**08 - Eventos y Oyentes de Eventos**

*Events, Event Listeners and more about Object Instances*

## **Preparación para enseñar este curso**

|  |  |
| --- | --- |
| :hourglass: **Instala** el [TKPJava](https://github.com/swandarina/TKP.Java) Material Didáctico | **1.** Burbuja Simple - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section08events/SimpleBubble.java) |
| :green_book: **Lee** la lección que está en esta página | **2.** Variación de Burbuja Simple - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section08events/SimpleBubbleVariation.java) |
| :computer:**Codifica** todas las recetas por ti mismo | **3.** Examen de Burbuja Simple - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section08events/SimpleBubbleQuiz.java) |
| :bulb: **Revisa** la [diapositivas](https://drive.google.com/a/wandarina.com/file/d/1MnJj1QvxYOELvR4U_L37ydP8-3-nApIE/view?usp=sharing) del Lenguaje de TKP | **4.** Conectando los puntos - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section08events/ConnectTheDots.java) |
| :fax: **Imprime** los [atajos de teclado](https://docs.google.com/document/d/1tzRac5nk5nkL0RcDQ0RiXQJhbRM5NxkNbmA81Dhb0dM/edit#bookmark=id.rgcmyj1v0fcr) | **5.** Inmersión Profunda 08 - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section08events/DeepDive08Events.java) |
| :fax: **Imprime** [las hojas de trabajo](https://drive.google.com/drive/folders/1P1eq_7YZl-2DKDitlFsVLftzUzQQfOpY?usp=sharing) | **6.** Laberinto de Tortuga - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section08events/TortoiseMaze.java) |
| :swimmer:Continúa aprendiendo |  |

## **Part 1 - Receta: Burbuja Simple**

Esta lección enseña sobre las instancias de un objeto, y también incluye cómo escribir en código en torno al alcance de una variable. Específicamente, la primera línea para ser traducida dice en español/inglés:

|  |
| --- |
| //Crea un ProgramaVentana titulado Mi Burbuja --#1 |

|  |
| --- |
| //Create a ProgramWindow titled My Bubble --#1 |

Para hacer esto, tendrá que guiar a los estudiantes a través de varios pasos (si tiene tiempo, mire el video que se muestra a continuación). Estos pasos son los siguientes (el código se muestra a continuación):

1) crea un nuevo objeto llamado ProgramWindow con el valor Bubbles

2) guarde ese objeto como una variable local llamada programWindow

3) declarar un campo (antes de la clase) de tipo ProgramWindow

|  |
| --- |
| private ProgramWindow programWindow; public SimpleBubble() {  //Create a ProgramWindow titled Bubbles --#1  programWindow = new ProgramWindow("Bubbles"); |

Usando lógica similar, guíe a los estudiantes a traducir los comentarios en español/inglés.

|  |
| --- |
| // Crea un círculo con el radio y el siguiente color de la rueda de colores --#2.1 |

|  |
| --- |
| // Create a circle with the radius and the next color from the color wheel --#2.1 |

Primero, cree un nuevo objeto de círculo, y luego incluya los parámetros requeridos. Es posible que desee ingresar un radio falso, como un paso intermedio. A continuación, guarde el nuevo círculo como una variable local llamada ‘círculo’. El código se vería así:

|  |
| --- |
| // Establece el radio para el círculo con un número aleatorio entre 10 y 50 - # 2.5  int radius = NumberUtils.getRandomInt (10, 50);  // Crea un círculo con el radio y el siguiente color de la rueda de colores - # 2.1  Circle circle = new Circle (radio, ColorWheel.getNextColor ()); |

|  |
| --- |
| // Set the radius for the circle to a random number between 10 and 50 --#2.5 int radius = NumberUtils.getRandomInt(10, 50); // Create a circle with the radius and the next color from the color wheel --#2.1 Circle circle = new Circle(radius, ColorWheel.getNextColor()); |

A continuación, guiará a los niños al siguiente nuevo concepto: codificar eventos a través de MouseClickListener. Escribimos un objeto *Listener* personalizado, derivado del objeto del evento MouseClick más genérico que es estándar en Java, para simplificar el registro de eventos. Observará que también limitamos al *listener* al botón izquierdo del mouse. La solución a esta sección de la lección que usa este objeto, se muestra a continuación:

|  |
| --- |
| @Override  public void onLeftMouseClick(int x, int y) { // createBubble (recipe below) --#8 createBubble(x, y); } |

