



ThoughtWorks®

# ESCREVENDO MENOS CÓDIGO COM *LOMBOK*

# Quem somos nós



***Thai Braga***  
*(DEV) @ Thoughtworks*



***Pati Ghiraldelli***  
*(DEV) @ Thoughtworks*

# "Lombo-O-Quê?"

## UM POUCO DE CONTEXTO



# O QUE É *LOMBOK*?

## ONDE VIVE, O QUE FAZ, COMO SE ALIMENTA?

- Java
  - Linguagem verbosa
  - Muito código necessário até para aplicações pequenas
- Biblioteca Lombok
  - Redução de *boilerplate*
  - Código gerado na compilação
  - Integração com IDEs
  - Maior produtividade, legibilidade e manutenibilidade



# INSTALANDO O LOMBOK

## INCLUINDO DEPENDÊNCIAS

- Maven

```
<dependencies>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.16.16</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

- Gradle

```
compileOnly "org.projectlombok:lombok:1.16.16"
```

# INSTALANDO O *LOMBOK*

## CONFIGURANDO SUA IDE

- Eclipse
- IntelliJ Idea
- Netbeans
- MyEclipse
- Spring Tools Suite
- JBoss Developer Studio
- Visual Studio Code

# ESCREVENDO MENOS CÓDIGO

## GETTERS E SETTERS

```
public class GetterSetterExample {  
  
    private int age = 10;  
    private String name;  
  
    public int getAge() {  
        return age;  
    }  
  
    public void setAge(int age) {  
        this.age = age;  
    }  
  
    protected void setName(String name) {  
        this.name = name;  
    }  
}
```

```
import lombok.AccessLevel.PROTECTED;  
import lombok.Getter;  
import lombok.Setter;  
  
public class GetterSetterExample {  
  
    @Getter @Setter private int age = 10;  
  
    @Setter(PROTECTED) private String name;  
  
}
```

# ESCREVENDO MENOS CÓDIGO

## CONSTRUTORES (com nenhum ou com todos os argumentos)

```
public class Example<T> {  
    private int x, y;  
    private T desc;  
  
    protected Example(int x, int y, T desc) {  
        this.x = x;  
        this.y = y;  
        this.desc = desc;  
    }  
  
    public Example() {  
    }  
}
```

```
import lombok.AccessLevel.PROTECTED;  
import lombok.AllArgsConstructor;  
  
@AllArgsConstructor (access = PROTECTED)  
public class Example<T> {  
    private int x, y;  
    private T desc;  
}
```



# ESCREVENDO MENOS CÓDIGO

## CONSTRUTORES (com apenas alguns argumentos obrigatórios)

```
public class Example<T> {  
    private int x, y;  
    private T desc;  
  
    public static <T> Example<T> of(T desc) {  
        return new Example<T>(desc);  
    }  
}
```

```
import lombok.AccessLevel;  
import lombok.RequiredArgsConstructor;  
  
@RequiredArgsConstructor (staticName = "of")  
public class Example<T> {  
    private int x, y;  
    @NonNull private T desc;  
}
```

# ESCREVENDO MENOS CÓDIGO

## MÉTODOS *equals()* & *hashCode()*

```
public class Example {  
    private transient int transientVar = 10;  
    private String name;  
    private double score;  
    private String[] tags;  
    private int id;  
  
    @Override public boolean equals(Object o) {  
        if (o == this) return true;  
        if (!(o instanceof Example)) return false;  
        Example other = (Example) o;  
        if (!this.getName().equals(other.getName()))  
            return false;  
        if (Double.compare(this.score, other.score) != 0)  
            return false;  
        if (!Arrays.deepEquals(this.tags, other.tags))  
            return false;  
        return true;  
    }  
  
    @Override public int hashCode() { /* ... */ }
```

```
import lombok.EqualsAndHashCode;  
  
@EqualsAndHashCode  
public class Example {  
  
    private transient int transientVar = 10;  
    private String name;  
    private double score;  
    private String[] tags;  
    @EqualsAndHashCode.Exclude private int id;  
  
}
```

# ESCREVENDO MENOS CÓDIGO

## MÉTODO *toString()*

```
import java.util.Arrays;

public class Example {

    private static final int STATIC_VAR = 10;
    private String name;
    private String[] tags;
    private int id;

