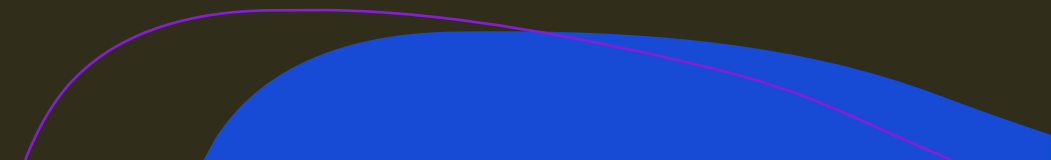


The image features a dynamic abstract composition. In the upper right, a series of parallel diagonal stripes in yellow, orange, red, purple, blue, and green create a sense of movement. A large, dark brown, curved shape dominates the right side, serving as a backdrop for the text. The left side is filled with various geometric elements: yellow and purple shapes with circular patterns, solid yellow and red circles, and a light gray background with a fine dot pattern. A thin pink line curves across the middle of the image, separating the colorful stripes from the dark shape.

Aplicando os
conceitos



Primeiro exercício

- Crie uma classe Carro com os seguintes atributos: marca, modelo, ano, e cor. Crie uma forma de imprimir todos os atributos do carro. Instancie três objetos da classe Carro e exiba seus detalhes.
- 

```
2
3 public class Carro {
4
5     private String marca;
6     private String modelo;
7     private int ano;
8     private String cor;
9
10    public Carro(String marca, String modelo, int ano, String cor) {
11        this.setMarca(marca);
12        this.setModelo(modelo);
13        this.setAno(ano);
14        this.setCor(cor);
15    }
16
17    @Override
18    public String toString() {
19        String retorno;
20        retorno = this.getMarca() + " " + this.getModelo() + "\n";
21        retorno = retorno + Integer.toString(this.getAno()) + " - " + this.getCor() + "\n";
22        return retorno;
23    }
24
25 }
```

Gera uma representação do obj como String

```
2
3 public class Carro {
4
5     private String marca;
6     private String modelo;
7     private int ano;
8     private String cor;
9
10    public Carro(String marca, String modelo, int ano, String cor) {
11        this.setMarca(marca);
12        this.setModelo(modelo);
13        this.setAno(ano);
14        this.setCor(cor);
15    }
16
17    @Override
18    public String toString() {
19        String retorno;
20        retorno = this.getMarca() + " " + this.getModelo() + "\n";
21        retorno = retorno + Integer.toString(this.getAno()) + " - " + this.getCor() + "\n";
22        return retorno;
23    }
24
25 }
```

```
1 public class Principal {
2
3     public static void main(String[] args) {
4         Carro c = new Carro("Wolksvagem", "UP", 2015, "Vermelho");
5         Carro c1 = new Carro("GM", "Onix", 2017, "Preto");
6         Carro c2 = new Carro("Toyota", "Yaris", 2017, "Vermelho");
7
8         System.out.println(c);
9         System.out.println(c1);
10        System.out.println(c2);
11
12    }
13
14 }
```



```
1 public class Principal {
2
3     public static void main(String[] args) {
4         Carro c = new Carro("Wolksvagem", "UP", 2015, "Vermelho");
5         Carro c1 = new Carro("GM", "Onix", 2017, "Preto");
6         Carro c2 = new Carro("Toyota", "Yaris", 2017, "Vermelho");
7
8         System.out.println(c);
9         System.out.println(c1);
10        System.out.println(c2);
11
12    }
13
14 }
```

