



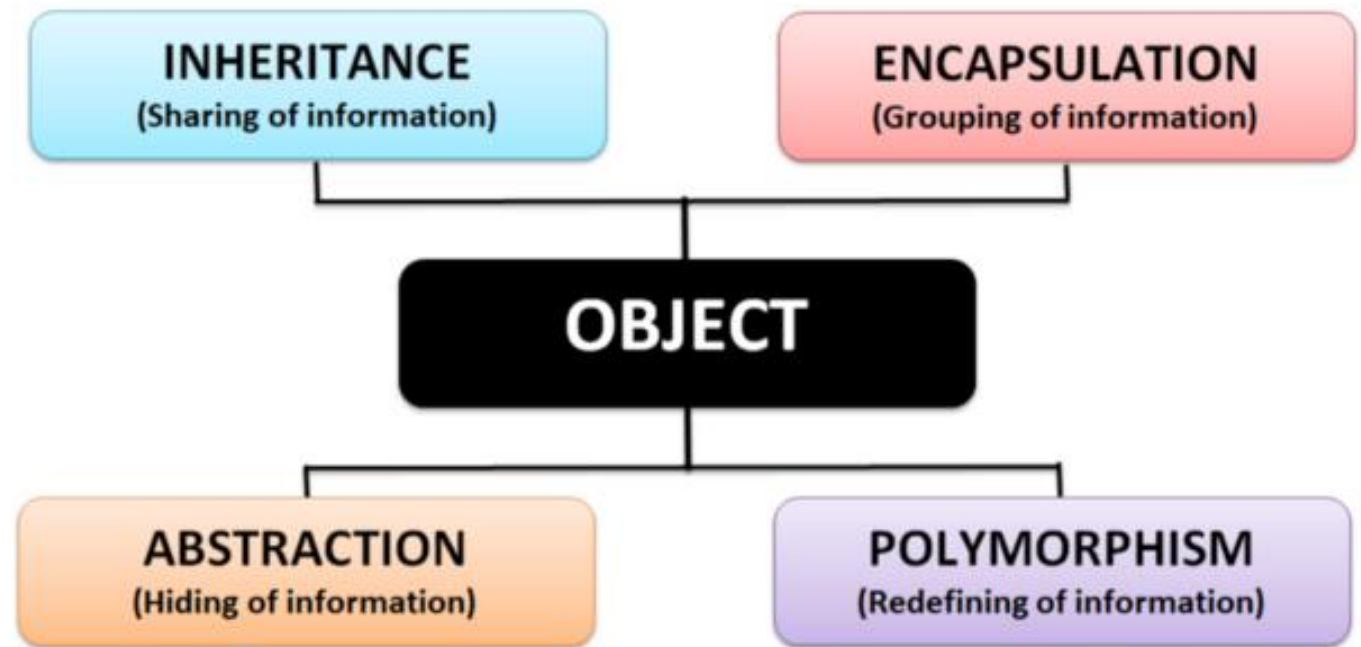
Ciência da **Computação**

Programação Orientada a Objetos
Prof. Luciano Rodrigo Ferretto

Uma breve revisão...

Abstração, Encapsulamento, Herança, Polimorfismo, Classe Object e Palavras chaves “super” e “abstract”

Pilares da Orientação a objetos



Abstração

- **Abstração** é o processo de identificar e definir características essenciais de objetos do mundo real, representando essas características em forma de **classes** e **objetos**.
- Uma **classe** é uma estrutura que define atributos e métodos representando um tipo de objeto, fornecendo um modelo para criar instâncias desse tipo.
- Um **objeto** é uma instância de uma classe, caracterizado por seus atributos e comportamentos definidos na classe.
 - Ele pode interagir com outros objetos por meio de métodos e troca de dados.

Objeto é uma **instância** concreta de uma classe na POO.

