



#Reskilling4Employment
Software Developer

Paradigmas de Programação

Aula 01 - Classes

Vitor Santos



Conteúdos

1. Palavras Reservadas
2. Conceito de Classe
3. Exemplos
4. Modificadores de Acesso
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

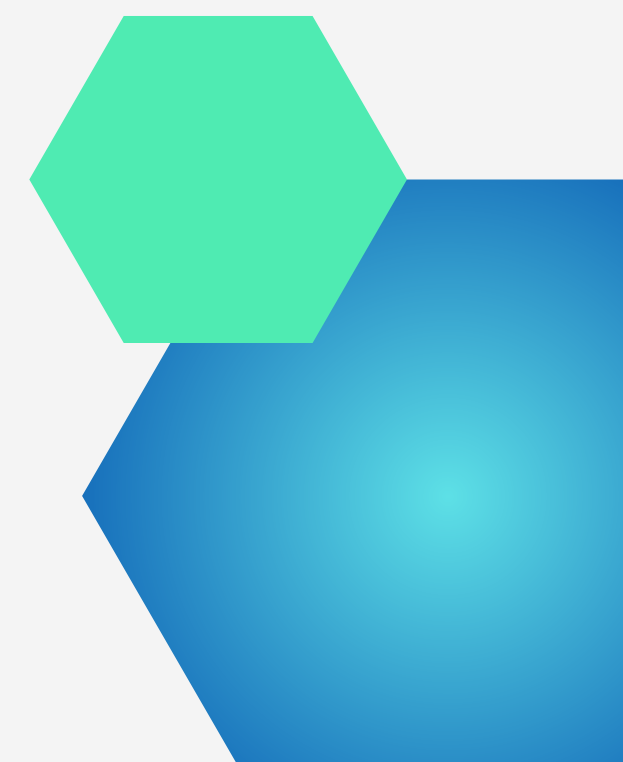
** added in 1.2

*** added in 1.4

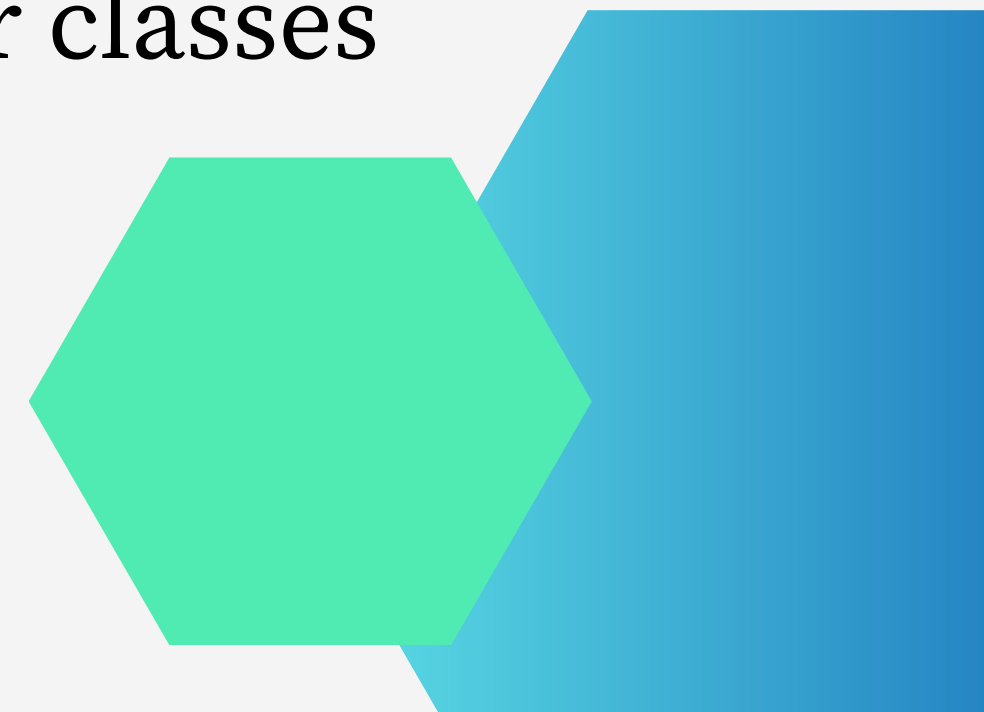
**** added in 5.0


Classes

- Uma classe é um módulo de software que impõe uma dada estrutura.
- Por norma, é uma especificação de um conjunto de características (atributos/propriedades) e/ou de um conjunto de comportamentos (métodos/funções e/ou procedimentos).
- Recorrendo a uma classe é possível reutilizar código.




- Reutilizar código definido noutras classes.
- Uma linguagem de programação baseada no paradigma **Object-Oriented**, como é o caso da linguagem Java, possui um conjunto de **APIs - Application Programming Interfaces** - que por sua vez são constituídas por classes.
- É normal as APIs não serem constituídas apenas por classes
- Os objetos são construídos a partir das classes.






```
public class Carro {  
    String marca;  
    int passageiros;  
    int portas;  
}
```



```
public class Pessoa {  
    String nome;  
    int idade;  
    double altura;  
    double peso;  
    int bi;  
}
```




```
public class Cao {  
    String nome;  
    String latido;  
    int idade;  
}
```


Controlo de acesso (ou visibilidade).

- Em Java podem ser especificados quatro tipos de acesso (ou de visibilidade), dos quais três usando palavras reservadas que se signam por modificadores de acesso, mais especificamente:

