

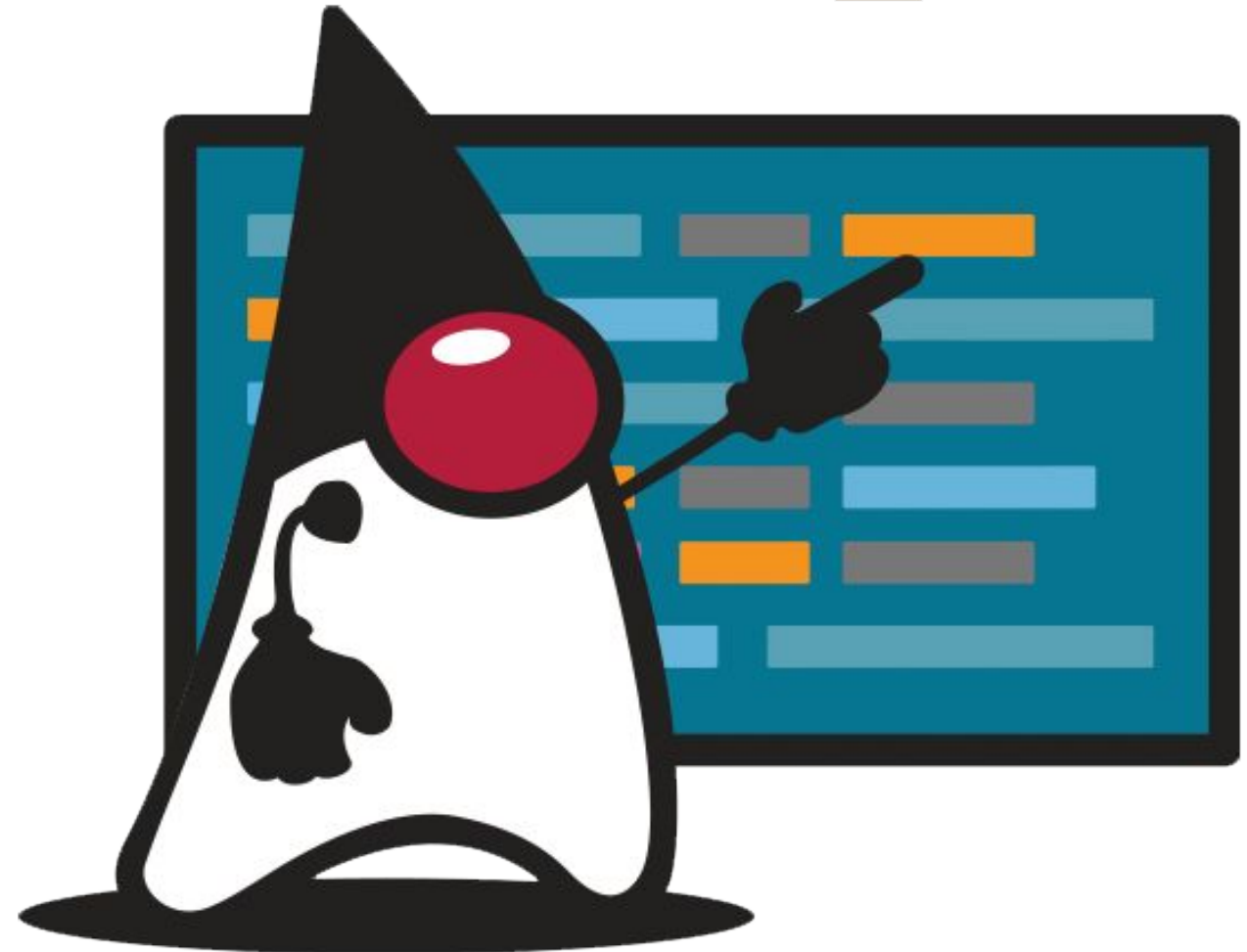
Java en la Educación

<Inserta tu nombre aquí, si estás presentando>

JUG Presentation

Para programadores junior y estudiantes

Preparado por **Ken Fogel** y el **JCP Executive Committee (EC) Java in Education Working Group**



