



JavaDoc

Práctica 3: JavaDoc

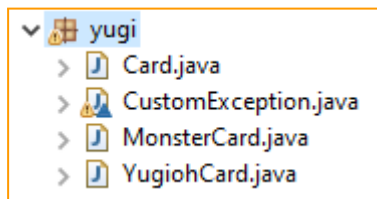
Alumnos

Edgar Chiquillo



1. Código realizado	1
1.1. Página principal	2
1.2. Objeto	2
1.3. Objeto heredado	3
1.4. Excepción	3
2. Creación JavaDoc	4
2.1. Primera fase (exportación)	4
2.2. Export wizard	4
2.3. JavaDoc Generation	5
2.4. Console	5
2.5. Resultado de la creación	6
3. Visualización	6
3.1. Proyectos creados	7
3.2. Información de Card	7
3.3. Información de MonsterCard	11
3.4. Información de YugiohCard	15

1. Código realizado



1.1. Página principal

```
1 package yugi;
2
3 /**
4  * Clase principal para ejecutar el programa
5  */
6 public class YugiohCard {
7     public static void main(String[] args) {
8         // Crear una carta generica
9         Card genericCard = new Card(1, "Carta genérica", "Esta es la carta genérica");
10
11         // Crear una carta de monstruo de Yu-Gi-Oh
12         MonsterCard monster = new MonsterCard("HÉROE Elemental Neos",
13         "¡Un nuevo HÉROE Elemental ha llegado desde el Neo Espacio! Cuando inicia una Fusión de Contacto con un Neo "
14         + "Espacial sus desconocidos poderes son liberados.",
15         7, "Luz", "Gerrero", 2500, 2000);
16
17         // Mostrar informacion de las cartas
18         System.out.println("Información de la carta genérica:");
19         System.out.println(genericCard.getInfoWithId(1));
20
21         System.out.println("\nInformación de la carta de monstruo de Yu-Gi-Oh:");
22         monster.displayInfo();
23
24         try {
25             genericCard.simulateException();
26         } catch (CustomException e) {
27             System.out.println("Excepción atrapada: " + e.getMessage());
28         }
29     }
30 }
31 }
```

1.2. Objeto

```
1 package yugi;
2
3 /**
4  * Clase principal que representa una carta genérica
5  */
6 public class Card {
7     /**
8      * Nombre de la carta.
9      */
10     private String name;
11     /**
12      * Descripción de la carta.
13      */
14     private String description;
15     /**
16      * Identificador de la carta.
17      */
18     private int cardId;
19 }
```

1.3. Objeto heredado

```
1 package yugi;
2
3 /**
4  * Clase secundaria que hereda de Card y representa una carta de monstruo.
5  */
6 public class MonsterCard extends Card {
7     /**
8      * Nivel del monstruo.
9      */
10    private int level;
11    /**
12     * Atributo del monstruo.
13     */
14    private String attribute;
15    /**
16     * Tipo del monstruo.
17     */
18    private String type;
19    /**
20     * Ataque del monstruo.
21     */
22    private int attack;
23    /**
24     * Defensa del monstruo.
25     */
26    private int defense;
```

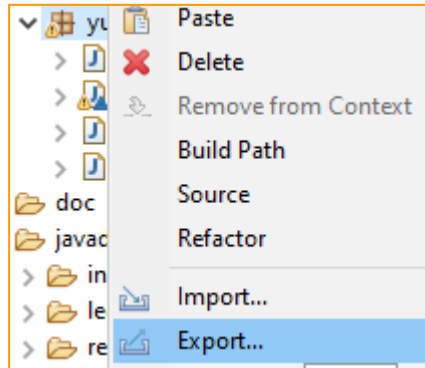
1.4. Excepción

```
1 package yugi;
2
3 /**
4  * Clase secundaria que gestionara las excepciones
5  */
6 class CustomException extends Exception {
7
8     public CustomException(String message) {
9         super(message);
10    }
11 }
```

2. Creación JavaDoc

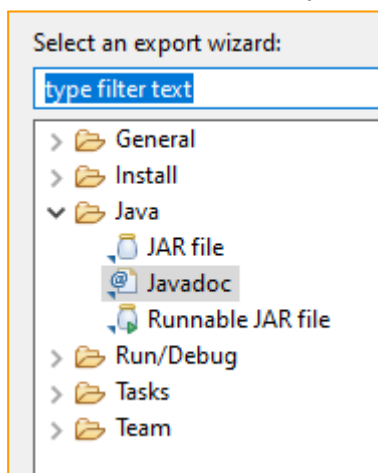
2.1. Primera fase (exportación)

