

Xarxes Curs 2016/2017

Pràctica 1: Introducció a Java - Wget

1 Objectiu

L'arquitectura Java fa possible crear aplicacions totalment portables a diferents plataformes. Les seves característiques la fan adequada per al desenvolupament d'aplicacions en entorns de xarxa. Des de 1995, any d'aparició pública de Java, aquest llenguatge ha anat guanyant acceptació entre la comunitat de programadors i fabricants de programari.

Amb aquesta pràctica l'alumne adquirirà els coneixements bàsics de programació en Java. Aprenderà a utilitzar la seva API i serà capaç d'utilitzar tècniques avançades de programació en Java com la programació paral·lela (*Multi-thread*) i les comunicacions en Internet.

2 La nostra aplicació Wget

Cada grup programarà una aplicació, *Wget*, per descarregar diversos documents binaris de forma simultània a través de la *Internet*.

Els documents els descarregareu mitjançant el protocol HTTP (*Hyper Text Transfer Protocol*).

Els documents, un cop descarregats, els desareu a disc.

La vostra aplicació *Wget*, serà similar a aplicacions com el *GNU Wget*¹ o el *cURL*², però amb una funcionalitat molt més reduïda.

¹<http://www.gnu.org/software/wget/wget.html>

²<http://curl.haxx.se/>

3 Com executar l'aplicació

Haureu d'executar l'aplicació fent servir els paràmetres:

```
-f <fitxerURLs> [-a] [-z] [-gz]
```

Noteu que els símbols '[' i ']' indiquen que el paràmetre és opcional. Per tant: podem fer la crida només indicant `-f <fitxerURLs>`, podem fer-la afegint-hi algun/s dels parametres opcionals, o podem fer-la afegint-los tots.

-f <fitxerURLs> Indica quin és el fitxer de configuració d'URLs a descarregar que s'ha de fer servir. Modifiqueu '<fitxerURLs>' pel path del fitxer.

-a Aplicar el filtre `ascii`.

-z Aplicar el filtre `zip`.

-gz Aplicar el filtre `gzip`.

4 Funcionament de la nostra aplicació Wget

Se us proporciona un fitxer `urls.txt` amb un contingut com el següent:

```
http://deic.uab.es/index.html
http://www.google.es
http://www.w3schools.com/php/php_intro.asp
http://deic.uab.es/images/deic-ct.png
```

Aquest fitxer l'heu de fer servir com a fitxer de configuració d'URLs a descarregar. Per a cada una de les URL's del fitxer, l'aplicació haurà de descarregar el recurs corresponent a la URL, aplicar potser algun filtre (depenent dels paràmetres), i desar-lo a disc en un fitxer.

Els fitxers es desen en local al directori actual des d'on heu executat la vostra aplicació. A la secció 4.1 s'especifica quin ha de ser el nom per a cada fitxer.

Tingueu molt present que el format de la *URL* a descarregar sempre serà:

```
http://servidor[:port]/[path/][fitxer]
```

Recordeu que els paràmetres entre els símbols '[' i ']' són opcionals.

4.1 Classes de la API de Java que utilitzarem

Per a llegir les URLs del fitxer de URLs us serà molt fàcil si utilitzeu la classe de la API de Java `java.io.BufferedReader`.

Cal que utilitzeu la classe `java.net.URL` per a realitzar les comunicacions, i s'haurà de vigilar les excepcions per informar a l'usuari de quina és la **causa exacta** de l'error en cas de produir-se.

A partir de l'objecte `java.net.URL`, associat a la *URL* que us heu de descarregar, obtindreu un fluxe de lectura: `java.io.InputStream` des d'on anireu llegint *bytes* i escrivint-los en el fitxer que desareu en local.

Per a fer l'escriptura a disc us caldrà la classe `java.io.FileOutputStream`.

