

Clase 2 - Elementos Básicos

Taller de preparación para la OCI - PUC

Hoy aprenderemos

- Crear un programa nuevo de Java que:
 - Pida valores al usuario
 - El programa ejecute acciones con los valores
 - Se entregue al usuario los resultados de las acciones

Hoy aprenderemos

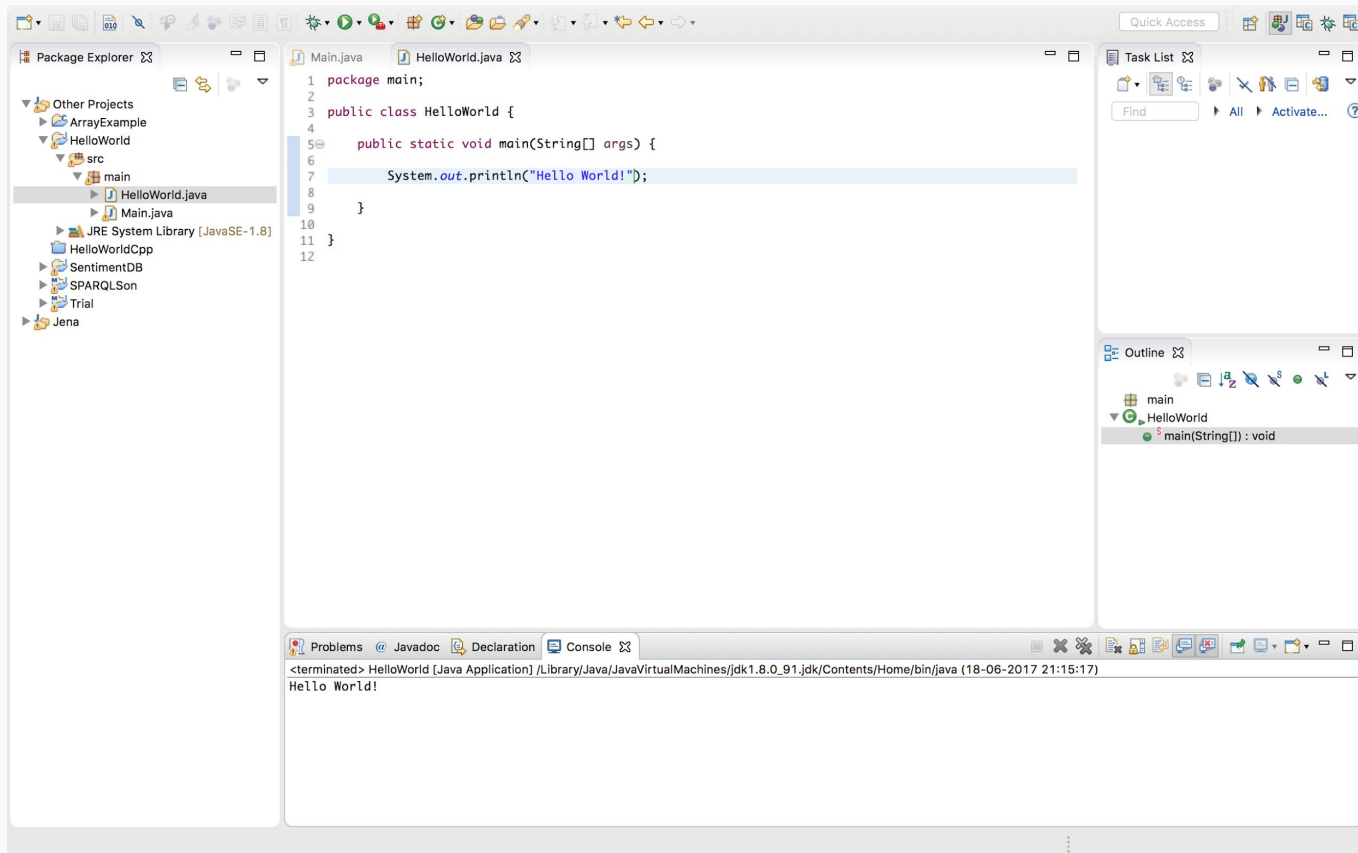


Eclipse

Eclipse es una herramienta que nos permite escribir nuestros programas en Java



Eclipse

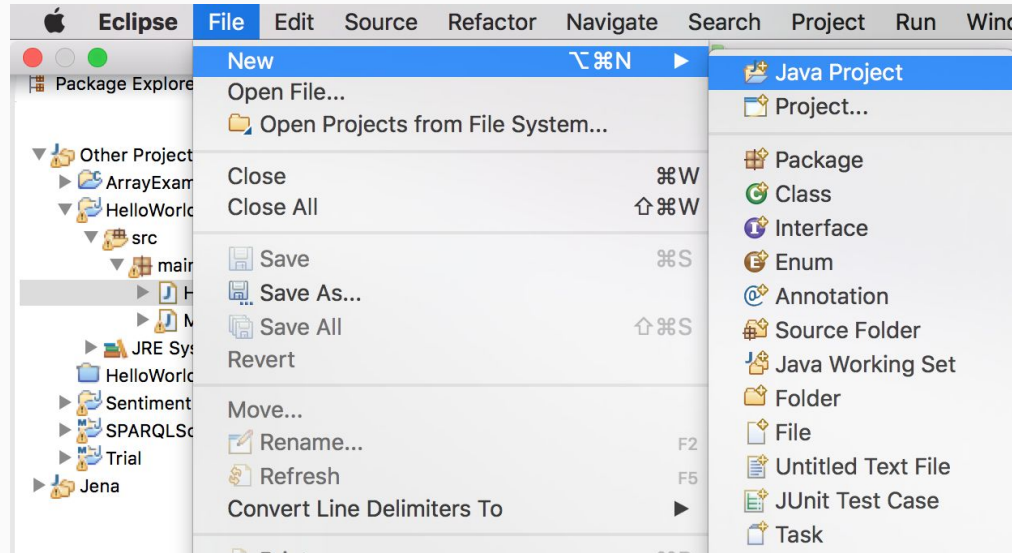


