Programando en Java

Día 1: introducción y tipos de datos básicos

Área de Ingeniería Telemática UPNA

Java

- Lenguaje desarrollado (~1995) por Sun microsystems (ahora propiedad de Oracle)
- Java es open source
- Inicialmente popular por integración con la web... ahora más por programas portables entre plataformas
- Flexible, simple y con una gran biblioteca de utilidades
- Adaptado a diferentes entornos
  - Aplicaciones de empresa
  - Web
  - Sistemas moviles y embebidos. Android

Yo prefiero el lenguaje <X>

- Esto es una iniciación a la programación

  Programas simples y fáciles de hacer en cualquier lenguaje

- Si os gusta programar: probar y aprender otros lenguajes por vuestra cuenta es bueno

  C, C++, Objective-C, Python, PHP, Javascript, Perl, Bash...

- Podéis preguntar dudas sobre otros si lo intentáis...

- Pero las prácticas y ejercicios solo pueden entregarse en Java

- Si pasar de un lenguaje a otro te parece difícil es señal de que deberías céntrate primero en Java

Vamos a usar el IDE <x>?

- IDE = Integrated Development Environment (Eclipse, NetBeans, Xcode...)
  - Son útiles pero al empezar ocultan el funcionamiento
  - Vais a ser ingenieros tenéis que entender como funciona
  - Los editores que resaltan la sintaxis si que vienen bien
- Debugers
  - Herramientas muy útiles para analizar y corregir errores
  - Pero para empezar es mejor pensar :-)
  - Encontrar los errores y entender por que funciona mal un programa es igual de importante que saber escribirlo
  - Los programas NUNCA funcionan a la primera

First learn stand, then learn fly...

Into Java

- Java es un lenguaje orientado a objeto
- Los programas son clases, una clase es una descripción de un objeto.
- Las clases deben definirse en un fichero que se llame nombredelaclase.java
- Las clases tienen metodos (funciones) que son instrucciones a seguir
- Las clases que tienen un metodo llamado `main` constituyen un programa que puede lanzarse para que se sigan esas instrucciones
- Eso es todo lo que veremos de programación orientada a objeto... de momento
- Queremos hacer programas para que el ordenador haga tareas

Primer ejemplo

- Un programa que escriba algo por pantalla
- Un fichero llamado Hello.java

  Esto es el código fuente

```
public class Hello {

    public static void main(String[] args) {
        // voy a escribir algo
         System.out.println("hola,");
         System.out.println("mi primer programa en java");
    }

}
```

- Una clase llamada Hello con un método llamado main

Lanzando el ejemplo

- Usamos el compilador `javac` para generar (compilar) el programa a partir del codigo fuente

```
$ javac Hello.java
$ ls
Hello.class  Hello.java
```

```
$ java Hello
hola,
mi primer programa en java
```

- Si hay errores en la compilación aparecerán en pantalla
- Que el programa compile no quiere decir que esté bien solo que esas instrucciones están correctamente expresadas como instrucciones y se puede generar código ejecutable a partir de ellas

El ejecutable

- Los ficheros Hello.class son el programa que se puede distribuir
- Las fuentes sólo hacen falta para obtener los .class si alguien va a usar el programa no las necesita
- Otros lenguajes normalmente generan ejecutable especifico para un procesador y un sistema operativo y no pueden usarse en otros.
- La gran idea de la plataforma java es que los .class son ejecutados por una maquina virtual.
  Se pueden ejecutar en cualquier procesador y sistema operativo para el que se disponga de la maquina virtual de java (Java Runtime Environment)
- Los programas complicados en java no son una sola clase sino un conjunto de clases. Ficheros Nombre.jar

Manejando información

- Lo principal que va a hacer un programa es procesar información de diferentes tipos
- Para manejarla tenemos que poder almacenarla
- Variables que permiten almacenar y etiquetar información para usarla en las instrucciones
- Diferentes tipos de variables para diferentes tipos de información

Tipos básicos (primitive types)

- Numérica

  `byte` - un entero almacenable en 8 bits entre -128 y 127

  `short` - entero 2bytes -32768 a 32767

  `int` - entero 4bytes -2147483648 a 2147483647

  `long` - entero 8bytes -9223372036854775808 a 9223372036854775807

  `float` - punto flotante 4 bytes según estandar IEEE-754

  `double` - punto flotante 8 bytes según estandar IEEE-754

- Caracteres

  `char` - 2 bytes puede almacenar una letra en codificacion Unicode

- **`boolean`** - 1 bit : puede almacenar `true` o `false`

- Y otros tipos mas complejos en próximas clases...

  Por ejemplo el String[] que aparece en el ejemplo...

