Universidad del Valle

Fundamentos de Programación Orientada a Eventos

Luis Yovany Romo Portilla, MsC.

Jueves 14:00 - 17:00 | Edf. B-13 -> SALA 4 -- MG -- MELENDEZ

10 Años Acreditación Institucional de Alta Calidad 2014 - 2024

Campus virtual

Agenda

- Presentación del curso.
- Ejercicio de Análisis y Diseños de POO
- Introducción al Lenguaje de Programación Java

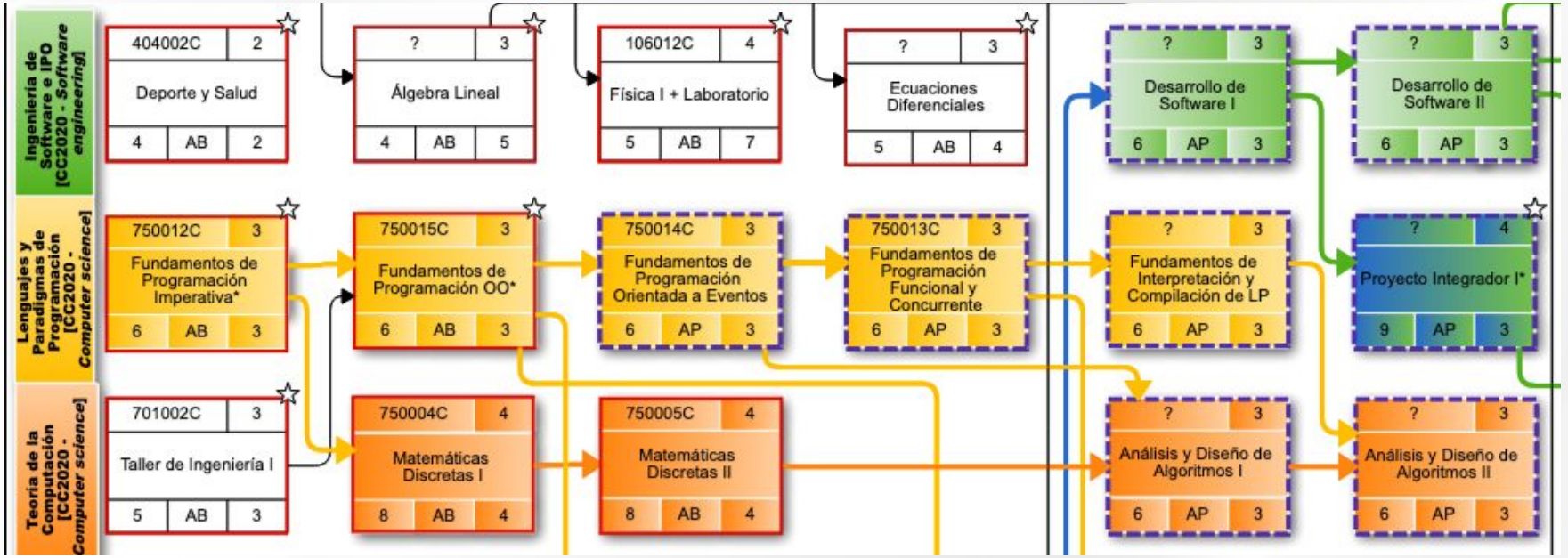
Fundamentos de Programación Orientada a Eventos

- **INFORMACIÓN GENERAL**

Código y Nombre				Fundamentos de programación orientada a eventos
Créditos				3
Horas de trabajo				Presenciales: 3 horas Trabajo independiente: 6 horas
Unidad(es) Académica(s)				Escuela de Ingeniería de Sistemas y Computación Facultad de Ingeniería
Programas Académicos				Programa académico de Ingeniería de sistemas
Prerrequisitos				Ninguno
Validable				Si
Habilitable				No
Tipo de Asignatura				Asignatura Básica (AB)
La asignatura favorece la Formación General				Científico Tecnológico
Si	x	**No**		

Fundamentos de Programación Orientada a Eventos

- **INFORMACIÓN GENERAL**



Fundamentos de Programación Orientada a Eventos

- **GENERAL**

  Desarrollar en el estudiante las habilidades para el uso de los lenguajes de programación orientados a objetos y el uso de una metodología de análisis, diseño e implementación de programas siguiendo la metodología orientada a objetos.

