

## Lista 10

1. Crie uma classe **Carro** contendo os atributos privados **marca**, **modelo** e **placa**. Logo após, leia do teclado os dados necessários para criar 3 carros e insira estes carros em array chamado **frota**. Finalmente, usando o array, imprima os dados sobre os carros.
2. Crie uma classe chamada **Carta** que deve conter dois atributos privados: naipe e número. Para esta classe, adicione um método construtor e os métodos get e set.

Crie uma segunda classe chamada **Exercicio2** contendo um método main.

Inicialmente, o método main deve criar um baralho contendo 52 cartas: 13 cartas de cada naipe.

Cada um dos 52 objetos **Carta** será inserido em uma posição de um array chamado baralho.

Você pode criar o array baralho com uma ou duas dimensões (você escolhe).

Finalmente, embaralhe as cartas contidas no array, e imprima os dados das cartas no baralho.

A formatação não precisa ser idêntica ao exemplo abaixo.

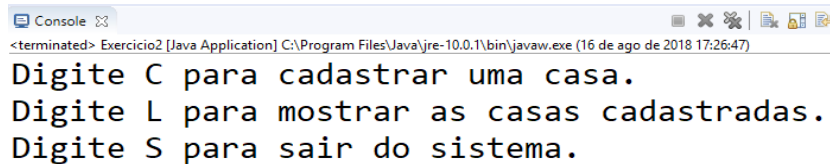
|                  |                 |                 |                  |
|------------------|-----------------|-----------------|------------------|
| Ouros : As       | Ouros : 2       | Ouros : 3       | Ouros : 4        |
| Ouros : 5        | Ouros : 6       | Ouros : 7       | Ouros : 8        |
| Ouros : 9        | Ouros : 10      | Ouros : Valete  | Ouros : Rainha   |
| Ouros : Rei      | Espadas : As    | Espadas : 2     | Espadas : 3      |
| Espadas : 4      | Espadas : 5     | Espadas : 6     | Espadas : 7      |
| Espadas : 8      | Espadas : 9     | Espadas : 10    | Espadas : Valete |
| Espadas : Rainha | Espadas : Rei   | Copas : As      | Copas : 2        |
| Copas : 3        | Copas : 4       | Copas : 5       | Copas : 6        |
| Copas : 7        | Copas : 8       | Copas : 9       | Copas : 10       |
| Copas : Valete   | Copas : Rainha  | Copas : Rei     | Bastos : As      |
| Bastos : 2       | Bastos : 3      | Bastos : 4      | Bastos : 5       |
| Bastos : 6       | Bastos : 7      | Bastos : 8      | Bastos : 9       |
| Bastos : 10      | Bastos : Valete | Bastos : Rainha | Bastos : Rei     |

3. **Pesquise**, no contexto da orientação a objetos, o que são:
  - a) **associação**
  - b) **composição**
  - c) **agregação**

4. Crie uma classe para representar uma **Casa** contendo pelo menos **3 atributos privados**.

Construa um arquivo chamado **Exercicio4.java** contendo um método **main** onde será criado um array de objetos da classe **Casa** representando um condomínio com 3 casas.

O seguinte menu deve ser criado:



```
<terminated> Exercicio2 [Java Application] C:\Program Files\Java\jre-10.0.1\bin\javaw.exe (16 de ago de 2018 17:26:47)
Digite C para cadastrar uma casa.
Digite L para mostrar as casas cadastradas.
Digite S para sair do sistema.
```

Opção:

- a) Se digitar **C** será cadastrada uma **Casa** na próxima posição livre do array.  
Os dados da nova casa são informados pelo usuário.  
Caso o array esteja completo, apenas informe que o condomínio está completo.  
Logo após, o menu deve ser exibido novamente.
  - b) Se digitar **L** serão listados todos os objetos **Casa** através da impressão dos seus conteúdos.  
Caso o condomínio esteja incompleto, imprima os dados **apenas** das casas existentes.  
Logo após, o menu deve ser exibido novamente.
  - c) Se digitar **S** o programa deve ser encerrado.
5. Tendo como base a classe **Matematica** nos slides da aula, crie os seguintes **métodos estáticos** para realizar cálculos:
- a) **subtração**:  $A-B$  se  $A \geq B$ , ou  $B-A$  se  $B > A$ .
  - b) **média aritmética** dos elementos de um array contendo números double.  
O retorno do método deve ser um número double.
  - c) Cálculo do n-ésimo termo  $a_n$  de uma **progressão aritmética** (PA) com os parâmetros:
    - p – primeiro termo da PA, chamado  $a_1$
    - r – razão da PA
    - n – índice do n-ésimo termo

O valor a ser retornado é o n-ésimo termo  $a_n$ .