Variables

- Dentro de un método (normalmente al principio) hay que declarar las variables con el tipo de almacenamiento que necesitamos.
- Podemos darle valor en la declaración o después con =

```
{
// quiero usar dos enteros
int x,y=4;
// un flotante
float r;
// una letra y una cadena
char c='b';
String nombre;
boolean preparado=false;

x=3;
r=2.0f;
nombre="Mikel";
...
```

Literales

- Los valores constantes que asignamos se llaman literales
- Literales numéricos

  1 -3 12313 123L 20 0x20 032

  1.0 1.2e+3 3.2f 3.2d
- Literales caracter: 'a' 'b' 'A' '?'
- Literales cadena:

  "Hola" "Cadena con\n varias lineas\n-----"
- Literales booleanos: true, false

Expresiones

- Todo es una expresión
- Aritemética normal...

```
y=x+5;
y=4-x;
r=3.0/4.0;
```

- Ojo con la conversion entre enteros y flotantes

```
r=3/4;      // da 0
r=3/2;      // da 1 division entera
r=3.0/4;    // da 0.75
r=1.0*x/y; // para asegurarse de float
```

- Expresiones con boleanos, cadenas...

```
boolean ymayor = y>x;
String usuario="Neo";
String a="Hola "+usuario;
```

Más expresiones

- Operaciones con tipos diferentes, los tipos básicos se promueven y convierten a tipos que se puedan operar (por eso 2/4 da 0 y 2.0/4 da 0.5)
- Los tipos básicos al sumarlos a cadenas se convierten en cadenas apropiadas.

  No es lo mismo el número 42 que la cadena "42"

```
int x=42;
```

En memoria

```
00000000 00000000 00000000 00101010
```

```
"42 "
```

En memoria

```
00000000 00110100 00000000 00110010 00000000 00100000
    '4'               '2'               ' '
     52                50                32
```

```
00110100 00110010 00100000
```

```
"[  "+x+"  ]"  ⟶  "[ 42 ]"
```

Imprimiendo resultados

- Al sumar a cadenas se convierte a cadena y se concatena
- Es muy útil para imprimir resultados

```
String aimprimir= "x=" + x ;
System.out.println(aimprimir);
System.out.println("usuario: "+usuario+"\nid: "+id);
```

```
x=42

usuario: Mikel
id: 1239y6a976
```

- Para imprimir con mas control

```
System.out.printf("cadena con formato", valor1, valor2... )
System.out.format("cadena con formato", valor1, valor2... )
```

`%f %d %s` se sustituyen por valores flotantes, enteros, cadenas...

```
Sytem.out.printlf("time %.2f segundos\n",time);
```

```
time 3.45 segundos
```

time se imprime como flotante
con 2 cifras decimales

¿Que podemos hacer con esto?

- Ejemplo: calculemos el tiempo de tarda un paquete en atravesar un enlace...

```
public class Enlace {

    public static void main(String[] argv) {
        float enlace_v_tx;      // en bps
        float enlace_v_prop;    // en m/s
        float enlace_longitud;  // en m
        int paquete_l;         // en bytes
        float t_tx, t_prop, t_total; // en s

        enlace_v_tx=10e6f;
        enlace_v_prop=200e6f;
        enlace_longitud=10000f;
        paquete_l=1500;

        System.out.println("Entrada -------------");
        System.out.println("v_tx= "+enlace_v_tx+" bps");
        System.out.println("v_prop= "+enlace_v_tx+" m/s");
        System.out.println("l= "+enlace_longitud+" m");
        System.out.println("s= "+paquete_l+" B");

        t_tx=paquete_l/enlace_v_tx;
        t_prop=enlace_longitud/enlace_v_prop;
        t_total=t_prop+t_tx;

        System.out.println("t= "+t_prop+" + "+t_tx+" = "+t_total);
    }
}
```

¿Que podemos hacer con esto?

- Imprimiendo un poco mejor

```
public class Enlace {

    public static void main(String[] argv) {
        float enlace_v_tx;      // en bps
        float enlace_v_prop;    // en m/s
        float enlace_longitud;  // en m
        int paquete_l;         // en bytes
        float t_tx, t_prop, t_total; // en s

        enlace_v_tx=10e6f;        enlace_v_prop=200e6f;
        enlace_longitud=10000f;        paquete_l=1500;

        System.out.printf("Entrada -------------\n");
        System.out.printf("v_tx= %.1f bps \n",enlace_v_tx);
        System.out.printf("v_prop= %.1f m/s \n",enlace_v_tx);
        System.out.printf("l= %.1f m s=%d B \n",enlace_longitud,paquete_l);

        t_tx=paquete_l/enlace_v_tx;
        t_prop=enlace_longitud/enlace_v_prop;
        t_total=t_prop+t_tx;

        System.out.printf("t= %f + %f = %f s\n",t_prop,t_tx,t_total);
        System.out.printf("t= %.3f + %.3f = %.3f ms\n",
                          t_prop*1000.0,t_tx*1000.0,t_total*1000.0);
    }
}
```

Ideas para recordar

- Java es un lenguaje
- Sabemos escribir código fuente en ficheros nombre.java
- Compilador javac
- Maquina virtual java (o java JRE) java
- Las clases se lanzan con `java nombreclase`

  Eso ejecuta las instrucciones del método `main`
- Se trata de hacer programas que hagan cálculos útiles y nos resuelvan problemas
- De momento sabemos manejar variables de diferentes tipos

  `byte, short, int, long, float, double, char, boolean` …