  Desarrollar aplicaciones con un diseño de interfaz gráfica que responda a las acciones de los eventos generados por los usuarios.

Fundamentos de Programación Orientada a Eventos

- **COMPETENCIAS Y CONTENIDOS**

COMPETENCIA	CONTENIDOS
Resolver problemas de ingeniería identificando diferentes alternativas de solución y desarrollando sistemas basados en TIC	Implementación de Interfaces, Escuchas (Listeners), Clases Adaptadores, Clases Internas
Aplicar conceptos y principios implicados en el proceso de diseño de interfaces gráficas de usuario durante el desarrollo de aplicaciones software.	Arquitectura Modelo Vista Controlador Aspectos Básicos de Diseño de Interacción y Usabilidad Excepciones Aplicación de Estructura de Datos y Colecciones Flujos de Datos

Fundamentos de Programación Orientada a Eventos

- **COMPETENCIAS Y CONTENIDOS**

COMPETENCIA	CONTENIDOS
Desarrollar proyectos de ingeniería analizando, modelando, diseñando, evaluando, gestionando, documentando, desplegando e implementando sistemas basados en TIC	Entorno Integrado de Desarrollo Estilo de Programación Estándar para documentación del código fuente
Utilizar un pensamiento crítico y creativo que le permite aprender de forma autónoma y permanente	Herramientas para gestión del tiempo y organización del trabajo. Herramientas para organización de ideas y representación de conceptos.

Fundamentos de Programación Orientada a Eventos

- **COMPETENCIAS Y CONTENIDOS**

COMPETENCIA	CONTENIDOS
Trabajar en equipo y aplicar diferentes formas y herramientas de comunicación durante la realización de proyectos de computación	Herramientas para gestionar los cambios del código fuente

Fundamentos de Programación Orientada a Eventos

COMPETENCIAS DEL PROGRAMA DE INGENIERÍA A LAS QUE APORTE ESTE CURSO

- C.E.13 Resolver problemas de ingeniería identificando diferentes alternativas de solución y desarrollando sistemas basados en TIC
- C.E.4 Aplicar conceptos y principios implicados en el proceso de diseño de interfaces gráficas de usuario durante el desarrollo de aplicaciones software
- C.E.14 Desarrollar proyectos de ingeniería analizando, modelando, diseñando, evaluando, gestionando, documentando, desplegando e implementando sistemas basados en TIC
- C.G.2 Utilizar un pensamiento crítico y creativo que le permite aprender de forma autónoma y permanente
- C.G.4: Trabajar en equipo y aplicar diferentes formas y herramientas de comunicación durante la realización de proyectos de computación

Fundamentos de Programación Orientada a Eventos

- **METODOLOGÍA**

El curso se desarrolla en las salas de cómputo, en clases semanales de 3 horas de las cuales 1 es

teórica y 2 son prácticas. En cada clase se aborda al menos un caso de estudio orientado a resolver

problemas relacionados con los resultados de aprendizaje que se quieren alcanzar. Dentro del proceso

formativo se considera como estrategia de aprendizaje la resolución de problemas, planteándose un

conjunto de mini proyectos en el contexto de juegos clásicos y arcades que de manera lúdica retan al

estudiante a poner en práctica los conceptos vistos en clase.