5 Com anomenar el fitxer que desem a disc.

El nom d'aquest fitxer, d'aquest recurs que ens hem descarregat, ha de correspondre amb el que s'especifica en la URL. És a dir, si en el fitxer `urls.txt` especifiquem la *URL*:

```
http://deic.uab.es/index.html
```

l'aplicació ha de descarregar-se la web `http://deic.uab.es/index.html` i desar, en local, el fitxer *index.html* que conté aquesta web.

Si una de les *URLs* a descarregar no conté el nom del fitxer, com a

```
http://www.uab.es/ o
```

```
http://www.deic.uab.es
```

es guardarà el contingut del document al fitxer `index.html`.

Encara que dos o més *URLs* especifiquin el mateix nom de fitxer, en cap cas es podran sobre escriure a disc els fitxers descarregats. Es buscarà algun mecanisme per resoldre aquest problema. Una possible solució seria afegir sufixes al nom del fitxer.

Idealment, a partir de les *URLs* del fitxer de URLs anterior, tindríem els fitxers: *index.html*, *index2.html*, *php_intro.asp* i *deic-ct.png*. Però per a fer la pràctica més senzilla, podeu mantenir un contador i associar sempre un sufixe al nom del fitxer que genereu. Així els noms dels fitxers anteriors quedarien: *index1.html*, *index2.html*, *php_intro3.asp* i *deic-ct4.png*.

6 Filtres

Quan invoquem a l'aplicació `Wget` ho podem fer tot indicant, com a paràmetres de l'aplicació, un conjunt de filtres. Aquests filtres s'aplicaran sobre el fitxer que s'està descarregant. Aquests filtres s'aplicaran *on the fly*. És a dir, **en cap cas heu de crear fitxers intermitjos**. Aquests filtres modificaran el format del fitxer que estem descarregant. Vegeu-los a les següents seccions.

6.1 Filtre `ascii`: paràmetre `-a`

Si s'ha indicat l'opció `-a` es treuran els *tags* del fitxer que s'ha especificat en la *URL* que ens estem descarregant. **Aquest filtre només és aplicable als fitxers que siguin de tipus *text/html*.**

Aquest filtre consisteix en ignorar les parts del document amb la sintaxis:

```
< {byte}* >
```

És a dir fragments del document entre els símbols: '`<`' i '`>`'. Per exemple, el següent document HTML:³

```
<html>\n
<head><title>Document de xarxes</title>\n
<body>\n
Aquest és el contingut del document\n
</body>\n
</html>\n
```

En aplicar el filtre, us quedaria el text:

```
\n
Document de xarxes\n
Aquest és el contingut del document.\n
\n
\n
```

En aplicar aquest filtre, al fitxer que desareu a disc li afegireu l'extensió `.asc`, i.e: `index.html.asc`.

Cal que l'algorisme per descarregar el contingut d'una *URL* sigui independent d'aquesta conversió. Per a tal efecte, cal que seguiu les següents passes:

1 No heu de crear fitxers intermitjos.

³Fem servir el símbol `\n` per a representar el salt de línia.

2 És molt important que no dupliqueu codi. Per a aconseguir això cal que implementeu la classe:

```
class HtmlToAsciiInputStream extends FilterInputStream
```

Caldrà que feu ús del **polimorfisme** de la classe

```
java.io.InputStream
```

Recordeu que una classe derivada com:

```
HtmlToAsciiInputStream
```

o

```
FilterInputStream
```

es pot tractar com si fos qualsevol de les seves superclasses.

Vegeu l'arbre de la jerarquia de la classe `java.io.FilterInputStream` a la API de Java[2]. Vegeu quina relació té amb la classe `java.io.InputStream`.

6.2 Filtre zip: paràmetre -z

Quan s'invoqui l'aplicació amb l'opció -z, els documents que s'estan descarregant s'han de desar comprimits utilitzant el format ZIP. Els fitxers comprimits han de tenir l'extensió '.zip'.

En aplicar aquest filtre, al fitxer que desareu a disc li afegireu l'extensió **.zip**, i.e: `index.html.zip`.

Per a la implementació podeu utilitzar la classe de la Api de Java: **`java.util.zip.ZipOutputStream`**.

