

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

ORACLE

Academy

Java Fundamentals

2-14

Métodos e Classes Java

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 um método, uma classe ou uma instância
 - Descrever um cenário em que uma estrutura de controle IF seria usada
 - Descrever um cenário em que uma estrutura de controle WHILE seria usada
 - Reconhecer a sintaxe de um método, uma classe, uma função ou um procedimento
 - Descrever a entrada e a saída



Alice 3 Versus Java

Alice 3	Java
O ambiente de programação 3D que usa representações visuais da linguagem de programação Java	Uma linguagem de programação com uma sintaxe que pode ser editada usando um IDE (Integrated Development Environment)
Usado para criar animações ou jogos interativos enquanto trabalha-se com construções de programação	Usado para criar aplicativos que são executados em qualquer plataforma, inclusive na web, usando a sintaxe Java
Interface de arrastar e soltar projetada para reduzir erros de sintaxe e fazer com que seja mais fácil aprender a programar	O IDE ajuda você a modelar objetos do mundo real, permitindo o reuso e facilitando a manutenção

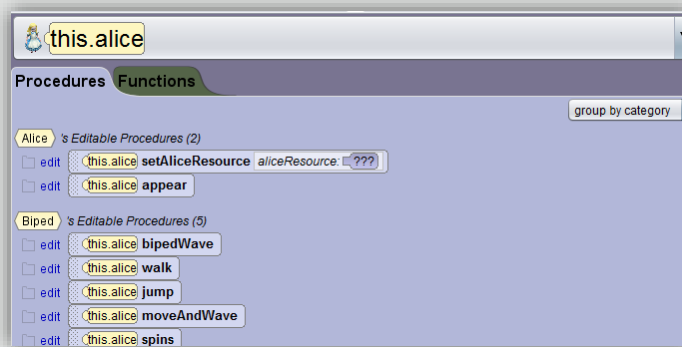
Procedimentos no Alice 3

- No Alice 3, um procedimento é um trecho de código que envia uma mensagem a um objeto pedindo a ele para executar uma ação
- Um procedimento não retorna um valor
- Um conjunto de procedimentos está disponível para cada classe

Os procedimentos são as informações usadas no Alice 3 para fazer com que nossos objetos executem uma ação.

Painel Methods no Alice 3

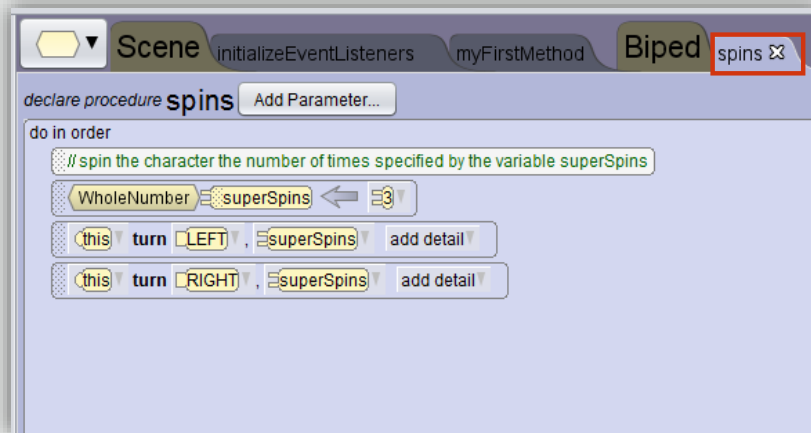
- O painel Methods lista todos os procedimentos e funções de um objeto
- Ele contém todas as funções e procedimentos incorporados e definidos pelo usuário disponíveis para cada objeto na animação



O painel Methods oferece a você procedimentos e funções relevantes ao objeto que você escolheu na lista drop-down.

Declarando Procedimentos no Alice 3

- Você pode declarar (criar) seus próprios procedimentos no Alice 3



Crie seus próprios procedimentos se tiver identificado vários objetos que realizarão a mesma ação ou se estiver escrevendo o mesmo código em mais de uma ocasião.

