

The logo for Oracle Academy is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by two horizontal dark gray bars, one at the top and one at the bottom.

ORACLE

Academy

Java Foundations

7-6

Variáveis e Métodos Estáticos

ORACLE
Academy



Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

Objetivos

- Esta lição abrange os seguintes objetivos:
 - Descrever uma variável estática e demonstrar seu uso dentro de um programa
 - Descrever um método estático e demonstrar seu uso dentro de um programa
 - Entender como usar a palavra-chave final com variáveis estáticas



Revisão de Referências a Objetos

- Um objeto deve ser instanciado antes que seja possível acessar seus métodos e campos
- A instanciação fornece uma referência a um objeto
- Uma referência a um objeto é usada para acessar campos e métodos de um objeto

```
Prisoner p01 = new Prisoner()  
p01.name           //Acessando um campo  
p01.display()      //Chamando um método
```

A Classe Math É Diferente

- Seria cansativo criar um novo objeto Math toda vez que desejássemos fazer um pequeno cálculo
- Felizmente, nunca precisamos instanciar um objeto Math
- Os campos e os métodos Math são acessados fazendo referência diretamente à classe Math
- Eles são conhecidos como variáveis estáticas e métodos estáticos

```
Math.PI  
Math.sin(0)
```

```
//Nada é instanciado  
//Acessando um campo estático  
//Chamando um método estático
```

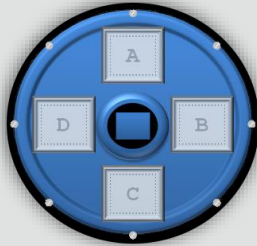
O Que Isso Significa?

- Por que esses dois fatores são importantes?
 - Uma referência a um objeto é usada para acessar campos e métodos de um objeto
 - Os campos e os métodos estáticos são acessados fazendo referência diretamente à classe
- Isso abrange mais do que simplesmente a conveniência de não precisar instanciar um objeto
- O próximo exercício permite que você explore um caso de uso de dados estáticos
 - Depois perguntaremos o que você observou

Exercício 1



- Execute os Basic Puzzles de 8 a 11
 - <https://objectstorage.uk-london-1.oraclecloud.com/n/lrvrlgaqj8dd/b/Games/o/JavaPuzzleBall/index.html>
- Considere o seguinte:
 - O que acontece quando você gira o BlueWheel?
 - Quais são as outras maneiras de interferir na rotação dos bumpers?



ORACLE
Academy

JFo 7-6
Variáveis e Métodos Estáticos (Static)

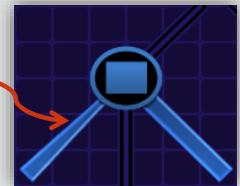
Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

7

Explicação Detalhada sobre o Java Puzzle Ball

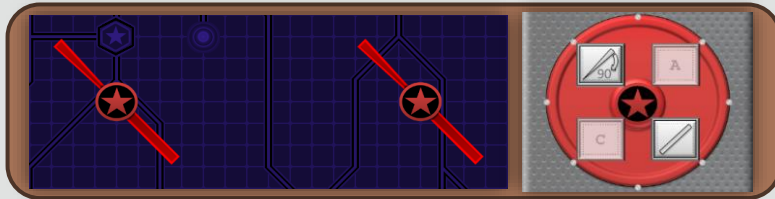
- O que acontece quando você gira o BlueWheel?
 - A orientação de todos os BlueBumpers muda
 - Todos os BlueBumpers compartilham a propriedade de orientação
 - A orientação pode ser representada por uma variável estática
- Quais são as outras maneiras de interferir na rotação dos bumpers?
 - Depois que a bola atinge uma parede de rotação, a rotação de um bumper individual muda
 - A rotação pode ser representada por uma variável de instância

Parede de rotação



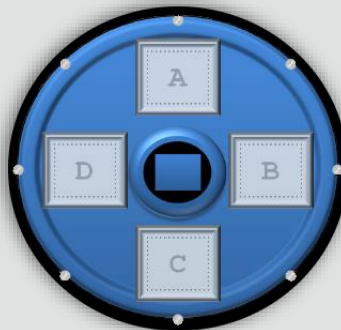
Variável estática: Orientation

- Essa variável estática é compartilhada por todas as instâncias
- As variáveis estáticas pertencem à classe, e não a uma instância individual
- Portanto, é necessário alterar uma variável estática uma única vez para que cada instância seja afetada
- No Basic Puzzle 11, girar o RedWheel muda a orientação de todos os objetos RedBumper



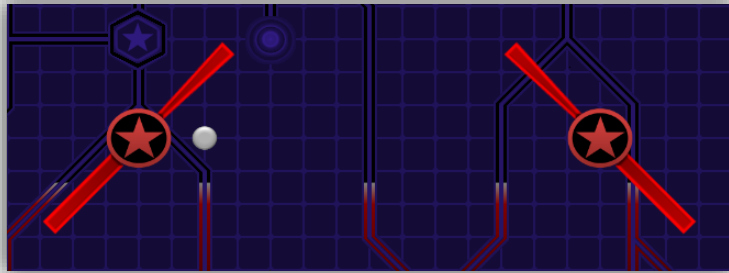
Variáveis Estáticas sem Instâncias

- É possível acessar variáveis estáticas, mesmo que nenhum objeto tenha sido instanciado
- No Basic Puzzle 11, o BlueWheel pode ser girado para alterar a propriedade orientation de todos os BlueBumpers
 - Não há BlueBumpers para mostrar os efeitos dessa alteração



Variáveis de instância Rotation

- Existem variáveis de instância exclusivas para cada instância de um objeto
- Portanto, as variáveis de instância precisam ser alteradas para cada objeto
- No Basic Puzzle 11, a rotação de um RedBumper individual muda depois de ser atingida pela bola



Por que um Campo Deve Ser Estático?

- Aqui estão alguns pontos a serem considerados:
 - O valor deste campo será diferente para cada objeto específico? Ou ele será o mesmo para todos os objetos?
 - O campo descreve a classe mais do que descreve qualquer objeto individual?
 - Você percebeu que estava digitando o mesmo valor em toda a classe?
 - Esse valor é uma constante que será usada nos cálculos?
 - Esse valor precisará ser acessado antes de quaisquer objetos serem instanciados?

Criando Variáveis Estáticas

- Uma variável torna-se estática quando sua declaração inclui a palavra-chave static
- Inicialize variáveis estáticas à medida que elas são declaradas
 - Caso contrário, as chamadas repetidas do construtor poderiam inicializar a mesma variável estática muitas vezes

```
public class RedBumper{  
    //Campos  
    public static int orientation = 45; //Variável estática  
    public int rotation;                //Variável de instância  
  
    //Construtor  
    public RedBumper(int rotation){  
        this.rotation = rotation;  
    }//fim construtor  
}//fim da classe RedBumper
```

Observação para os Instrutores: a caixa de código nesses dois slides devem estar na mesma posição.

Acessando Variáveis Estáticas em Sua Classe

- Mesmo que as variáveis estáticas não sejam inicializadas no construtor, ainda assim elas poderão ser acessadas
- Como qualquer outra variável, as variáveis estáticas podem ser acessadas dentro da respectiva classe

```
public class RedBumper{  
    //Campos  
    public static int orientation = 45; //Variável estática  
    public int rotation;                //Variável de instância  
    ...  
    //Métodos  
    public void display(){  
        System.out.println(orientation); //Acessar variável estática  
        System.out.println(rotation);    //Acessar variável de instância  
    }//fim do método display  
}//fim da classe RedBumper
```

Observação para os Instrutores: a caixa de código nesses dois slides devem estar na mesma posição.

Acessando Variáveis Estáticas em Outro Lugar

- As variáveis estáticas podem aparecer nos construtores, nos métodos ou fora da respectiva classe
- Chamar variáveis estáticas fora de sua classe baseia-se em fazer referência ao nome da classe, em vez de a uma variável de referência específica

```
public class TestClass {  
    public static void main(String[] args){  
        int x;  
        x = RedBumper.orientation;    //Acessar variável estática  
  
        RedBumper rb01 = new RedBumper(90); //Instância  
        int y;  
        y = rb01.rotation;            //Acessar variável de instância  
    }//fim do método main  
}//fim da classe TestClass
```

Exercício 2

- Continue a editar o projeto `PrisonTest`
 - Uma versão deste programa é fornecida nos arquivos `PrisonTest_Student_7_6.java`, `Prisoner_Student_7_6.java` e `Cell_Student_7_6.java`
- Modifique a classe `Prisoner`:
 - Inclua um campo `prisonerCount` de valor inteiro estático
 - Esse campo conta o número total de prisioneiros instanciados
 - Inicialize esse campo como 0
 - Aumente-o toda vez que um prisioneiro for instanciado
 - Inclua um campo `bookingNumber` de valor inteiro
 - Esse campo será inicializado com o valor atual de `prisonerCount`
 - Imprima `bookingNumber` e `prisonerCount` como parte do método `display()`
- Instancie alguns prisioneiros e exiba as respectivas informações

Você não precisa escrever getters para este exercício

Introdução aos Métodos Estáticos

- Você deve ter percebido o seguinte no exercício anterior:
 - O método `display()` pode acessar uma variável estática
 - As variáveis estáticas podem ser acessadas em métodos não estáticos
- A maioria dos métodos que você escreveu neste curso (exceto o método `main`) é considerada métodos de instância
 - Os métodos de instância são métodos não estáticos
- Os métodos também podem ser transformados em estáticos

Quando um Método Deve Ser Estático?