Encapsulamento

- **Encapsulamento** é a prática de ocultar os detalhes internos de um objeto, expondo apenas as operações relevantes para manipular esses detalhes. Isso promove a modularidade, a segurança e a manutenibilidade do código.
- Na maioria das linguagens de programação orientada a objetos, o encapsulamento é alcançado através da definição de **modificadores de acesso**.
 - **public, private, protected e default** (sem modificador)

Herança

- **Herança** é o princípio que permite que uma classe (subclasse) herde os atributos e métodos de outra classe (superclasse), facilitando a reutilização de código, a organização hierárquica e a promoção de relações entre objetos.
- Quando há duas ou mais classes com atributos e métodos em comum, a herança facilita a criação de uma estrutura hierárquica. Nessa estrutura, uma classe "pai" define os elementos comuns que serão herdados pelas classes "filhas".

Polimorfismo

- **Polimorfismo** é um princípio da programação orientada a objetos que permite que funções ou métodos assumam comportamentos diferentes, dependendo do contexto.
- Com o polimorfismo, é possível reutilizar e adaptar comportamentos em diferentes contextos, permitindo que classes e métodos sejam usados de maneira flexível, sem perder a consistência da interface ou lógica comum.
- Ele se divide em dois tipos principais:
 - Polimorfismo Estático
 - Polimorfismo Dinâmico

Polimorfismo

- **Polimorfismo dinâmico** (em tempo de execução): Permite que objetos de sub-classes de uma mesma classe base invoquem métodos que têm a mesma assinatura (nome e parâmetros), mas que se comportam de maneira diferente, de acordo com o tipo específico do objeto instanciado.
 - Sobrescrita de métodos
- **Polimorfismo estático** (em tempo de compilação): Refere-se à capacidade de métodos com o mesmo nome, mas com assinaturas diferentes (número ou tipos de parâmetros), serem utilizados de formas distintas.
 - Sobrecarga de métodos.

Polimorfismo - Estático x Dinâmico

Estático	Dinâmico
Também conhecido como Sobrecarga de Método .	Também conhecido como Sobrescrita de Método .
Ocorre quando há vários métodos com o <u>mesmo nome</u> , mas com diferentes assinaturas (<u>diferentes tipos ou números de parâmetros</u>) dentro da mesma classe.	Ocorre quando uma <u>subclasse</u> fornece uma implementação <u>específica</u> de um método que <u>já é definido na sua superclasse</u> .
A ligação (binding) das chamadas de método ao código correspondente é feita em tempo de <u>compilação</u> .	A ligação (binding) das chamadas de método ao código correspondente é feita em tempo de <u>execução</u> .
Permite que métodos com o mesmo nome (assinaturas diferentes) possam ser chamados, e a versão correta do método é invocada de acordo com os parâmetros fornecidos na chamada	Permite que o mesmo método (mesma assinatura) possa ser chamado em <u>diferentes tipos de objetos</u> , e a versão correta do método é invocada de acordo com o <u>tipo real do objeto</u> em tempo de execução.

Acesso via “super”

Usando super para acessar membros da superclasse

Há uma segunda forma de **super** que age um pouco como **this**, exceto por referenciar sempre a superclasse da subclasse em que é usada. Essa aplicação tem a forma geral a seguir:

`super.membro`

Aqui, *membro* pode ser um método ou uma variável de instância.

```
public class Livro {  
    @Override  
    public String toString() {  
        String descricao = "Título: " + this.getTitulo()  
        + " - Autor: " + this.getAutor()  
        + " - Ano: " + this.getAnoPublicacao();  
        return descricao;  
    }  
}
```

```
public class LivroFisico extends Livro {  
    @Override  
    public String toString() {  
        String descricao = super.toString();  
        return "Livro Físico: " + descricao + " - Dimensões: " + this.getDimensoes();  
    }  
}
```

```
public class LivroDigital extends Livro {  
    @Override  
    public String toString() {  
        String descricao = super.toString();  
        return "Livro Digital: " + descricao + " - Formato: " + this.getFormatoArquivo();  
    }  
}
```

Fonte: SCHILDI, H. Java para iniciantes: crie, compile e execute programas Java rapidamente. 6. ed. Porto Alegre: Bookman, 2015..





