

ROTEIRO

- Sobrecarga
 - De métodos
 - De construtores
- Package
- Import
- Modificadores

A declaração package

- Quando um programador utiliza classes feitas por outro programador, surge um problema clássico: Como escrever duas classes com o mesmo nome?

A declaração package

- A linguagem de programação da tecnologia Java fornece a declaração package como uma forma de agrupar classes relacionadas. A declaração package tem o seguinte formato:
 - `package <top_pkg_name>[.<sub_pkg_name>]*;`
- Você pode indicar que as classes em um arquivo fonte pertencem a determinado pacote usando a declaração package como por exemplo:
 - `package br.com.submarino.informatica;`
 - `public class Teclado {...`

A declaração package

- A declaração package, quando existir, deve aparecer no início do arquivo fonte. Antes dela você só pode incluir espaços em branco e comentários, nada mais. Só é permitida uma declaração de pacote e ela governa todo o arquivo fonte. Se um arquivo fonte da tecnologia Java não tiver uma declaração de pacote, a(s) classe(s) declarada(s) naquele arquivo pertence(m) ao pacote não nomeado (default).

A declaração package

- Os nomes dos pacotes são hierárquicos e separados por pontos. É comum que os elementos do nome do pacote sejam informados em letras minúsculas. Entretanto, o nome da classe geralmente começa com letra maiúscula e você pode colocar a primeira letra de cada palavra adicional em maiúscula, para diferenciar as palavras dentro do nome da classe.
- Se não estiver incluída no arquivo nenhuma declaração de pacote, todas as classes declaradas em tal arquivo “pertencem” ao pacote default (ou seja, um pacote sem nome).

A declaração import

- A declaração import diz ao compilador onde encontrar as classes a serem utilizadas.
- A declaração import tem o seguinte formato:
 - `import <pkg_name>[.<sub_pkg_name>].<class_name>;`
 - ou
 - `import <pkg_name>[.<sub_pkg_name>].*;`
- Na verdade, o nome do pacote(por exemplo, `br.com.submarino`) é parte do nome das classes dentro do pacote. Você poderia se referir a classe `Telefone` como `br.com.submarino.Teclado`, ou então poderia usar a declaração import e apenas o nome da classe `Teclado`.
- OBS: As declarações import devem preceder todas as declarações de classe.

A declaração import

- O fragmento de arquivo a seguir usa a declaração import:
- `package shipping.reports;`
- `import shipping.domain.*;`
- `import java.util.List;`
- `public class VehicleCapacityReport {...`
- Lembre-se que a declaração import é usada para colocar classes de outros pacotes à disposição da classe usada no momento.
- A declaração import especifica a classe que você deseja acessar. Por exemplo, se você quiser incluir somente a classe `Writer` (do pacote `java.io`), use:
- `import java.io.Writer;`

A declaração import

- Se você quiser ter acesso a todas as classes de um pacote, use “.*”. Por exemplo, para acessar todas as classes no pacote java.io use:
- `import java.io.*;`
- OBS: Uma declaração import não faz o compilador carregar nada a mais na memória de trabalho.
- *Os pacotes são simplesmente diretórios.

A declaração import

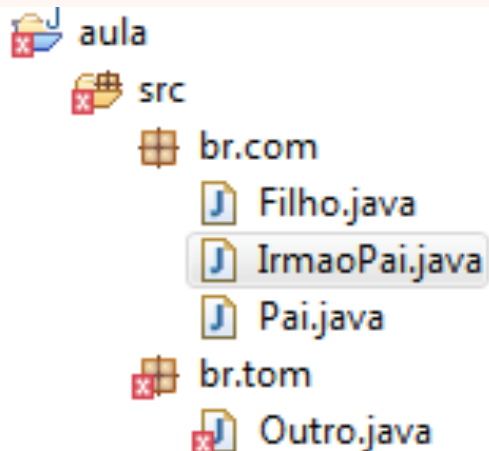
- Uma variável ou método marcados com o modificador `protected` é, na verdade, mais acessível do que uma variável que tenha acesso `default`.
- Um método ou variável `protected` é acessível a partir de métodos em classes que são membros do mesmo pacote, e a partir de qualquer método em qualquer subclasse.
- Uma variável ou método marcados com o modificador `public`, podem ser acessados universalmente.

MODIFICADORES

Cr terios de acessibilidade

Modificador	Mesma classe	Mesmo Pacote	Subclasse	Universo
private	sim	n�o	n�o	n�o
default	sim	sim	n�o	n�o
protected	sim	sim	sim	n�o
public	sim	sim	sim	sim

MODIFICADORES



```
package br.com;
```

```
public class Pai {
    void metodoDefault() {

    }
    protected void metodoProtected() {

    }
    public void metodoPublico() {

    }
}
```

```
package br.com;
```

```
public class IrmãoPai {
    public static void main(String[] args) {
        Pai p = new Pai();
        p.metodoDefault();
        p.metodoProtected();
    }
}
```

```
package br.com;
```

```
public class Filho extends Pai{
    public static void main(String[] args) {
        Filho f = new Filho();
        f.metodoDefault();
        f.metodoProtected();
    }
}
```

```
package br.tom;
```

```
import br.com.Pai;
```

```
public class Outro extends Pai{
    public static void main(String[] args) {
        Outro o = new Outro();
        o.metodoProtected();
        o.metodoDefault();
    }
}
```

A declaração import

- As variáveis e os métodos podem estar em um dos quatro níveis de acesso: `public`, `protected`, `default` ou `private`. As classes podem estar no nível `public` ou `default`.
- Uma variável ou método marcados como `private`, somente podem ser acessados por métodos que sejam membros da mesma classe.
- Uma variável, um método ou uma classe têm acessibilidade `default` quando não possuem um modificador de proteção explícito como parte de sua declaração. Essa acessibilidade significa que o acesso é permitido a partir de qualquer método em classes que são membros do mesmo pacote que o destino. Isso muitas vezes é chamado de “`package-friendly`”.