Mitos y Beneficios de aprender Java

- Java puede tener una pronunciada curva de aprendizaje para un principiante.
 - *Sólo si el instructor en su etapa de aprendizaje también la tuvo.*
- Java no es adecuado para tareas rápidas y ligeras.
 - Es más adecuado para aplicaciones grandes y complejas.
 - *¿Has visto el Single-File Source-Code y en Linux has probado la ejecución de shebang?*
 - *Recién incluidos! **Unnamed Classes and Instance Main Methods***
- Oracle Java Development Kit (JDK), no es código abierto
 - *OpenJDK es una implementación del JDK completamente de código abierto.*
 - *El desarrollo continuo de Java se realiza en el proyecto OpenJDK por parte de los desarrolladores Java de Oracle.*

Mitos y Beneficios de aprender Java

- Java es un lenguaje **viejo** (Java 1996 y Python 1991)
 - *También, significa que está establecido, ampliamente utilizado y bien documentado.*
- Más programadores **Java** que cualquier otro tipo de programador en el mundo.
 - Es fácil encontrar personas que puedan ayudarte y orientarte.
- Java deriva su sintáxis de **C**
 - *Aprende Java, te hará aprender lenguajes como Javascript, C#, C++ e incluso Python más fácil.*
- *Java (y su variante en el JVM **Kotlin**) son la base del desarrollo Android.*

Mejoras del Lenguaje *Java*

Esta presentación
analiza varias mejoras
al lenguaje Java.

Estas mejoras ayudan
a disipar algunos de
los mitos que rodean a
Java.

Se trata de por qué Java
debería ser el lenguaje a
enseñar en todos los
niveles educativos al día
de hoy.

Incluso hay una
comparación de un
programa en Java y
Python.

JShell - Read-Evaluate- Print Loop (REPL) JDK 9

Una herramienta para simplificar las instrucciones.

Ejecución a medida que ingresa el código y presiona Intro.

Respuesta inmediata, línea a línea.

También puedes escribir tus métodos y luego ejecutarlos.

Ideal para enseñar o aprender Java una línea a la vez.

JEP 330 - Launch Single-File Source-Code Programs JDK 11

- Aborda la sobrecarga de ejecutar código.
 - Estilo tradicional:
 - Dos pasos para la ejecución:
 - javac
 - java -jar
 - Estilo usando → Single-File Source-Code
 - Un solo paso para la ejecución
 - java
 - Si el archivo tiene una clase pública con un método main, se compila y ejecuta.
 - Funciona también con preview-features además de funciones ya establecidas.
 - Un solo archivo puede contener múltiples clases.
- Esta puede ser la segunda* nueva capacidad más importante para escribir Java para aquellos que deseen aprender el lenguaje.
- No necesitas ser un master en un IDE, para aprender Java.

* La primera se mostrará en el Slide 18


Java 21 - Unnamed Classes and Instance Main Methods *Preview*

- La queja más común sobre Java es que no es adecuado para principiantes en comparación (usualmente) con Python.
- JEP 330 - Launch Single-File Source-Code fue un primer paso hacia la simplificación.
- JEP 445 - Unnamed Classes and Instance Main Methods son los siguientes pasos
 - No necesita declarar ninguna clase.
 - Declarar *imports* está permitido.
 - Declarar *packages* **NO** está permitido.
 - Constructores **NO** están permitidos.

Java 21 - Unnamed Classes and Instance Main Methods *Preview*

- Aquí, tenemos un programa en **Java** completo, que puede ser compilado y ejecutado.

```
void main() {  
    System.out.println("Hello World");  
}
```

- Se espera que en JDK22 se elimine el `System.out`
- 

El nuevo y
mejorado
main!

Utilizable
en cualquier
lugar!