Podemos destacar:

- 1) **Package Explorer:** nos permite navegar a través de nuestros distintos proyectos
- 2) **Editor de texto:** es la ventana central, donde escribimos nuestros programas
- 3) **Consola:** lugar donde se interactúa con el usuario, mostrando resultados o pidiendo valores

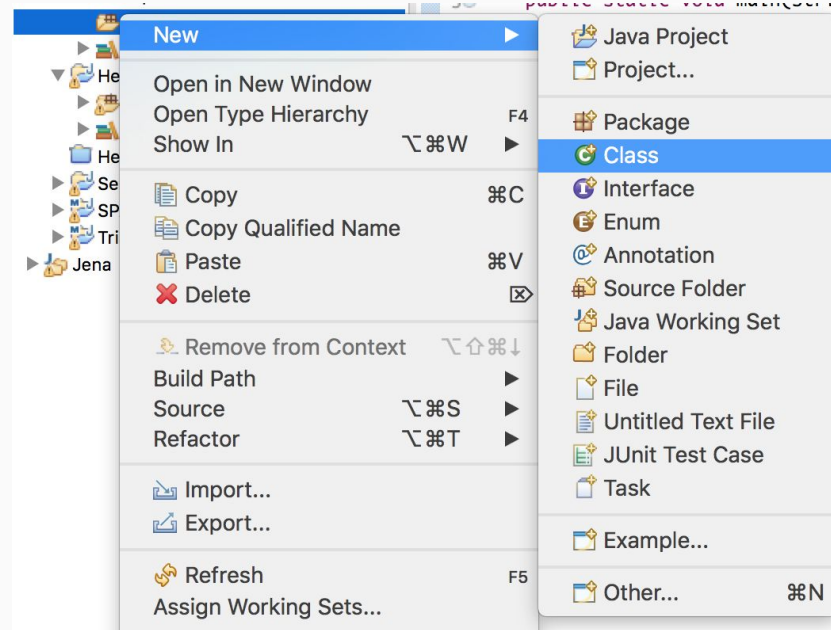
Crear un proyecto en Eclipse

Para crear un nuevo proyecto nos vamos a Archivo (File), después a New y luego Java Project