public, private, protected e nenhum



- 
- Os modificadores de acesso são palavras reservadas que alteram a visibilidade das entidades (classes, interfaces, atributos, métodos, tipos enumerados, blocos de código) às quais estão associados

Modificador de Acesso: `private`

- Os membros que possuem o modificador `private` só podem ser acedidos dentro do código da própria classe onde foram assinados.
- O modificador `private` é o que permite uma menor visibilidade.



O modificador `private` é aplicável a:

- Atributos e métodos
- Classes, interfaces, e tipos enumerados que não sejam `Top Level (Outer)`, isto é, a classes aninhadas, interfaces aninhadas, e tipos enumerados aninhados*.

*Aninhado (`Nested`) - Definido dentro de outro.



Modificador de Acesso: `protected`

- Os membros que possuem o modificador `protected` só podem ser acessados dentro do código onde foram assinados ou em entidades que derivam daquela a que os mesmos pertencem.
- A aplicabilidade deste modificador é semelhante à do modificador `private`, o seu estudo será aprofundado quando se abordar o conceito de `herança`.

Modificador de Acesso: nenhum

- Os membros que **não possuem** um dos modificadores private, protected, ou public possuem visibilidade ao nível da **package**.
- A ausência de modificador é aplicável a qualquer entidade (atributos, métodos, classes, interfaces, e tipos enumerados).
- A ausência de modificador permite um grau de visibilidade igual ao que o modificador **public** permite mas só para uma dada **package**.



