Javadoc

Design de classes

- Especificação da classe
  - Oque um objeto deve/pode realizar.
  - O usuário da classe deve conhecer (do contrário, a classe não é “útil”).
- Implementação da classe
  - Como a especificação é realizada.
  - Não é importante para o usuário da classe.
  - Deve ser escondida do usuário da classe, pode conter informação/conhecimento proprietário.

Javadoc

Compartilhando/escondendo informação

- Em algumas linguagens, separamos o que pode ser compartilhado e o que deve ser escondido em arquivos distintos:
  - C/C++:
    - Informação compartilhada: arquivos de interface **`*.h`**
    - Infotmação escondida: arquivos de implementação **`*.c`**/ **`*.cpp`**
- Java
  - Usa uma abordagem distinta
  - Como a informação de “interface” está explícita na implementação, utilizamos um programa, javadoc, que gera automaticamente a documentação da classe.

Javadoc

- Documentação gerada em arquivos **`.html`**
- Os arquivos de implementação podem conter comentários javadoc:
  - Começam com **`/**`**
  - Terminam com **`*/`**
- Podemos incluir alguns tags especiais:
  - @param
  - @return
  - @throws

Javadoc

```
/**
 Classe utilizada para exemplificar o uso do Javadoc
 */
 public class Teste{
    private int a, b;
    private String nome;
    public double db;

    /**
     Inicializa objeto
     @param a descrição do parametro a
     @param b descrição do parametro b
     @param nome descrição do parametro nome
    */
    public Teste(int a, int b, String nome){
    }
```

Javadoc

```
   /**
      @return descrição do retorno do método
   */
    public int metodoA(int b, int c, double db){
        return 0;
    }
 }
```

Javadoc

```
/**
 * Comentários para a classe ClasseA
 */
public class ClasseA extends Teste{
   private int i_a,i_b;
   public double db_a;
   /**
    Metodo publico metodoPublico da classe ClasseA
    @throws Exception
    */
   public void metodoPublico()throws Exception{
   }
   /**
    metodo não compartilhado
    */
    private int metodoPrivado(){
        return 0;
   }
}
```