**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 ejercicio04;

import java.net.URL;

import java.net.MalformedURLException;

public class Actividad4 {

public static void main(String[] args) {

try {

System.out.println("Constructor simple para una URL:");

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

visualizar(url1);

System.out.println("Constructor para protocolo, host y directorio:");

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

visualizar(url2);

System.out.println("Constructor para protocolo, host, puerto y directorio:");

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

visualizar(url3);

System.out.println("Constructor para un objeto URL y su directorio:");

URL url4 = new URL(url1, "/javase/7/docs/api/java/net/URL.html");

visualizar(url4);

} catch (MalformedURLException e) {

System.err.println(e.getMessage());

}

}

public static void visualizar(URL url) {

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

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

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

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

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

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

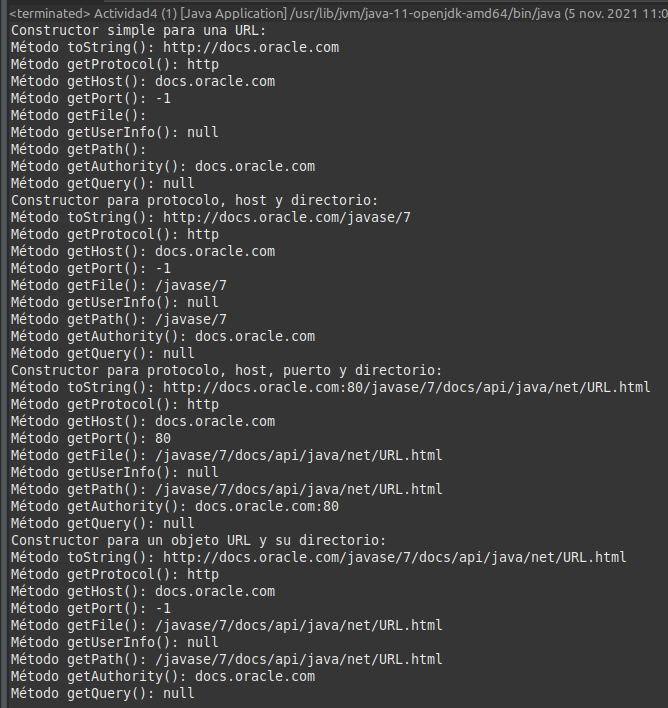
System.out.println("Método getPath(): " + url.getPath());

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

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

}

}

****

**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.kaixo.com.

El resultado esperado es el siguiente:

Conexión con www.kaixo.com

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

Método toString():sun.net.www.protocol.http.HttpURLConnection:http://www.kaixo.com

Método Fecha():1509301263000

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

Campos Cabecera con getHeaderField:

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

Línea 1:HTTP/1.1 302 Found

Linea 2:Sun, 29 Oct 2017 18:21:03 GMT

Linea 3:Apache/2.4.18 (Ubuntu)

Linea 4:https://www.kaixo.com/

Linea 5:285

Campos Cabecera con getHeaderFields:

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

Keep-Alive:[timeout=5, max=100]

null:[HTTP/1.1 302 Found]

Server:[Apache/2.4.18 (Ubuntu)]

Connection:[Keep-Alive]

Content-Length:[285]

Date:[Sun, 29 Oct 2017 18:21:03 GMT]

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

Location:[https://www.kaixo.com/]

**Solución**

package ejercicio05;

import java.io.IOException;

import java.net.URL;

import java.net.URLConnection;

public class Actividad5 {

public void main(String[] args) {

try {

URL url = new URL("www.kaixo.com");//Creamos la url

URLConnection con = url.openConnection();//Creamos la conexión

visualizarConexion(con);//Llamamos a la función para ver los métodos

}

catch (IOException e) {

System.out.println(e.getMessage());

}

}

private void visualizarConexion(URLConnection urlConection) {

System.out.println("Conexión con " + urlConection.getURL().getHost());

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

System.out.println("\t"+"Método toString(): " + urlConection.toString());

System.out.println("\t"+"Método getDate(): " + urlConection.getDate());

System.out.println("\t"+"Método getContentType(): " + urlConection.getContentType());

System.out.println();

System.out.println("\t"+"Campos Cabecera con getHeaderField:");

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

System.out.println("\t"+"Línea 1: " + urlConection.getHeaderField(0));

System.out.println("\t"+"Línea 2: " + urlConection.getHeaderField(1));

System.out.println("\t"+"Línea 3: " + urlConection.getHeaderField(2));

System.out.println("\t"+"Línea 4: " + urlConection.getHeaderField(3));

System.out.println("\t"+"Línea 5: " + urlConection.getHeaderField(4));

System.out.println();

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

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

System.out.println("\t"+"Keep-Alive: " + urlConection.getHeaderField("Keep-Alive"));

System.out.println("\t"+"null: " + urlConection.getHeaderField(null));

System.out.println("\t"+"Server: " + urlConection.getHeaderField("Server"));

System.out.println("\t"+"Connection: " + urlConection.getHeaderField("Connection"));

System.out.println("\t"+"Content-Length: " + urlConection.getHeaderField("Content-Length"));

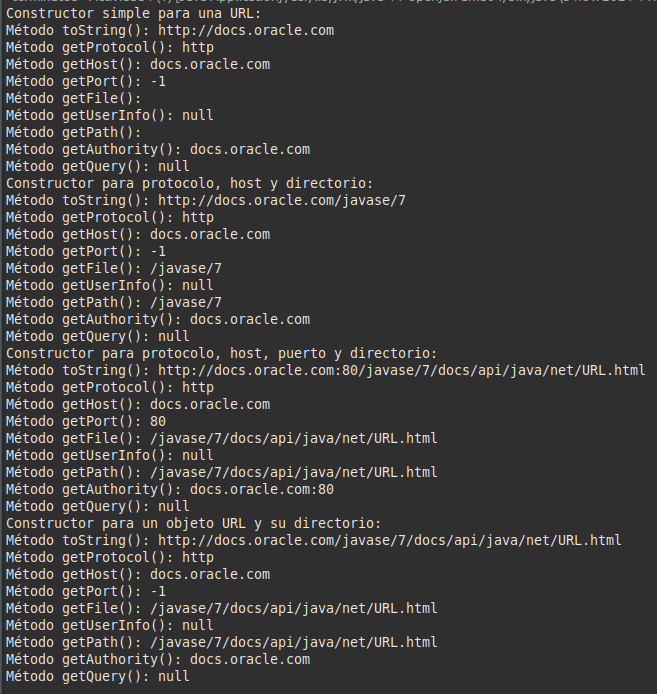
System.out.println("\t"+"Date: " + urlConection.getHeaderField("Date"));

System.out.println("\t"+"Content-Type: " + urlConection.getHeaderField("Content-Type"));

System.out.println("\t"+"Location: " + urlConection.getHeaderField("Location"));

}

}

****

**Resolución de problemas**

**REQUISITOS**

1. Entregar las soluciones de las actividades en un único archivo comprimido donde tendremos:
   1. Un archivo de extensión .odt o .docx que contendrá
      1. Una copia del código del programa
      2. Una captura de pantalla del resultado de la ejecución del programa
   2. Los archivos \*.java empleados
2. El único archivo con la soluciones contendrá el nombre original de la actividad, seguido de un guión bajo y del primer apellido del alumno.

**CRITERIOS DE CORRECCIÓN**

1. Cumplimiento de los requisitos (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%).