- Aqui estão alguns pontos a serem considerados:
 - O método lerá ou modificará campos estáticos?
 - O método não lerá nem modificará os campos de um objeto específico?
 - O método precisará ser chamado antes de quaisquer objetos serem instanciados?
- Os métodos estáticos são projetados para tratar dados estáticos
 - As variáveis estáticas podem ser acessadas em métodos estáticos

Criando Métodos Estáticos

- Um método torna-se estático quando sua declaração inclui a palavra-chave static

```
public class Prisoner{  
    //Campos  
    private static int prisonerCount = 0; //Variável estática  
    private int bookingNumber;           //Variável de instância  
  
    //Métodos  
    public static void displayPrisonerCount(){ //Método estático  
        System.out.println(prisonerCount);  
    } //fim do método displayPrisonerCount  
} //fim da classe Prisoner
```

Observação para os Instrutores: a caixa de código nesses dois slides devem estar na mesma posição.

Chamando Métodos Estáticos na Respectiva Classe

- Como acontece com qualquer outro método, os métodos estáticos podem ser chamados dentro da respectiva classe
- Os métodos estáticos ou de instância podem chamar um método estático

```
public class Prisoner{  
    private static int prisonerCount = 0; //Variável estática  
    private int bookingNumber;           //Variável de instância  
  
    public static void displayPrisonerCount(){ //Método estático  
        System.out.println(prisonerCount);  
    }//fim do método displayPrisonerCount  
    public void callAnotherMethod(){           //Método de instância  
        displayPrisonerCount();  
    }//fim do método callAnotherMethod  
}//fim da classe Prisoner
```

Observação para os Instrutores: a caixa de código nesses dois slides devem estar na mesma posição.

Chamando Métodos Estáticos em Outro Lugar

- Os métodos estáticos podem ser chamados dos construtores, de outros métodos ou de fora de sua classe
- Chamar métodos estáticos fora de sua classe baseia-se em fazer referência ao nome da classe, em vez de a uma variável de referência específica

```
public class TestClass {  
    public static void main(String[] args){  
        Prisoner.displayPrisonerCount(); //Chamar método estático  
  
        Cell cA1 = new Cell("A1", false, 1234);  
        Prisoner bubba = new Prisoner("Bubba", 2.08, 4, cA1);  
        bubba.display(); //Chamar método de instância  
    } //fim do método main  
} //fim da classe TestClass
```

Exercício 3

- Continue a editar o projeto `PrisonTest`
- Modifique a classe `Prisoner`:
 - Encapsule o campo `prisonerCount` Torne esse campo privado e crie um método getter estático
 - Tente tornar o método de exibição estático
 - Quais são as reclamações de seu IDE?
- No método `main`:
 - Chame o método getter que você acabou de criar e imprima o valor retornado

Por que seu IDE reclamou?

- Os campos e os métodos estáticos podem ser chamados sem instanciar um objeto
- Mas as variáveis de instância devem ser associadas a uma instância específica
- Um paradoxo será criado se um método estático tentar acessar informações sobre uma instância antes de ser criado
- Portanto, o Java não permite que métodos estáticos contenham variáveis de instância ou métodos de instância

```
public static void display(){  
    System.out.println(prisonerCount);  
    System.out.println(bookingNumber);  
}//fim do método display
```

Escrevendo campos static final

- Você é estimulado a fazer com que variáveis static sejam final
 - Mas os motivos vão além do escopo deste curso
- Lembre-se: os nomes de variáveis final...
 - São capitalizados por convenção
 - Usam um sublinhado (_) para separar palavras

```
public class Prisoner{  
    //Campos  
    ...  
    private int bookingNumber;  
    private static int prisonerCount = 0;  
    public static final int MAX_PRISONER_COUNT = 100;  
}//fim da classe Prisoner
```

ORACLE
Academy

JFo 7-6
Variáveis e Métodos Estáticos (Static)

Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

24

Variáveis estáticas são variáveis compartilhadas por todas as instâncias de uma classe.

Exemplo: Uma variável que é um contador o qual será incrementado cada vez que uma instância da classe for criada.

Variáveis finais estáticas são variáveis compartilhadas por todas as instâncias da classe, mas o valor não pode ser alterado.

Exemplo: Interest Rates ou Tax Rates permanece constante para todas as instâncias da classe. Elas também podem ser

acessadas sem criar uma instância da classe e também devem ser públicas.

Transformando campos static final em campos public

- O encapsulamento impede que as variáveis sejam manipuladas de uma maneira indesejável
- Mas não há risco de as primitivas public static final serem alteradas porque é impossível que seus valores mudem
- Isso é de grande utilidade para constantes como π , e ou outros valores que são usados constantemente nos cálculos
- Essas variáveis são chamadas diretamente, e não por meio dos getters

```
System.out.println(Math.PI);  
System.out.println(Math.E);
```

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Descrever uma variável estática e demonstrar seu uso dentro de um programa
 - Descrever um método estático e demonstrar seu uso dentro de um programa
 - Entender como usar a palavra-chave final com variáveis estáticas



ORACLE
Academy

JFo 7-6
Variáveis e Métodos Estáticos (Static)

Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

26

The Oracle Academy logo is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by two horizontal dark gray bars, one at the top and one at the bottom.

ORACLE

Academy