
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		PRÁCTICA DE LABORATORIO	
CARRERA: COMPUTACIÓN		ASIGNATURA: Programación aplicada	
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Investigación de las versiones de Java	
OBJETIVO ALCANZADO: <p>Identificar los cambios importantes de Java</p> <p>Diseñar e Implementar las nuevas técnicas de programación</p> <p>Entender la cada una de las características nuevas en Java</p>			
ACTIVIDADES DESARROLLADAS			
1. Java 11: <p>En la plataforma Java era habitual que entre versión y versión mayor pasen varios años normalmente 3 o más. Este ciclo de publicaciones no se adapta a las necesidades de todas las empresas, organizaciones y usuarios. Algunas empresas y usuarios desean ciclos de publicaciones más cortos quizá con no tantas novedades en cada uno de ellos, pero sí de forma más constante incorporando mejoras. Otras organizaciones necesitan confiar en una versión que tenga un ciclo de soporte largo para sus aplicaciones.</p> <p>En esta nueva versión de Java 11 publicada el 25 de septiembre de 2018 las novedades no son tan relevantes como lo fueron las versiones Java 8 con las <i>lambdas</i> y Java 9 con los módulos, pero continúa con las mejoras incrementales y proporciona una versión LTS.</p> <p>Esto significa que tendrá soporte y actualizaciones durante varios años, dicho soporte de Java 11 está planificado que dure hasta 2023 y hasta 2026 de forma extendida lo que son 8 años de soporte.</p>			
2. Novedades y nuevas características de Java 11: <p>Dentro de las nuevas características y mejoras que implementa la versión JAVA 11 podemos destacar:</p> <p>En esta versión Oracle no proveerá El JDK de forma libre para uso comercial:</p> <ul style="list-style-type: none"> • Es decir, si deseamos usar Java 11 para uso comercial deberíamos pagar una licencia. • El cambio de licencia a partir de Java 11 no implica que Java deje de ser gratuito. Quien no desee entrar en el esquema de licencias de Oracle podrá trasladarse a OpenJDK o a otras JDK alternativas y con buen soporte de la comunidad. Claro si se trata de una empresa quizás considere más fiable abonarse con Oracle. 			

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Ejecutar archivos Java directamente sin javac:

- Este cambio lo que nos permite ahora es ejecutar un file java directamente sin necesidad de compilarlo previamente con javac.
- Ahora podemos directamente correr el programa java y esto ya implica una compilación previa.

Java 11 introduce algunos métodos nuevos para los Strings:

- **isBlank()** -> Este método retorna un valor booleano según si el string contiene solo espacios en blanco y si está vacío.
- **Lines()** -> Este método devolverá un array de strings “spliteando” el texto por líneas.
- **strip(), stripLeading(), stripTrailing()** -> Remueve los caracteres en blanco. A diferencia del viejo trim(), este nuevo método strip si tiene en cuenta los caracteres Unicode.
- **Character.isWhitespace(c)** -> nos permite saber si un carácter es considerado un blanco.
- **repeat(int)** -> Repite el string las veces que se le indique.

Lectura y escritura de String en archivos:

- En Java 11 se agregaron estos dos métodos que nos simplificarán la lectura y escritura de texto sobre archivos.
- `readString(Path)`
- `readString(Path, Charset)`
- `writeString(Path, CharSequence, OpenOption...)`
- `writeString(Path, CharSequence, Charset, OpenOption...)`

Variables Locales para los parámetros Lambda:

- En Java 10 se introdujo el uso de **var** con inferencia del tipo. Ahora podemos usar esto en la declaración de lambdas.
- Esto a su vez permite el uso de anotaciones en los parámetros de la función por ejemplo `@NotNull`
- Si se usa **var** se debe usar en todas las variables.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Acceso a clases internas:

- En Java 11 se agregaron nuevos métodos para acceder via reflexión a los métodos internos.
- La clase `java.lang.Class` introduce tres metodos nuevos `getNestHost()`, `getNestMembers()`, y `isNestmateOf()`.
-

Mejoras En Los *Arch64 Intrinsics*:

- Los *intrinsics* son usados para aprovechar código ensamblador que es específico de una arquitectura que se ejecuta en lugar de código java para un método dado, y mejorar así su desempeño.
- Específicamente, se optimizaron los siguientes métodos que aún estaban pendientes para que funcionaran con *intrinsics* del paquete `java.lang.math`.

