

Sesión 19: Sobre la Práctica Obligatoria

Programación 2

Ángel Herranz

Abril 2019


Universidad Politécnica de Madrid

En el capítulo de hoy

 Simulacro de entrega de práctica obligatoria

El Club del Lector

(by Jaime Ramírez y Ángel Lucas)

 Sistema de entrega

<http://triqui2.fi.upm.es/entrega/>

(by profes + sonarque)

 Bibliotecas necesarias

CorrectorOffline + ConsolaIO + tads1819 +
JUnit

El Club del Lector

- Ejercicio usado en otros grupos
- Tarea en Moodle: “Simulacro entrega practica obligatoria”

Enunciado

+

Solución **incorrecta**

 ¡Leer, corregir y entregar!

Leer el enunciado



y luego preguntas

UML: diagramas de clases

- **UML**: *Unified Modelling Language*
- Los diagramas de clase permiten representar visualmente diseños de clases

<i>nombre de la clase</i>
<i>+ atributo público : su tipo</i>
<i>- atributo privado : su tipo</i>
<i># atributo protected : su tipo</i>
<i>+ método público(tipo parámetros) : tipo resultado</i>
<i>- método privado(tipo parámetros) : tipo resultado</i>
<i># método protected(tipo parámetros) : tipo resultado</i>



UML Distilled (*Martin Fowler*)

Código de apoyo (.java y .class)

```
src/  
    ee1/  
        Lector.java (ee1.Lector)  
        Libro.java (ee1.Libro)  
    test/  
        TestLector.java (test.TestLector)  
        TestLibro.java (test.TestLibro)  
bin/ (donde irán a parar los .class)
```

Código de apoyo (bibliotecas .jar)

lib/

consolaIO.jar

Fecha.jar

hamcrest-core-1.3.jar

junit-4.12.jar

tads1819.jar

Código de apoyo (documentación)

```
doc/  
  consola/  
    index.html  
    ...  
  fecha/  
    index.html  
    ...  
  tads1819/  
    index.html  
    ...
```

La documentación de JUnit en

<https://junit.org/junit4/>

Línea de comandos

- Ir al directorio donde hayas descomprimido el .zip, pongamos `Sesion19`
- Compilación (Unix y Windows):

```
javac -d bin -cp src:lib/Fecha.jar:lib/hamcrest-core-1.3.jar:lib/junit-4.12.jar -s src src/test/TestLibro.java
```

```
javac -d bin -cp src:lib\Fecha.jar;lib\hamcrest-core-1.3.jar;lib\junit-4.12.jar -s src src\test\TestLibro.java
```

- Ejecución de los tests (Unix y Windows):

```
java -cp bin:lib/Fecha.jar:lib/hamcrest-core-1.3.jar:lib/junit-4.12.jar org.junit.runner.JUnitCore test.TestLibro
```

```
java -cp bin;lib\Fecha.jar;lib\hamcrest-core-1.3.jar;lib\junit-4.12.jar org.junit.runner.JUnitCore test.TestLibro
```

Línea de comandos (disección) i

javac **Compilar**

-d bin **Directorio donde dejar los .class**

-cp **Classpath**

src **Otros ficheros .java**

:lib/Fecha.jar **Biblioteca de fechas**

:lib/hamcrest-core-1.3.jar **Biblioteca auxiliar**

:lib/junit-4.12.jar **Biblioteca JUnit 4**

-s src **Dónde buscar los .java**

src/test/TestLibro.java **Fichero a compilar**

Línea de comandos (disección) ii

```
java Ejecutar  
  -cp Classpath  
      bin .class  
      :lib/Fecha.jar Biblioteca de fechas  
      :lib/hamcrest-core-1.3.jar Biblioteca auxiliar  
      :lib/junit-4.12.jar Biblioteca JUnit 4  
org.junit.runner.JUnitCore Programa principal  
test.TestLibro argumento de JUnitCore
```

IDE (ej. Eclipse o IntelliJ IDEA)

- Crear un nuevo proyecto
- Descomprimir el `.zip`
- Copiar lo que hay en el directorio `src` del `.zip` en el directorio `src` del proyecto
- Añadir las bibliotecas en el directorio `lib` del `.zip` al proyecto
- El IDE suele entender de JUnit y te permite ejecutar los tests desde algún botón

Entregar con errores

`http://triqui2.fi.upm.es/entrega/`

Corregir errores (entregas sucesivas)

- Libro: en `esIgual` falta mirar la fecha
- Lector: en `getLibroLeido` falta `pos-1`
- Lector: en `leerLibro` falta llamar a `prestado()`
- Lector: en `esIgual` falta mirar el último libro del histórico

 hasta el final de la clase