

Taller de Lenguajes II

Práctica nº 2

Tema: Temas avanzados en Eclipse

1. Ejecute Eclipse y cree un workspace WSPueba. ¿Cuál es la idea detrás de uso del Workspace?
2. **Importación y exportación de Proyectos en Eclipse**
 - a. El archivo *Proyecto2.zip* fue generado en eclipse y exportado usando la opción: *Export > General > Archive File* de modo que se exporte como un archivo comprimido.
 - I. Haga doble click sobre el archivo .zip y verifique que contiene un descriptor de proyecto.
 - i. ¿Qué nombre tiene ese archivo?
 - ii. ¿Cómo debería realizar la importación si no existiera ese archivo descriptor?
 - II. Considerando que existe el descriptor de proyecto, importe el proyecto usando la opción *Import > General > Existing Projects into Workspace*. En la ventana de selección, impórtelo usando la opción “*Select archive File*”. Verifique que el proyecto se importó correctamente (si aparecen errores de librerías o ambiente de ejecución corrijalo) y ejecute el programa.
3. **Compilación**
 - a. Haga click con el botón derecho del mouse sobre el nombre del proyecto. Elija *Propiedades*, y en la ventana desplegada seleccione “*Resources*”, **responda:**
 - i. ¿Cuál es la ubicación de su proyecto?
 - ii. ¿Qué cambios debería realizar si quisiera que su proyecto fuera compilado para ser compatible con las características de Java 1.8?

NOTA: en esta cátedra usaremos las características básicas de programación en Java

 - iii. De acuerdo a la información de la opción “*Java Build Path*”, **responda:**
 1. ¿En qué directorio quedará el resultado de la compilación -código bytecode- de sus archivos?
 2. Use el explorador de Windows y vaya a dicho directorio, ¿Qué contenido tiene?
 - b. Usando el menú superior, Project →Build Automatically, desactive la opción “*Build Automatically*” que compila automáticamente los archivos.
 - c. Usando el menú superior, Project →Clean, haga click en “*Clean*” y luego vaya al directorio donde eclipse dejará los archivos compilados. Nota: en la ventana de diálogo abierta por el “*Clean*” deberá deshabilitar la opción de “*start a build immediately*”. **Responda:**
 - i. ¿Qué archivo compilado encuentra?
 - ii. ¿Para qué sirve la opción Clean?
 - iii. ¿Para qué sirve la opción Clean combinada con el “*Build Automatically*”?
 - d. Vuelva a chequear la opción “*Build Automatically*”.
4. **Uso de Librerías**
 - a. Importe el proyecto *usolib.zip* provisto por la cátedra.

- b. El proyecto Dibujo necesita la librería `stdlib.jar` para compilar y funcionar correctamente. Agregue la librería al proyecto:
 - i. Cree una carpeta `libs` dentro del proyecto
 - ii. Copie y pegue el archivo `.jar`
 - iii. ¿Compila? si no compila, verifique que el proyecto **conoce** la nueva librería (puede verificarlo en la opción `Build Path`).
 - iv. Ejecute la clase que contiene el método `main` y verifique que funciona correctamente.

5. **Opción Renombre**

- a. Sobre el proyecto y usando el menú contextual haga click en la opción `Rename`, y cambie el nombre del proyecto a `ProyectoBouncyBalls`
- b. Renombre la clase `Dibujo` por `BouncyBalls`
- c. Renombre las siguientes variables dentro de la clase `BouncyBalls` haciendo uso de las facilidades que brinda eclipse:
 - `rx --> posX`
 - `ry --> posY`
 - `vx --> veloX`
 - `vy --> veloY`

6. **Opción Búsqueda y Reemplazo**

- a. Usando la combinación de teclas `Ctrl + H` se habilitará el panel de búsqueda y reemplazo en varios archivos (`search and replace`).
- b. Seleccione el tab `"File Search"` y busque todas las ocurrencias de `posX` (verifique el *filtro* de archivos y el *alcance* de la búsqueda)
- c. Use esta misma opción para reemplazar todas las ocurrencias de `posX` por `posicionX` y todas las ocurrencias de `posY` por `posicionY`.
- d. Verifique que la clase continúe ejecutándose correctamente.

7. **Refactorización de código**

- a. Seleccione con el mouse todas las líneas de código que comprende el bucle `while(true)`.
- b. Con el botón derecho del mouse haga click en `"Refactor → Extract Method"`. ¿puede extraer esa porción de código y convertirla en un método? **Justifique**
- c. Realice la misma acción pero seleccionando sólo las líneas que aparecen a continuación. El nuevo método se llamará `dibujaCirculo`:

```
// clear the background
StdDraw.clear(StdDraw.GRAY);

// draw ball on the screen
StdDraw.setPenColor(StdDraw.BLACK);
StdDraw.filledCircle(rx, ry, radius);
```

```
// display and pause for 20 ms  
StdDraw.show(20);
```

8. Debugging

- a. Agregue un breakpoint en la línea:
`StdDraw.clear(StdDraw.GRAY);`
- b. Ejecute la clase en modo **debug**. Indique que teclas usará para:
 - i. avanzar hasta el próximo breakpoint
 - ii. avanzar una línea
- c. Avance hasta el próximo breakpoint e indique:
 - i. ¿En que Vista visualiza los valores de las variables?
 - ii. ¿En que Vista visualiza los breakpoints indicados?
 - iii. ¿tiene alguna opción para deshabilitar los breakpoints? (no eliminarlos)
- d. Si se posiciona sobre la variable `posicionX` debería visualizar su valor. También lo puede visualizar usando el menú contextual: opción *Inspect* ó *Ctrol+Shift+I*.
- e. **Deshabilite** el breakpoint y continúe la ejecución.
- f. Vuelva a **habilitar** el breakpoint.
- g. Sobre el breakpoint (margen izquierdo) haga click con el botón derecho del mouse.
 - i. Usando el menú contextual haga click en *"Breakpoint Properties"*
 - ii. Chequee la opción : *Conditional*
 - iii. Complete en el panel la condición: *que el valor de la posicionX sea menor a 0*
 - iv. Verifique que efectivamente la aplicación se detiene cuando se cumple la condición.

9. **Javadoc.** Los comentarios javadoc permiten documentar código y prepararlo para general su manual en formato HTML. Puede obtener información acerca de uso del javadoc en las siguientes URLs ó pregunte a su ayudante:

Información breve y general:

- https://www.tutorialspoint.com/java/java_documentation.htm

Sitio oficial:

- <https://www.oracle.com/technetwork/articles/javase/index-137868.html>
- <http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html#javadoctags>
- <https://docs.oracle.com/javase/8/docs/technotes/tools/unix/javadoc.html> (sección tags descriptions)

- a. Genere los comentarios javadoc para la clase `BouncyBalls` de acuerdo a la especificación (al menos de la clase, algún método con sus parámetros).
- b. Genere la documentación HTML usando las opciones de eclipse.