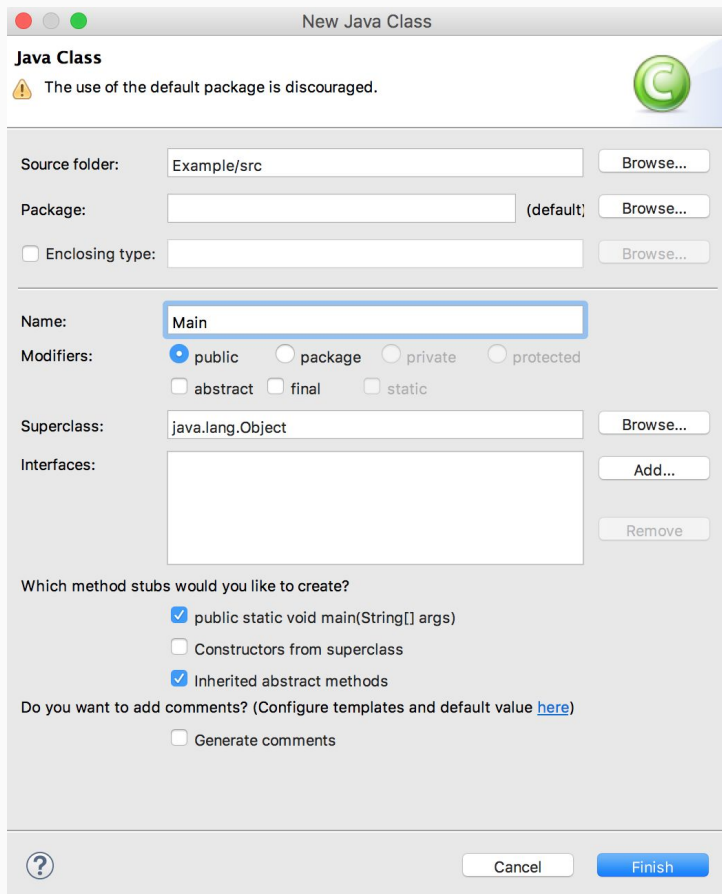


Crear una nueva clase en Eclipse

Una vez creado un proyecto, hacemos click con el botón derecho y hacemos click a New y luego Class



Crear una nueva clase en Eclipse



The screenshot shows the 'New Java Class' dialog box in Eclipse. The title bar says 'New Java Class'. Below the title bar, there's a 'Java Class' section with a warning icon and the text 'The use of the default package is discouraged.' and a green 'C' icon. The dialog has several fields and buttons:

- Source folder:** A text field containing 'Example/src' and a 'Browse...' button.
- Package:** A text field containing '(default)' and a 'Browse...' button.
- Enclosing type:** A checkbox labeled 'Enclosing type:' followed by a text field and a 'Browse...' button.
- Name:** A text field containing 'Main'.
- Modifiers:** A group of radio buttons and checkboxes: 'public' (selected), 'package', 'private', 'protected', 'abstract', 'final', and 'static'.
- Superclass:** A text field containing 'java.lang.Object' and a 'Browse...' button.
- Interfaces:** A text area and an 'Add...' button, with a 'Remove' button below it.
- Which method stubs would you like to create?:** A group of checkboxes: 'public static void main(String[] args)' (checked), 'Constructors from superclass', and 'Inherited abstract methods' (checked).
- Do you want to add comments?:** A checkbox labeled 'Generate comments'.