Para generar el javaDoc le daremos “click” derecho al proyecto y *exportar*



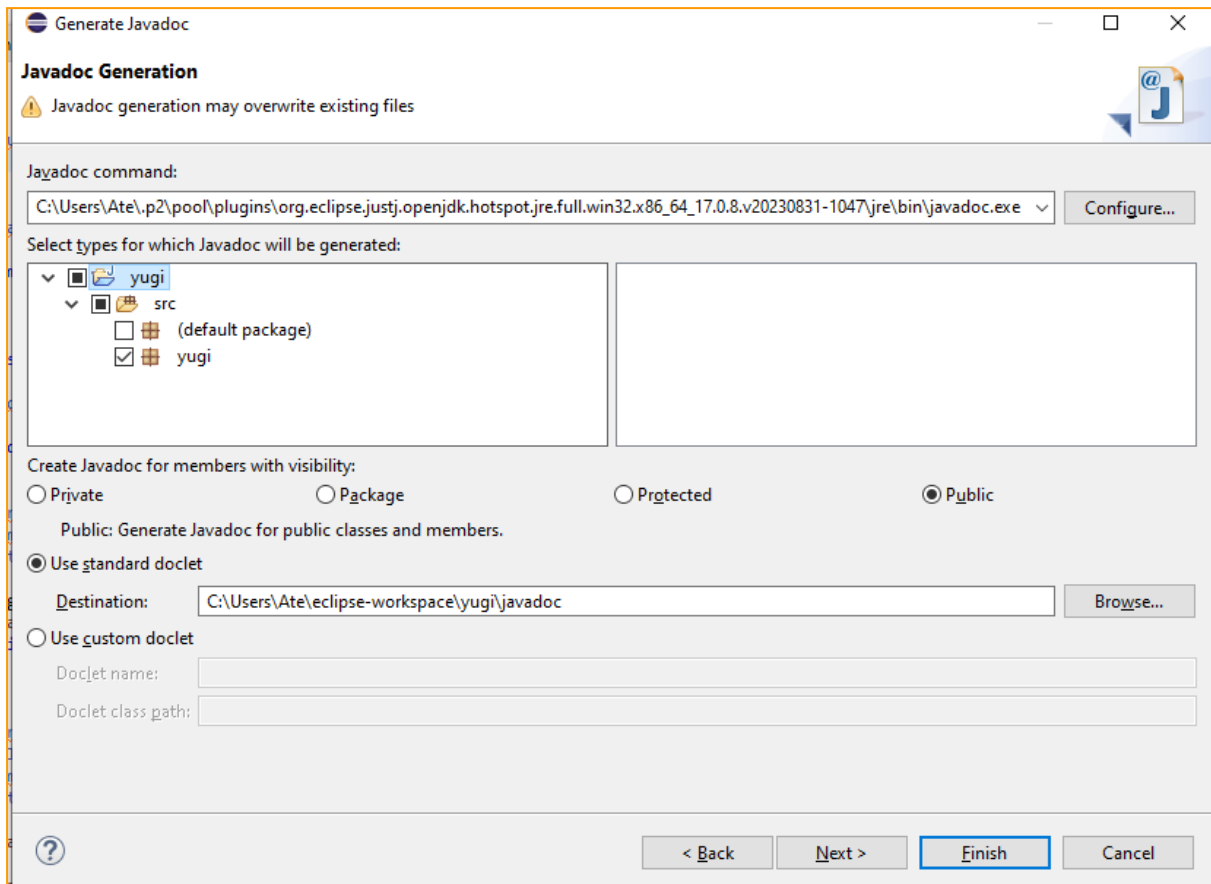
2.2. Export wizard

Seleccionaremos *Java* y *JavaDoc* y le daremos a next



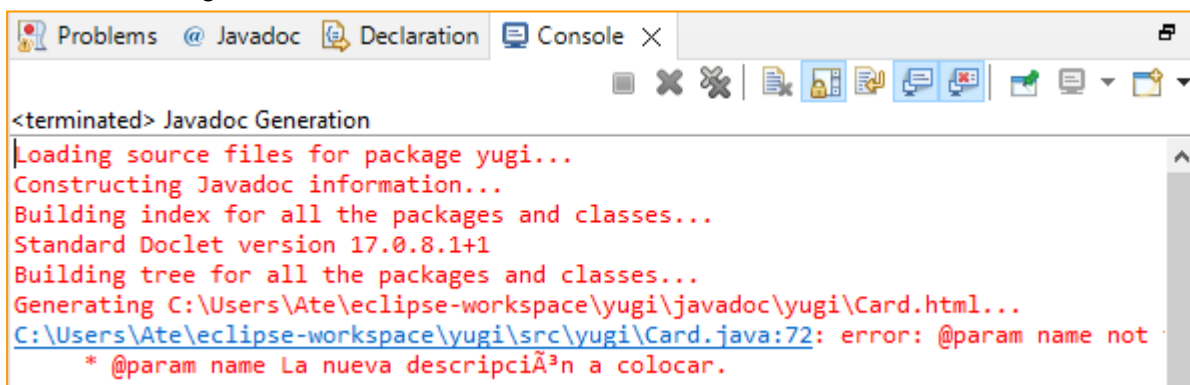
2.3. Javadoc Generation

Elegimos todo lo que necesitamos para generarlo, la *ruta* en la que se generará y *finish*

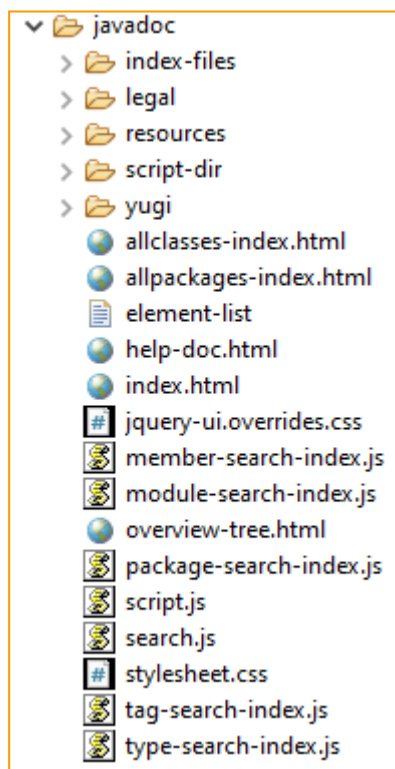


2.4. Console

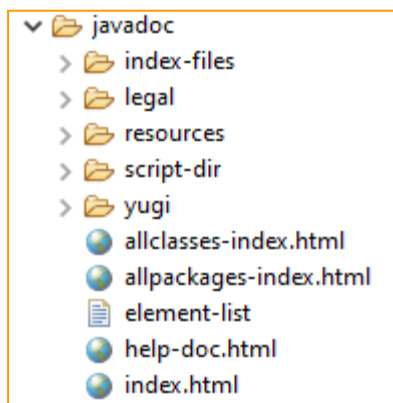
La *consola* nos mostrará los *procesos* que va ejecutando. Aquí nos muestra un error por culpa de un acento, pero, se va a generar igualmente.



2.5. Resultado de la creación



3. Visualización



Package yugi	
package yugi	
Classes	
Class	Description
Card	Clase principal que representa una carta genérica
MonsterCard	Clase secundaria que hereda de Card y representa una carta de monstruo.
YugiohCard	Clase principal para ejecutar el programa

3.1. Proyectos creados

Tendremos un apartado por cada proyecto que tenemos

Package yugi	
package yugi	
Classes	
Class	Description
Card	Clase principal que representa una carta genérica
MonsterCard	Clase secundaria que hereda de Card y representa una carta de monstruo.
YugiohCard	Clase principal para ejecutar el programa

3.2. Información de Card

Package yugi
Class Card
java.lang.Object ↗ yugi.Card
Direct Known Subclasses:
MonsterCard
public class Card extends Object ↗

Clase principal que representa una carta genérica

Constructor Summary

Constructors

Constructor	Description
Card(int cardId, String ↗ name, String ↗ description)	Constructor para una carta genérica.
Card(String ↗ name, String ↗ description)	Constructor para una carta genérica.

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	<code>displayInfo()</code>	
<code>String</code>	<code>getCardId()</code>	Get the name of the object.
<code>String</code>	<code>getDescription()</code>	Get the name of the object.
<code>String</code>	<code>getInfoWithId(int cardId)</code>	Funci3n que toma un par3metro de entrada y devuelve un valor.
<code>String</code>	<code>getName()</code>	Get the name of the object.
void	<code>setCardId(int cardId)</code>	Set the name of the object.
void	<code>setDescription(<code>String</code> description)</code>	Set the name of the object.
void	<code>setName(<code>String</code> name)</code>	Set the name of the object.
void	<code>simulateException()</code>	Funci3n que lanza una excepci3n personalizada.

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Details

Card

```
public Card(String name,  
           String description)
```

Constructor para una carta gen3rica.

Parameters:

name - Nombre de la carta.

description - Descripci3n de la carta.

Card

```
public Card(int cardId,  
           String name,  
           String description)
```

Constructor para una carta gen3rica.

