

# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Modificadores de visibilidade



# Você já perguntou sobre...



2

- ... o significado dos modificadores de visibilidade “-” e “+” nas classes modeladas com UML?
- ... a utilidade dos modificadores de acesso “**private**” e “**public**” em Java?

```
public class Aluno ex
    private String ma
    private String cu

    public void setMa
        this.matricul
```

- cp  
- da  
- no

+ se  
+ ge  
+ se

# Visibilidade e modificadores de acesso



3

Notação visual	Modificador de acesso	A parte é visível...
+	public	dentro da própria classe e para qualquer outra classe
-	private	somente dentro da própria classe
#	protected	somente dentro do próprio pacote e das subclasses em outros pacotes
~	package	somente dentro da própria classe e das classes dentro do mesmo pacote

# Visibilidade “package” em Java



4

- ❑ Para representar a visibilidade “package”, você não deve especificar um modificador de acesso
  - ▣ Quando não é especificado um modificador de acesso, o Java assume que a visibilidade é do tipo “package”

```
package br.univali.poo.calculo;
```

```
class ClasseVisibilidadePackage {  
    int atributoVisibilidadePackage;  
    ...  
}
```

# Visibilidade “package” em Java



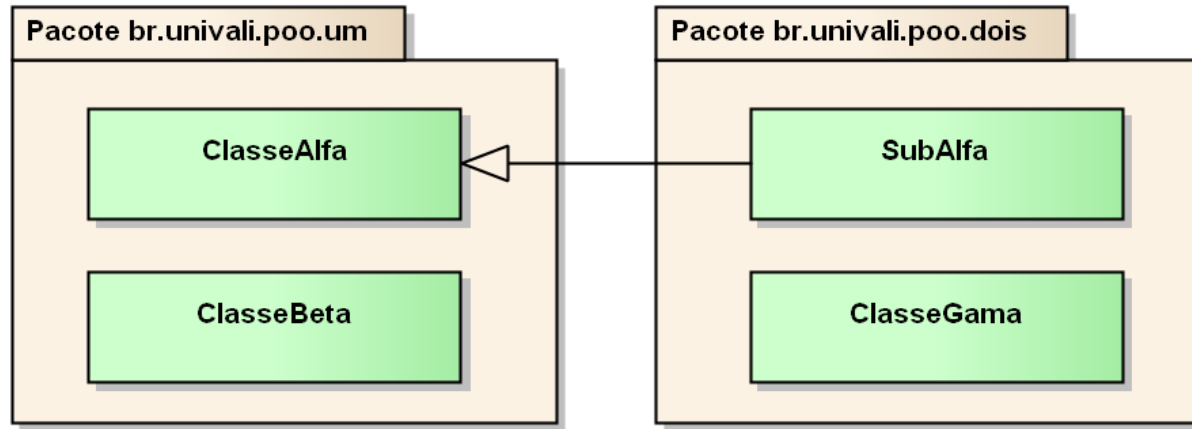
5

- ❑ Para cada arquivo “.java” deve haver, pelo menos, uma classe “public” com o mesmo nome do arquivo
- ❑ As demais classes declaradas no mesmo arquivo devem ser “package”
  - ▣ Ou seja, não devem especificar um modificador de acesso

# Visibilidade de atributos e operações



6



Modificador na ClasseAlfa	ClasseAlfa	ClasseBeta	SubAlfa	Gama
public	S	S	S	S
protected	S	S	S	N
sem modificador	S	S	N	N
private	S	N	N	N

# Na prática, adotaremos...



- As classes serão públicas
  - ▣ Uma única classe em cada arquivo “.java”
  - ▣ Logo, o nome da classe e do arquivo deve ser o mesmo
- **Atributos** serão **sempre privados**
- Operações podem ser “públicas”, “privadas” ou “protegidas”

# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Modificadores de visibilidade