Un fitxer *zip* és un archive que conté fitxers comprimits. Cada un d'aquests fitxers és una entrada d'aquest archive. Vegeu el mètode de la classe `java.util.zip.ZipOutputStream`:

```
public void putNextEntry(ZipEntry e) throws IOException
```

Des d'una consola, executant la comanda `unzip <nom_fitxer>` podreu comprovar si el fitxer s'ha comprimit correctament.

Noteu que la idea és que imbriqueu el flux de sortida d'aquesta classe i el de la classe `java.io.FileOutputStream` que utilitzareu per generar el fitxer que desareu a disc. No heu de crear cap fitxer intermig. A la vegada que estem descarregant-nos una URL anem comprimint els bytes i els anem desant a disc.

De nou caldrà que feu ús del **polimorfisme** de la classe `java.io.OutputStream`.

Aquest és un exemple d'una invocació amb la opció -z:

```
# cat urls.txt
# http://ccd.uab.es/prova.html
#
#
# java Wget -f urls.txt -a -z
# ls
prova.html.asc.zip
# unzip -v prova.html.asc.zip
Archive:  prova.html.zip
 Length  Method  Size  Ratio  Date      ...   Name
-----  -
```

1928	Defl:N	849	56%	02-25-00 ...	prova.html.asc
-----		----	----		-----
1928		849	56%		1 file

6.3 Filtre gzip: paràmetre -gz

L'opció `-gzip` farà una compressió de manera similar a l'anterior, però utilitzant l'algorisme GZIP.

Per a la implementació de la compressió podeu utilitzar la classe de la API de Java: `java.util.GZIPOutputStream`.

Si s'utilitza aquesta opció caldrà afegir l'extensió `'.gz'` al fitxer que deseu a disc.

Des de consola, executant la comanda `gzip -d <nom_fitxer>` podreu comprovar si el fitxer s'ha comprimit correctament.

6.4 Filtre zip+gzip: Paràmetres `z` i `gz`

En cas d'indicar-se ambdues opcions de compressió simultàniament, el fitxer resultant ha de ser un fitxer GZIP que contingui un ZIP, dins del qual hi haurà el fitxer original.

El fitxer resultant tindrà el nom: `nom_fitxer.html.zip.gz`.

Fixeu-vos que el tipus del fitxer resultant és un `gzip`. Noteu que l'extensió més externa és `gz`.

Això ho haureu de fer imbricant els fluxes de sortida de les classes de la API `java.util.zip.GZIPOutputStream`, `java.util.zip.ZipOutputStream` i `java.io.FileOutputStream`. Jugant, com sempre, amb el polimorfisme de la classe pare de totes tres: `java.io.OutputStream`. No heu de crear fitxers intermitjos.

6.5 Tos els filtres actius

Finalment, una petició a on s'especifiquen els paràmetres: `-a`, `-z` i `-gz` ha de retornar un fitxer amb el nom: `nom_fitxer.html.asc.zip.gz`

De nou fixeu-vos que el tipus del fitxer resultant és un `gzip`. Noteu que l'extensió més externa és la de `gz`.

Per a generar aquest fitxer el que hem fet ha sigut connectar-nos a una URL i establir-hi un fluxe de dades de lectura. Per a cada byte llegit hem aplicat l'algorisme per a treure tags, cosa que ha implicat llegir més d'un byte, al byte llegit, que no té tags, li hem aplicat la compressió `zip` i al byte resultant li hem aplicat la compressió `gzip`. Al final aquest byte resultant l'hem desat a disc. **Hem imbricat els fluxes de lectura, de compressió en format `zip` i de compressió en format `gzip` per a generar un fitxer `.html.zip.gz`.**

Quan parlem de comprimir un byte (nosaltres ho programarem així), penseu que l'operatiu, a l'hora de desar els bytes a disc, els guarda en un *buffer*, de manera que el que realment anem comprimint és aquest *buffer*, no pas un sol byte. A efectes de programació això no es afectarà: **byte que llegim, byte que comprimim.**

6.5.1 Comanda `file`

El sistema operatiu GNU-linux ens ofereix una comanda, que executareu des d'una consola, que rep com a paràmetre el nom d'un fitxer i ens retorna de quin tipus és el fitxer. Aquí teniu un exemple d'utilització:

```
file index.html.asc.zip.gz
index.html.asc.zip.gz: gzip compressed data, from Unix
```

En aquest exemple estem segurs que el fitxer `index.html.asc.zip.gz` és de tipus `gzip`. Per tant la imbricació de fluxes que hem utilitzat és la correcta.

