**ACTIVIDAD 4**

Enunciado

Realizar un programa con un método de nombre Visualizar(), que tenga un parámetro que sea un objeto URL y que visualice el resultado de aplicar diferentes métodos de la clase URL a un objeto URL creado con diferentes constructores. Se usarán las siguientes URLs:

* http://docs.oracle.com
* http://docs.oracle.com/javase/7
* http://docs.oracle.com/javase/7/docs/api/java/net/URL.html

El nombre de la clase principal será Actividad4.

El resultado esperado es el siguiente:

Constructor simple para una URL:

Método toString():http://docs.oracle.com

Método getProtocol():http

Método getHost():docs.oracle.com

Método getPort():-1

Método getFile():

Método getUserInfo():null

Método getPath():

Método getAuthority():docs.oracle.com

Método getQuery():null

Constructor para protocolo, host y directorio:

Método toString():http://docs.oracle.com/javase/7

Método getProtocol():http

Método getHost():docs.oracle.com

Método getPort():-1

Método getFile():/javase/7

Método getUserInfo():null

Método getPath():/javase/7

Método getAuthority():docs.oracle.com

Método getQuery():null

Constructor para protocolo, host, puerto y directorio:

Método toString():http://docs.oracle.com:80/javase/7

Método getProtocol():http

Método getHost():docs.oracle.com

Método getPort():80

Método getFile():/javase/7

Método getUserInfo():null

Método getPath():/javase/7

Método getAuthority():docs.oracle.com:80

Método getQuery():null

Constructor para un objeto URL y su directorio:

Método toString():http://www.oracle.com/javase/7/docs/api/java/net/URL.html

Método getProtocol():http

Método getHost():www.oracle.com

Método getPort():-1

Método getFile():/javase/7/docs/api/java/net/URL.html

Método getUserInfo():null

Método getPath():/javase/7/docs/api/java/net/URL.html

Método getAuthority():www.oracle.com

Método getQuery():null

**Solución**

package code;

import java.net.MalformedURLException;

import java.net.URL;

public class Actividad4 {

// Se pasa un parámetro más llamado msg para asi ahorrar líneas de código

private void Visualizar(URL url,String msg) {

System.out.println("Constructor "+msg);

System.out.println("Metodo toString(): " + url.toString());

System.out.println("Metodo getProtocol(): " +url.getProtocol());

System.out.println("Metodo getHost(): "+url.getHost());

System.out.println("Metodo getPort(): "+url.getPort());

System.out.println("Metodo getFile(): "+url.getFile());

System.out.println("Metodo getUserInfo(): "+url.getUserInfo());

System.out.println("Metodo getPath(): "+url.getPath());

System.out.println("Metodo getAuthority(): "+url.getAuthority());

System.out.println("Metodo getQuery(): "+url.getQuery());

System.out.println();

}

public static void main(String[] args) {

Actividad4 a4=new Actividad4();

try {

URL url1 = new URL("http://docs.oracle.com");

a4.Visualizar(url1,"simple para una URL");

URL url2 = new URL("http","docs.oracle.com","javase/7");

a4.Visualizar(url2,"para protocolo, host y directorio:");

URL url3 =new URL("http", "docs.oracle.com", 80, "javase/7");

a4.Visualizar(url3,"para protocolo, host, puerto y directorio:");

URL url4 = new URL(new URL("http://docs.oracle.com/javase/7/docs/api/java/net/URL.html"), "javase/7");

a4.Visualizar(url4,"para un objeto URL y su directorio:");

} catch (MalformedURLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

**Resolución de problemas**

**ACTIVIDAD 5**

**Enunciado**

Realizar un programa con un método privado llamado VisualizarConexión(), que tenga un parámetro que sea un objeto URLConnetion y que visualice el resultado de aplicar diferentes métodos de la clase URLConnection.

El nombre de la clase principal será Actividad5.

