

- I. Implemente com o uso adequado de threads o seguinte problema abaixo.

Simulador de Aeroporto

Crie a classe `Aviao` para a simulação de voos em um aeroporto com as seguintes características

- Deverá receber no construtor o nome do voo e o objeto da classe `Aeroporto` (será explicado adiante).
- Deverá possuir os métodos `decolar`, `voar` e `aterriar`.
- No método `voar`, coloque uma espera de 2 segundos.
- No método `decolar` e `aterriar` deverá verificar se na classe `Aeroporto` possui pista disponível por meio do método `aguardarPistaDisponivel`.

Crie a classe `Aeroporto` para simulação de um aeroporto com as seguintes características.

- Deverá receber no construtor o nome do aeroporto.
- Deverá ter um método chamado `aguardarPistaDisponivel` onde verificará a disponibilidade da pista. Caso a pista esteja indisponível, deverá aguardar até ficar disponível.
- Deverá ter um método chamado `alterarEstadoPista` que será responsável por alterar o estado da pista (verdadeiro/falso). A cada atualização do estado da pista deverá notificar as threads desta condição para continuidade das operações (aterriagem e decolagem).
- O estado da pista deverá ser alterado a cada 3 segundos.

As classes `Aeroporto` e `Aviao` deverão herdar a classe `Thread`.

Crie uma classe `Main` para simular o comportamento das *threads*. Para isso, crie 3 objetos da classe `Aviao` para simular os voos: “TAN 123”, “OLÉ 111” e “LINHA 222” e analise o comportamento deles no aeroporto de acordo com o estado da pista.

- II. Crie uma classe chamada `Funcionario`. Nesta classe inclua atributo de nome (*String*), idade (*int*), habilidades (*String*) e nome dos filhos (*String*). Nos atributos onde há mais que um valor (habilidades e nome dos filhos) utilize o *Varargs*. Depois crie um *ArrayList* para incluir os seguintes funcionários da tabela abaixo. No *ArrayList* utilize adequadamente a funcionalidade *Generics*.

Por fim, imprima em tela seguindo o layout abaixo, utilizado o novo laço `for` (Java 5).

```

----- Nome Funcionário -----
                Marcos Alves
----- Idade -----
                    33
----- Filhos -----
                Luis Henrique
----- Habilidades -----
                    Java
                    Visual Basic
                Java Server Faces
----- Total Filhos -----

```

1
----- Total Habilidades -----
3

Nome	Idade	Habilidades	Filhos
Marcos Alves	33	Java	Luis Henrique
		Visual Basic	
		Java Server Faces	
Maria Clara	22	Javascript	Marcos Pontes
		SQL Server	Marcelo Pontos
Henrique Pedro	55	Java HTML5	José Pedro
			Maria Pedro
			Neli Pedro

III. Crie uma classe *enum* chamada Cores que informe ao sistema o código de determinadas cores que serão mostrados a seguir. Esta classe deve possuir:

Atributos

- Nome da Cor
- Código da Cor

Métodos

- Recuperar Nome da Cor
- Recuperar Código da Cor
- Construtor deve receber como parâmetro o código e o nome da Cor

Depois de criada a classe *enum* Cores, crie uma classe Main para mostrar em tela todas as cores, informando seu nome e seu código. Por fim, mostre em tela especificamente o código da cor Branca.

Cor	Código
Branco	21
Preto	22
Vermelho	23
Amarelo	24
Azul	25