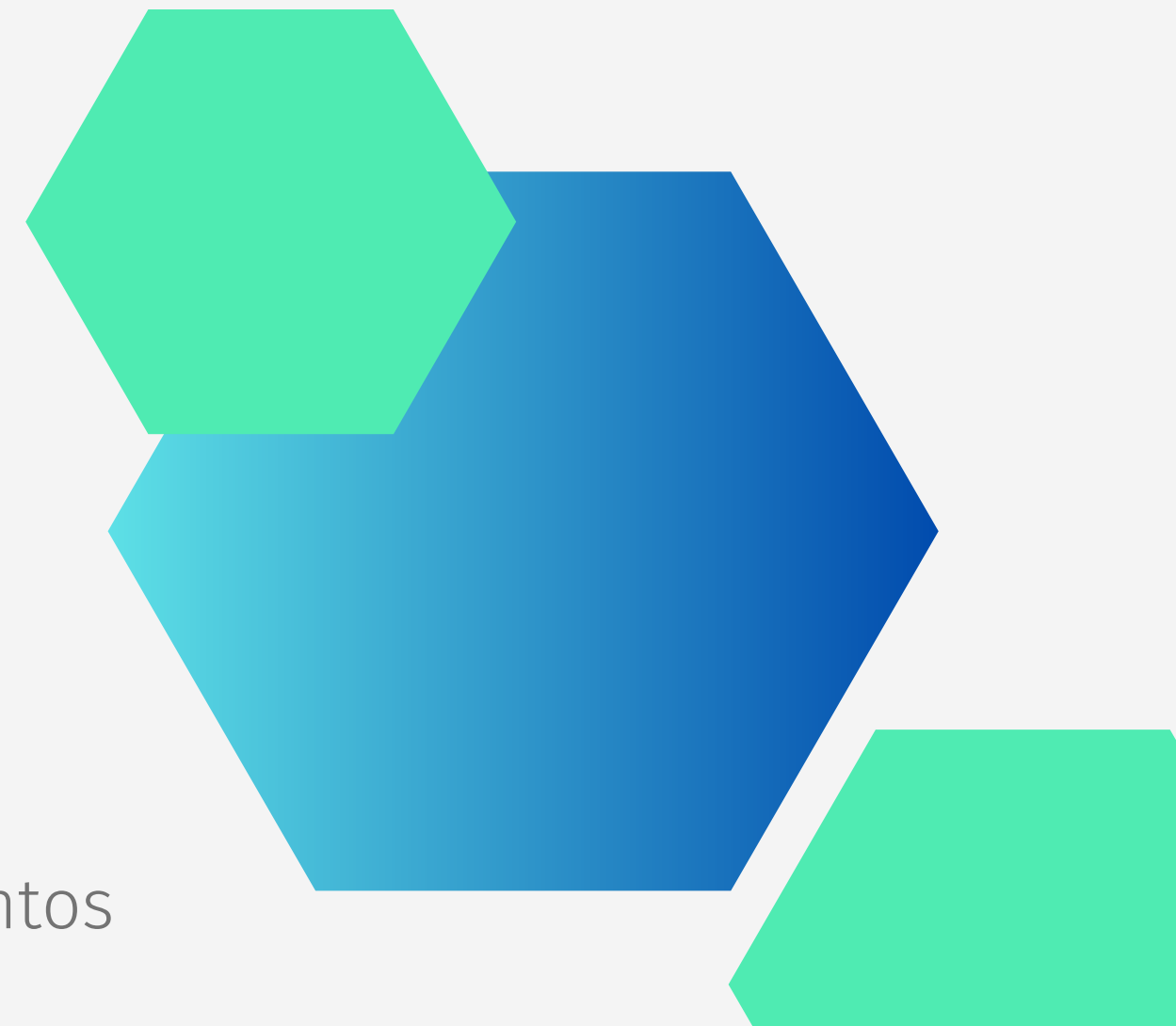
- Os membros que possuem o modificador `public` podem ser acedidos a partir de qualquer código, independentemente da `package`.
- O modificador `public` é aplicável a qualquer entidade (atributos, métodos, classes, interfaces, e tipos enumerados).
- O modificador `public` é o que permite uma **maior** visibilidade.

Modificador static

- Não é um modificador de acesso.
- O modificador `static` altera o membro ao qual é aplicado, esses membros passam a chamar-se `membros de classe`.
- Os membros de classe não necessitam de uma instância (objeto) para serem acedidos, para tal basta o nome da classe.



- **Um membro de classe é compartilhado por todas as instâncias de uma dada classe.**
- Os membros que não possuem o modificador static chamam-se **membros de instância** e, como é óbvio, **só podem ser acedidos através de um(a) objeto(instância).**



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 01 - Classes

Vitor Santos