Imagineu el següent cas:

```
file index.html.asc.zip,gz
index.html.asc.zip.gz: Zip archive data, at least v1.0 to extract
```

En aquest exemple, tot i que l'extensió més externa del fitxer sigui `.gz`, i per tant, entenguem que el fitxer està comprimit en format `gzip`, la comanda `file` ens diu que el fitxer està comprimit en format `zip`. Aquest és un clar exemple a on hem comés un error en la imbricació dels fluxes de sortida.

7 Concurrència

L'aplicació haurà de **descarregar en paral·lel totes les URLs** del fitxer de URLs. Per a tal efecte es llençarà un `Thread` per descarregar cada una de les *URL*'s. Per a implementar la concurrència en Java, vegeu la classe `java.lang.Thread` a la API de Java.

Per a cada *URL* del fitxer de URLs, doncs, la vostra aplicació llençarà un *Thread* que durà a terme les següents tasques:

1. Connexió a la URL i obtenció d'un flux de lectura sobre aquesta URL.
2. Transformació del byte o conjunt de bytes, que anem llegint del flux de lectura, tot aplicant els filtres especificats com a paràmetres en l'execució de l'aplicació: `-a`, `-z` i `-gz`.
3. Desar en un fitxer el contingut de la URL després d'haver aplicat els filtres: *byte* que llegim, *byte* que filtrem, *byte* que escrivim a disc.

8 Opcional

8.1 Modificació de funcionalitat del filtre `ascii`

Modificar el comportament del filtre `ascii` per a que, a més a més de la funcionalitat d'eliminar els tags HTML, elimini també els comentaris HTML:

`<!-- Un comentari HTML -->`.

Recordeu que poden haver símbols `'<'` o `'>'` dins dels comentaris, per exemple codi Javascript.

Aquesta part opcional està valorada amb **1 punt**.

8.2 Generar la documentació de les vostres classes amb `javadoc`

Java ofereix un format molt específic per comentar el vostre codi font. Cal que utilitzeu aquest format per comentar les classes que definiu i tots els mètodes d'aquestes classes.

Amb l'aplicació `javadoc` de Java, a partir dels comentaris que heu fet al vostre codi, es genera, en format HTML, la documentació de la vostra aplicació seguint el *look and feel* de la API de Java.

No es valorarà que només executeu aquesta aplicació havent afegit molt pocs o cap comentari al vostre codi.

Aquesta part opcional està valorada amb **0.5 punts**.

9 Observacions

- La classe principal de la vostra aplicació (la que té el *main*) ha de tenir el nom **Wget** i ha d'estar al fitxer **Wget.java**.
- Per a compilar des d'una consola utilitzareu la comanda:

```
javac fitxers.java
```

i per a executar l'aplicació des d'una consola utilitzareu la comanda:

```
java fitxer_ja_compilat_sense_extensio <paràmetres>
```

Mitjançant l'**eclipse** també podreu compilar i executar l'aplicació de forma visual.

- L'equivalent a la funció en C `printf(char*)`, en Java és:
 - `System.out.print(String)` o bé
 - `System.out.println(String)` per afegir un salt de línia.

Per a més informació, mireu a la API de Java l'atribut `out` en la classe `System`.

Podeu generar aquesta línia de codi des de l'eclipse tot escrivint: `syso` i prement `Ctrl Space`.

10 Planificació de la pràctica

L'organització de les sessions de pràctiques sempre serà la mateixa, excepte la sessió del lliurament final.