Fundamentos de Programación Orientada a Eventos

- **EVALUACIÓN**

Actividades Evaluativas	%
Diario del Curso (Tareas)	10
Parcial	10
Mini proyecto 1	10
Mini proyecto 2	10
Mini proyecto 3	15
Mini proyecto 4	20
Mini proyecto 5	25
TOTAL	100

Curso Pildoras informaticas

Fundamentos de Programación Orientada a Eventos

- **PLANEACIÓN**

SES	FECHA	TEMA
1	7 Abril	Introducción y Presentación Introducción a Java
2	21 Abril	Diseño de Interfaces Gráficas y Eventos
3	28 Abril	Mini Proyecto 1 (Asesoría)
4	5 Mayo	Versionamiento de Código - Layouts
5	12 Mayo	Mini Proyecto 2 (Asesoría)
6	19 Mayo	Eventos del Mouse y Teclado
7	26 Mayo	Mini Proyecto 3 (Asesoría)
8	2 Junio	Estructuras de Datos
9	9 Junio	Estructuras de Datos

Fundamentos de Programación Orientada a Eventos

- **PLANEACIÓN**

SES	FECHA	TEMA
10	16 Junio	Mini Proyecto 5 (Asesoría)
11	23 Junio	Manejo de Archivos y Excepciones
12	30 Junio	Manejo de Archivos y Excepciones
13	7 Julio	MVC
14	14 Julio	MVC
15	21 Julio	Mini Proyecto 5 (Asesoría)
16	28 Julio	Examen Parcial
17	3 Agosto	Examen Opcional

Fundamentos de Programación Orientada a Eventos

BIBLIOGRAFÍA

- ✓ Deitel H. M., Deitel P.J. *Cómo programar en Java*. Prentice-Hall, Hispanoamericana, S.A. Quinta edición, 2003, México.
- ✓ Lynn, S. Andrea. *Interactive programming in Java*. MIT, 1999, USA. http://www.cs101.org/ipij/
- ✓ Lewis Bil, Berg Daniel J. *Multithread programming with Java technology*. Sun Microsystems, Inc. 2000, USA.
- ✓ Martin, Robert C. UML para programadores Java. Pearson Prentice Hall. 2004.
- ✓ Wang. Paul. *Java con programación orientada a objetos y aplicaciones en la WWW*. Internacional. Thompson, 2000, México.

Otros Recursos

- ✓ Instalación de Java: Aquí puede descargar el JDK http://www.oracle.com/technetwork/java/javase/downloads/index.html
- ✓ **IDE Eclipse:** Debe instalar previamente el JDK. http://www.eclipse.org/ide/
- ✓ Documentación de Java: https://docs.oracle.com/javase/8/
- ✓ https://github.com/
- ✓ https://www.atlassian.com/es/software/jira

Ejercicio Práctico

Recordando el paradigma de POO

Análisis y Diseño

- **Restaurante Univalle**

El Gerente del restaurante requiere de un Ingeniero de Sistema para que diseñe bajo el paradigma de programación orientado a objetos el proceso de reserva de mesas.

El cliente puede llamar y solicitar la reserva de la mesa, para una fecha y hora específica, y mencionando la cantidad de personas que asistiran.

A la reserva se le asigna un mesero para que el cliente sea atendido una vez llegue al restaurante