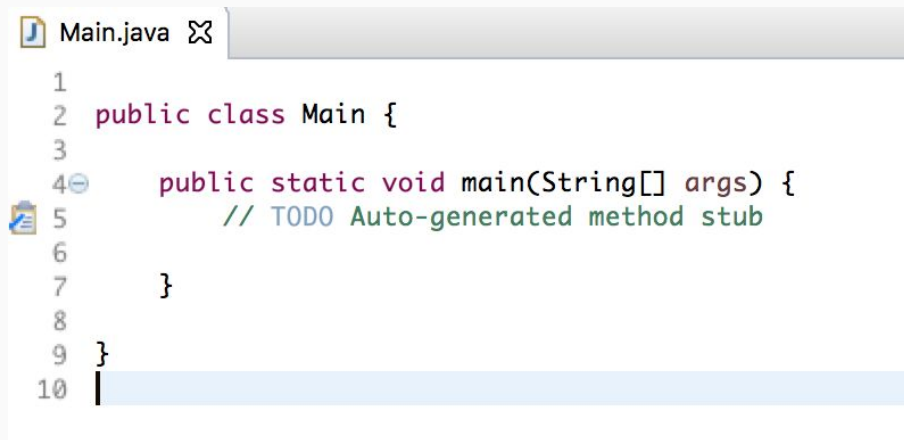
At the bottom, there's a question mark icon, a 'Cancel' button, and a 'Finish' button.

Luego le agregamos un nombre (generalmente **Main** o **Principal**), y hacemos click al casillero:

```
public static void main(String[] args)
```

Nueva clase principal

Verás algo como la imagen a continuación, podemos empezar a programar dentro de la sección **public static void** `main(String[] args)`



```
Main.java
1
2 public class Main {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6
7     }
8
9 }
10
```

Hello World

Ahora haremos un programa que imprima “Hola mundo!” al usuario (esto es un clásico en cualquier lenguaje de programación)

```
public class Main {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hola mundo!");  
  
    }  
  
}
```

Hello World

```
Main.java  🔍  
1  
2 public class Main {  
3  
4     public static void main(String[] args) {  
5  
6         System.out.println("Hola mundo!");  
7  
8     }  
9  
10 }  
11
```

Problems @ Javadoc Declaration Console 🔍

<terminated> Main (3) [Java Application] /Library/Java/JavaVirtualMach
Hola mundo!

Elementos básicos de Java

Variables

- Las variables son la base de todo lenguaje de programación
- Son elementos a los que se les asignan valores para poder trabajar con ellos
- Las variables pueden tener distintos tipos

Variables (analogía)



- Distintas cajas para distintos tipos
- Le puedo asignar una etiqueta a cada caja
- Puedo cambiar el contenido de la caja sin cambiar su etiqueta

Veamos un programa simple

```
public class Main {  
    public static void main(String[] args) {  
  
        int number;  
        number = 0;  
  
    }  
}
```

En este programa estamos creando una variable de **tipo** `int` (número entero), asignándole el **valor** 0.

Ahora además estamos imprimiendo el valor de number al usuario

```
public class Main {  
    public static void main(String[] args) {  
  
        int number;  
        number = 0;  
        System.out.println(number);  
  
    }  
}
```

Variables

Podemos asignar el valor en el momento que creamos la variable

```
public class Main {  
    public static void main(String[] args) {  
        int number = 0;  
        System.out.println(number);  
    }  
}
```

Variables

Podemos cambiar el valor de la variable, pero se sigue llamando igual

```
public class Main {  
    public static void main(String[] args) {  
  
        int number = 0;  
        number = 42;  
        System.out.println(number);  
  
    }  
}
```

Tipos de datos

Atributos asociados a cada dato que indican al computador cómo administrarlos.
Se destacan:

- 1) Número entero (**int**)
- 2) Números reales (**double**)
- 3) Caracteres (**char**)
- 4) Texto (**String**)
- 5) Valores de verdad (**boolean**)

