
A yellow L-shaped line is positioned in the top-left corner of the slide.

# CC3642

# Orientação a Objetos

A short blue horizontal line is centered below the title.

Prof. Danilo H. Perico

A yellow L-shaped line is positioned in the bottom-right corner of the slide.

# Arrays, ArrayLists e Documentação

---

# Exercício 1 - Classe Estudante

- Construa uma classe *Estudante* com os seguintes atributos: nome, sobrenome, identificador e proximold. O atributo proximold é um número sequencial e é disponível para todos os estudantes.
- Faça os métodos get e set e um construtor parametrizado para a classe Estudante.
- Crie um ArrayList para representar uma turma com 30 alunos.
- Faça a impressão das informações dos estudantes.

# Exercício 2 - Classe Matematica

- Construa uma classe *Matematica*, que contém os seguintes métodos estáticos:
  - *max3()* - recebe três valores int como argumentos e retorna o valor do maior.
  - *impar()* - recebe três entradas booleanas e retorna verdadeiro se um número ímpar de entradas for *true* e falso caso contrário.
  - *maioria()* - recebe três argumentos booleanos e retorna verdadeiro se pelo menos dois dos argumentos tiverem o valor *true* e falso caso contrário. Não utilize if.
- Teste os métodos da classe no *main()*.

# Exercício 3 - Agenda

Escreva uma classe *Pessoa* que tenha como variáveis de instância o nome (private) de uma pessoa, o seu telefone (private) e um id sequencial único. O construtor dessa classe deve ter como parâmetros o nome e o telefone. O id deve ser gerado automaticamente. Em seguida, escreva um programa que tenha como entrada as seguintes opções:

- *n*: Cria uma nova pessoa na agenda:  
solicita o nome: *entre com o nome da pessoa que será criada*  
solicita o telefone: *entre com o telefone da pessoa*
- *d*: Deleta uma pessoa da agenda:  
solicita o nome: *entre com nome da pessoa que será excluída*
- *p*: Imprime toda a agenda
- *q*: sai do programa

```
Entre com uma das seguintes opções:
```

```
n [nova entrada]  
d [apaga registro da agenda]  
p [imprime toda a agenda]  
q [sai do programa]
```

```
n
```

```
Entre com o nome:
```

```
Fulano
```

```
Entre com o telefone:
```

```
12345678
```

```
Entre com uma das seguintes opções:
```

```
n [nova entrada]  
d [apaga registro da agenda]  
p [imprime toda a agenda]  
q [sai do programa]
```

Utilize um loop infinito até que *q* seja a entrada. Armazene todos os objetos do tipo *Pessoa* em um *ArrayList* e manipule os objetos pelo *ArrayList*.

PS: considere que as pessoas nunca terão nomes iguais para facilitar a busca.

# Exercício 4 - Carro

- Construa uma classe *Carro*, composta por objetos de várias outras classes: *Roda*, *Motor*, *Porta*, *Banco*, *Volante* etc.
- Crie pelo menos 10 classes para compor Carro. Cada classe deve ter 2 atributos no mínimo, métodos get e set e toString.
- Muitos objetos que compõe carro serão manipulados por meio de *ArrayLists* (dentro da classe Carro), como objetos da classe Roda, por exemplo, já que a maioria dos carros é composto por 4 rodas.
- No main, crie um *ArrayList* de Carros e instancie 3 Carros.
- Faça a impressão das informações de todos os 3 Carros.