe nenhum dano, mas causará -25 pontos de dano no oponente que é da Terra.

Observações:

- Implemente uma classe **PokemonTest.java** para realizar os testes necessários.
- Sua solução deve ser desenvolvida, obrigatoriamente, utilizando herança e os conceitos relacionados vistos em sala de aula.