Automaticamente transforma o objeto para String

Segundo exercício

- Compare dois objetos da classe Carro e diga se eles são iguais ou não

```
3 public class Carro {
4
5     private String marca;
6     private String modelo;
7     private int ano;
8     private String cor;
9
10    public Carro(String marca, String modelo, int ano, String cor) {
11        this.setMarca(marca);
12        this.setModelo(modelo);
13        this.setAno(ano);
14        this.setCor(cor);
15    }
16
17    @Override
18    public boolean equals(Object obj) {
19        Carro cComp = Carro.class.cast(obj);
20        if (
21            (this.getAno() == cComp.getAno()) &&
22            (this.getMarca().equals(cComp.getMarca())) &&
23            (this.getModelo().equals(cComp.getModelo())) &&
24            (this.getCor().equals(cComp.getCor()))
25        ) {
26            return true;
27        } else {
28            return false;
29        }
30    }
31
32 }
```



```
3 public class Carro {
4
5     private String marca;
6     private String modelo;
7     private int ano;
8     private String cor;
9
10    public Carro(String marca, String modelo, int ano, String cor) {
11        this.setMarca(marca);
12        this.setModelo(modelo);
13        this.setAno(ano);
14        this.setCor(cor);
15    }
16
17    @Override
18    public boolean equals(Object obj) {
19        Carro cComp = Carro.class.cast(obj);
20        if (
21            (this.getAno() == cComp.getAno()) &&
22            (this.getMarca().equals(cComp.getMarca())) &&
23            (this.getModelo().equals(cComp.getModelo())) &&
24            (this.getCor().equals(cComp.getCor()))
25        ) {
26            return true;
27        } else {
28            return false;
29        }
30    }
31
32 }
```

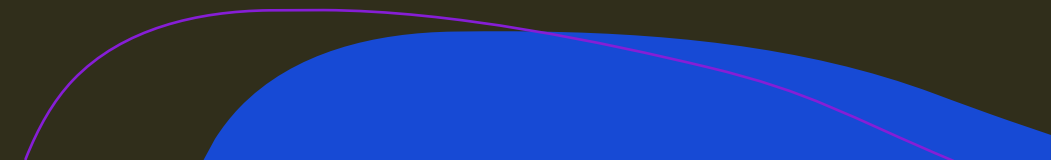
Converte objeto do tipo Object para o tipo Carro

Strings são objetos, então é necessário chamar o método equals

```
6 public class Principal {
7
8     public static void main(String[] args) {
9         Carro c = new Carro("Wolksvagem", "UP", 2015, "Vermelho");
10        Carro c1 = new Carro("Wolksvagem", "UP", 2015, "Vermelho");
11
12        if (c.equals(c1)) {
13            System.out.println("Iguais");
14        } else {
15            System.out.println("Diferentes");
16        }
17
18    }
19
20 }
```




Segundo exercício

- Transforme o atributo em uma classe e faça seu vínculo com a classe carro.
- 

```
3 public class Carro {  
4  
5     private Marca marca;  
6     private String modelo;  
7     private int ano;  
8     private String cor;
```

```
3 public class Carro {
```

Cria uma referência a um objeto da classe
Marca com carro



```
4  
5     private Marca marca;
```

```
6     private String modelo;
```

```
7     private int ano;
```

```
8     private String cor;
```



```
1 package aplicacaoPrincipal;
2
3 import dados.Veiculo;
4 import dados.Carro;
5 import dados.Marca;
6
7 public class Principal {
8
9     public static void main(String[] args) {
10         Marca m = new Marca("Wolksvagem");
11         Carro c = new Carro(m, "UP", 2015, "Vermelho");
12         Carro c1 = new Carro(m, "UP", 2015, "Vermelho");
13
14         if (c.equals(c1)) {
15             System.out.println("Iguais");
16         } else {
17             System.out.println("Diferentes");
18         }
19     }
20 }
21
22 }
```