Durant la sessió, al laboratori, treballareu sobre una funcionalitat de la pràctica i demanareu tots els dubtes que us sorgeixin.

Per a cada sessió **cal que feu un treball previ indispensable** per aprofitar la sessió. Aquest treball previ normalment consistirà en que us mireu com funciona algun aspecte de programació que ens caldrà per desenvolupar la funcionalitat de la pràctica que toqui a la sessió.

La idea és que feu una feina prèvia a casa i que durant la sessió pregunteu els dubtes que us hagin sorgit sobre aquest aspecte de programació i que continueu treballant en la funcionalitat que us demanem per la sessió.

A continuació us indiquem què treballarem durant cada sessió de pràctiques.

10.1 Sessió 1

Setmana del 20/02.

Feina prèvia que heu de fer abans de venir a la sessió

- Repasseu un tutorial de Java. N'hi ha molts online. Us n'hem recomanat alguns.
- Fer els exercicis de Java del treball previ.
- Respondre les qüestions prèvies sobre l'enunciat de la pràctica.

Feina que heu de fer durant la sessió i acabar a casa abans de la sessió següent

- Crear la classe `Wget` que tingui el mètode `main`. Aquesta serà la classe executable de la nostra aplicació.
- Des del mètode `main` obrir el fitxer `urls.txt` i llegir cada línia i imprimir-la per pantalla. Utilitzeu la classe `java.io.BufferedReader`.
- Crear un mètode a la classe `Wget` que donada una URL obri una connexió cap a aquesta URL. Aquesta connexió li permetrà obtenir un flux de lectura per tal de poder descarregar el recurs web a disc. Heu d'utilitzar les classes `java.net.URL`, `java.io.InputStream` i `java.io.FileOutputStream`.

10.2 Sessió 2

Setmana del 27/02.

Feina prèvia que heu de fer abans de venir a la sessió

- Portar feta la feina proposada en la sessió anterior.
- Mirar-vos com funcionen els `Threads` en Java. Podeu mirar l'exemple de la API de Java `java.lang.Thread`.
- Mirar l'explicació que hi ha en la Api de Java de la classe `java.io.FilterInputStream`.

Feina que heu de fer durant la sessió i acabar a casa abans de la sessió següent

- Crear una classe de tipus `Thread` i moure-hi el mètode que donat una URL se la descarrega a disc.
- Modificar la classe `Wget` per a que el mètode `main` obri el fitxer de les URLs i, per a cada una, llenci un `Thread` que se la descarregui a disc.
- Tractar els paràmetres d'execució de l'aplicació: `-f fitxerURLs i [-a]`.
- Modificar el codi per a que en lloc d'utilitzar el fitxer `urls.txt` utilitzi el fitxer especificat mitjançant el paràmetre `-f`.
- Crear la classe `Html2AsiiInputStream`.

10.3 Sessió 3

Setmana del 06/03.

Feina prèvia que heu de fer abans de venir a la sessió

- **Lliurament parcial:** lliurament de la feina proposada en les sessions 1 i 2, excepte la implementació i utilització de la classe `Html2AsciiInputStream`. Aquest lliurament està valorat amb 0.5 punts.
- Mirar l'explicació que hi ha en la Api de Java de la classe `java.io.FilterOutputStream`.
- Mirar l'explicació que hi ha en la Api de Java de la classe `java.util.zip.ZipOutputStream`.
- Mirar l'explicació que hi ha en la Api de Java de la classe `java.util.zip.GZipOutputStream`.
- Entendre la relació que hi ha entre els filtres de sortida `ZipOutputStream`, `GZipOutputStream` respecte la classe `FilterOutputStream`.
- Pensar com relacionareu els filtres `ZipOutputStream` i `GZipOutputStream` amb el fluxe de sortida que implementeu amb la classe `FileOutputStream` i que utilitzeu per a desar a disc els bytes que aneu llegint.
- Entendre l'equivalència entre el que heu après sobre com implementar el filtre `HTML2AsciiInputStream`, que s'aplica al fluxe que utilitzem de lectura, respecte els filtres `ZipOutputStream` i `GzipOutputStream`, que són filtres que apliquem en el fluxe de dades d'escriptura.

