

USJT – 2019/1 - Programação Orientada a Objetos - Laboratório Dojo

Professores: Machion e Mario

**Aula: 01**

Assunto: Dojo, Estrutura básica de um programa Java, Entrada e Saída, Variáveis, Atribuição, Conversão de Strings

**Dojo - Conceitos básicos**

**Definição:** a palavra Dojo (pronuncia-se Dojô), em japonês, significa "local de treinamento". O termo, amplamente utilizado em artes marciais como o Kendo e o Karatê, vem também sendo usado na prática de programação voltada à aprendizagem, chamado de *Coding Dojo*.

Existem alguns formatos do Dojo, como o *Kata* e o *Kake*. No nosso curso, usaremos o *Randori* e o *Kata*, os quais explico abaixo.

**Randori**

- 1) Todos os alunos participam. É proposto um ou mais problemas para serem resolvidos e a programação é realizada por pares em apenas uma máquina (as outras ficam preferencialmente desligadas), projetando a tela para que todos vejam.
- 2) A solução deve ser gradual, obtida passo a passo, sendo explicada pelo par que está programando e deve ser entendida por todos. Nos métodos ágeis, isto é conhecido por *baby steps*.
- 3) A pessoa que está no teclado é o piloto e, seu par, o co-piloto. A cada cinco minutos o piloto volta para a plateia e o co-piloto assume a condição de piloto. Uma pessoa da plateia passa a assumir a posição de co-piloto. Caso a plateia seja muito grande, ambos podem ser substituídos por outra dupla.
- 4) O professor é o *sensei*. Seu papel é o de moderar o encontro, propor o problema, ajudar a dupla e a plateia encontrarem a solução, manter a ordem e a dinâmica do Dojo e avaliar a participação dos alunos.
- 5) A plateia deve prestar atenção atentamente na solução e se manter em silêncio. A plateia pode ajudar na solução caso a dupla que está no teclado esteja com dificuldades, mas isso deve ocorrer sempre mediante o estímulo do professor.

6) Ao final o professor deve fazer uma retrospectiva do Dojo, comentando as soluções obtidas, complementando o que foi ensinado e dando um feedback para a turma sobre a participação dela no Dojo.

O objetivo é fazer com que todos escrevam código, peguem gosto pela programação e aprendam a trabalhar coletivamente. O aluno ficará exposto e não poderá se omitir nem guardar as dúvidas para ele.

### **Kata**

1) O professor (*sensei*) inicia o Dojo apresentando a solução de um problema para os alunos, resolvendo-o no JGrasp e projetando o código na tela.

2) Os alunos, com os computadores ligados, vão copiando o código e entendendo a solução proposta.

O objetivo é preparar o aluno para a sessão de *Randori* que virá em seguida.

### **Dojo - Regras:**

**NOTA: APESAR DAS REGRAS, O DOJO DEVE SER UM AMBIENTE DESCONTRAÍDO E ESTIMULANTE, NO QUAL O ALUNO SE SINTA IMPELIDO A PARTICIPAR NÃO PELOS PONTOS, MAS SIM PELO GOSTO DE PROGRAMAR E DE ENCONTRAR UMA SOLUÇÃO DE FORMA COLABORATIVA.**

1. Todas as máquinas do computador ficam preferencialmente desligadas, com exceção do computador que está sendo usado pelo piloto.

2. Todos os alunos da turma, se possível, devem se revezar no teclado a cada sessão de Dojo. Os alunos que não teclarem em uma aula tem preferência na seguinte.

3. O professor deve dar uma nota de 0,0 a 20,0 para cada aluno, em incrementos de 5,0, em cada sessão de Dojo. A nota final do laboratório será a média de todos os Dojos.

4. Todo aluno entra na sala com 20,0 pontos, que serão descontados segundo o seguinte critério de avaliação:

- **pontualidade:** tolerância máxima de 20 minutos de atraso (penalidade de 5,0 pontos); mais de 40 minutos de atraso, não entra no laboratório (penalidade de 20,0 pontos).
- **atenção quando na plateia:** estar atento, ajudar quando solicitado, ficar em silêncio, não usar o celular, não ligar o computador, não conversar com o colega. (penalidade de 5,0 pontos).
- **desempenho como co-piloto:** ajudar o piloto na construção da solução (penalidade de 5,0 pontos).

- **desempenho como piloto:** construir efetivamente a solução; a tolerância do professor com alunos que não tenham ideia do que fazer deve ser mais alta no começo e ir diminuindo conforme as aulas vão avançando (penalidade de 5,0 pontos).

5. Fica a critério do professor excluir um aluno do Dojo por mau comportamento a qualquer momento e retirar dele todos os pontos daquele Dojo.

### Problemas Propostos:

- todos os problemas podem ser resolvidos diretamente no método main.
- faça uma classe para cada solução.
- a nome da classe pode ser Solucao1a, Solucao1b, e assim por diante.
- não use `Scanner` para ler dados.
- não se preocupe em explicar o `import`, o `static`, o `String[]`; isso será feito adiante.
- os alunos podem consultar qualquer material (menos o Google).

### Kata

- 1) Apresente o JGrasp para os alunos e mostre suas principais funcionalidades.
- 2) Crie um algoritmo bem simples, que escreva o nome da disciplina na tela.
  - a. Comece escrevendo usando um `System.out.println` e um literal de String.
  - b. Faça outra solução, agora atribuindo o literal a uma variável String e imprimindo a variável usando o `System.out.println`.
  - c. Faça outra, agora lendo uma entrada via `JOptionPane.showInputDialog`, atribuindo à variável e imprimindo ainda com o `System.out.println`.
  - d. Faça outra análoga à anterior, mas desta vez imprima o resultado via `JOptionPane.showMessageDialog`.

### Randori

- 3) Crie um algoritmo para ler 2 strings, concatená-las, atribuindo a uma terceira variável e escrever o resultado na tela.
- 4) Crie um algoritmo para ler um número inteiro, um número real, uma letra (caractere) e um booleano e imprimir todos na tela. Os valores lidos devem ser atribuídos a variáveis do tipo correspondente (inteiro para `int`, real para `double`, etc.). Faça em dois passos, primeiro lendo para uma variável do tipo `String` e depois convertendo a `String` para o tipo correto.

## **Bibliografia**

BONFIM, MÁRCIO (ex aluno USJT); **O que é Coding Dojo**; disponível em <http://www.devmedia.com.br/o-que-e-o-coding-doj/30517>. Acessado em 16/08/2015.

LOPES, ANITA. GARCIA, GUTO. Introdução à Programação: 500 algoritmos resolvidos. Rio de Janeiro: Elsevier, 2002.

DEITEL, P. DEITEL, H. Java: como programar. 8 Ed. São Paulo: Prentice – Hall (Pearson), 2010.