1. `static void main(String[]args)`
2. `static void main()`
3. `void main(String[] args)`

Y nuestro favorito:

4. `void main()`
 - Ejecución de un solo archivo:
 - `java --enable-preview --source 21 file.java`
 - Compilar y Ejecutar:
 - `javac --enable-preview --source 21 file.java`
 - `java --enable-preview file`
 - Puede utilizar cualquiera o todos los nuevos formatos del método *main*.
 - Se ejecutarán en el orden que se muestra aquí, 1, luego 2, luego 3, luego 4

var – Reducción de redundancia JDK 10

No más:

- `MyClass m = new MyClass();`

Ahora se convierte en:

- `var m = new MyClass();`

Fomenta solo la creación de objetos con su inicialización

- Will reduce the occurrence of the dreaded `NullPointerException`

Text Blocks JDK 15

Finalmente, lo que ingresas en tu código fuente es lo que obtienes.



Especialmente útil para cadenas que contienen HTML, XML y JSON.



¿A quién no le gusta escribir tres comillas seguidas?

```
''''''
```

```
''''''
```

Concatenación de la vieja forma

```
String htmlStr = "<html><head><link rel='stylesheet' "
+ "href='styles/main.css' "
+ "type='text/css' /><title>The Learning Servlet</title></head>"
+ "<body><h1>GET method</h1>"
+ "<form id='form:index' action='index.html'>"
+ "<br/><input type='submit' value='Return to Home page' /></form>"
+ "</body></html>";
```

Bloque de Texto - Nueva forma JDK 15

```
String htmlStr = ""
<html>
  <head>
    <link rel='stylesheet' href='styles/main.css' type='text/css' />
    <title>The Learning Servlet</title>
  </head>
  <body>
    <h1>GET method</h1>
    <form id='form:index' action = 'index.html'>
      <br/>
      <input type= 'submit' value='Return to Home page' />
    </form>
  </body>
</html>"";
```

Pero espera, hay más . . . String formateado JDK 15

```
out.println("""
<html>
  <head>
    <title>Just Servlet Output</title>
    <link rel='stylesheet' href='styles/main.css' type= 'text/css' />
  </head>
  <body>
    <h1>Thanks for joining our email list</h1>
    <p>Here is the information that you entered:</p>
    <label>Email:</label>
    <span>%s</span>
  </body>
</html>""").formatted(user.getEmailAddress());
```

Y aún más . . . *String template* JDK 21 *preview*

```
String email = "person@mail.com";
String page = STR. """
    <html>
        <head>
            <title>Just Servlet Output</title>
            <link rel='stylesheet' href='styles/main.css' type= 'text/css'/>
        </head>
        <body>
            <h1>Thanks for joining our email list</h1>
            <p>Here is the information that you entered:</p>
            <label>Email:</label>
            <span>\{email}</span>
        </body>
    </html>""";
System.out.println(page);
```

switch – una expresión y sin break JDK 14

Un switch que se puede explicar con sencillez

Reducción de la duplicación de código cuando se utiliza para establecer un valor

El switch ya no necesita ser el original switch de C.

El break llegó a su fin, todos los casos terminan!

¿Cuál prefieres aprender o enseñar?

```
double value = 0;
switch (point) {
    case NORTH:
        value = 12.12;
        break;
    case SOUTH:
        value = 14.14;
        break;
    case EAST:
        value = 16.16;
        break;
    case WEST:
        value = 18.18;
        break;
```

```
}
```

```
double value = switch (point)
{
    case NORTH -> 12.12;
    case SOUTH -> 14.14;
    case EAST -> 16.16;
    case WEST -> 18.18;
    default -> 0.0;
};
```

Pattern Matching *Switch*. JDK 21

```
Object x = "4";
String designation = switch (x) {
    // case Integer i when i > 4 && i < 12 -> "child";
    case Integer i when i < 12 -> "child";
    case Integer i when i < 18 -> "teenager";
    case Integer i when i < 25 -> "young adult";
    case Integer i when i < 65 -> "adult";
    case Integer i when i >= 65 -> "senior";
    default -> "Not an Integer";
};
System.out.printf("Designation is %s\n", designation);
```

records – Reducción del boilerplate requerido con inmutabilidad agregada al manejo de la data. JDK 16

Los objetos de datos son conocidos por su código repetitivo:

- Inicializar constructores, setters, getters, equals, hashCode, and toString

Al rescate, llegan los records inmutables

Más que una simple simplificación de un bean

Es el camino hacia objetos inmutables por defecto

Y luego está el constructor compacto.