- **Realizar el análisis y construir el Diagrama de Clase**

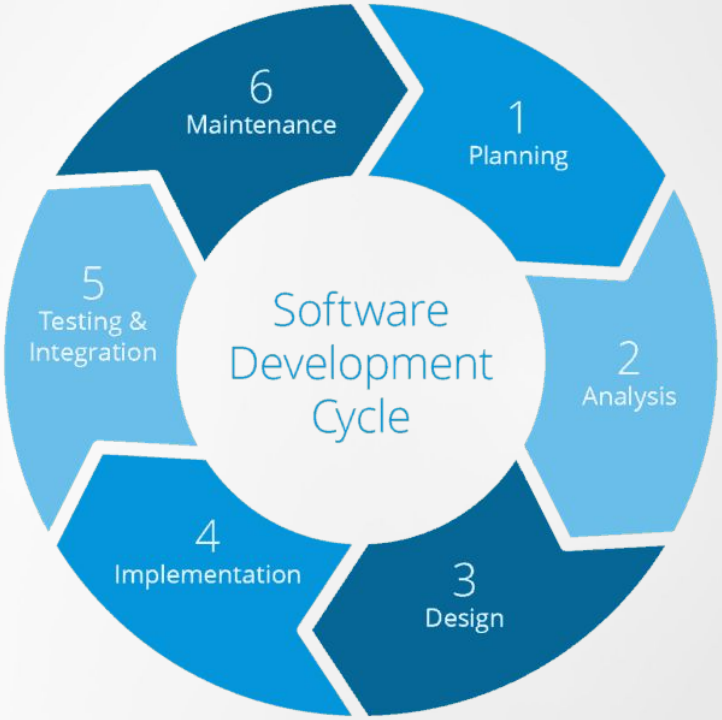
Introducción al Lenguaje de Programación java

LENGUAJES DE PROGRAMACION

- Un **lenguaje de programación** es un idioma artificial diseñado y creado para expresar algoritmos que puedan ser interpretados por una computadora.

- El lenguaje utilizado por la computadora se denomina lenguaje máquina, que consiste en una serie de 0 y 1 (datos binarios).

Ciclo de Vida de Desarrollo del Software

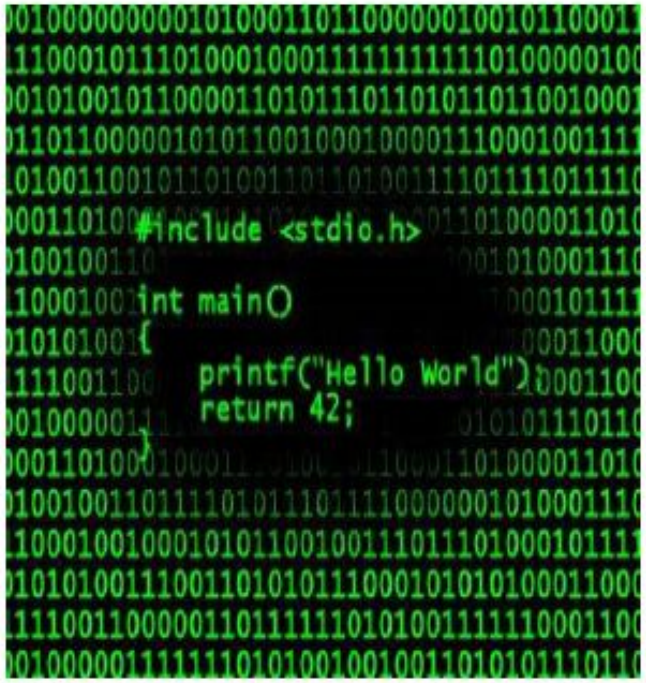


Fuente: https://www.mendix.com/blog/pursuing-a-full-agile-software-lifecycle/

LENGUAJES DE PROGRAMACION

- El lenguaje máquina es difícil de entender y programar, razón por la cual se han desarrollado otros lenguajes más sencillos de comprender (Java, C++, PHP, …).

- El código escrito en este tipo de lenguajes se transforma en código máquina para que la computadora pueda procesarlo.



LENGUAJES DE PROGRAMACIÓN

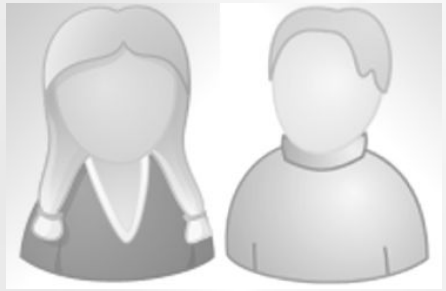
Código binario

Lenguaje de programación

Pseudocódigo

```
Inicio
   base, altura: entero
   area: real

   leer (base)
   leer (altura)

   area = (base*altura)/2
   imprimir (area)
Fin
```

Computador



Programadores

Java

- Java es un lenguaje de programación, desarrollado por Sun Microsystems a principios de los años 90.