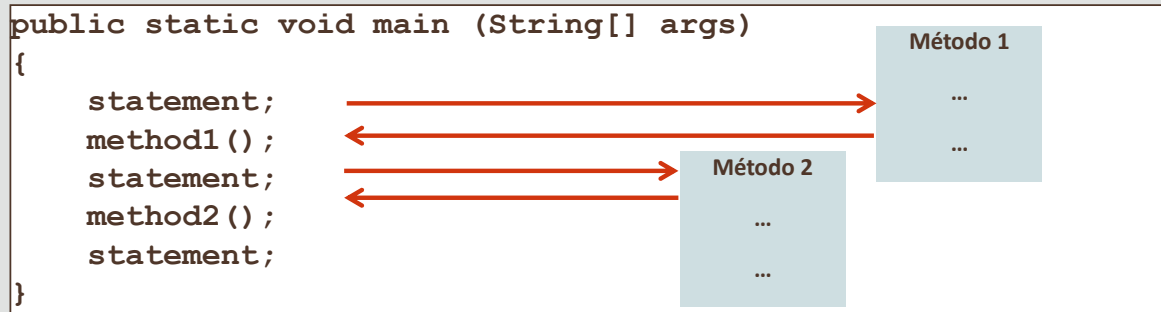
Métodos no Java

- Os métodos no Java equivalem aos procedimentos no Alice 3
- Um método é um trecho de código que envia uma mensagem a um objeto pedindo a ele para executar uma ação
- Métodos:
 - Pertencem a uma classe
 - São chamados por um nome
 - Podem ser chamados a qualquer momento em um programa usando o nome do método
- Quando é encontrado um nome de método em um programa, ele é executado

Quando você passar para o Greenfoot e para o Eclipse, todo o seu código será descrito como métodos. Você terá métodos void (procedimentos) que executam ações e métodos não void (funções) que retornam informações.

Métodos no Java - Exemplo

- Quando um método é concluído, a execução retorna para a área do código do programa de onde foi chamada, e o programa continua na linha de código seguinte



As chamadas de métodos são executadas da mesma maneira que ocorre quando usamos funções ou procedimentos definidos pelo usuário no Alice 3.

Decisões de Cada Método

- Há três decisões para tomar para cada método:
 - O que o método deve fazer
 - De quais entradas o método precisa
 - Qual resposta ele deve retornar
- A sintaxe Java de um método:

```
[modifiers] dataType methodName(parameterList) {  
    //methodBody  
    return result;  
}
```

O modificador, o tipo de dados, o nome e a lista de parâmetros são conhecidos como a assinatura do método no Java.

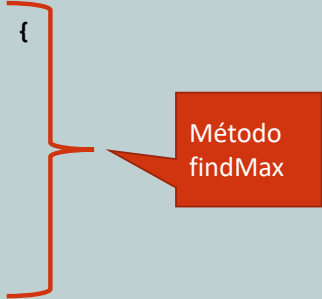
Propriedades do Método

Propriedade do Método	Descrição
modifiers	São opcionais e podem ser públicos, privados, protegidos ou ficar em branco
dataType	Tipo de dados, como int
methodName	Nome do seu método
parameterList	Lista de nomes de parâmetros separados por vírgula com os respectivos tipos de dados Cada parâmetro é listado primeiro com o tipo de dados e, depois, com o nome
methodBody	Conjunto de instruções que executam a tarefa
return	Palavra-chave que envia o valor do resultado de volta para o código que chamou o método

Todos esses termos serão explicados detalhadamente nas próximas seções deste curso. Por enquanto, basta saber para que eles são usados.

Método findMax()

```
public class TestFindMax {  
    /** Main method */  
    public static void main(String[] args) {  
        int i = 5;  
        int j = 2;  
        int k = findMax(i, j);  
        System.out.println("The maximum between " + i + " and " + j +  
                           " is " + k);  
    } //fim do método main  
  
    /** Return the max between two numbers */  
    public static int findMax(int num1, int num2) {  
        int result;  
        if (num1 > num2)  
            result = num1;  
        else  
            result = num2;  
        //fim if  
        return result;  
    } //fim do método findMax  
}
```



Quando estiver lendo o código nesta seção, tente identificar as seções com que você está familiarizado. Neste exemplo, existe uma declaração de variável (`int i = 5;`), uma chamada de método (`findMax(i, j);`) e uma instrução If Else. Você usou todas essas instruções no Alice 3.

Classes no Alice 3

- Um Dálmata é um cão
- Quando um objeto Dálmata é adicionado a uma cena, ele tem as propriedades da classe Dálmata:
 - quatro patas, duas orelhas, pêlo branco com bolas pretas e a capacidade de andar

Uma classe é uma especificação, como um plano gráfico ou um padrão, de como construir um objeto



ORACLE
Academy

JF 2-14
Métodos e Classes Java

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.

