**05 - Recursión**

*Working with Recursion and Introducing Collections with HashMaps*

## **Preparación para enseñar este curso**

|  |  |
| --- | --- |
| :hourglass: **Instala** el [TKPJava](https://github.com/swandarina/TKP.Java) Material Didáctico | **1.** Árbol de Tortuga - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section05recursion/TurtleTree.java) |
| :green_book: **Lee** la lección que está en esta página | **2.** Variación del Árbol de Tortuga - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section05recursion/TurtleTreeVariation.java) |
| :computer:**Codifica** todas las recetas por ti mismo | **3.** Examen del Árbol - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section05recursion/TreeQuiz.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.** Telaraña - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section05recursion/SpiderWeb.java) |
| :fax: **Imprime** los [atajos de teclado](https://docs.google.com/document/d/1tzRac5nk5nkL0RcDQ0RiXQJhbRM5NxkNbmA81Dhb0dM/edit#bookmark=id.rgcmyj1v0fcr) y [las hojas de trabajo](https://drive.google.com/drive/folders/1P1eq_7YZl-2DKDitlFsVLftzUzQQfOpY?usp=sharing) | **5.** Cuadrado Recursivo - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section05recursion/RecursiveSquare.java) |
| :bulb: **Revisa** las recetas/ archivos de respuestas | **6.** Profundiza más 05 - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section05recursion/DeepDive05Recursion.java) |
| :swimmer:Continúa aprendiendo |  |

## **Part 1 - Recipe: Turtle Tree**

-Have them read the line out loud.

-Guide them to figuring out the code themselves, to whatever extent possible.

-Once they figure it out, MAKE SURE THEY RUN THE CODE BEFORE PROCEEDING. (Use <http://virtualproctor-tkp.appspot.com/> to show student windows on screen when they close)

#### **Teacher notes**

-Red box will kill the program.

-XX will reset the program, if only ONE is running. (useful if machine is being weird.)

### **LINE NOTES**

Concept reinforced here is 'extract method'. Recursion is introduced through recipe translation. You can complete the conditional portion of this recipe using multiple 'If' statements (for example by using the pattern shown below) - this solution is also shown in the video:

|  |
| --- |
| if (length == 10) { Tortoise.setPenColor(PenColors.Browns.Brown); //etc |

An alternative is to use a collection. This is due to the quantity of If statements needed to satisfy the logic. We recommend using the HashMap object for this. Alternatively, you can introduce the HashMap as refactoring (of the multiple 'if' statements during the variation for this recipe.

The Hashmap is the first introduction of a collection object. Use of a HashMap is shown in the lines below (solution to the AdjustColor method):

|  |
| --- |
| HashMap<Integer, Color> colors = new HashMap<Integer, Color>(); colors.put(10, Colors.Greens.Lime); colors.put(20, Colors.Greens.ForestGreen); colors.put(30, Colors.Greens.DarkGreen); colors.put(40, Colors.Greens.Olive); colors.put(50, Colors.Browns.Sienna); colors.put(60, Colors.Browns.SaddleBrown); |

To use the values in the HashMap, you'll need to call the 'get' method on it. This code is shown below:

|  |
| --- |
| Tortoise.setPenColor(colors.get(branchLength)); |

### **Video Recap: TurtleTree Recap**

Video for teacher preparation

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

## **Part 2 – Variation: Turtle Tree**

To kick off this section, you may choose to use the TKP Kata Question:

*"How would you make a second tree?"*

Example variation shows drawing more than one tree (extract method), also changing properties, background color, branch color, branch width. Explore changing the HashMap key (integer) values to let kids 'see' how the recursive calls affect the color(s) of the trees.

## **Part 3 – Quiz: TreeQuiz.java**

Cmd-shift-T: TQ (TreeQuiz)

Quiz questions 3 and 4 validate the HashMap concept using the pre-created object called 'angles' and the HashMap method 'put' for the (Key, Value) arguments.

|  |
| --- |
| //question 3 - don't forget to refactor to extract the method  angles.put(turn,90);   //question 4 - don't forget to refactor to extract the method  angles.put(5,36) |

## **Part 4 – Recipe: SpiderWeb.java**

This recipes uses Doubles rather than Integers for precision. Have the kids point out the recursion, by noting which methods call which other methods . This is a also a great recipe to use the Java debugger to have the kids 'see' the program execution flow.

Note: Thanks to Katrina Owen for authoring the SpiderWeb recipe.

## **Part 5 – Recipe: RecursiveSquare.java**

As with the SpiderWeb recipe, this is another great one to use the debugger to show the kids the program flow. Again, you may want to ask the kids 'which method(s) call themselves and how many times do they do so?'

Note: Thanks to Matt Ballin for authoring the RecursiveSquare recipe.

## **Part 6 – Deep Dive: DeepDive05Recursion**

### **SETUP**

Chairs at front, circle/oval/whatever – computer at desk to the side.

When going through Deep Dive, one student will sit @ computer, another will be standing and will tell them what to type.

After each test, standing student gives observation (/explains why they entered what they did), then they rotate.

If they ask “Does it work if…” then try it and see!

The concepts taught here are as follows:

1) Chained methods, that is Tortoise.firstMethod().secondMethod(params);

-we created the new method .getCursor specifically for this deepDive

2) Collections - HashTables - using get/put with typed key/value pairs

3) Collections - Arrays - zero-based retrieval

4) Collections - Lists - as above

### **Part 7 – Worksheet:**

Print the '[5\_TurtleTreeWorksheet.docx](https://drive.google.com/open?id=17l1468lxV-zakZp9nT4NrwgXv4_JM-iZIdUD0LBsWXg)'