Fundamentos de Java

Organização do Código Java

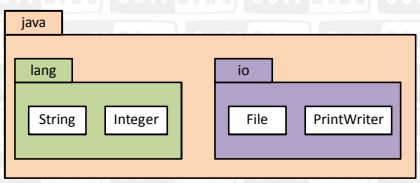
SoftBlue
www.softblue.com.br

Tópicos Abordados

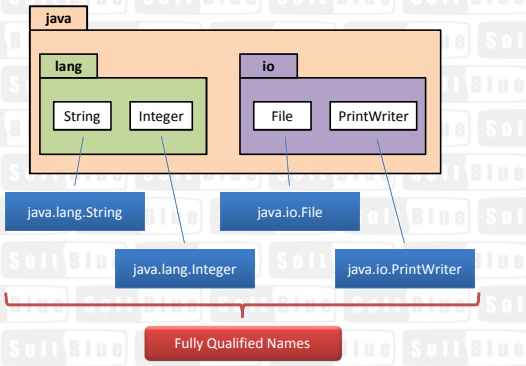
- Pacotes
  - Mapeamento de um pacote
  - Convenção no nome dos pacotes
- O uso do *import*
- Visibilidades *package* e *protected*
- Javadoc
- Criação de arquivos JAR
- Convenções do código Java

Pacotes

- As classes podem ser organizadas em pacotes
- Objetivos
  - Organização
  - Possibilitar que classes possam ter o mesmo nome



Pacotes



Mapeamento de um Pacote

- O nome do pacote é mapeado para um diretório no sistema de arquivos

Pacote	Diretório
curso	curso
com.treinamento.java	com/treinamento/java

Classes Dentro de um Pacote

- Devem estar na estrutura correta no sistema de arquivos
- O seu pacote deve ser declarado usando o *package*
  - Esta declaração deve ser feita na primeira linha

```
package com.treinamento.java;

public class Exemplo {
      ...
}
```

Convenção de Nome dos Pacotes

- Contém apenas letras minúsculas
- Normalmente é definido um nome que não terá conflito com pacotes criados por terceiros

Como Encontrar as Classes

- Usar o *fully qualified name*

```
com.java.Exemplo e = new com.java.Exemplo();
```

- Usar o *import*

```
import com.java.Exemplo;

...

Exemplo e = new Exemplo();
```

Uso do *import*

- Os imports devem ser usados logo após o uso do *package* (caso exista)
- É possível importar todas as classes de um pacote

```
import com.java.*;
```

- Importar classe por classe é preferível por questões de organização de código

A Visibilidade Package

- Quando não definimos modificadores para classes, atributos, métodos ou construtores, eles assumem a visibilidade **package** por padrão
- Package significa que todas as classes do mesmo pacote possuem o acesso

A Visibilidade *protected*

- Quando um método, atributo ou construtor possui o modificador **protected**
  - As subclasses têm acesso ao atributo
  - Outras classes do mesmo pacote também têm acesso

Exportando o Javadoc

- O Javadoc é a documentação do seu projeto
  - Classes, construtores, métodos, pacotes, etc.
- Gerado a partir de comentários no código
- O Java possui uma ferramenta para exportar o Javadoc
- O Javadoc da API do Java 7 pode ser encontrado em:
  - http://docs.oracle.com/javase/7/docs/api

Criando JARs

- Java ARchive
- Conjunto de classes compactadas no padrão ZIP, mas com extensão JAR
- O JAR é um componente de software
  - Agrupa código comum
  - Possui relativa independência
- O JDK possui um utilitário para gerar arquivos JAR

Criando JARs

Convenções do Código Java

- Códigos escritos em Java devem seguir algumas convenções
  - Esta padronização ajuda na manutenção do código
  - Facilita a leitura do código por outros desenvolvedores
- Documentação oficial da Oracle
  - http://www.oracle.com/technetwork/java/codeconv-138413.html

Convenções de Nomes

- Classe e interface
  - Deve ser um substantivo
  - Começa com letra maiúscula e segue a notação *camel case*

```
class Estado
interface DVDPlayer
class CasaDeMadeira
```

Convenções de Nomes

- Método
  - Deve ser um verbo
  - Começa com letra minúscula e segue a notação *camel case*

```
void comer()
int getIdade()
void processarPagamento()
```

Convenções de Nomes

- Variável
  - Deve ter um nome que descreva seu propósito de forma clara
  - Começa com letra minúscula e segue a notação *camel case*

```
double nota
int qtdeItens
Casa casaDaPpraia
```

Convenções de Nomes

- Constante
  - Todas as letras são maiúsculas e o caractere "_" é usado para separar as palavras
  - A regra se aplica a elementos de enums e atributos com os modificadores *static final*

```
int VALOR
```

```
String ARQUIVO_CONFIG
```

```
enum Prioridade {
    ALTA,
    MEDIA,
    BAIXA
}
```

Convenções para Blocos de Código

- Convenção para estruturas que usam "{" e "}" para delimitar um bloco de código

```
if (valor > 10) {
    //…
}
```

```
for (int i = 0; i < 10; i++) {
    //…
}
```

```
public class Caneta {
    //...
}
```

Use a identação adequada

Colocando em Prática...

Agora que você já aprendeu a teoria, acesse as vídeo-aulas práticas e pratique os assuntos abordados neste módulo!

Clique aqui para acessar as vídeo-aulas práticas