Acesso via “super” - constructors

Usando super para chamar construtores da superclasse

Uma subclasse pode chamar um construtor definido por sua superclasse usando a forma de **super** a seguir:

`super(lista-parâmetros);`

```
public class Livro {  
    public Livro(String titulo, String autor, int anoPublicacao, int nPaginas) {  
        this.titulo = titulo;  
        this.autor = autor;  
        this.anoPublicacao = anoPublicacao;  
        this.nPaginas = nPaginas;  
    }  
  
    public class LivroFisico extends Livro {  
        public LivroFisico(String titulo, String autor, int anoPublicacao, int nPaginas, int nExemplares,  
            String dimensoes) {  
            super(titulo, autor, anoPublicacao, nPaginas);  
            this.nExemplares = nExemplares;  
            this.dimensoes = dimensoes;  
        }  
    }  
}
```



Classes Abstratas

As classes que não são abstratas e herdam classes abstratas são denominadas classes concretas. Sobre as classes abstratas, podemos destacar os seguintes conceitos.

- Uma classe abstrata serve como modelo para uma classe concreta.
- Como são apenas modelos, não podem ser instanciadas diretamente.
- Por não ser instanciadas, devem ser herdadas por classes concretas.
- Uma classe abstrata pode, ou não, conter métodos abstratos, ou seja, pode ou não implementar um método.
- Contudo, métodos abstratos definidos em uma classe abstrata devem, obrigatoriamente, ser implementados em uma classe concreta.

Métodos Abstratos

- Uma **classe abstrata** pode conter **métodos abstratos**, que são declarados sem uma implementação.
- Os métodos abstratos servem como "contratos" que as subclasses devem implementar.
- A lógica é dizer o que precisa ser feito, e não como fazer.

```
public abstract class Livro {  
    // Método abstrato para definir o formato do livro  
    //      (ex.: "Físico" ou "Digital")  
    public abstract String getFormato();  
    // Método abstrato para definir o tipo de mídia  
    //      (ex.: "Capa Simples", "Capa Dura", "ePub", "PDF", etc)  
    public abstract String getTipoMidia();  
}
```

Palavra chave “final”

Criando constantes e impedindo a herança e a sobrescrita

Palavra chave “final”

- Mesmo com a sobreposição de métodos e a herança sendo tão poderosas e úteis, em alguns casos pode ser desejável **evitar que ocorram**.
- Por exemplo, no nosso Sistema de Gestão de Biblioteca, na classe “Livro” podemos ter um método para calcular a “idade” do livro, ou seja, a quantos anos já fazem desde a publicação.
- Esse é um exemplo prático de método que NÃO deve ser sobrescrito, pois o cálculo não muda independente do tipo de livro.
- Neste exemplo, para evitar que alguém da equipe sobrescreva o método e possa, “sem querer ou não”, comprometer o cálculo, podemos utilizar a palavra chave “**final**”.

Palavra chave “final”

- Nas imagens podemos constatar que ao tentar sobrescrever um método definido com “final” na sua assinatura, temos um erro em tempo de compilação.

```
public abstract class Livro {  
    public final int calcularTempoPublicacao() {  
        int anoAtual = LocalDate.now().getYear();  
        return anoAtual - this.getAnoPublicacao();  
    }  
}
```

```
public class LivroFisico extends Livro {  
    public int calcularTempoPublicacao() {  
        int ano  
        return  
    }  
    ...  
}
```

Cannot override the final method from Livro Java(67109265)

[Go to Super Implementation](#)

```
int LivroFisico.calcularTempoPublicacao()
```


Palavra chave “**final**”

- Quando queremos que uma classe não seja instanciada e sirva apenas para ser herdada utilizamos a palavra chave “**abstract**”.
- Com a palavra chave “**final**” temos o contrário, podemos definir que uma classe não possa ser herdada.
- Como era de se esperar, é inválido declarar uma classe como “**abstract**” e “**final**”.

Palavra chave “final”

```
public final class LivroFisico extends Livro {  
    private int nExemplares;  
    private String dimensoes;  
}
```

```
public class SubLivroFisico extends LivroFisico{  
|
```

The type SubLivroFisico cannot subclass the final class LivroFisico :

LivroFisico

Palavra chave “final”

- Você também pode usar a palavra-chave “**final**” para definir constantes dentro do seu sistema. Isso é útil para valores que não devem ser modificados após sua inicialização.

```
public class Biblioteca {  
    // BD em memória  
    private List<Livro> acervo = new ArrayList<>();  
  
    private final int ANO_PUBLICACAO_MINIMO = 1400;  
}
```

