



#Reskilling4Employment
Software Developer

Paradigmas de Programação

Aula 02 - Objetos

Vitor Santos



Conteúdos

1. Palavras Reservadas
2. Criação de Objetos
3. Métodos Construtores
4. Variáveis de Instância
5. Palavras Reservadas Usadas
6. Links Úteis

Palavras Reservadas

abstract	continue	for	new	switch
assert***	default	goto*	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum****	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp**	volatile
const*	float	native	super	while

*** not used**

**** added in 1.2**

***** added in 1.4**

****** added in 5.0**

Objeto

- Um objecto é uma instância de uma classe.
- Construído a partir da especificação de uma classe.
- Com uma identidade única.
- Em Java para criarmos um objeto usamos o operador de alocação `new` como podemos ver no slide seguinte.




- Os objectos são alocados dinamicamente e o operador de alocação é o `new`:

Sintaxe de Alocação

`new <data-type> (<arguments> ...)`

**`<data-type> <variable_name> =
 new <data-type> (<arguments> ...)`**



```
public class Cao {  
    String nome= "Fido";  
    String raca = "Boerboel"  
    String latido = "woooof!";  
    int idade = 6;  
}
```

Os atributos nome, latido e idade da classe Dog são denominados variáveis de instância

```
class ADogsLife {  
    public static void main(String[] args) {  
        Cao fido = new Cao( );  
        System.out.println(fido.nome);  
        System.out.println(fido.raca);  
        System.out.println(fido.latido);  
        System.out.println(fido.idade);  
    }  
}
```

new é o operador
de alocação.

Cao() é o método
construtor.

Consola IDE

```
> Fido  
> Boerboel  
> woof!  
> 6
```

```
class ADogsLife {  
    public static void main(String[] args) {  
  
        Cao fido = new Cao();  
        Cao spot = new Cao();  
  
        System.out.println(fido.nome);  
        System.out.println(fido.raca);  
        System.out.println(fido.latido);  
        System.out.println(fido.idade);  
  
        System.out.println(spot.nome);  
        System.out.println(spot.raca);  
        System.out.println(spot.latido);  
        System.out.println(spot.idade);  
  
    }  
}
```

Consola IDE

```
> Fido  
> Boerboel  
> woof!  
> 6  
> Fido  
> Boerboel  
> woof!  
> 6
```



Métodos Construtores

- Os construtores de uma classe são todos os métodos especiais que são declarados na classe tendo por identificador o nome exacto da classe.



Métodos Construtores

- Os métodos construtores podem ter **argumentos** de **qualquer tipo de dados** e cujo objetivo é criar instâncias de tal classe que sejam de imediato manipuláveis.
- Os construtores, dado criarem instâncias de uma dada classe, não têm obviamente, que especificar qual o resultado, pois será sempre uma instância da respectiva classe.
- É possível e útil construir mais do que um construtor de instâncias de uma dada classe.



```
public class Cao {  
    String nome = "Fido";  
    String raca = "Boerboel";  
    String latido = "woooof!";  
    int idade = 6;  
  
    Cao(String nome_temp) {  
        this.nome = nome_temp;  
    }  
}
```

Cao (String nome_temp)
é o método construtor

```
class ADogsLife {  
    public static void main(String[] args) {  
  
        Cao fido = new Cao("Fido");  
        Cao spot = new Cao("Spot");  
  
        System.out.println(fido.nome);  
        System.out.println(fido.raca);  
        System.out.println(fido.latido);  
        System.out.println(fido.idade);  
  
        System.out.println(spot.name);  
        System.out.println(spot.raca);  
        System.out.println(spot.latido);  
        System.out.println(spot.idade);  
    }  
}
```



```
public class Cao {  
    String nome= "Fido";  
    String raca = "Boerboel";  
    String latido = "woooof!";  
    int idade = 6;  
  
    Cao(String nome_temp) {  
        this.nome = nome_temp;  
    }  
}
```

```
    Cao(String nome_temp, String raca_temp, String latido_temp, int idade_temp){  
        this.nome = nome_temp;  
        this.raca = raca_temp;  
        this.latido = latido_temp;  
        this.idade = idade_temp;  
    }  
}
```

Criamos um novo método construtor

```
class ADogsLife {  
    public static void main(String[] args) {  
  
        Cao spot = new Cao("Spot" , "Pastor Alemão","ruff!", 3);  
  
        System.out.println(spot.nome);  
        System.out.println(spot.raca);  
        System.out.println(spot.latido);  
        System.out.println(spot.idade);  
    }  
}
```

Variáveis de Instância

- Até agora aprendemos a definir instâncias de objectos mas cada objeto pode também ter variáveis de estado.
- Para cada instância de Dog do exemplo anterior podemos ter alguma variabilidade de objetos de acordo com as suas características tais como cor do pêlo, peso, etc.
- A estas variáveis damos o nome de variáveis de instância.



```
public class Cao {
```

```
    String nome;
```

```
    String raca;
```

```
    String latido;
```

```
    int idade;
```

Variáveis de Instância

```
    Cao(String nome_temp) {
```

```
        this.name = nome_temp;
```

```
    }
```

```
    Cao(String nome_temp, String raca_temp, String latido_temp, int idade_temp){
```

```
        this.nome = nome_temp;
```

```
        this.raca = raca_temp;
```

```
        this.latido = latido_temp;
```

```
        this.idade = idade_temp;
```

```
    }
```

```
}
```

Métodos construtores

Palavras Reservadas

abstract	continue	for	new	switch
assert***	default	goto*	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum****	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp**	volatile
const*	float	native	super	while

*** not used**

**** added in 1.2**

***** added in 1.4**

****** added in 5.0**



#Reskilling4Employment
Software Developer

Paradigmas de Programação

Aula 02 - Objetos

Vitor Santos

