



Laboratório 04 – Relacionamentos

Atividade individual.

Crie um novo projeto no NetBeans para a parte de implementação (código) de cada exercício a seguir. Caso necessário, peça orientação ao professor e/ou consulte os materiais complementares listados ao final deste documento.

1. Construir um sistema de controle de cursos de informática equivalente ao módulo de matrícula segundo os seguintes requisitos:
 - O curso pode ter mais de uma turma, no entanto, uma turma se relaciona exclusivamente com um único curso.
 - Uma turma pode ter diversos alunos matriculados. No entanto, uma matrícula refere-se exclusivamente a uma determinada turma. Cada turma tem um número mínimo de 5 matrículas para iniciar o curso.
 - Um aluno pode realizar muitas matrículas, mas cada matrícula refere-se exclusivamente a uma turma específica e a um único aluno.
 - (a) **Modelagem do problema:** Fazer o diagrama de classes desse sistema.
 - (b) **Código** (NetBeans)
2. Considere que um vetor v contém o número de notas dos alunos de POO agrupadas em categorias, ou seja: $v[0]$ contém o número de notas de 0.0 a 9.9, $v[1]$ contém o número de notas de 10.0 a 19.9, $v[2]$ contém o número de notas de 20.0 a 29.9, e assim por diante, até $v[9]$ – que contém o número de notas entre 90.0 e 100.0. Montar uma classe com um método para mostrar a distribuição de notas por meio de um gráfico de barra de asteriscos (*). A classe principal deverá ler as notas de n alunos (n é fornecido pelo usuário) e distribuí-las nas categorias correspondentes. Em seguida, mostre o gráfico de barras correspondente.

- (a) **Modelagem do problema:** Fazer o diagrama de classes desse sistema.
 - (b) **Código** (NetBeans)
3. Considere uma classe `Aluno` com os atributos `nome`, `nota` e `idade` e uma classe `Turma`. `Turma` deverá manipular um vetor de objetos da classe `Aluno` (uma turma contém vários alunos). Faça métodos para: incluir e listar alunos; e procurar alunos por nome. Fazer um menu de opções com `switch-case` em uma classe principal onde o usuário pode escolher cadastrar alunos, procurá-los ou listá-los.

Exemplo de `switch-case`, para referência:

```
1 teste = entrada.nextInt();
2 switch(teste){
3     case 1:
4         System.out.println("Janeiro");
5         break;
6     case 2:
7         System.out.println("Fevereiro");
8         break;
9     case 3:
10        System.out.println("Março");
11        break;
12    case 4:
13        System.out.println("Abril");
14        break;
15    default:
16        System.out.println("Digite SOMENTE números
17        entre 1 e 4");
17 }
```

- (a) **Modelagem do problema:** Fazer o diagrama de classes desse sistema.
 - (b) **Código** (NetBeans)
4. Construir um programa que simula o lançamento de dois dados. O programa deve usar um objeto da classe `Random` (da Java API) para realizar tal lançamento. Como cada dado pode mostrar um valor de 1

a 6, a soma dos dois dados varia de 2 a 12. O programa deve lançar os mesmos 30000 vezes. Utilizar um vetor para guardar o número de vezes que cada possível valor da soma ocorreu. Mostrar o valor da soma mais e da menos frequente. Mais detalhes sobre Random: <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>.

Materias complementares (Java)

- <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html> (em inglês)
- <https://www.caelum.com.br/apostila/apostila-java-orientacao-objetos.pdf> (em português)