Sistema de Gerenciamento de Frotas

- O Sistema deverá apresentar ao usuário o seguinte menu:
- **Sistema de Gestão de Frotas**
Opções:
 - 1 – Cadastrar Novo Veículo**
 - 2 - Listar todos Veículos cadastrados**
 - 3 - Pesquisar um Veículo por placa**
 - 4 - Remover um Veículo**
 - 0 - Sair**



Sistema de Gerenciamento de Frotas

- O Sistema deverá seguir estas Regras de Negócio:
- Não permitir cadastrar veículos com os campos em branco.
- Não permitir cadastrar dois veículos com a mesma placa.
- O sistema deverá conter no mínimo duas camadas:
 - Camada “View” (esta irá conter junto a camada Controller)
 - Camada “Service”
 - Não se preocupem, vou explicar melhor na aula.



Sistema de Gerenciamento de Frotas

- O Sistema deverá apresentar ao usuário o seguinte menu:

Sistema de Gestão de Frotas

Opções:

- 1 – Cadastrar Novo Veículo
- 2 - Listar todos Veículos cadastrados
- 3 - Pesquisar um Veículo por placa
- 4 - Remover um Veículo
- 0 - Sair



Sistema de Gerenciamento de Frota

- O Sistema deverá seguir estas Regras de Negócio:
- Não permitir cadastrar veículos com os campos em branco.
 - Não permitir cadastrar com ano “0”
 - Não permitir carro com o nº de portas “0”
- Não permitir cadastrar dois veículos com a mesma placa.
- O sistema deverá conter no mínimo duas camadas:
 - Camada “View” (Apresentação)
 - Camada “Service” (Negócio)



Tela Inicial – Menu

- Sempre ao exibir a tela inicial, a qual consta o nosso Menu principal, deverá ser limpa a tela: **`System.out.print("\033[H\033[2J");`**
- Observe a posição do cursor!!!

```
Sistema de Gerenciamento de Frotas
MENU DE OPÇÕES:
1 - Cadastrar Novo Veículo
2 - Listar todos os Veículos
3 - Pesquisar Veículo por Placa
4 - Remover Veículo
0 - Sair
Digite a opção desejada: 
```


Tela Inicial – Menu → Opção Inválida

- Fazer o tratamento de erro caso o usuário digite um valor não numérico, ou que esteja fora do range de opções.

```
Sistema de Gerenciamento de Frotas
MENU DE OPÇÕES:
1 - Cadastrar Novo Veículo
2 - Listar todos os Veículos
3 - Pesquisar Veículo por Placa
4 - Remover Veículo
0 - Sair
Digite a opção desejada: opcaoerrada
Digite um número dentro das Opções acima: 455
Digite um número dentro das Opções acima: 333
Digite um número dentro das Opções acima: █
```

Cadastro de Veículo - Início

- Limpe a tela e faça o tratamento de erro igual ao Menu Principal

```
CADASTRO DE VEÍCULO
Qual o tipo de veículo
1 - Carro
2 - Moto
Digite a opção desejada: dd
Digite um número dentro das Opções acima: ff
Digite um número dentro das Opções acima: 22
Digite um número dentro das Opções acima: █
```

Cadastro de Veículo – Validação de dados

- Todos os campos que são numéricos (ano, númeroPortas, etc) devem ter o tratamento para exceção, caso o usuário digite um valor inválido.

```
CADASTRO DE VEÍCULO
Qual o tipo de veículo
1 - Carro
2 - Moto
Digite a opção desejada: 2
Digite a marca da moto: Honda
Digite o modelo da moto: Falcon
Digite o ano da moto: abc
Digite um ano válido: def
Digite um ano válido: f333
Digite um ano válido: █
```

Cadastro de Veículo – Cadastro de Carro

- Observe que o usuário deve digitar os dados de entrada ao lado da mensagem.
- Deve aparecer a mensagem “Veículo cadastrado com sucesso!” caso não ocorra exceções.
- Após aparecer a mensagem “Pressione Enter para voltar ao Menu Inicial”

```
CADASTRO DE VEÍCULO
Qual o tipo de veículo
1 - Carro
2 - Moto
Digite a opção desejada: 1
Digite a marca do carro: Kia
Digite o modelo do carro: Cerato
Digite o ano do carro: 2011
Digite a placa do carro: ABC-1234
Digite o número de portas: 4
Veículo cadastrado com sucesso!
Pressione Enter para voltar ao Menu Inicial
```

