**03 - Lógica Condicional**

*Más sobre Métodos, e introducción al Declaraciones Condicionales Simples y Complejas*

## **Preparación para enseñar este curso**

|  |  |
| --- | --- |
| :hourglass: **Instala** el [TKPJava](https://github.com/swandarina/TKP.Java) Material Didáctico | **1.** AltoBajo - [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.** VariacionesAltoBajo - [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.** Examen AltoBajo - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section03ifs/HiLowQuiz.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.** LazosConcéntricos - [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.** Escoge tu propia aventura - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section03ifs/ChooseYourOwnAdventure.java) |
| :fax: **Imprime** y **descarga** [las hojas de trabajo](https://drive.google.com/drive/folders/1P1eq_7YZl-2DKDitlFsVLftzUzQQfOpY?usp=sharing) | **6.** Profundiza más 03 - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section03ifs/DeepDive03Ifs.java) |
| :bar_chart: Continúa aprendiendo |  |

## **Part 1 - Recipe: HiLow Game**

This course introduces conditional logic with increasingly complex Java if..., if...else If...., etc...control flow structures. We also introduce the idea of 'exit' points, or end of program code.

Additionally, in this course, we use a random number generator that we wrote for TKPJava.

The line referring to using a random number can be translated in two ways. We make use of a utility library (NumberUtils) that we wrote to simplify the creation of a random number (as shown below).

|  |
| --- |
| // Choose a random number between 1 and 100 --#4.1 (fake!) & --#13  int answer = NumberUtils.getRandomInt(1, 100); |

Alternatively you could use the "classic" Java pattern as shown below:

|  |
| --- |
| Random randomGenerator = new Random(); int randomInt = randomGenerator.nextInt(100); |

There are also two ways to translate the 'and exit --#10' line, this depends on how many if statements are used. When using a single If statement (with two included 'else if' conditions), then you can translate 'and exit' as shown below:

|  |
| --- |
| // and exit --#10  System.exit(0); |

Alternatively, if you choose to use two if statements, then you can use a Java 'break' statement rather than 'System.exit(0)' to exit the program on the winning condition.

### **Viedo Recap: HiLow**

Teacher preparation video HiLow recipe recap linked below.



Also, new teacher tip video 'How to use the Eclipse Debugger' (shown with the HiLow Variation) linked below.

<https://www.youtube.com/watch?v=M8F6ziuccmw>

## **Part 2 – Variation: HiLow**

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

*"How would you make sure that the guess is a positive number?"*

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

Elicit from kids – *what features are there?*

Example (with all later steps completed):

Feature - Value

Text of questions - i.e. "What is your guess?"

Answer - Integer (40)

Number of Times Guess - 8

Range of possible guesses - any integer

### **Variation Preparation**

1. Prepare and then update the text of questions
2. Easy to change right now
3. Change to update the game - can make it funny
4. Run, see that it’s the same. [THIS IS IMPORTANT.]
5. Put X in “prepare” for 'text of questions'.
6. Can change text of input questions and messages out (including adding error messages - i.e. 'you can't guess more than 100 silly...', etc)
7. Prepare and then update answer
8. Simple refactoring. Go to where it says 40, highlight, right click, refactor, extract local
9. variable.
10. Name it “answer”
11. Run – show that it does exactly the same thing.
12. [Don’t explain everything here – just show that it runs.]
13. X it off in the graph.
14. Prepare and update the number of attempts to guess
15. *Which line determines this? the for loop stop value*
16. You are just checking for understanding
17. Go back to grid, X it off
18. Range of possible guesses
19. Can verify that currently the guess can be ANY integer, or within a range of integers (using additional conditional logic, i.e. additional and/or nested 'if' blocks)
20. Can change number of guesses allows and wrap in an if-block to select easy, medium or hard game (with progressively fewer guesses possible)

### **Sample Variation Grid**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Value** | **Refactor** |
| Number of Guesses | 8 | no |
| Guess Value | number | no |
| Answer Value | number | no |
| Guess Value Range | integers | positive integers only |
| Guess Values | as above | reduce range of possible inputs |
| Guess Value Message | none | custom |
| Program Exit Point | simple | with message |

## **Part 3 – Quiz: HiLowQuiz.java**

Working with Tortoise methods, such as getX( ), getY( ), and detail -- includes 'turn to the left' and more to verify that kids are actually reading the English comments before they translate into Java

## **Part 4 – Recipe: ConcentricLoop**

Combines nested for loops with if statement and introduces scope of variables in loops, i.e. 'i' in first for loop and 'j' in nested for loop.

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

Open “ChooseYourOwnAdventure”

Each kid is on a single computer or this can be done in pairs if you prefer.

This recipe uses a new type of comparision in the if clauses - specifically it uses the syntax:

"string1".equalsIgnoreCase(string2).

This recipe teaches 'control of (program) flow' via complex conditional logic. Concepts included are these:

* Conditionals (if...elseif...else)
* Nested Conditionals (if....if...)
* Redirect to main method

Also using methods to encapsulate reused code is demonstrated by both having kids write code and then extract it as a method (and then reuse that method elsewhere in the code) and also by introducing the idea of convenience methods for repeated code. Convenience methods represent code libraries that are used in production code to produce more human-readable code, particularly in cases of long methods/classes.

The convenience methods used are as follows:

MessageBox.showMessage(message); --> tellMoreStory("blah...");

MessageBox.askForInput(question); --> askAQuestion("blah...");

Other teaching tips: if the students are having difficulty, be sure to have them 'match' the braces for their 'if' statements. If you feel comfortable, you may want to add a breakpoint (click in the left margin on the code editor window on a line to 'break' [or stop] during execution of code on and then run the code in debug mode to open the debugger. Here's a link to a short tutorial on how to use the debugger - <http://www.vogella.com/tutorials/EclipseDebugging/article.html>

## **Part 6 – Deep Dive: DeepDive03Ifs.java**

Covers conditional logic, various implementations of 'if...' clauses

### **SETUP**

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

Give kids post-its.

Each kid writes an observation and the last thing they just did. No talking – they just write. Then collect, and read them to the class.

PROCESS

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!

SHOW THEM THE PROCESS

## **Part 7 – Worksheet:**

Print '[3\_HiLowWorksheet.docx](https://docs.google.com/document/d/1XzwgBlYKEs_7vbr7YWKVtYfjIVvfPM1ctsMJo0JmL7Y/edit?usp=sharing)' and have the kids complete it.