- Las aplicaciones Java están compiladas en un *bytecode,* que luego es traducido a lenguaje de máquina.

- Una de las principales ventajas de Java, es que es **independiente de la plataforma**, lo que significa que un algoritmo escrito en Java, funcionará en diferentes sistemas operativos como Linux, Windows, Unix,….

JDK - JDE

- JDK (Java Develompmet Kit - Kit de desarrollo de Java): Es una colección de herramientas que le permiten al desarrollador realizar una serie de tareas, las más comunes o principales son: compilar (javac) e interpretar o ejecutar (java).

- JRE (Java Runtime Environment – Entorno en tiempo de ejecución de Java) : Es una colección de utilidades que permiten la ejecución de una aplicación escrita en Java.

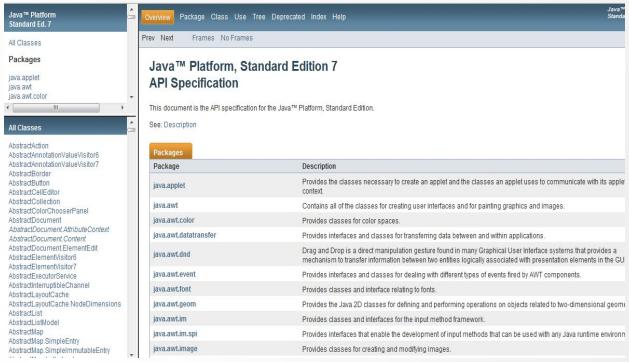
NOTA: Para ejecutar una aplicación en Java basta con tener instalado el JRE, pero para desarrollarla se requiere el JDK (al instalar el JDK se instala también el JRE).

API

- Interfaz de programación de aplicación: Provee una colección de clases que le permite al programador construir todo tipo de programas de acuerdo a sus necesidades.

- Existen varia API´s de acuerdo a las operaciones que se deseen realizar, normalmente las más utilizadas (y a usar en este curso) son las de Standar Edition (SE), su documentación se puede encontrar en:

http://download.oracle.com/javase/7/docs/api/



Ejemplo de la documentación del API para JSE7

JAVA en tres pasos

JAVA en tres pasos

- Escritura de algoritmos en Java

  Todos los algoritmos en Java se deben guardar en archivos con la extensión .java (*Triangulo.java*) (*código fuente*)

- Compilación

  Para compilar se usa la instrucción **javac** (*archivo de clase*)

```
javac  Triangulo.java
```