Cadastro de Veículo – Cadastro de Moto

- Observe que o usuário deve digitar os dados de entrada ao lado da mensagem.
- Deve aparecer a mensagem “Veículo cadastrado com sucesso!” caso não ocorra exceções.
- Após aparecer a mensagem “Pressione Enter para voltar ao Menu Inicial”

```
CADASTRO DE VEÍCULO
Qual o tipo de veículo
1 - Carro
2 - Moto
Digite a opção desejada: 2
Digite a marca da moto: Honda
Digite o modelo da moto: Falcon NX400
Digite o ano da moto: 2005
Digite a placa da moto: ABC-4321
A moto possui partida elétrica: 1-Sim, 2-Não: 1
Veículo cadastrado com sucesso!
Pressione Enter para voltar ao Menu Inicial
█
```

Cadastro de Veículo – Regra de Negócio

- Caso alguma das regras de negócio seja quebrada, então deverá aparecer a mensagem “NÃO FOI POSSÍVEL CADASTRAR O VEÍCULO”.
- Abaixo dessa mensagem deverá aparecer a exceção retornada pela camada Service.
- Após aparecer a mensagem “Pressione Enter para continuar ...”

```
CADASTRO DE VEÍCULO
Qual o tipo de veículo
1 - Carro
2 - Moto
Digite a opção desejada: 1
Digite a marca do carro:
Digite o modelo do carro: Fusca
Digite o ano do carro: 1995
Digite a placa do carro: ABC-4321
Digite o número de portas: 2

NÃO FOI POSSÍVEL CADASTRAR O VEÍCULO
Campo Marca não pode ser em branco
Pressione Enter para voltar ao Menu Inicial
█
```

Listar Todos Veículos

- Limpe a tela
- Deverá ser listados todos os veículos seguindo o padrão da imagem abaixo.
- Deverá ser utilizado o “toString()” para os objetos veículos, aplicando corretamente a sobrescrita de métodos (polimorfismo).

```
LISTA DE VEÍCULOS CADASTRADOS
```

```
Veículo 1 - Tipo: Carro - Kia Cerato 2011 - Placa: ABC-1234 - N. Portas: 4
```

```
Veículo 2 - Tipo: Moto - Honda Falcon NX400 2005 - Placa: ABC-4321 - Part. Elétrica: Sim
```

```
Pressione Enter para voltar ao Menu Inicial
```



Pesquisa de Veículo por placa

- Limpe a tela
- Deverá ser apresentado as mensagens seguindo o padrão da imagem abaixo.
- Lembre-se que como não podemos registrar dois veículos com a mesma placa, então o retorno será sempre somente UM veículo.
- Deverá ser utilizado o “toString()” para os objetos veículos, aplicando corretamente a sobrescrita de métodos (polimorfismo).

```
PESQUISA DE VEÍCULOS POR PLACA
```

```
Informe a placa que deseja pesquisar: ABC-1234
```

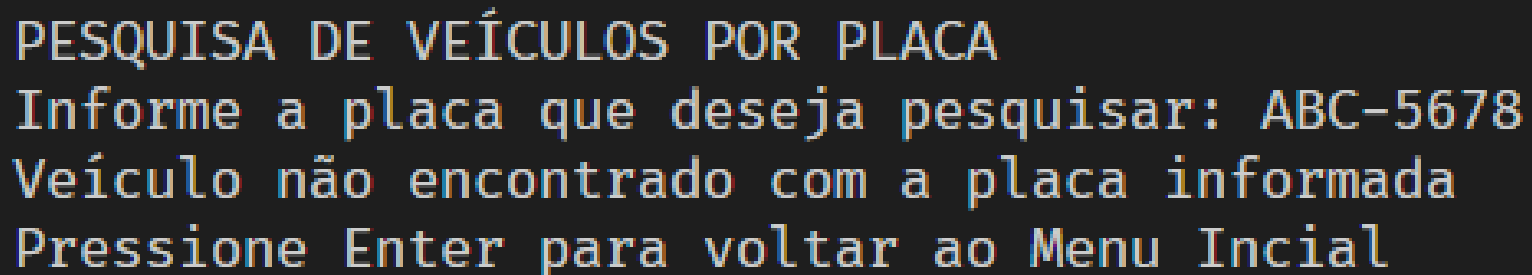
```
Veículo Tipo: Carro - Kia Cerato 2011 - Placa: ABC-1234 - N. Portas: 4
```

```
Pressione Enter para voltar ao Menu Inicial
```



Pesquisa de Veículo por placa – Veículo não encontrado

- Quando não encontrar deverá ser apresentado a mensagem conforme a imagem abaixo.