13

Uma classe é o código que define quais informações sobre um objeto e a função dele devem ser armazenadas. Uma classe torna-se um objeto quando atribuímos valores a ela. Um cão é uma classe, mas um Dálmata é um objeto ou uma "instância" da classe cão porque ele é um cão com valores específicos.

Classes no Java

- O primeiro grupo:
 - Opcional; refere-se à visibilidade de outros objetos
 - public significa visível em qualquer lugar
 - O padrão é package ou visible dentro do pacote atual apenas

```
[ "public" ] [ "abstract" | "final" ] "class" Class_name  
    [ "extends" object_name ] [ "implements" interface_name ]  
{  
    // properties declarations  
    // behavior declarations  
}
```

Os próximos slides são bastante técnicos. Portanto, não se preocupe se não os entender por completo. As informações serão descritas mais detalhadamente mais adiante no curso.

Classes no Java

- O segundo grupo:
 - Opcional; define se a classe pode ser herdada ou estendida por outras classes
 - As classes abstratas devem ser estendidas e as classes finais nunca podem ser estendidas por herança
 - O padrão indica que a classe pode ou não ser estendida, a critério dos programadores

```
[ "public" ] [ "abstract" | "final" ] "class" Class_name  
    [ "extends" object_name ] [ "implements" interface_name ]  
"{"  
    // properties declarations  
    // behavior declarations  
"}"
```

Classes no Java

- Class_name é o nome da classe
- A terceira opção de extensões está relacionada à herança
- A quarta opção de implementações está relacionada às interfaces

```
[ "public" ] [ "abstract" | "final" ] "class" Class_name  
    [ "extends" object_name ] [ "implements" interface_name ]  
"{  
    // properties declarations  
    // behavior declarations  
}"
```


Código para Criar uma Classe Gato no Java

```
class Cat{
    int catAge;
    String catName;

    public Cat(String name){
        catName = name;
    }//fim do constructor

    public void setAge(int age){
        catAge = age;
    }//fim do método setAge

    public int getAge(){
        return catAge;
    }//fim do método getAge

    public static void main(String []args){
        Cat myCat = new Cat("Garfield");
        myCat.setAge(6);
        System.out.println("Cat age: " + myCat.getAge());
    }//fim do método main
}//fim da classe Cat
```

ORACLE
Academy

JF 2-14
Métodos e Classes Java

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.

17

Esta classe tem duas variáveis para o nome e a idade do gato. Ela também tem dois métodos: um que permite definir a idade do gato e um que informa a idade do gato. Toda variável em uma classe deve ter um método set and get anexado a ela.

Instâncias no Alice 3

- Quando você adiciona um objeto a uma cena, isso cria uma instância da classe

Um objeto é uma instância de uma classe



ORACLE
Academy

JF 2-14
Métodos e Classes Java

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.

18

Você pode ter várias instâncias da mesma classe dentro de uma cena. Neste exemplo, há uma instância de um gato, mas há várias instâncias da carta de baralho. Todas as instâncias de uma classe terão o mesmo comportamento.

Instâncias no Java

- No método principal, a instância myCat da classe gato é criada

```
class Cat{
    int catAge;
    String catName;
    ...
    ...
    ...
    public static void main(String []args){
        Cat myCat = new Cat("Garfield");
        myCat.setAge(6);
        System.out.println("Cat age: " + myCat.getAge());
    }//fim do método main
}//fim da classe Cat
```

Você cria uma instância de uma classe usando a nova instrução no Java.

Criar uma Instância de uma Classe

- É criada uma instância de uma classe usando o novo operador, seguido do nome da classe

```
class Cat{
    int catAge;
    String catName;
    ...
    ...
    ...
    public static void main(String []args){
        Cat myCat = new Cat("Garfield");
        myCat.setAge(6);
        System.out.println("Cat age: " + myCat.getAge());
    } //fim do método main
} //fim da classe Cat
```

Criar uma nova instância

Estruturas de Controle

- As estruturas de controle permitem que você altere a ordem em que as instruções são executadas nos seus programas
- E tanto o Alice quanto o Java permitem esses tipos de estruturas de controle

Tipo	Descrição	Exemplo
Estruturas de Controle de Decisão	Permitem que você escolha seções específicas do código para serem executadas	if ... then ... else
Estruturas de Controle de Repetição	Permitem que você execute seções específicas de código repetidamente	loop while

Você acessou as estruturas de controle no Alice arrastando-as para a barra do code editor na parte inferior da tela.