- Ejecución

  Para ejecutar un programa se usa la instrucción **java**

```
java Triangulo
```

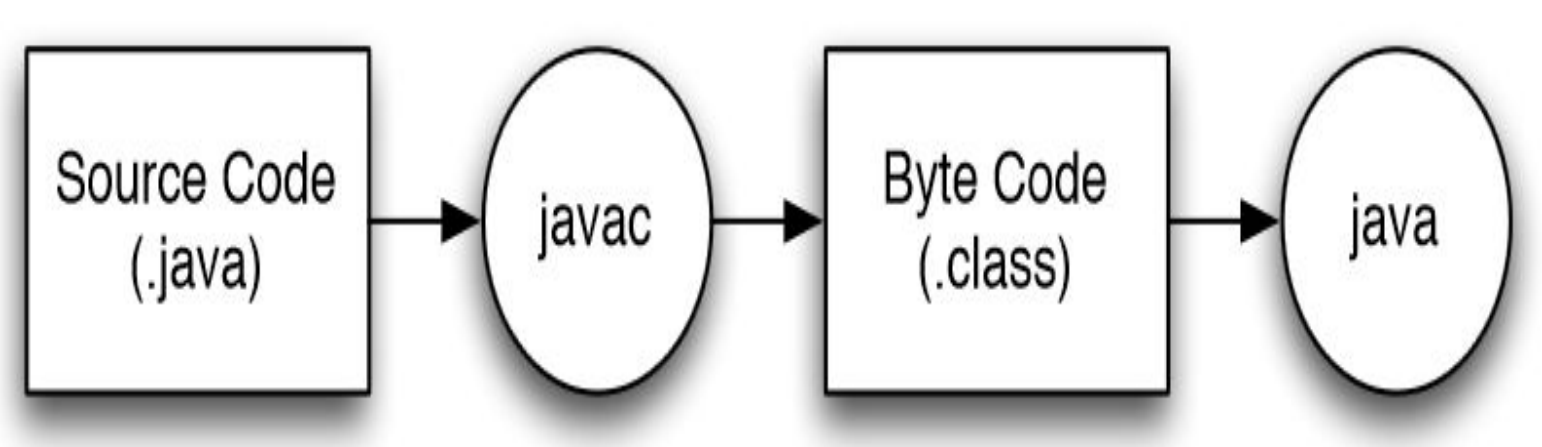
Ejecución de un programa

- Para Java una clase ejecutable es aquella que es **pública** y tiene **un método main**()
- Todos los archivos con código Java deben tener la extensión .java

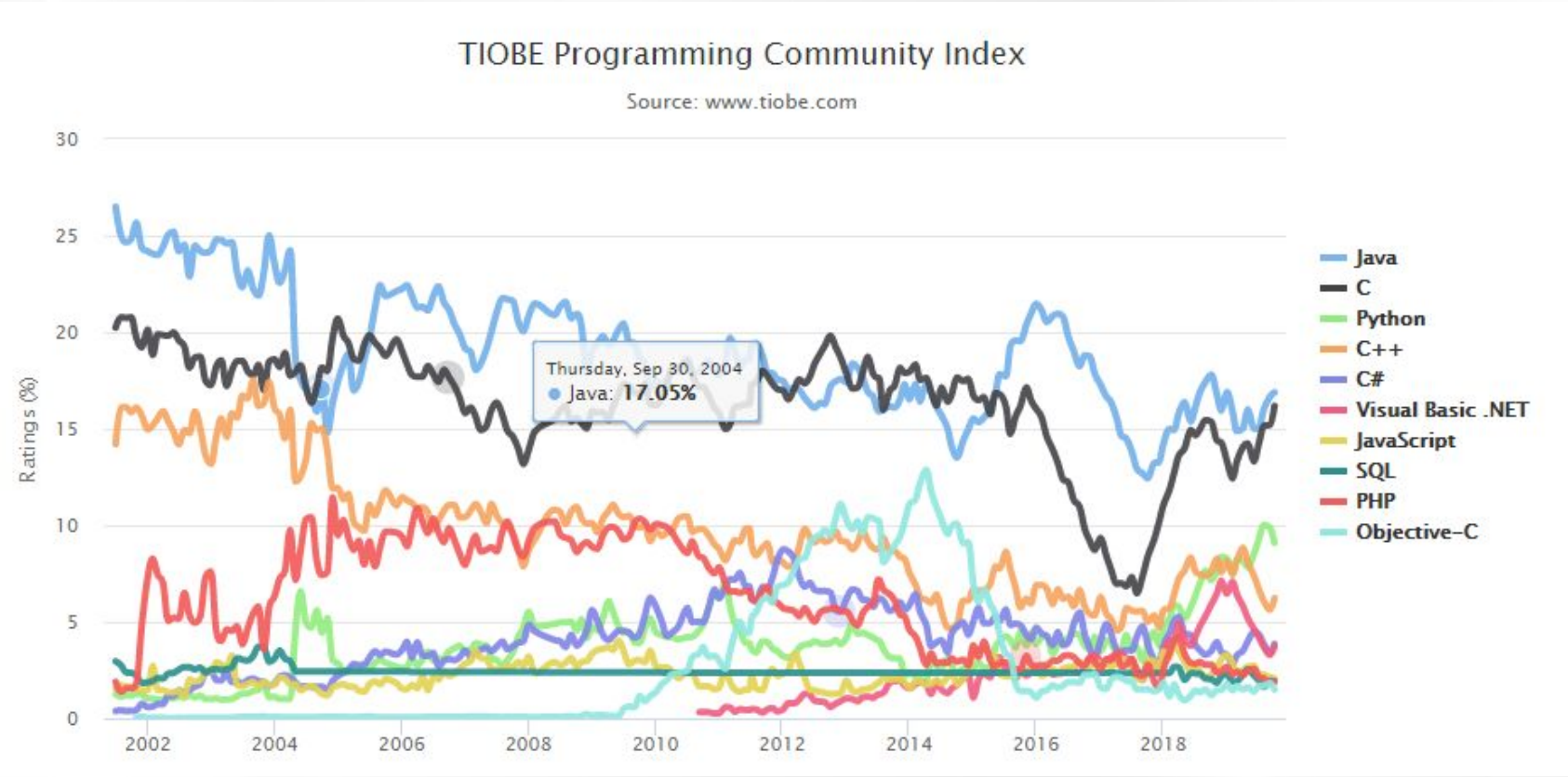
```
public class HolaMundo {

     public static void main(String args []){

        System.out.print ("Hola Mundo");

      }

}
```

