**Entregable #3.1 [ejercicios] [semana 26 Feb 2018]**

* **Objetivos específicos** que persigue la realización del trabajo.
  + - Practicar con el entorno de Scala sobre eclipse
    - Utilizar el depurador de eclipse en un programa escrito en Scala
* **Competencias**: CC14

**Apellidos y nombre**

|  |
| --- |
| Gutierrez Ospina Anderson |
| Altamirano Cargua Fabricio |

1. **Conceptos básicos de Scala**

Introduzca la clase Point y el programa Demo en un fichero llamado Demo.

class Point(val xc: Int, val yc: Int) {

var x: Int = xc

var y: Int = yc

def move(dx: Int, dy: Int) {

x = x + dx

y = y + dy

println ("Point x location : " + x);

println ("Point y location : " + y);

}

}

object Demo {

def main(args: Array[String]) {

val pt = new Point(10, 20);

// Move to a new location

pt.move(10, 10);

}

}

3.1.1 ¿Qué hace el programa al ejecutarse?

Primeramente en el main() se instancia el objeto **pt** de la clase Point la cual tiene los un constructor que toma dos parámetros X e Y como valores enteros. Posteriormente se llama al método move(dx, dy) que se encuentra en la clase Point que suma sus atributos con los argumentos pasados, finalmente muestra por pantalla el resultado de X e Y.

1. **Trabajo con el depurador en Scala**

Para trabajar con el depurador, debes seguir estos pasos:

* Para abrir por primera vez la perspectiva de depuración, deberás seleccionar la siguiente entrada partiendo de la entrada “Window” del menú superior:

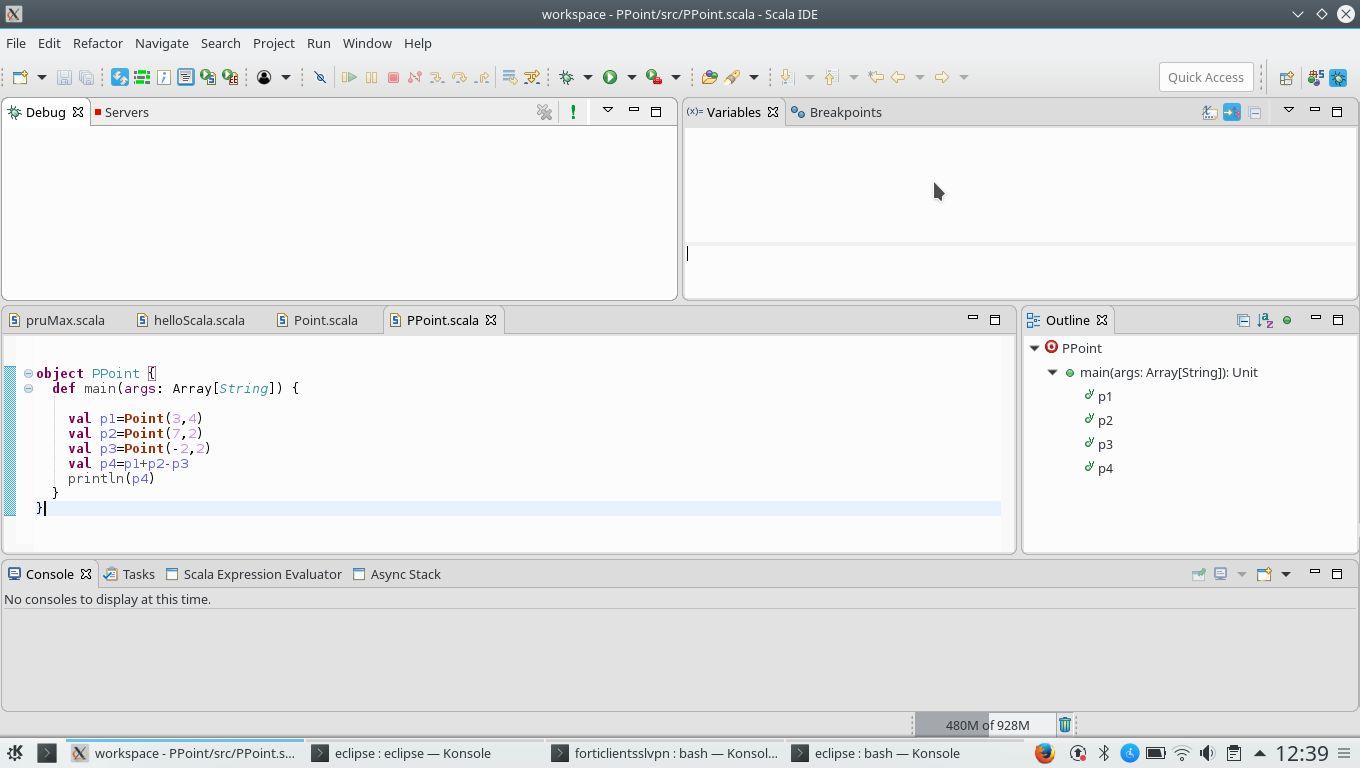
Window -> Perspective -> Open perspective -> Debug