La URL con la que se realizará la conexión es www.vitoria-gasteiz.org

El resultado esperado es el siguiente:

Conexión con www.vitoria-gasteiz.org

==========================

Método toString():sun.net.www.protocol.http.HttpURLConnection:http://www.vitoria-gasteiz.org

Método Fecha():1636109544000

Método getContentType():text/html; charset=iso-8859-1

Campos Cabecera con getHeaderField:

====================================

Linea 1:HTTP/1.1 302 Found

Linea 2:Fri, 05 Nov 2021 10:52:24 GMT

Linea 3:Apache

Linea 4:https://www.vitoria-gasteiz.org/

Linea 5:216

Campos Cabecera con getHeaderFields:

====================================

Keep-Alive:[timeout=10, max=100]

null:[HTTP/1.1 302 Found]

Server:[Apache]

Connection:[Keep-Alive]

Content-Length:[216]

Date:[Fri, 05 Nov 2021 10:52:24 GMT]

Content-Type:[text/html; charset=iso-8859-1]

Location:[https://www.vitoria-gasteiz.org/]

**Solución**

package code;

import java.io.IOException;

import java.net.MalformedURLException;

import java.net.URL;

import java.net.URLConnection;

import java.util.Collection;

import java.util.Iterator;

import java.util.Map;

public class Actividad5 {

private void VisualizarConexion(URLConnection uc) {

// Metodos simples

System.out.println("Conexion con www.vitoria-gasteiz.org");

System.out.println("====================================");

System.out.println("\tMetodo toString(): " + uc.toString());

System.out.println("\tMetodo Fecha(): " + uc.getDate());

System.out.println("\tMetodo getContentType(): " + uc.getContentType());

System.out.println();

// Se crea un bucle para usar el getHeaderField en una soloa línea

System.out.println("Campos cabecera con getHeaderField: ");

System.out.println("====================================");

for (int i=1;i<6;i++) {

System.out.println("\tLinea "+i+": "+uc.getHeaderField(i));

}

System.out.println();

/\*

\* m: mapa que obtiene los valores de la cabecera.

\* c: coleccion que solo tiene las claves.

\* it: iterador de c para recorrerlo.

\* clave: string obtenido de cada vuelta del iterador.

\*/

System.out.println("Campos cabecera con getHeaderFields: ");

System.out.println("====================================");

Map m=uc.getHeaderFields();

Collection c=m.keySet();

Iterator<String>it = c.iterator();

while(it.hasNext()) {

String clave = it.next();

System.out.print(clave+": ");

System.out.println(m.get(clave));

}

}

public static void main(String[] args) {

Actividad5 a5 = new Actividad5();

URL url1;

try {

url1 = new URL("http://www.vitoria-gasteiz.org");

URLConnection uc= url1.openConnection();

a5.VisualizarConexion(uc);

} catch (MalformedURLException e1) {

e1.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**Resolución de problemas**

‘uc.getHeaderFields.values()’ te da el valor en una colección que dentro tiene una lista de Strings. Lo que ahorraba de espacio de código haciéndolo así, más adelante (a la hora de sacar los valores String) complicaba el código, por lo que hice paso a paso para así sacar tanto el mapa como la colección a variables y poder interactuar directamente con ellas.

**CRITERIOS DE CORRECCIÓN**

1. Nivel de ajuste a la nomenclatura de objetos definida en el enunciado de la actividad y nivel de ajuste al almacenamiento de la información requerida en las subcarpetas definidas para la actividad y unidad didáctica correspondientes (15%).
2. Hacer lo que se indica en el enunciado (55%).
3. Claridad del código Java (15%).
4. Documentación del código Java (15%).
5. Explicación detallada de cómo se han resuelto problemas que hayan podido surgir en la realización de la actividad (hasta un 15% adicional sobre la nota obtenida).   
   Dicha explicación se incluirá al principio de los archivos de código fuente Java mediante líneas de comentarios.   
   La primera línea de comentarios sólo incluirá lo siguiente:  
    // Resolución de problemas.
6. Variables no usadas (-15%).
7. Librerías no usadas (-15%).