HolaMundo.java

Ejecución de un programa



¿Por qué java?


TIOBE Programming Community Index
Source: www.tiobe.com
Ratings (%)
30
25
20
15
10
5
0
2002
2004
2006
2008
2010
2012
2014
2016
2018
Thursday, Sep 30, 2004
Java: 17.05%
Java
C
Python
C++
C#
Visual Basic .NET
JavaScript
SQL
PHP
Objective-C

Variables en Java

Una variable en Java es un **identificador** que representa una dirección de memoria que contiene información

Declaración	identificador	tipo
int i;	i	entero
String s;	s	referencia a string
int a[];	a	referencia a arreglo de enteros
int[] b;	b	referencia a arreglo de enteros

Tipos de variables

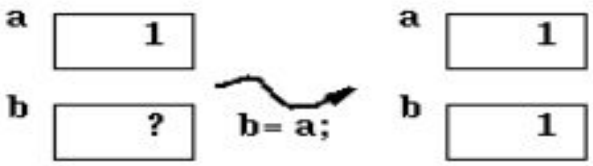
- ***Tipo de datos primitivos:*** Almacenan directamente un valor que siempre pertenece al rango de ese tipo.

- ***Referencias a objetos:*** Almacenan direcciones y no valores directamente. Una referencia a un objeto es la dirección de un área en memoria destinada a representar ese objeto

Tipos de variables

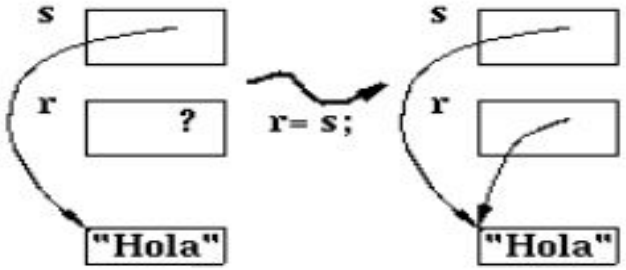
Datos primitivos

```
int a=1;
int b;

b= a;
```



Referencia a objetos

```
String s= "Hola";
String r;

r= s;
```



Variables en Java

En Java una variable no puede almacenar directamente un objeto, como ocurre en C y C++.

Por lo tanto cuando se dice en Java que una variable es un String, lo que se quiere decir en realidad es que la variable es una referencia a un String.

