**04 - Unidad de Dominio**

*Clases, Métodos, Lazos For, Variables, Declaraciones If*

*Objetos, Métodos y Lazos For*

## **Preparación para enseñar este curso**

|  |  |
| --- | --- |
| :hourglass: **Instala** el [TKPJava](https://github.com/swandarina/TKP.Java) Material Didáctico | **1.** Petágono Loco - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java//blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section03ifs/HiLow.java) |
| :green_book: **Lee** la lección que está en esta página | **2.** Examen de Pentágono Loco - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section03ifs/HiLowVariation.java) |
| :computer:**Codifica** todas las recetas por ti mismo | **3.** DigiFlor - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section04mastery/DigiFlower.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.** Variación de DigiFlor - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section03ifs/ConcentricLoop.java) |
| :fax: **Imprime** los [atajos de teclado](https://docs.google.com/document/d/1tzRac5nk5nkL0RcDQ0RiXQJhbRM5NxkNbmA81Dhb0dM/edit#bookmark=id.rgcmyj1v0fcr) | **5.** Anillo Enredado - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section03ifs/ChooseYourOwnAdventure.java) |
| :fax: **Imprime** [las hojas de trabajo](https://drive.google.com/drive/folders/1P1eq_7YZl-2DKDitlFsVLftzUzQQfOpY?usp=sharing) | **6.** Foto de Fondo - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section04mastery/BackgroundPhoto.java) |
| :swimmer:Continúa aprendiendo | **7.** Profundiza más 04 - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section04mastery/DeepDive04Mastery.java) |

## **Part 1 - Recipe: Pentagon Crazy**

This recipe is the first in this course. This course contains practices which review concepts that the students have already been introduced to in TKP Courses 1-3.

### **LINE NOTES**

The first review concept is the multi-step creation of variables. Remind students to use the 'fake it until you make it' technique. They will use this first on the length of a side by using a actual number, i.e. 42. Also remind the students that they should not delete the English as they have not finished translating it into Java.

The line 'the current length of a side' - the kids should be using the refactoring 'extract variable' to create the variable length. You may have to help them to find the correct variable from the for loop.

|  |
| --- |
| int length = 1; |

Do another 'fake' on the setPenColor (reinforces) with any color, then follow the info below.

The only tricky lines are the two shown below as they introduce a variation on the concept of passing parameters to methods using objects rather than simple variables or primitives (strings or ints):

|  |
| --- |
| Tortoise.setPenColor(ColorWheel.getNextColor()); Tortoise.setPenWidth(Tortoise.getPenWidth() + 1.0); |

Also this recipe is designed to reinforce using the 'extract method' tool in Eclipse - you should verify that students understand HOW and WHY they are extracting several methods in this recipe. The core concept is that well-written methods should be concise (do one thing) so that those methods are readable, have fewer bugs, are easier to maintain and update.

### **Video Recap: Pentagon Crazy**

Video for teacher preparation



### **Variation Grid: Pentagon Crazy**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Value** | **Refactor** |
| Sides | 5 | no |
| Length | i | no |
| Color | ColorWheel | no |
| Rotation | 1 | no |
| Width | 1-4 | no |
| Background | SolidWhite (default) | expose default |
| Number of LInes | 200 | no |

## **Part 2 – Quiz: PentagonCrazyQuiz.java**

Cmd-shift-T: PCQ (PentagonCrazyQuiz)

This quiz verifies that kids can create a method (by using the 'refactor>extract method' in Eclipse and also can use (or call) an existing method. Also the colors passed in to the quiz are used in creating the 'reward shape'. After kids PASS all 4 questions, they can alter the color values, which will FAIL the quiz questions, but will change the reward shape colors.

## **Part 3, 4 and 5 – Extra Recipes: DigiFlower.java and BackgroundPhoto.java**

See the TKP Instructional Design (Teaching Variation) section for more information.

In DigiFlower we introduce the idea of a group or list of variables by providing the kids with a list of Colors from the Java Color library, which are mapped to TKPPenColors for the ColorWheel. As with many TKP lessons, we first show the kids the code pattern and have them use this code. We do this because this is a common code pattern and also in TKP Course 5 kids will create and use a custom list to hold colors that is a different kind of list (here it's and abstraction over the Java ArrayList (using the TKPColorWheel) in Course 5 we will use the Java HashMap and in general in Course 5 we cover Java collections more extensively.

There is a also a DigiFlowerVariation. It uses more complex math in the loop to render a more intricate pattern.

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

*"How would you change the shaper of a flower petal?"*

## **Part 7 – Recipe: KnottedRing.java**

This is a review recipe - there is a reinforcement of a 'deliberate' error (or exception) that the students saw in an earlier recipe (Spiral). This 'error' is when the students try to 'get the next color' from the ColorWheel before they have added any colors to the wheel.

The subtle difference in this error is that we do continue to show a custom error message i.e. 'I call shinanigans, there are no colors on the ColorWheel' (rather than the default message which would show a array out of bounds exception for the ArrayList object (upon which the ColorWheel is derived). However, unlike in the Spiral recipe, where we 'elevated' the custom error message to a Message Box (to make sure the kids see it), in this case, we've let the error message show up in the default location (that Console window at the bottom of the IDE). This is intentional.

At this point the kids should have the habit of READING the error messages that show up in the Console so that they can understand and fix programming errors.

## **Part 7 – Deep Dive: DeepDive04**

### **SETUP**

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

PROCESS

When going through Deep Dive, one student will sit @ computer, another will be standing and will tell them what to type. Encourage them the run each test FIRST to get more information about what change they need to make to get the test to pass.

After each test passes (green bar in the result), the standing student gives observation (/explains why they entered what they did and what s/he learned), then they rotate.

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

In this DeepDive we are reinforcing these concepts:

1) Each variable has a type

2) Types explored are Integer, Double, String, Number and Object

3) 'Light' introduction to type inheritance via Number and Object

4) Variable initializers and type creation, i.e.

|  |
| --- |
| Number myNumber = 2; //produces an Integer-typed variable Number1 myNumber1 = 2.1 //produces a Double-typed variable |

5) Type conversion methods by example (i.e. to toString() method and others)

6) Operations on variables of same or mixed types (i.e. addition, division, concatenation)

## **Part 8 – Worksheet: 4\_PentagonCrazyWorksheet.docx**

Print [this](https://docs.google.com/document/d/1mAiC2UkoqS3RyoZq7Zee-FF-wd-lOoavFhE3e1aQGNE/edit?usp=sharing) worksheet