Validando los valores iniciales sin un constructor por separado.

No setters, solo simple getters.
Implícitos equals, hashCode and toString.
Y, un compacto constructor para validación

```
public record Person(String firstName,  
                    String lastName,  
                    int age,  
                    String postion,  
                    LocalDate birthday) {  
    public Person{  
        if (age < 18) {  
            throw new IllegalArgumentException("Too young");  
        }  
    }  
}
```

Virtuosos Virtual Threads JDK 21.

```
public class VirtualThreadClass extends Thread { . . }  
  
public void perform() {  
    for (int i = 0; i < 5; ++i) {  
        Thread.ofVirtual().name("Thread # " + i).  
            start(new VirtualThreadClass());  
    }  
}
```



¿Que está dejando a Java a un lado?

JavaScript

- Pequeño para descargar.
- Disponible en cada navegador de cada computador.
- Numerosos entornos de programación en línea

Python

- Asociado a las dos grandes tendencias:
 - Big Data
 - AI/ML
- Jupyter notebook en línea es muy popular

¿Por qué Python está ganando popularidad en la educación?

El patrón de diseño más temido:

- Flujo de conciencia

El programa fluye a medida que las tareas vienen a mi.

Atrae a personas que necesitan codificar pero que no necesariamente quieren aprender a codificar profesionalmente.

Comparemos
Python con
Java.

Discute las
comparaciones
mientras las
revisas.

En las siguientes diapositivas está el mismo programa en Python y Java.

Estos programas solicitan tres valores decimales (float).

- Cantidad de dinero prestada llamada loan.
- La tasa de porcentaje anual (APR) de interés sobre el dinero prestado.
- La duración del préstamo expresada en meses se llama plazo (term).

A partir de estos valores el programa calcula el pago mensual y lo muestra.


```
loan = 1000.0
interest 0.05
term = 5
tempInterest = float(interest) / 12
result = float(loan) * \
    (tempInterest / (1.0 - ((1.0 + tempInterest) ** -float(term))))
print("Monthly Payment: %.2f" % result)
```

```
public class JavaCalculator01 {  
  
    public static void main(String[] args) {  
  
        double loan = 1000.0;  
        double interest = 0.05;  
        double term = 5;  
  
        var tempInterest = interest / 12.0;  
        var result = loan * (tempInterest / (1.0 - Math.pow((1.0 + tempInterest), -term)));  
  
        System.out.printf("Monthly Payment: %.2f%n", result);  
    }  
}  
  
// Single Source File Code example  
// runs with java JavaCalculator01.java
```