Estrutura de Controle if

- As estruturas de controle if são instruções que permitem a você selecionar e executar blocos de código específicos enquanto ignora outras seções
- Essas estruturas têm o seguinte formato

```
if (boolean_expression) {  
    doSomething();  
}  
else {  
    doSomethingElse();  
} //endif
```

Uma instrução If será executada se a condição for verdadeira. Lembre-se de que a instrução Else é opcional e não precisa ser incluída.

Estrutura de Controle if - Exemplo

- Por exemplo:
 - Se um aluno obtiver uma pontuação de **90%** ou mais no teste, então dê a ele **"A"**
 - Se um aluno obtiver uma pontuação que é maior ou igual a 80% e menor que **90%**, dê a ele **"B"**
 - Se um aluno obtiver uma pontuação que é maior ou igual a 65% e menor que **80%**, dê a ele **"C"**
 - Se um aluno obtiver uma pontuação inferior a **65%**, dê a ele **"F"**

Código da Estrutura de Controle if - Exemplo

```
public class Grade {  
    public static void main( String[] args ){  
        //variable declaration section  
        double grade = 89.0;  
  
        if( grade >= 90 ){  
            System.out.println( "A" );  
        }  
        else if( (grade < 90) && (grade >= 80) ){  
            System.out.println("B" );  
        }  
        else if( (grade < 80) && (grade >= 65) ){  
            System.out.println("C" );  
        }  
        else{  
            System.out.println("F");  
        } //fim if  
    } //fim do método main  
} //fim da classe Grade
```

Neste código, assim que for encontrada uma condição verdadeira no programa, ele parará de verificar as instruções Else If aninhadas. Esta é a finalidade de aninhá-las na instrução Else.

Estrutura de Controle while

- Uma estrutura de controle while, ou loop while, é uma instrução ou um bloco de instruções Java que permite a você executar blocos específicos de código repetidamente desde que alguma condição seja satisfeita
- A condição é testada antes de cada loop
- formato do loop while:

```
while( boolean_expression ){  
    statement1;  
    statement2;  
    . . .  
}
```

Estrutura de Controle while - Exemplo

```
class WhileDemo {  
    public static void main(String[] args){  
        //variable declaration section  
        int count = 1;  
  
        while (count < 11) {  
            System.out.println("Count is: " + count);  
            count++;  
        } //fim while  
  
    } //fim do método main  
} //fim da classe WhileDemo
```

Em um loop while, você precisa assegurar que a condição final possa ser atendida; caso contrário, você terminará em um loop infinito. É por isso que o valor da contagem é aumentado em uma unidade cada vez que o código entra em loop. Assim, quando o valor da contagem exceder onze, a condição será falsa e o loop terminará.

Entrada e Saída

- Os programas Java funcionam em muitas plataformas
- Eles podem ser programas simples, que são executados na linha de comando, ou podem ter interfaces gráficas de usuário complexas
- Quando estiver aprendendo a programar, você criará programas que usam a linha de comando para entrada e saída exclusivamente
- Ao adquirir mais experiência, você poderá construir interfaces gráficas de usuário e aprender sobre as bibliotecas necessárias para construí-las

Código de Entrada e Saída

- Exemplo de impressão na tela

```
System.out.println("Hello World!");
```

- `System.out.println()` é um trecho de código que fica disponível automaticamente para todo programa Java

É semelhante ao procedimento Say ou Think no Alice 3.

Código de Entrada e Saída - Exemplo

- Exemplo de entrada do usuário usando o pacote `java.util.Scanner`

```
import java.util.Scanner;
public class ReadString {
    public static void main (String[] args) {
        System.out.print("Enter your name: ");
        Scanner in = new Scanner(System.in);
        String userName = null;
        userName = in.nextLine();
        System.out.println("Thanks for the name, " + userName);
    } //fim do método main
} //fim da classe ReadString
```

Veja a seguir que um objeto da classe `Scanner` está sendo usado para permitir a entrada. A maioria dos programas exigirá interação com o usuário; portanto, a entrada e a saída são muito importantes. No Alice, as funções `getWholeNumber`, `getDecimalNumber` etc. realizaram a entrada.

Terminologia

- Estes são os principais termos usados nesta lição:
 - Classe
 - Estrutura de controle
 - Estrutura de controle IF
 - Instância
 - Método
 - Estrutura de controle WHILE

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Descrever um método, uma classe ou uma instância
 - Descrever um cenário em que uma estrutura de controle IF seria usada
 - Descrever um cenário em que uma estrutura de controle WHILE seria usada
 - Reconhecer a sintaxe de um método, uma classe, uma função ou um procedimento
 - Descrever a entrada e a saída