* En ocasiones sucesivas, podrás abrir la perspectiva de depuración pulsando en el icono del escarabajo en la esquina superior derecha:



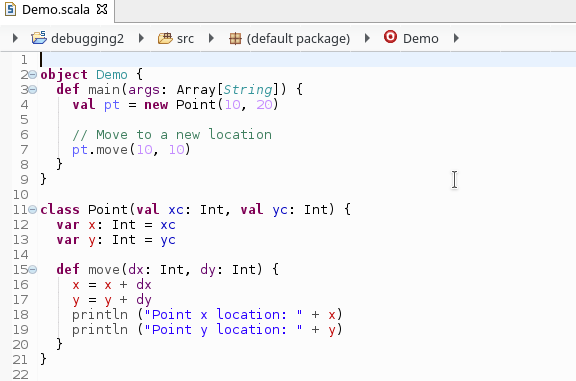
Cuando quieras volver a la perspectiva de código, debes seleccionar el icono que aparece a la izquierda del escarabajo.

La pantalla correspondiente a la perspectiva de depuración es la siguiente:



En dicha pantalla tenemos 5 áreas: la de seguimiento de ejecución (arriba-izquierda), la de variables y breakpoints (arriba-derecha), la de código (centro-izquierda), la de Outline (centro-derecha) y la de consola (abajo).

* Sitúate en la ventana de código:

****

Para visualizar la numeración del código, coloca el cursor en la barra vertical (azul) izquierda del código, pulsa en el botón derecho del ratón, y elige la opción de numerar. Después, pon un punto de parada en las líneas 4 y 7 del código del programa Demo. Para poner el punto de parada (breakpoint), debes colocarte de nuevo en la barra vertical de la izquierda del código, a la altura de cada una de estas líneas y hacer doble click sobre ellas. Si vuelves a hacer doble click en el mismo sitio, se quitarán los puntos de parada.

Elige el menú Run de la parte de arriba, y elige la opción “Debug as -> scala application”. El programa empezará a ejecutarse y se detendrá en el primer punto de parada.

En la zona de iconos de la segunda línea de pantalla, hay 3 flechas amarillas relacionadas con los puntos de parada:

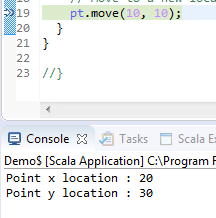


3.1.2 Selecciona la flecha amarilla del centro (step over o F6), que avanzará hasta el próximo punto de parada. Ve a la ventana de Variables, pulsa en el triángulo a la izquierda de la variable pt e indica qué valor tienen ahora las coordenadas **x = 10** e **y = 20** de Point.

3.1.3 Ahora selecciona la flecha amarilla de la izquierda (step into o F5). ¿Qué ves en la zona de código de la pantalla? ¿Por qué?

Veo que esta subrayada esta línea x = x + dx pues es el siguiente punto de ejecución del programa.

3.1.4 Selecciona la flecha amarilla de la derecha (step Return o F7), que saldrá al punto de llamada del método que se está ejecutando y finalizará la ejecución. Mira en la ventana de Consola, e indica qué valores tienen las coordenadas x e y de Point al finalizar la ejecución.



Lee la documentación relacionada con la clase Array de scala (<https://www.scala-lang.org/api/2.12.2/scala/Array.html>). Crea un nuevo scala object, introduce el siguiente programa y pon un punto de parada antes de rellenar cada entrada del array "saludos". Selecciona la flecha amarilla del centro (step over o F6) para ejecutar el programa.

**object** saluda {

**def** main(args: Array[*String*]) {

**val** saludos = **new** Array[*String*](3)

saludos.update(0, "Hello")

saludos.update(1, "Hola")

saludos.update(2, "Ciao")

**for** (i <- 0.to(2))

print (saludos.apply(i))

}

3.1.5 ¿Qué valor contienen las posiciones 1, 2 y 3 del array "saludos" antes de la sentencia:

saludos.update(2, "Ciao")? ¿En qué parte de la pantalla has podido verlo?