A screenshot of a terminal window with a black background and light blue text. The text is as follows:

```
PESQUISA DE VEÍCULOS POR PLACA  
Informe a placa que deseja pesquisar: ABC-5678  
Veículo não encontrado com a placa informada  
Pressione Enter para voltar ao Menu Inicial
```

A small white cursor is visible at the end of the last line of text.

PESQUISA DE VEÍCULOS POR PLACA
Informe a placa que deseja pesquisar: ABC-5678
Veículo não encontrado com a placa informada
Pressione Enter para voltar ao Menu Inicial

Remoção de Veículo por placa

- Limpe a tela
- O Sistema deverá apresentar as mensagens conforme a imagem abaixo.
- Deverá ser listado todos os veículos assim como o processo de Listar Todos os Veículos.
- Após o sistema deverá pedir para o usuário digitar a placa do veículo que ele deseja remover.

```
REMOÇÃO DE VEÍCULOS
```

```
LISTA DE VEÍCULOS CADASTRADOS
```

```
Veículo 1 - Tipo: Carro - Kia Cerato 2011 - Placa: ABC-1234 - N. Portas: 4
```

```
Veículo 2 - Tipo: Moto - Honda Falcon NX400 2005 - Placa: ABC-4321 - Part. Elétrica: Sim
```

```
Veículo 3 - Tipo: Carro - Volkswagen Fusca 1995 - Placa: ABC-9876 - N. Portas: 2
```

```
Veículo 4 - Tipo: Moto - Honda CB400 1998 - Placa: ABC-7890 - Part. Elétrica: Sim
```

```
Informe a placa do veículo que deseja REMOVER: █
```

Remoção de Veículo por placa – Veículo não encontrado

- Quando não encontrar deverá ser apresentado a mensagem conforme a imagem abaixo.

```
REMOÇÃO DE VEÍCULOS
LISTA DE VEÍCULOS CADASTRADOS
Veículo 1 - Tipo: Carro - Kia Cerato 2011 - Placa: ABC-1234 - N. Portas: 4
Veículo 2 - Tipo: Moto - Honda Falcon NX400 2005 - Placa: ABC-4321 - Part. Elétrica: Sim
Veículo 3 - Tipo: Carro - Volkswagen Fusca 1995 - Placa: ABC-9876 - N. Portas: 2
Veículo 4 - Tipo: Moto - Honda CB400 1998 - Placa: ABC-7890 - Part. Elétrica: Sim
Informe a placa do veículo que deseja REMOVER: ABD-4444
Veículo não encontrado com a placa informada
Pressione Enter para voltar ao Menu Inicial
```



Remoção de Veículo por placa – Veículo não encontrado

- Caso encontre o veículo então deverá ser removido da Lista (BD).
- Lembre-se que os processos de remoção e pesquisa devem ser feitos na camada Service.

```
REMOÇÃO DE VEÍCULOS
LISTA DE VEÍCULOS CADASTRADOS
Veículo 1 - Tipo: Carro - Kia Cerato 2011 - Placa: ABC-1234 - N. Portas: 4
Veículo 2 - Tipo: Moto - Honda Falcon NX400 2005 - Placa: ABC-4321 - Part. Elétrica: Sim
Veículo 3 - Tipo: Carro - Volkswagen Fusca 1995 - Placa: ABC-9876 - N. Portas: 2
Veículo 4 - Tipo: Moto - Honda CB400 1998 - Placa: ABC-7890 - Part. Elétrica: Sim
Informe a placa do veículo que deseja REMOVER: ABC-9876
Veículo removido com sucesso!
Pressione Enter para voltar ao Menu Inicial
```