Terceiro exercício

- Crie um atributo quantidade que diga quantos carros já foram instanciados.

```
3 public class Carro {
4
5     private Marca marca;
6     private String modelo;
7     private int ano;
8     private String cor;
9     private static int qtdCriados = 0;
10
11     public static void aumentaCriados() {
12         Carro.qtdCriados += 1;
13     }
14
15     public static int getQtdCriados() {
16         return Carro.qtdCriados;
17     }
18
19     public Carro(Marca marca, String modelo, int ano, String cor) {
20         this.setMarca(marca);
21         this.setModelo(modelo);
22         this.setAno(ano);
23         this.setCor(cor);
24         Carro.aumentaCriados();
25     }
26 }
```

```
3 public class Carro {
4     private static int qtdCriados;
5     private int ano;
6     private String cor;
7     private static int qtdCriados = 0;
8
9     public static void aumentaCriados() {
10         Carro.qtdCriados += 1;
11     }
12
13     public static int getQtdCriados() {
14         return Carro.qtdCriados;
15     }
16
17     public Carro(Marca marca, String modelo, int ano, String cor) {
18         this.setMarca(marca);
19         this.setModelo(modelo);
20         this.setAno(ano);
21         this.setCor(cor);
22         Carro.aumentaCriados();
23     }
24 }
25
```

Modificador estático

Atributo da classe

Método estático (da classe)

Método estático (da classe)

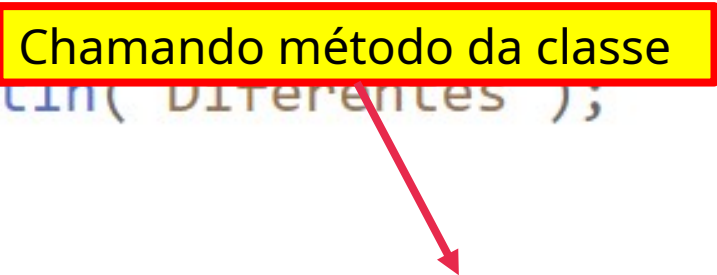
Métodos e
atributos da
Classe

Não pode ser
chamado com this

Pertencem a
classe, não ao
objeto


```
7 public class Principal {
8
9     public static void main(String[] args) {
10         Marca m = new Marca("Wolksvagem");
11         Carro c = new Carro(m, "UP", 2015, "Vermelho");
12         Carro c1 = new Carro(m, "UP", 2015, "Vermelho");
13
14         if (c.equals(c1)) {
15             System.out.println("Iguais");
16         } else {
17             System.out.println("Diferentes");
18         }
19
20         System.out.println(Carro.getQtdCriados());
21
22     }
23
24 }
```

```
7 public class Principal {
8
9     public static void main(String[] args) {
10         Marca m = new Marca("Wolkswagen");
11         Carro c = new Carro(m, "UP", 2015, "Vermelho");
12         Carro c1 = new Carro(m, "UP", 2015, "Vermelho");
13
14         if (c.equals(c1)) {
15             System.out.println("Iguais");
16         } else {
17             System.out.println("Diferentes");
18         }
19
20         System.out.println(Carro.getQtdCriados());
21
22     }
23
24 }
```



Trabalhando com arrays em Java



É possível criar arrays em Java assim como em C



Eles podem ser do tipo de uma classe, assim possibilitando o armazenamento de um conjunto de objetos

```
1 public class Principal {
2
3     public static void main(String[] args) {
4         Carro[] cs = new Carro[5];
5         Marca m = new Marca("Wolksvagem");
6         for (int i = 0; i < cs.length; i++) {
7             cs[i] = new Carro(m, "Teste", 2015 + i, "Vermelho");
8         }
9         System.out.println(Carro.getQtdCriados());
10    }
11
12 }
```

Define um array de Carro

Cria um array de 5 posições de carro

```
1 public class Principal {
2
3     public static void main(String[] args) {
4         Carro[] cs = new Carro[5];
5         Marca m = new Marca("Wolksvagem");
6         for (int i = 0; i < cs.length; i++) {
7             cs[i] = new Carro(m, "Teste", 2015 + i, "Vermelho");
8         }
9         System.out.println(Carro.getQtdCriados());
10    }
11
12 }
```