    @Override
    public String toString() {
        return "Example(" + this.name + "," +
                this.shape + "," +
                Arrays.deepToString( this.tags) +
                ")";
    }
}
```

```
import lombok.ToString;

@ToString
public class Example {

    private static final int STATIC_VAR = 10;
    private String name;
    private String[] tags;
    @ToString.Exclude private int id;
}
```

# ESCREVENDO MENOS CÓDIGO

## UMA ÚNICA NOTAÇÃO PARA POJOS

```
public class DataExample {  
    private final String name;  
    private int age;  
    private double score;  
    private String[] tags;  
    public DataExample(String name) {  
        this.name = name;  
    }  
    public String getName() {  
        return this.name;  
    }  
    void setAge(int age) {  
        this.age = age;  
    }  
    public int getAge() {  
        return this.age;  
    }  
    public void setScore(double score) {  
        this.score = score;  
    }  
    public double getScore() {  
        return this.score;  
    }  
    public String[] getTags() {
```

```
import lombok.AccessLevel;  
import lombok.Setter;  
import lombok.Data;  
  
@Data  
public class DataExample {  
  
    private final String name;  
    @Setter(AccessLevel.PACKAGE) private int age;  
    private double score;  
    private String[] tags;  
  
}
```

# ESCREVENDO MENOS CÓDIGO

## CONSTRUINDO OBJETOS

```
public class Example {
    private String name;
    private int age;
    private Set<String> occupations;
    public static ExampleBuilder builder() {
        return new ExampleBuilder();
    }
    public static class ExampleBuilder {
        private String name;
        private int age;
        private ArrayList<String> occupations;
        public ExampleBuilder name(String name) {
            this.name = name;
            return this;
        }
        public ExampleBuilder age(int age) {
            this.age = age;
            return this;
        }
        public ExampleBuilder occupation(String occupation) {
            if (this.occupations == null)
                this.occupations = new java.util.ArrayList<String>();
            this.occupations.add(occupation);
            return this;
        }
    }
}
```

```
import lombok.Builder;
import lombok.Singular;
import java.util.Set;

@Builder
public class Example {

    private String name;
    private int age;
    @Singular private Set<String> occupations;

}
/*
Example.builder().name("Fulano da Silva")
    .age(25)
    .occupation("Consultant")
    .occupation("Developer").build();
*/
```

# ESCREVENDO MENOS CÓDIGO

## CHECANDO ARGUMENTOS NULOS

```
public class NonNullExample extends Something
{
    private String name;

    public NonNullExample(Person p) {
        super("Hello");
        if (p == null) {
            throw new NullPointerException(
                "p is marked @NonNull but is null" );
        }
        this.name = p.getName();
    }
}
```

```
import lombok.NonNull;

public class NonNullExample extends Something
{
    private String name;

    public NonNullExample( @NonNull Person p) {
        super("Hello");
        this.name = p.getName();
    }
}
```

# ESCREVENDO MENOS CÓDIGO

## LOGANDO SÓ QUE REALMENTE IMPORTA

```
import org.slf4j.*;

public class LogExample {
    private static final Logger log =
        LoggerFactory.getLogger(LogExample.class);

    public static void main(String... args) {
        log.error("Something else is wrong");
    }
}
```

```
@Slf4j

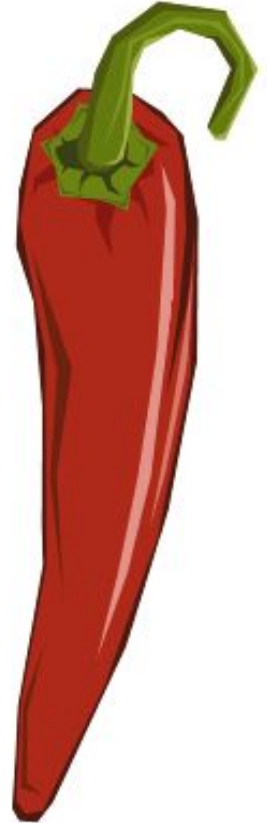
public class LogExample {

    public static void main(String... args) {
        log.error("Something else is wrong");
    }

}
```

O que acharam?

**Prontas pra sair usando Lombok?**





# VALEU :)

***Patricia Guiraldelli***

*psantosg@thoughtworks.com*

***Thaiane Braga***

*thbraga@thoughtworks.com*

**ThoughtWorks®**