Parameters:

cardId - Id que va ha tener la carta carta.

name - Nombre de la carta.

description - Descripci3n de la carta.

Method Details

getName

```
public String getName()
```

Get the name of the object.

Returns:

El nombre del objeto.

setName

```
public void setName(String name)
```

Set the name of the object.

Parameters:

name - El nuevo nombre a colocar.

getDescription

```
public String getDescription()
```

Get the name of the object.

Returns:

El nombre de la descripción.

setDescription

```
public void setDescription(String description)
```

Set the name of the object.

Parameters:

name - La nueva descripción a colocar.

getCardId

```
public String getCardId()
```

Get the name of the object.

Returns:

El identificador de la descripción.

setCardId

```
public void setCardId(int cardId)
```

Set the name of the object.

Parameters:

name - El nueva identificador a colocar.

getInfoWithId

```
public String getInfoWithId(int cardId)
```

Función que toma un parámetro de entrada y devuelve un valor.

Parameters:

cardId - El ID de la carta a buscar.

Returns:

Información de la carta con el ID especificado o un mensaje de error si no se encuentra.

simulateException

```
public void simulateException()  
        throws yugi.CustomException
```

Función que lanza una excepción personalizada.

Throws:

yugi.CustomException - Cuando se lanza una excepción personalizada.

displayInfo

```
public void displayInfo()
```

3.3. Información de MonsterCard

Package `yugi`

Class `MonsterCard`

`java.lang.Object`
`yugi.Card`
`yugi.MonsterCard`

```
public class MonsterCard  
extends Card
```

Clase secundaria que hereda de `Card` y representa una carta de monstruo.

Constructor Summary

Constructors

Constructor

`MonsterCard(String name, String description, int level, String attribute, String type, int attack, int defense)`

Description

Constructor para crear la carta de monstruo

Method Summary

All Methods	Instance Methods	Concrete Methods
-------------	------------------	------------------

Modifier and Type	Method	Description
boolean	<code>canAttack()</code>	Función que verifica si el monstruo puede atacar.
boolean	<code>canDefend()</code>	Función que verifica si el monstruo puede defender.
void	<code>displayInfo()</code>	
int	<code>getAttack()</code>	Get the name of the object.
<code>String</code>	<code>getAttribute()</code>	Get the name of the object.
int	<code>getDefense()</code>	Get the name of the object.
int	<code>getLevel()</code>	Get the name of the object.
<code>String</code>	<code>getType()</code>	Get the name of the object.

Methods inherited from class `yugi.Card`

`getCardId`, `getDescription`, `getInfoWithId`, `getName`, `setCardId`, `setDescription`, `setName`, `simulateException`

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Details

MonsterCard

```
public MonsterCard(String name,  
                   String description,  
                   int level,  
                   String attribute,  
                   String type,  
                   int attack,  
                   int defense)
```

Constructor para crear la carta de monstruo

Parameters:

name - El nombre del monstruo.

description - La descripción de este.

level - El nivel del monstruo.

attribute - El atributo del monstruo.

type - Tipo del monstruo.

attack - Ataque del monstruo.

defense - Defensa del monstruo.

Method Details

getLevel

```
public int getLevel()
```

Get the name of the object.

Returns:

El nivel del objeto.

getAttribute

```
public StringⒶ getAttribute()
```

Get the name of the object.

Returns:

El atributo del objeto.

getType

```
public StringⒶ getType()
```

Get the name of the object.

Returns:

El typo del objeto.

getAttack

```
public int getAttack()
```

Get the name of the object.

Returns:

El ataque del objeto.

getDefense

```
public int getDefense()
```

Get the name of the object.

Returns:

La defensa del objeto.

displayInfo

```
public void displayInfo()
```

Overrides:

`displayInfo` in class `Card`

canAttack

```
public boolean canAttack()
```

Función que verifica si el monstruo puede atacar.

Returns:

true si el monstruo puede atacar, false en caso contrario.

canDefend

```
public boolean canDefend()
```

Función que verifica si el monstruo puede defender.

Returns:

true si el monstruo puede defender, false en caso contrario.

3.4. Información de YugiohCard

Package `yugi`

Class `YugiohCard`

`java.lang.Object`
`yugi.YugiohCard`

```
public class YugiohCard  
extends Object
```

Clase principal para ejecutar el programa

Clase principal para ejecutar el programa

Constructor Summary

Constructors

Constructor	Description
<code>YugiohCard()</code>	

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method	Description
static void	<code>main(String[] args)</code>	

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Details

`YugiohCard`

```
public YugiohCard()
```

Method Details

`main`

```
public static void main(String[] args)
```