

GCC178 – Práticas de Programação Orientada a Objetos

PRÁTICA SOBRE PROGRAMAÇÃO GENÉRICA

Um professor deseja implementar um sistema de geração de diários de classe para suas disciplinas. Para auxiliar esse professor nessa tarefa, seguindo as boas práticas em programação orientada a objetos, implemente em Java a parte desse sistema responsável por gerar o diário de classe (em ordem decrescente da nota de prova obtida pelos alunos) de uma disciplina ministrada pelo professor, conforme mostra o exemplo a seguir:

Nome	Nota Prova	Faltas	Situacao
Jose	85	12	Aprovado
Regina	75	15	Aprovado
Pedro	45	10	Reprovado

Esse sistema engloba dois conceitos principais: **diario** e **aluno**. Um **diário** contém somente a **lista de alunos** que estão matriculados em uma disciplina. Já os **alunos** podem ser da graduação ou da pós-graduação. Apesar de cursarem a mesma disciplina, os critérios para aprovação na disciplina são diferentes para alunos de graduação e pós-graduação. Todos os alunos são caracterizados apenas pelo seu **nome** (String), **nota obtida na prova** (inteiro no intervalo [0,100]) e **quantidade de faltas** (int). Enquanto os alunos da graduação também possuem a **nota do trabalho prático** realizado (inteiro no intervalo [0,100]), os alunos da pós-graduação são caracterizados pela **quantidade de seminários** apresentados na disciplina (int). Desse modo, para que um aluno da graduação seja aprovado na disciplina ele precisa obter nota média (média aritmética simples das notas da prova e trabalho prático) maior ou igual a 60 e quantidade de faltas inferior a 19. Já para um aluno da pós-graduação obter aprovação na disciplina, ele precisa ter apresentado pelo menos dois seminários, ter obtido nota na prova maior ou igual a 70 e ter no máximo 15 faltas.

Vale observar que a classe *Principal* apresentada a seguir, que contém o método *main* e é responsável por instanciar os objetos e chamar os métodos necessários para geração do **diário** do exemplo dado anteriormente, não pode sofrer qualquer alteração no código já disponível.

Os seguintes requisitos também devem ser atendidos para essa implementação:

- Apresente uma solução funcione para qualquer quantidade de alunos instanciados e passados por parâmetro na chamada do método *adicionarAluno* na função *main*, ou seja, nenhuma outra alteração no código deverá ser necessária para se ajustar a diferentes quantidades de alunos.
- Somente a classe *Diario* poderá utilizar algum código para impressão de dados na tela.
- Faça uso do método *toString* para retornar o(s) valor(es) do(s) atributo(s) dos objetos para os quais se deseja obter o seu estado.
- Faça uso da classe *Collections* para realizar a ordenação dos dados.

```
public class Principal {  
    public static void main(String[] args){  
        Diario diario = new Diario();  
        Aluno a;  
        a = new AlunoGrad(90, "Jose", 85, 12);  
        diario.adicionarAluno(a);  
        a = new AlunoGrad(65, "Pedro", 45, 10);  
        diario.adicionarAluno(a);  
        a = new AlunoPos(3, "Regina", 75, 15);  
        diario.adicionarAluno(a);  
        diario.gerarDiarioTurma();  
    }  
}
```