

1ª Prova de POO

Elabore os códigos referentes às questões abaixo e envie os fontes Java, descompactados, via e-mail, para sbacala@gmail.com, com o assunto **“Prova 1 de POO – 2009-2 nome do aluno”**. Não se esqueça de incluir seu nome no interior do e-mail. Entregas diferentes dessa definição serão desconsideradas. Inclua em cada arquivo java um comentário com o nome do autor.

1 – (1,0) Elabore uma classe Disciplina contendo os atributos “int codigo” e “String nome”, e os métodos de acesso públicos getCodigo(), getNome() e setNome(String). O valor do codigo é autoincremental definido no construtor padrão.

2 – (2,0) Elabore uma classe Aluno contendo os atributos “int matricula” e “String nome”, e os métodos de acesso públicos getMatricula() e getNome(). Crie também o método setNome(String) que deve ser privado e só deve atualizar o nome se a String for diferente de null e com tamanho maior que zero. Caso contrário, deve levantar uma exceção. O construtor deve receber nome como parâmetro e chamar o setNome() para atualizar o nome.

3 – (2,0) Criar as exceções “AlunoMatriculadoException” e “AlunoInexistenteException” para exibir as mensagens “Aluno já matriculado!” e “Aluno não encontrado!”, respectivamente.

4 – (3,0) Adicionar, como atributo na classe Disciplina, um array para conter a lista de alunos matriculados.

- a) Criar o método matricular (Aluno aluno) para adicionar um aluno na lista de matriculados. Caso já exista um aluno com mesma matricula na lista, levantar uma exceção AlunoMatriculadoException.
- b) Criar o método cancelarMatricula (int matricula) para remover um aluno da lista de matriculados. Caso não exista um aluno com a matricula na lista, levantar uma exceção AlunoInexistenteException.

5 – (1,0) Sobrecarregue o construtor para que receba o parâmetro nome da disciplina.

5 – (2,0) Adicionar na classe Aluno array de Disciplina como atributo disciplinaMatriculadas. Inclua na classe os métodos adicionarDisciplina(Disciplina) e removerDisciplina(Disciplina) para serem chamados pela classe Disciplina nos métodos criados na questão 4. Em caso de erros, levantar uma exceção com a mensagem “Dados inconsistentes!”

6 – (1,0) Sobrepor o método toString() na classe Aluno para retornar uma String com o seguinte formato: “Aluno – 99999 : nome-do aluno”, onde 99999 é a matrícula do aluno.

7 – (2,0) Crie o método toString() na classe Disciplina para retornar uma String com os dados das disciplinas com a relação dos alunos matriculados usando o toString() da classe Aluno. Utilize o StringBuffer e inclua “\n” para controlar o salto de linhas.

8 – (6,0) Faça um método main() em qualquer classe, para executar um procedimentos de teste:

- Crie três disciplinas
- Crie cinco alunos
- Matricule os alunos nas disciplinas
- Matricule novamente um aluno para testar a exceção
- Exiba as disciplinas
- Cancele a matricula de um aluno
- Cancele a matricula de um aluno que não esteja matriculado na disciplina
- Exibir novamente as disciplinas após o cancelamento.

Copie a saída do processamento em um arquivo “saida.txt” e encaminhe juntamente com as classes Java no e-mail sbacala@gmail.com.