El concepto de llamar a un *listener,* que usa la palabra clave "this", también es nuevo en esta lección:

|  |
| --- |
| //Haga que SimpleBubble escuche cuando se hace clic con el botón izquierdo del mouse en la ventana de su programa - # 2.2 programWindow.addMouseLeftClickListener(this); |

|  |
| --- |
| //Have SimpleBubble listen for when the left mouse button is clicked in your program window --#2.2 programWindow.addMouseLeftClickListener(this); |

Adicionalmente, hacemos un objeto de un Círculo (Circle) para lograr que la creación de una burbuja sea más directa. El objeto Círculo (Circle) es una abstracción de nivel superior a la parte más general biblioteca Pintable. Esta es la razón por la cual los estudiantes deben llamar a 'removePainable' en la ventana del programa para eliminar los círculos que fueron creados.

En el método createBubble además del uso del objeto Círculo (Circle), hay varios conceptos nuevos de programación intermedia. Incluyen el uso de nuestra biblioteca NumberUtils para crear un número aleatorio y también el uso de una instancia del objeto Círculo (Circle).

La idea de la creación de una instancia (como un campo o una variable local) frente a la inicialización de objetos globales debe reforzarse (círculo círculo = círculo nuevo ...) (Circle circle = new Circle...). Además, la idea de 'usar tu editor para explorar' está incorporada ya que creamos métodos personalizados en el objeto Circle, como por ejemplo, Circle.setCenter, y Circle.addTo para lograr que el proceso de traducción esté más cerca del español/inglés original.

The idea of instance creation (as a field or a local variable) vs. global object initialization should be reinforced (Circle circle = new Circle...). Also, the idea of 'use your editor to explore' is built-in as we created custom methods on the Circle object, such as Circle.setCenter... and Circle.addTo to make the translation process closer to the original English.

### **Recapitulación: Burbuja**

A continuación un video para la preparación del maestro, puede encontrarse [aquí](https://www.youtube.com/watch?v=3aihvg44gNs). Los conceptos para enfatizar son el alcance de las variables, es decir, el campo vs. la variable local; y “simula” hasta que lo logres, es decir, usar falsificaciones (constantes, tales como un número, en nuestro caso 9 para el radio, etc.) para que pueda ejecutar y probar su código en pasos intermedios.

También es posible que desee reforzar, SIN borrar las líneas de español/inglés hasta que cada línea se traduzca COMPLETAMENTE.

Haga una pregunta a los alumnos: '¿Por qué extrajimos dos métodos?'

También es posible que usted desee preguntar a los alumnos sobre el método principal -en las recetas anteriores lo colocamos siempre en una de las primeras líneas-, en esta ocasión, no lo hicimos. La razón es que queríamos que los niños usen un método constructor, tal es el caso de, SimpleBubble (), el cual después, es llamado atípicamente por el método principal al FINAL de esta receta. También reforzamos esta idea de un método constructor en la receta adicional: 'Connect the Dots'.

[](https://www.youtube.com/watch?feature=player_embedded&v=3aihvg44gNs)

## **Parte 2 – Variación: SimpleBubbleVariation (aka 'BubbleSplat')**

Para iniciar esta sección, puede optar por utilizar la pregunta Kata de TKP:

*"¿Cómo agregarías la capacidad de cambiar tu burbuja en un unicornio?"*

Ejemplo (con todos los pasos posteriores completados):

Variación 1: crea un 'splat' de burbujas haciendo lo siguiente:

--agregando otra anulación en el onLeftMouseClick la cual incluye un lazo *for* del tamaño 7. Dentro del método (y para el lazo) crea un nuevo método llamado createBubbles, copiando el método createBubbles original y eliminando la línea programWindow.removePaintable - una solución potencial se muestra a continuación:

|  |
| --- |
| public class SimpleBubble implements MouseLeftClickListener  {  private ProgramWindow programWindow;   public SimpleBubble()  {  programWindow = new ProgramWindow("Bubbles");  programWindow.addMouseLeftClickListener(this);  prepareColorPalette();  }   private void prepareColorPalette()  {  ColorWheel.addColor(Colors.Blues.AliceBlue);  ColorWheel.addColor(Colors.Blues.Blue);  ColorWheel.addColor(Colors.Blues.DarkBlue);  ColorWheel.addColor(Colors.Purples.Purple);  } // @Override // public void onLeftMouseClick(int x, int y) // { // createBubble(x, y); // }  @Override public void onLeftMouseClick(int x, int y) { int size = 7; for (int i = 1; i <= size; i++)  {  createBubbles(x + i \* 15, y + i \* 15);  } } private void createBubble(int x, int y) {  programWindow.removePaintable();  int radius = NumberUtils.getRandomInt(10, 50);  Circle circle = new Circle(radius, ColorWheel.getNextColor());  circle.setCenter(x, y);  circle.addTo(programWindow); } private void createBubbles(int x1, int y1)  {  //programWindow.removePaintable();  int radius = NumberUtils.getRandomInt(10, 50);  Circle circle = new Circle(radius, ColorWheel.getNextColor());  circle.setCenter(x1, y1);  circle.addTo(programWindow);  } public static void main(String[] args)  {  new SimpleBubble();  }  } |

Variación 2: refactoriza para crear un array de burbujas

## **Parte 3 – Examen: Simple Burbuja (SimpleBubbleQuiz)**

Complete el examen - IMPORTANTE: los estudiantes deben interactuar (hacer clic) con la pantalla, además de completar el código para aprobar esta prueba. Deben hacer clic en el botón izquierdo o derecho del mouse para completar el paso (después de haber escrito el código correspondiente) para aprobar el examen.

## **Parte 4 – Conecta los Puntos**

Esta receta reforzó la idea de un constructor en la primera línea de comentarios:

|  |
| --- |
| //Crea una nueva ventana de 'Connect The Dots'. --[#1](https://www.penflip.com/lynnlangit/tkp-lesson-plans/discussions/1).1 |

|  |
| --- |
| //Create a new 'Connect The Dots' window. --[#1](https://www.penflip.com/lynnlangit/tkp-lesson-plans/discussions/1).1 |

El cual, similar al proceso de traducción de la receta SimpleBubble hecho anteriormente, se traduce usando los mismos pasos, esto es:

1) Crea un nuevo objeto ConnectTheDots

2) Crea una variable local llamada puntos del tipo ConnectTheDots (objecto), la cual es traducida a lo siguiente:

|  |
| --- |
| ConnectTheDots dots = new ConnectTheDots(); |

Nuevamente usamos el método de encadenamiento, como lo hicimos en la receta Turtle Tree, pero ahora combinado con los oyentes del evento en la línea de español/inglés:

|  |
| --- |
| // Escucha por los clicks de la izquierda en la ventana para la tortuga |

|  |
| --- |
| // Listen for left clicks on the window for the tortoise |

El cual es traducido a este código:

|  |
| --- |
| Tortoise.getBackgroundWindow().addMouseLeftClickListener(this); |

Esta lección refuerza el concepto de eventos, recientemente introducido con el uso de los de los eventos oyentes (event listeners): onLeftMouseClick y onRightMouseClick.

|  |
| --- |
| @Override  public void onRightMouseClick(int x, int y) { clearTheScreen(); } @Override public void onLeftMouseClick(int x, int y) { addDot(x, y); } |

También el uso del objeto Tortuga para borrar la ventana del programa, en lugar de la ventana del programa en sí, puede ser confuso para los estudiantes. El uso del objeto Texto como una traducción de 'Escribir' {estas palabras} 'en la pantalla es un cambio sutil que es importante resaltar, por ejemplo, del verbo español/inglés al sustantivo Java.

|  |
| --- |
| private static void clearTheScreen()  { // ------------- Recipe for clearTheScreen // Clear the Program Window Tortoise.clear(); // Write "Right click to clear" on the screen at position 100, 100 new Text("Right click to clear").setTopLeft(100, 100).addTo(Tortoise.getBackgroundWindow()); } |

## **Parte 5 - Inmersión Profunda 08: TBD**

Conceptos incluirán eventos y más instancias de objetos

## **Parte 6 - Laberinto de tortuga**

Eventos de clic de botón: esta lección no está completa, por lo que aún no está incluida en la biblioteca.

## **Parte 7 – Hoja de Trabajo**

-Imprime la siguiente Hoja de Trabajo '[7\_SimpleBubbleWorksheet.docx](https://drive.google.com/open?id=1JRIJqHgw_tHkFEAoX2mikdSdtGYGmZoNcGT7QZ6gMb0)'