Feina que heu de fer durant la sessió i acabar a casa abans de la sessió següent

- Tractar els paràmetres d'execució de l'aplicació: `[-z]` i `[-gz]`.
- Desar a disc el contingut de la URL aplicant-li el filtre `ascii`.
- Desar a disc el contingut de la URL aplicant-li el filtre `zip`.
- Desar a disc el contingut de la URL aplicant-li el filtre `gzip`.
- Considerar tots els filtres a la vegada:

```
java Wget -f urls.txt -a -z -gz
```

10.4 Sessió 4

Setmana del 13/03.

Feina prèvia que heu de fer abans de venir a la sessió

- **Lliurament parcial:** Que l'aplicació funcioni amb qualsevol combinació dels possibles filtres: $-a$, $-z$ i $-gz$.

Feina que heu de fer durant la sessió i acabar a casa abans de la sessió següent

- Fer les parts opcionals de la pràctica.

10.5 Sessió 5

Setmana del 20/03.

Lliurament final i avaluació de la pràctica. Heu de desar la pràctica en el vostre directori de lliuraments (entregues) una hora abans de la vostra sessió de pràctiques.

11 Entorn de la pràctica

- El sistema operatiu instal·lat a les màquines del laboratori és GNU-Linux.
- Utilitzarem el IDE eclipse NEON[4] per a desenvolupar, compilar i executar la pràctica.
- La versió del JDK que utilitzarem serà la 8 build 121[1].
- Cada grup de treball disposarà d'un compte de pràctiques. En el vostre compte tindreu creat el directori *java* a on trobareu el fitxer *urls.txt* i les carpetes corresponents als lliuraments que haureu de fer.
- Per accedir al vostre compte de pràctiques remotament ho heu de fer via **ssh**.

Si esteu en un entorn Linux, directament heu d'executar:

```
ssh <usuari>@deic-dc10.uab.cat  
o  
ssh <usuari>@deic-dc26.uab.cat
```

Si esteu en un entorn Windows podeu utilitzar l'aplicació **putty**[3].

- El professor us proporcionarà un password temporal per accedir al vostre compte. Un cop hagueu entrat per primera vegada, canvieu-lo utilitzant la comanda *yppasswd*
- Per a copiar fitxers remotament, si esteu en un entorn Linux, podeu utilitzar les comandes *scp* i *sftp*. En entorns Windows podeu utilitzar l'aplicació *winscp* [5], Filezilla [6], etc...
- A casa vostra podeu instal·lar-vos l'eclipse i treballar normalment. No us cal l'operatiu GNU-Linux. De totes maneres, abans de lliurar el codi assegureu-vos que tot funciona correctament als ordinadors dels laboratoris.

12 Condicions de lliurament

Per tal de lliurar la pràctica caldrà fer el següent:

- Exportar, des de l'eclipse, el projecte Wget en format `jar` fent servir el nom de fitxer **Wget.jar**. És molt important que respecteu aquest nom. **No us descuideu de marcar el check box per afegir al jar el codi font, no només els executables!**. Comproveu que els fitxers `.java` són a dins.

Cal que deseu aquest fitxer `.jar` en el directori `$HOME/entregues/java/final/` una hora abans de la sessió de pràctiques de la setmana del **20 de Març**

- El codi dels lliuraments parcials l'heu de desar també en un `jar` als directoris:
`$HOME/entregues/java/control1/Wget.jar`
`$HOME/entregues/java/control2/Wget.jar`
del vostre compte de pràctiques una hora abans de la sessió de pràctiques en qüestió.
- És molt important que respecteu els noms dels fitxers que heu de lliurar i els directoris de lliurament. En cas de no respectar-ho **el professor es reserva el dret de no corregir-vos la pràctica**.

13 Nota de pràctiques

Per a poder aprovar la pràctica de Java cal complir amb totes les següents restriccions:⁴

- Haver assistit a almenys un 80% de les sessions d'aquesta pràctica.
- Haver superat la prova de validació.
- Obtenir almenys un 5.00 en calcular la nota del lliurament final (LLF).

