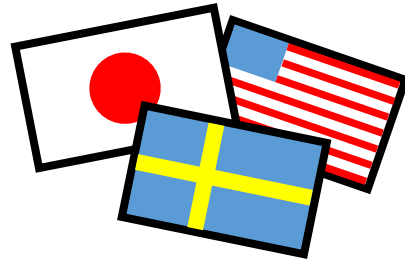


**Laboratório de Programação 2 - 2015.1**  
**Turma 3: Francisco Oliveira Neto - LCC1**

**Prova 2 - 22/06/2015**

> HelloWorld!\_



Você foi contratada(o) para implementar o sistema da **escola de Idiomas HelloWorld!** Além do típico e importantíssimo curso de Inglês, a escola também oferece cursos de Sueco e Japonês. Cada curso fornece uma pontuação que reflete a facilidade que a aluna(o) terá em aprender o idioma. Para isso, são usadas as habilidades de proficiência de escuta, fala, leitura e escrita.

- a) Crie uma entidade para Aluno que possua um nome e uma mensalidade.
- b) Cada aluno possui uma Proficiência que é representada por quatro valores reais **entre 0 e 10**. Esses valores representam as habilidades dos alunos em **escuta, fala, leitura e escrita**. Por exemplo, um aluno com Proficiência: 7, 8.5, 10, 4.5 possui, respectivamente, uma pontuação 7 em escuta, 8.5 em fala, 10 em leitura e 4.5 em escrita. A proficiência é definida na *criação* da aluna(o), porém, cada habilidade pode ser alterada posteriormente.
- c) Permita que cada aluno possua apenas um curso de Idiomas que esteja matriculado. Essa matrícula deve ser feita pelo método `matricula("Curso")` onde a palavra **"Curso"** será: **Sueco, Inglês ou Japonês**. Ao ser criado, uma aluna(o) não está matriculada(o) em nenhum curso. Não é possível se matricular em outro curso que não seja esses três.
- d) Uma das vantagens do HelloWorld é fornecer à estudante uma pontuação relacionada à facilidade que ela terá ao estudar um idioma específico, a partir das suas capacitações de proficiência. **Essa pontuação será um valor entre 0 e 10 calculado de formas diferentes de acordo com o Curso** em que a aluna(o) está matriculada(o). Todos os cursos possuem duas formas para calcular essa pontuação. A primeira é igual para todos os cursos e corresponde à média aritmética dos pontos de proficiência da aluna(o).

```
facilidadeGenerica = (escuta + leitura + escrita + fala) / 4.0;
```

No entanto, cada idioma possui dificuldades específicas. Por exemplo, Japonês é um idioma cuja leitura é mais difícil que o Sueco. Abaixo seguem as formas específicas de calcular a facilidade:

```
japones=(0.4*leitura)+ (0.6*escrita);  
inglês =(0.1*leitura)+ (0.5*escrita)+ (0.4*fala);  
sueco =(0.1*leitura)+ (0.1*escrita)+ (0.3*fala) + (0.5*escuta).
```

Dessa forma, ao matricular a aluna(o) em cada curso podemos, a partir de então, obter a respectiva pontuação de acordo com a proficiência da aluna(o). Permitindo que uma aluna(o) possa trocar seu curso matriculado, atingimos um comportamento dinâmico na pontuação de proficiência.

- e) A primeira lição de cada curso são as saudações. Cada aluna(o) deve saber se apresentar de acordo com o idioma que está estudando. Então, a entidade Aluno deve possuir um método `cumprimenta()` que retorna uma saudação diferente. Para cada curso, temos:

**Inglês:** *"Hello! I am <nomeAluno>. Nice to meet you. "*

**Sueco:** *"Hej hej! Jag heter <nomeAluno>. Trevligt att träffas. "*

**Japonês:** *"Hajimemashite. Watashi ha <nomeAluno> desu. "*

- f) Escreva seu código de forma que facilite para outro programador escrever a classe que armazena e ordena a lista de Alunos. Para isso, permita que os alunos sejam comparáveis por meio do seu respectivo **nome**.

#### **Opcional (0,5):**

Indique como **comentário no seu código** quais as composições que estão sendo utilizadas. Para isso indique qual o seu **wrapper**, e quais os **objetos compostos**. Indique também, quais desses objetos possuem **polimorfismo**, e onde estão as respectivas **chamadas polimórficas e trocas de comportamento**. (0.5)

**Seu código será analisado de acordo com a sua funcionalidade e seu design.** Diante disso, é altamente recomendável ficar atento aos padrões GRASP e KISS para obter um bom código como resultado. Utilize a classe de testes fornecida para ajudar com a implementação. **Faça o tratamento de Exception** para dados de proficiência inválidos e para alunos que usam um curso porém não forma ainda matriculados.

**Extra (1,0):** Mais testes, Hierarquia de Exceptions, Javadoc, Factory de Cursos

---

Link para o Arquivo de Teste: <https://goo.gl/pZ6z5H>

#### **Instruções da Entrega:**

**Nomeie seu projeto** de acordo com o formato: **Nome\_Sobrenome\_Matricula**

A entrega será para o e-mail: [juiz.lp2@gmail.com](mailto:juiz.lp2@gmail.com)

Envie um zip do seu projeto. Cuidado para empacotar a pasta correta. Confira.

**Formato:** MATRICULA\_NOME\_SOBRENOME\_TURMA\_3\_PROVA\_2.zip

No assunto do e-mail coloque: **[Prova 2 - 2015.1] Nome Completo**

**Boa prova!**