1. `sin`
2. `cos`
3. `log`

Se Removieron Los Módulos Java EE y CORBA:

- Estos módulos fueron marcados como descontinuados en java 9 y la intención era quitarlos en las versiones futuras.
 1. Como resultado los siguientes paquetes, dejará de compilar en java 11:
 2. `java.xml.ws` (JAX-WS, y las tecnologías relacionadas SAAJ y Web Services Metadata)
 3. `java.xml.bind` (JAXB)
 4. `java.activation` (JAF)
 5. `java.xml.ws.annotation` (Anotaciones Comunes)
 6. `java.corba` (CORBA)
 7. `java.transaction` (JTA)
 8. Y módulos de Java SE 9 que también se marcaron como obsoletos:
 9. `java.se.ee` (Modulo agregador de los seis módulos anteriores)
 10. `jdk.xml.ws` (Herramientas para JAX-WS)
 11. `jdk.xml.bind` (Herramientas para JAXB)

3. Ejecución de las nuevas características implementadas

- Ejecutar archivos Java directamente sin javac:

```
3    public class HelloWorldExample {
4        public static void main(String[] args) {
5            System.out.println("Hello World!!");
6        }
7    }
```

```
1    $ java HelloWorldExample.java
2    Hello World!!
```

- Nuevos métodos para los strings:

```
//isBlank() example
System.out.println(" ".isBlank());
String s = "Hi Word!";
System.out.println(s.isBlank());
String s1 = "";
System.out.println(s1.isBlank());
System.out.println();

// lines() example
String str = "Hi\nWord\nHi\nUniverse";
System.out.println(str);
System.out.println(str.lines().collect(Collectors.toList()));
System.out.println();

// strip() example
String hello = " HELLO ";
System.out.println("SAY" + hello.strip() + "PLEASE");
System.out.println("SAY" + hello.stripLeading() + "PLEASE");
System.out.println("SAY" + hello.stripTrailing() + "PLEASE");
System.out.println();

Character c = '\u2000';
String stringWithUnicode = c + "abc" + c;
System.out.println("Contain unicode " + Character.isWhitespace(c));
System.out.println("Trim test " + (stringWithUnicode == stringWithUnicode.trim() ));
System.out.println("Strip test " + ("abc"== stringWithUnicode.strip()));
System.out.println();

// repeat() example
String rep = "HI".repeat(2);
System.out.println(rep);
System.out.println();
}
```

Salida del código

```
true
false
true

Hi
Word
Hi
Universe
[Hi, Word, Hi, Universe]

SAYHELLOPLEASE
SAYHELLO PLEASE
SAY HELLOPLEASE

Contain unicode true
Trim test true
Strip test false

HIHI
```

- Lectura y escritura de String en archivos

```
1  import java.io.IOException;
2  import java.nio.file.Files;
3  import java.nio.file.Path;
4
5  public class FileReadStringExample {
6
7      public static void main(String[] args) throws IOException {
8          Path path = Files.writeString(
9              Files.createTempFile(
10                 "myfile", ".txt"),
11                 "This was posted on JD");
12          String s = Files.readString(path);
13          System.out.println(s);
14      }
```

RESULTADO(S) OBTENIDO(S):

Realizar procesos de investigación sobre los cambios importantes de Java

Entender las aplicaciones de codificación de las nuevas características

Entender las funcionalidades adicionales de Java.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

CONCLUSIONES:

En esta practica se pudo conocer las nuevas características y mejoras que se han implementado en la versión JAVA 11 y las nuevas funcionalidades que estas nos permiten al momento de programar.

RECOMENDACIONES:

Realizar el trabajo dentro del tiempo establecido.

Nombre de estudiante: Ariel Vazquez

Firma de estudiante:

