

# Programación Orientada a Objetos

## Curso 2023/2024

### Prácticas – Sesión 1

#### Entorno de desarrollo Java

##### 1. Trabajo en el entorno Eclipse.

- Crea un directorio (en un *pendrive*, en el escritorio, etc.) que se denomine por ejemplo *workspace*, que será el que utilices **SIEMPRE** como espacio de trabajo de Eclipse. En este espacio de trabajo serán donde se cree el proyecto Java.
- Arranca el entorno de desarrollo Eclipse y selecciona el espacio de trabajo creado en el paso anterior.
- Crea un proyecto Java en el entorno Eclipse con la siguiente configuración (Figura 1):
  - o Nombra el proyecto siguiendo la plantilla: GrupoXX-Apellidos-Nombre. Por ejemplo, Grupo11-MartinezPerez-Alfonso.
  - o Selecciona el entorno de ejecución (JRE) **JavaSE-1.8**

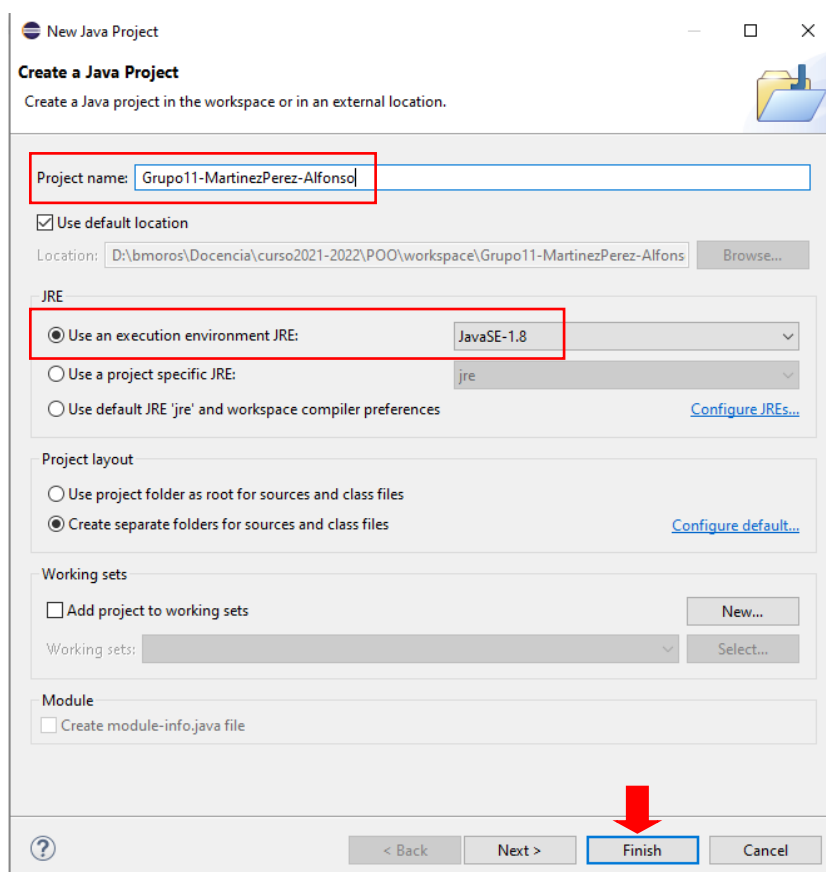
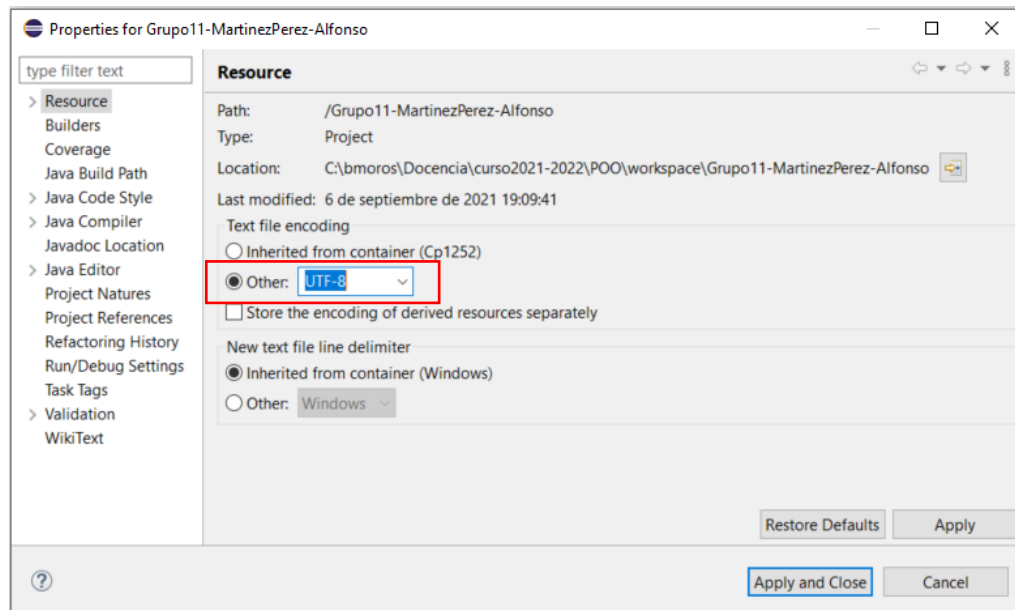


Figura 1 Creación de un proyecto Java.

- Accede a las propiedades del proyecto (menú Project → Properties) y selecciona la codificación UTF-8 para el fichero de texto (Figura 2) en la sección “Resource”.



**Figura 2 Propiedades del proyecto.**

- Crea el paquete `practica0` y añade la clase `Programa` a ese paquete.
- Dentro del código de la clase `Programa` escribe `main` y después pulsa `ctrl+Espacio` para probar las **plantillas de código** que te ofrece el entorno de desarrollo. Aparece el esqueleto del método `main` que sólo hay que “rellenar”. Implementa el método `main` para que se muestre un mensaje por la consola (plantilla `sysout` → `System.out.println`).
- Ejecuta la aplicación.
- Observa la estructura del proyecto en el directorio correspondiente al espacio de trabajo. Encontrarás una carpeta con el nombre del proyecto que contiene a su vez otras dos carpetas: 1) `src` con los ficheros que contienen el código fuente (ficheros `.java`) y 2) `bin` con los ficheros que contienen el código compilado (ficheros `.class`).
- Obtén el **primer argumento del programa** y muéstralo por la consola. Configura la ejecución del programa para establecer este argumento.

## Ejercicios de programación

### 1. Implementación y pruebas de la clase `Punto`.

- Crea el paquete `geometria` y añade la clase `Punto`. Esta clase representa un punto en el plano XY de los números enteros.
- Las **propiedades** que caracterizan a un punto son las coordenadas `x` e `y`. Estas propiedades se pueden consultar y modificar.
- Los objetos punto pueden ser **construidos** de tres modos:
  - *Constructor con dos argumentos*: inicializa las coordenadas del objeto punto a partir de los parámetros `x` e `y`.
  - *Constructor sin argumentos*: inicializa el punto en el origen de coordenadas (0, 0). Por tanto, este constructor no tiene ningún parámetro.
  - *Constructor de copia*: establece las coordenadas del punto a partir de las coordenadas de otro punto. Así pues, el parámetro de este constructor es otro objeto punto. Nótese que un objeto tiene acceso a las declaraciones privadas de otro objeto de su misma clase.
- La **funcionalidad** que ofrece esta clase es la siguiente:
  - *Desplazar* un incremento en `x` y un incremento en `y`. Esta operación modifica los valores de las coordenadas `x` e `y` sumando las cantidades que se especifican como parámetros.
- Define la clase `PruebaPunto` que incluya un método `main` en el paquete `pruebas`. El código del programa debe estar documentado con comentarios que expliquen los pasos que se van realizando.
  - Declara la variable local `punto1` y asigna un objeto `Punto` en las coordenadas (1, 3). Muestra el valor de las propiedades `x` e `y` por la consola.
  - Declara la variable `punto2` y asigna un objeto `Punto` construido a partir del constructor sin parámetros. Muestra por la consola el valor de sus propiedades.
  - Declara la variable `punto3` y asigna un objeto `Punto` construido a partir del `punto1` (constructor de copia). Muestra sus propiedades por la consola.
  - Declara la variable `punto4` y asígnale la variable `punto2`.
  - Modifica las coordenadas de `punto2` y muestra el valor de las propiedades por la consola. Muestra también el valor de las propiedades del `punto4`. Observa que tienen los mismos valores porque son referencias al mismo objeto (*aliasing*).
- Utiliza el **depurador** de Eclipse para analizar los objetos a los que referencian las variables y su contenido.
- Una vez que acabes la sesión debes **guardar tu espacio de trabajo**. Si está situado en una carpeta del ordenador, puedes comprimirlo y guardarlo en un *pendrive* o en un servicio de almacenamiento en la nube.
- Para seguir trabajando en el proyecto en cualquier ordenador que disponga de Eclipse y Java solo necesitas descomprimir el espacio de trabajo y al abrir Eclipse seleccionar la carpeta descomprimida como espacio de trabajo. Dentro del entorno se abrirán automáticamente los proyectos contenidos en el espacio de trabajo.