Java - Classe sin nombre

```
void main() {  
  
    var loan = 1000.0;  
    var interest = 0.05;  
    var term = 5;  
  
    var tempInterest = interest / 12.0;  
    var result = loan * (tempInterest / (1.0 - Math.pow((1.0 + tempInterest), -term)));  
  
    System.out.printf("Monthly Payment: %.2f%n", result);  
}  
  
// Single Source File Code example  
// runs with java --enable-preview --source 21 JavaCalculator01N.java
```

```
class PythonCalculator03:

    def func_input(self):
        loan = float(input("          loan: "))
        interest = float(input("        interest: "))
        term = float(input("          term: "))
        return loan, interest, term

    def func_process(self, input_data):
        (loan, interest, term) = input_data
        temp_interest = float(interest) / 12.0;
        return loan * (temp_interest / (1.0 - ((1.0 + temp_interest) ** -term)));

    def func_output(self, result):
        print('Monthly Payment: %.2f' % result)

    def func_work(self):
        input_data = self.func_input()
        result = self.func_process(input_data)
        self.func_output(result)

worker = PythonCalculator03()
worker.func_work()
```

```
public class JavaCalculator03 {

    private LoanRecord inputData() {
        var sc = new Scanner(System.in);
        System.out.printf("          Loan: ");
        var loan = sc.nextDouble();
        System.out.printf("          Interest: ");
        var interest = sc.nextDouble();
        System.out.printf("          Term: ");
        var term = sc.nextDouble();
        return new LoanRecord(loan, interest, term);
    }

    private double processData(LoanRecord loan) {
        var tempInterest = loan.interest() / 12.0;
        var result = loan.loan() * (tempInterest / (1.0 - Math.pow((1.0 + tempInterest), -loan.term())));
        return result;
    }

    private void outputResult(double result) {
        System.out.printf("Monthly Payment: %.2f%n", result);
    }

    public void perform() {
        var loan = inputData();
        var result = processData(loan);
        outputResult(result);
    }

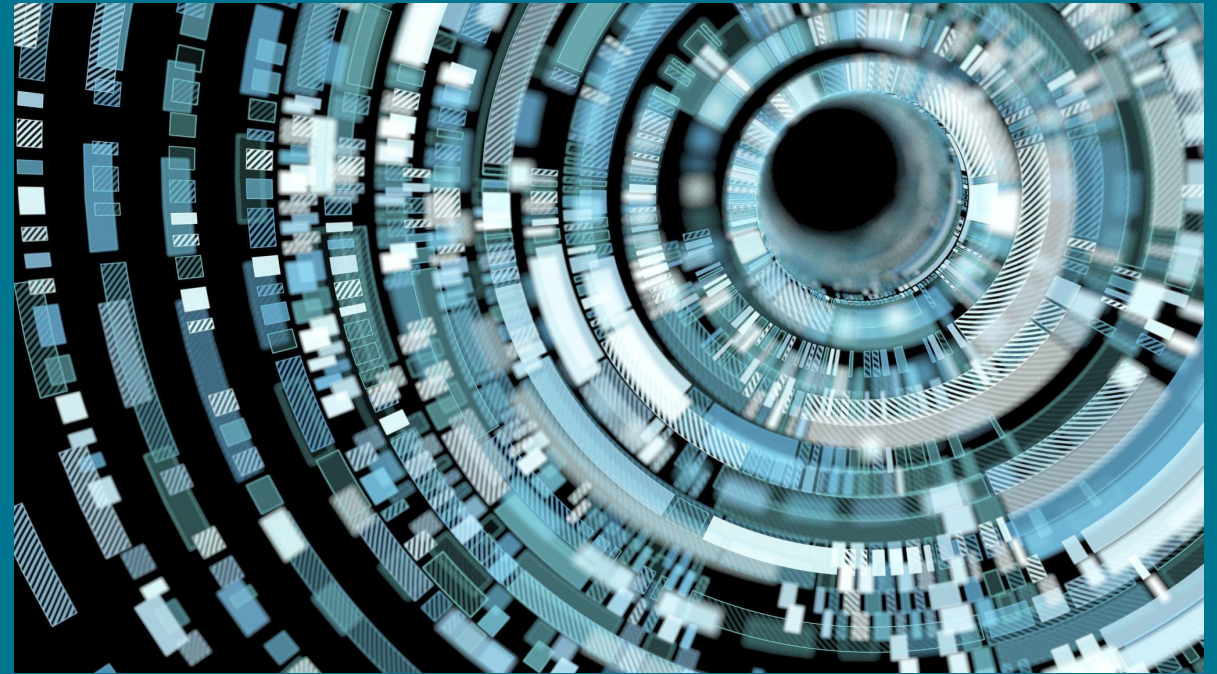
    public static void main(String[] args) {
        new JavaCalculator03().perform();
    }
}

record LoanRecord(double loan, double interest, double term) {}
```