Tipos de datos primitivos

Tipo	Tamaño y formato	Rango
	enteros	
byte	8 bits - complemento a 2	-2^7 *al* $2^7 - 1$
short	16 bits - complemento a 2	-2^{15} *al* $2^{15} - 1$
int	32 bits - complemento a 2	-2^{31} *al* $2^{31} - 1$
long	64 bits - complemento a 2	-2^{63} *al* $2^{63} - 1$
	números reales	
float	32 bits - IEEE 754	
double	64 bits - IEEE 754	
char	16 bits - caracteres UNICODE	'\u0000' al '\uffff'
boolean	1 bit	true o false

Variables en Java

Valores por defecto

- ***false*** para los ***boolean***
- ***0*** para los atributos de tipo numérico (***int***, ***double***, ***float***, etc)
- ***null*** para las referencias a objeto

Buenas prácticas de programación

- Variables con significado
- Espacios en blanco
- Identación

Variables con significado

```
String a1;
int a2;
double b;              // BAD!!

String firstName;      // GOOD
String lastName;       // GOOD
int temperature;       // GOOD
```

Espacios en blanco

```
double cel=fahr*42.0/(13.0-7.0);
```

```
double cel = fahr * 42.0 / (13.0 - 7.0);
```

Identación

```
public static void main (String[] arguments) {
    int x = 5;
    x = x * x;
    if (x > 20) {
        System.out.println(x + " is greater than 20.");
    }
    double y = 3.4;
}
```

Operadores

- Asignación: =
- Adición: +
- Substracción:
- Multiplicación: *
- División: /
- Residuo: % (mod)

Operador División

La división opera de manera diferente entre enteros int y double.

```
double a = 5.0/2.0; //  a = 2.5
int b = 4/2; //  b = 2
int c = 5/2; //  c = 2
double d = 5/2; //  d = 2.0
```

Funciones matemáticas

Java posee la clase **Math**, la cual contiene una serie de métodos para el manejo de funciones matemáticas básicas. Entre sus métodos se encuentran:

Math.sin(x)
Math.cos(Y)
Math.pow(x,y)
Math.log(x)

http://docs.oracle.com/javase/7/docs/api/

Operadores

`X > Y`	Mayor que
`X < Y`	Menor que
`X >= y`	Mayor o igual que
`X <= Y`	Menor o igual que
`X == Y`	Igualdad
`X = Y`	Asignación

```
X > Y   &&   X > Z
```

(AND)

```
X > Y   ||   X > Z
```

(OR)

Condicionales

```
if ( logical condiction) {

    statements

}
```

```
if ( logical condiction) {

    statements;
}
else {

    statements;

}
```

Condicionales

```
public static void test(int x) {
        if (x > 5) {
            System.out.println(x + " is > 5");
        }
}
```

```
public static void main(String[] arguments) {
        test(6);
        test(5);
        test(4);
}
```

Condicionales

```
if (CONDITION) {
    STATEMENTS
} else if (CONDITION) {
    STATEMENTS
} else if (CONDITION) {
    STATEMENTS
} else {
    STATEMENTS
}
```

```
public static void test(int x) {
    if (x > 5) {
        System.out.println(x + " is > 5");
    } else if (x == 5) {
        System.out.println(x + " equals 5");
    } else {
        System.out.println(x + " is < 5");
    }
}
```

Ciclos

```
while   (condition)   {
    statements
}
```

```
int i = 0;
while (i < 3) {
    System.out.println("Rule #" + i);
    i = i+1;
}
```

Ciclos

```
for ( init; condition; update) {

      statements
}
```

```
for ( i=0; i <3;  i=i+1) {
      System.out.println ( i );
}
```

Ciclos

```
do {
     statements

}while ( condition);
```

```
int i=0;
do {
     System..out.println(”Hello word“);

          i++;

} while ( i<3);
```

Introducción

Gracias!!!

Luis Yovany Romo Portilla, MsC.

Fundamentos de Programación Orientada a Eventos