Ejemplo de asignaciones

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int myNumber = 0;  
  
        String myText = "Soy un texto";  
        myText = "Ahora soy otro texto";  
  
        boolean myBoolean;  
        myBoolean = true;  
        myBoolean = false;  
  
    }  
  
}
```

Veremos expresiones básicas para hacer operaciones entre variables

- 1) Suma, resta, multiplicación y división (+, −, *, /)
- 2) Módulo, que es el resto de la división entera entre dos números (%)
- 3) Comparaciones (==, <=, >=, >, <)

Ejemplo de expresiones

```
public class Main {  
    public static void main(String[] args) {  
  
        int a = 7;  
        int b = 4;  
  
        int result = a + b;  
        // El valor de result es 11  
  
        result = a - b;  
        // Ahora el valor de result es 3  
  
        result = 2 * a;  
        // Ahora el valor de result es 14  
  
        result = a / b;  
        // Ahora el valor de result es 1  
  
        result = a % b;  
        // ahora el valor de result es 3  
  
    }  
}
```


La división entre dos números enteros debe ser un entero. Si deseamos obtener la división real podemos hacer lo siguiente:

```
public class Main {  
    public static void main(String[] args) {  
        double a = 7;  
        double b = 4;  
  
        double result = a / b;  
        // El valor de result es 1.75  
    }  
}
```

También es posible pedir que trate a números enteros como si fuesen de tipo **double**:

```
public class Main {  
    public static void main(String[] args) {  
        int a = 7;  
        int b = 4;  
  
        double result = ((double) a) / ((double) b)  
        // El valor de result es 1.75  
    }  
}
```

Ejemplo de una asignación a sí mismo

Veamos un ejemplo donde defino el valor de una variable en base a sí misma, con ayuda de expresiones aritméticas

```
public class Main {  
    public static void main(String[] args) {  
        int a = 3;  
        a = a+1;  
        // ahora a vale 4  
  
        a = 2*a;  
        // ahora a vale 8  
    }  
}
```

Comparaciones

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int a = 3;  
        boolean comparison = (a == 3);  
        // comparison es true  
  
        int b = 4;  
        comparison = (a == b);  
        // comparison ahora es false  
  
        comparison = (a >= 10);  
        // comparison es false, porque a no es mayor o igual a 10  
  
    }  
  
}
```

Ojo! El resultado de una comparación es de tipo `boolean` (verdadero o falso)

Vamos a ver algunas funciones ya implementadas en Java

Para mostrar información en pantalla al usuario usamos la función `System.out.println`

```
public class Main {  
    public static void main(String[] args) {  
  
        int a = 2;  
        int result = 2*a;  
  
        System.out.println(result);  
  
    }  
}
```

Podemos juntar lo que queremos imprimir junto a texto

```
public class Main {  
    public static void main(String[] args) {  
  
        int a = 2;  
        int result = 2*a;  
  
        System.out.println("El resultado es: " + result);  
  
    }  
}
```


Recibir input del usuario

Para pedir valores al usuario utilizamos Scanner

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        System.out.println("Ingrese su nombre: ");
        Scanner scanner = new Scanner(System.in);
        String username = scanner.nextLine();
        System.out.println("Hola " + username);
    }
}
```

Notemos que hay que importar el Scanner, y que el resultado de nextLine se guarda como un String.

Transformar strings en números

Como Scanner nos guarda valores strings, queremos convertir **Strings** en **números**

```
// En número entero
String number = "42";
int result = Integer.parseInt(number);
System.out.println(result);

// En número real
String number = "10.3";
double result = Double.parseDouble(number);
System.out.println(result);
```

En Java están implementadas algunas funciones matemáticas en la librería Math

```
// Calculamos 3 elevado a 4  
int number = 3;  
double pow = Math.pow(3, 4);  
System.out.println(pow);
```

Clase 2 - Elementos Básicos

Taller de preparación para la OCI - PUC