Aprendizaje automático y Big Data

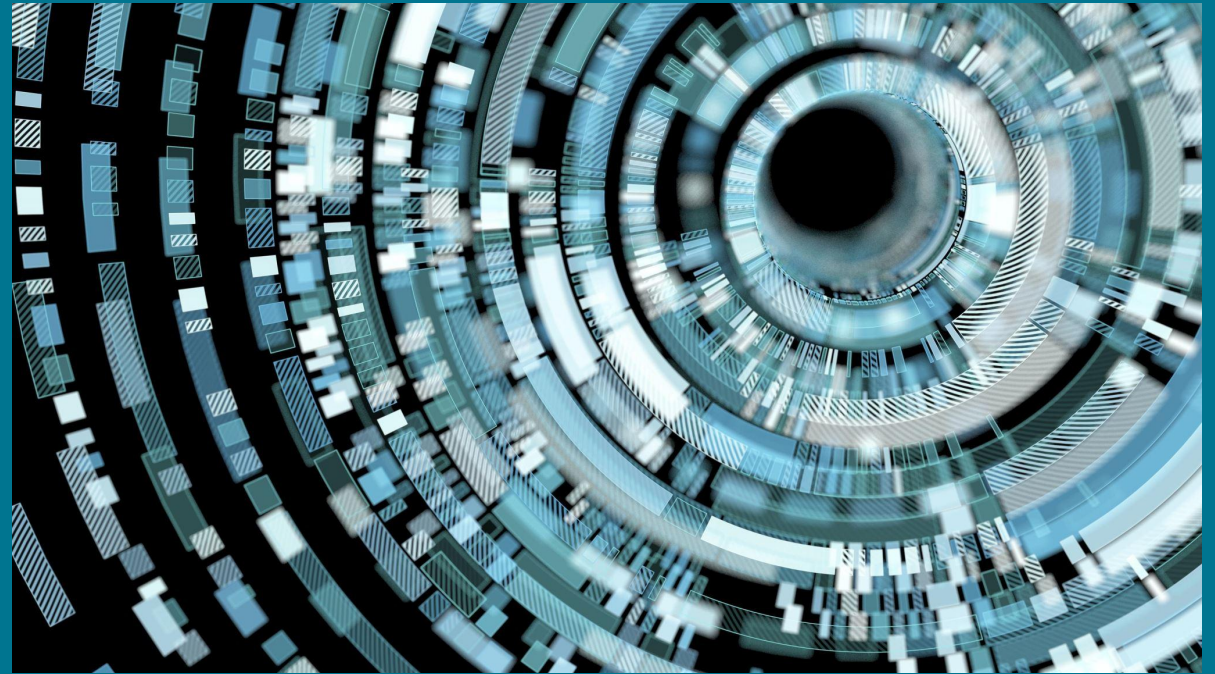
Reconocimiento visual *JSR 381*

- Java está haciendo aprendizaje automático ahora!
- La biblioteca Deep Java (DJL) de Amazon es una de varias implementaciones de este nuevo JSR.
- La profundidad y amplitud de las herramientas *Java* la convierten en la mejor plataforma para ML.



¿Por qué Python es ampliamente usado para AI/ML?

- Python está escrito en C.
- La mayoría de librerías para AI/ML están escritas en C.
- Esto simplifica el uso de estas librerías en Python.
- Con Java 21 tenemos Foreign Linker API & Foreign Memory Access API que simplifican el acceso a librerías escritas en C.



La máquina
virtual Java

JVM:

hogar de más
que Java

- Kotlin, Scala, Groovy, Clojure y más.
- Incluso existe un Python llamado Jython que se ejecuta en la JVM y admite interoperabilidad entre Java y Python.

¿Cuáles son tus expectativas laborales si aprendes Java?



Muchas instituciones financieras dependen de Java para ejecutar su core bancario.



Twitter, LinkedIn, Amazon y otros usan Java.



Tus expectativas son una función de qué tan bien puedes escribir código



Aprender Java es el mejor idioma para aprender y prepararte para trabajar con cualquier idioma durante tu carrera.



Es el mejor lenguaje para enseñar y brindarles a los estudiantes una comprensión clara de lo que significa programar.

Conclusión –

Comuníquese con las escuelas y los docentes/ profesores de todos los niveles.



*Descuentos para todos los miembros del grupo de usuarios de parte de Oracle University aquí:

<https://education.oracle.com/usergroupchampions>

Currently 25% discount through 12/21/2020.

El código de ejemplo se puede
encontrar en:

[https://github.com/omniprof/JCP_EC
Education_WG_Presentation](https://github.com/omniprof/JCP_EC_Education_WG_Presentation)