En cas de complir-los tots, la nota de la pràctica de Java (PJ) la calculem seguint la següent fórmula:

$$PJ = LLP1 \times 0.5 + LLP2 \times 0.5 + LLF$$

LLP1: nota lliurament parcial 1

LLP2: nota lliurament parcial 2

LLF: nota lliurament final

L'avaluació del lliurament final (LLF) es farà seguint la següent rúbrica:

⁴Si alguna restricció no es compleix, la nota de la pràctica de Java (PJ) és 0.

Tipus criteri	Criteri amb descripció del necessari per a superar-lo	Pes
imprescin.	Obrir el fitxer de les URLs i descarregar-se-les a disc.	-

Si no es supera el criteri imprescindible no cal seguir corregint. El lliurament final (LLF) és 0.

	Considerar els paràmetres -f -a -z i -gz especificats en la crida a l'aplicació: <code>Wget -f file.txt -a -z -gz</code>	0.5
	Utilitzar Threads per a descarregar-se les URLs	1.0
	Considerar sufixes pels noms dels fitxers descarregats	0.5
	Extensions dels fitxers descarregats correctes: Si tots els filtres actius l'extensió del fitxer descarregat per la URL: <code>www.google.es</code> → <code>index.html.asc.zip.gz</code>	0.5
	Utilitzar la classe <code>HTML2AsciiInputStream</code> per a treure tags HTML utilitzant sobre escriptura dels mètodes que toquin. Recordar <code>extends FilterInputStream</code> .	1.25
	Comprimir en Zip	0.5
	Comprimir en Gzip	0.5
	Comprimir en Zip i Gzip	0.75
	No utilitzar fitxers intermitjos	0.5
	Control Errors: Recurs web no existeix.	0.5
	Control Errors: Recurs web no és de tipus <code>text/html</code> → no aplicar el filtre	0.5
	Polimorfisme <code>InputStream</code>	1.0
	Polimorfisme <code>OutputStream</code>	1.0

Si la suma dels criteris anteriors ponderats pel seu pes és igual o superior a 5.00, llavors seguim corregint. Sinó, ja tenim la nota pel lliurament final (LLF).

opcional	JavaDoc	0.5
opcional	Filtre <code>HTML2AsciiInputStreram</code> elimina comentaris html	1.0

La nota del lliurament final (LLF) és la suma dels criteris anteriors ponderats pel seu pes.

A la rúbrica apareixen un seguit de criteris amb el seu pes. Atenció!! El primer criteri és **IMPRESINDIBLE**. S'ha de superar satisfactoriament per tal de poder avaluar el lliurament final. Si no es satisfà, la nota del lliurament final és 0.

Si havent avaluat les funcionalitats obligatòries (no opcionals) la nota és igual o superior a 5.00, llavors s'avaluen també les opcionals per a poder pujar nota.

Com podreu veure, les funcionalitats opcionals de la pràctica permeten obtenir una nota del lliurament final (LLF) de fins a 10.5 que suposa una nota de la pràctica (PJ) de fins a 11.5 punts. Aquests 1.5 punts extres poden ser molt beneficiosos en calcular la nota final de pràctiques (P) juntament amb la nota de les altres dues pràctiques de l'assignatura.

14 Recuperació

Aquells alumnes que no aprovin la pràctica podran recuperar-la al Juny. Per a poder recuperar la Pràctica de Java cal complir amb totes les següents restriccions:

- Haver assistit a almenys un 80% de les sessions d'aquesta pràctica.
- Haver superat la prova de validació de la recuperació.
- Obtenir almenys un 7.00 seguint la rúbrica.

Referències

- [1] <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- [2] <http://docs.oracle.com/javase/8/docs/api/>
- [3] <http://www.putty.org/>
- [4] <http://www.eclipse.org>
- [5] <https://winscp.net/eng/download.php>
- [6] <https://filezilla-project.org/>