

Trabalho de LPOO: NP2 (grupo máximo de 6 pessoas)

Neste trabalho o grupo deve fazer um programa de acesso aos dados de uma faculdade, com informações sobre cadastro de alunos e cursos; e notas dos alunos.

Dados

o seu programa deve processar dados relativos a alunos, cursos, notas e rendimentos da faculdade.

- Um `Aluno` tem:
 - um `id`, do tipo `String`;
 - um `nome` do tipo `String`
 - Cada aluno tem um `id` único e pode ser identificado exclusivamente por seu `id`; dois alunos podem ter o mesmo nome, mas nunca o mesmo `id`
- Um `Curso` tem:
 - um `nome`, do tipo `String`;
 - um `nivel` de dois possíveis, podendo ser de "graduacao", ou "pos_graduacao";
 - um `ano`, do tipo inteiro, representando o ano que o curso foi dado;
 - Cada curso deve ser identificado pela combinação de seus três campos; dois cursos são iguais se, e somente se, seus três campos forem iguais.
- Uma relação `Aluno` e `Curso` chamada de `Rendimento`:
 - Um `Rendimento` representa a qualidade das notas que *um* `Aluno` obteve em *um* `Curso`;
 - um `rendimento` está atrelado à *um* `aluno` e *um* `curso`
 - um `rendimento` tem:
 - Contêm 4 valores do tipo `double`: a nota da NP1, da NP2, da reposição, e do exame.

Relatórios

O seu programa deve gerar alguns relatórios baseados nas seguintes regras:

- Cada `Rendimento` tem os valores associados
 - um valor `aprovado` do tipo `boolean`
 - um valor `media`, do tipo `double`, associada às notas do curso, esta média deve ser calculada da seguinte maneira:
 - deve ser calculada uma média inicial (variável local) da seguinte maneira: caso a nota da reposição seja maior que a nota da NP1 ou da NP2, a menor nota, entre a NP1 e a NP2, deve ser substituída pela nota da reposição, (esta substituição deve ser feita apenas no cálculo da média, a nota em da NP1 ou da NP2 não devem mudar de valor.) O valor da média entre a NP1 e NP2, depois de ter sido feita a substituição, será a média inicial
 - caso o curso seja um curso de graduação:
 - se a média inicial for maior ou igual a 7.0, então a média se mantém no valor da média inicial e o valor de aprovado é verdadeiro.

- se a média inicial **não** for maior que 7.0, então a média final será calculada a partir da média entre a nota do exame e a média inicial. A média terá o valor da média final e se o seu valor for maior ou igual a 5.0, o valor de aprovado será verdadeiro, caso esta média final seja menor que 5.0 o valor de aprovado é falso
- caso o curso seja um curso de pós graduação:
 - se a média inicial for maior ou igual a 5.0, então a média se mantém no valor da média inicial e o valor de aprovado é verdadeiro.
 - se a média inicial **não** for maior que 5.0, então a média final será calculada a partir da média entre a nota do exame e a média inicial. Se o seu valor for maior ou igual a 5.0, o valor de aprovado será verdadeiro, e a média terá o valor de 5.0. Caso esta média final seja menor que 5.0 o valor de aprovado é falso e a média terá o valor da média final.

Estrutura dos Dados Persistentes

Os dados persistentes deverão ser lidos e salvos em um arquivo `csv`. (2pts)

Serão 3 tipos de arquivos

- Um arquivo CSV com os dados dos alunos, na ordem "id;nome"

Exemplo:

alunos.csv

```
123;Ana
124;Beto
125;Carlos Magno
b26;Danilo E.
127;Fernando
128;Gabriel M.
129;Hugo S.
```

- Um arquivo CSV com os dados dos cursos na ordem "nome;nível;ano"

Exemplo:

cursos.csv

```
LP00;GRADUACAO;2018
ALP00;GRADUACAO;2018
ALP00;GRADUACAO;2019
ALP00;POS_GRADUACAO;2019
```

- Para cada curso, um arquivo com o nome "nome_nível_ano.csv", com as informações das notas dos alunos naquele curso, na ordem "id_do_aluno;nota_NP1;nota_NP2;nota_reposicao;nota_exame". Havendo a possibilidade de o arquivo não ter nenhum dado, representando o fato de que ninguém cursou aquela matéria naquele ano.

Exemplos:

ALPOO_GRADUACAO_2018.csv

```
123;8.0;7.0;0.0;0.0
124;3.0;4.0;5.0;2.0
125;2.0;4.0;1.0;9.0
```

ALPOO_GRADUACAO_2019.csv

ALPOO_POS_GRADUACAO_2019.csv

```
123;7.0;9.0;0.0;0.0
124;2.0;3.0;4.0;2.0
125;2.0;4.0;5.0;9.0
b26;1.0;5.0;5.0;0.0
```

LPOO_GRADUACAO_2019.csv

```
129;10.0;0.0;2.0;0.0
```

Sistema

O seu sistema deve acessar e manipular os dados persistentes.

De modo que, ao iniciar o programa, o usuário final tenha acesso aos dados persistentes.

Ao terminar o programa, qualquer adição que o usuário tenha feito esteja representada pelos dados persistentes no mesmo formato descrito.

Neste sistema deve ser possível que um usuário final:

- Cadastre alunos (1pt)
- Listar alunos (1pt)
- Cadastre as matérias(1pt).
- Listar todas as matérias cadastradas (1pt).
- Listar matérias de um ano específico (1pt).
- Escolher um aluno (pelo RA) e mostrar o relatório do aluno com:(1pt)
 - Todas as matérias cursadas por este aluno com as notas e o status (se passou ou não)
 - Mostrar a média do histórico do aluno.
- Escolher uma matéria (pelo nome, ano e nível) e mostrar o relatório da matéria com:(1pt)
 - Cada aluno que cursou a matéria, com a sua média e status (se passou ou não)
 - A média das médias da matéria.
- Escolher um Aluno e um Curso e cadastrar um Rendimento .

Interface com o Usuário Final

O seu programa deve se comunicar com o usuário final através do terminal, o usuário deve poder executar todos os comandos listados em "Sistemas".

Pontos Adicionais

- Estruturas de Dados Interna: Caso o seu programa use estruturas de dados como `Map` e/ou `Set` para organizar os dados internos: (+1pt)
- Herança e polimorfismo: Caso seu programa use herança para organizar os diferentes tipos de `Rendimento` (cálculo da média final de `GRADUACAO` e `POS_GRADUACAO`). Por exemplo: Faça com que `Rendimento` seja uma classe abstrata que tenha duas subclasses que tenham override do cálculo da média. (+1pt)
- Controle de Erros: Caso o seu programa tenha controle de erros usando `try-catch` da entrada do usuário final (+1pt)