**06 - MVC (Model-View-Controller)**

*More on String Concatenation, Message Box - Using MVC (as a variation)*

## **Preparación para enseñar este curso**

|  |  |
| --- | --- |
| :hourglass: **Instala** el [TKPJava](https://github.com/swandarina/TKP.Java) Material Didáctico | **1.** AdLibs - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section06modelviewcontroller/AdLibs.java) |
| :green_book: **Lee** la lección que está en esta página | **2.** Variación de AdLibs Excepcionales - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section06modelviewcontroller/ExceptionalAdLibsVariation.java) |
| :computer:**Codifica** todas las recetas por ti mismo | **3.** AdLibsRtf - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section06modelviewcontroller/AdLibsRtf.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.** Examen AdLibs - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section06modelviewcontroller/AdLibsQuiz.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.** Profundiza más en Modelo de Controlador de Vista 06 - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section06modelviewcontroller/DeepDive06ModelViewController.java) |
| :bulb: **Revisa** las recetas/ archivos de respuestas | **6.** Un Pez Dos Peces - [Respuesta](https://github.com/swandarina/TKP.WANDARINA.Fuente.Java/blob/master/src/main/java/org/teachingkidsprogramming/recipes/completed/section06modelviewcontroller/OneFishTwoFish.java) |
| :swimmer:Continúa aprendiendo |  |

## **Part 1 - Recipe: Ad Libs**

Pretty basic recipe, but fun! String concatenation requires careful reading of the English and using quotes and concatenation characters ('+') correctly.

We are also reinforcing the creation and use of variables with the use of 'current{word}' repeatedly in this recipe.

|  |
| --- |
| //Ask the user to enter an adverb, save it as currentAdverb --#2 String currentAdverb = MessageBox.askForTextInput("What is the adverb?"); |

We layer the string concatenation practice on top of the use of mutiple variables, for example:

|  |
| --- |
| //Add the words "I woke " + currentAdverb + ". " to the currentStory --#3 currentStory = currentStory + "I woke " + currentAdverb + ". "; |

### **Video Recap: AdLibs Recap**

Video recap for teacher preparation.

[](https://www.youtube.com/watch?feature=player_embedded&v=1HSWdT4-czc)

## **Part 2 - Variation: ExceptionalAdLibsVariation**

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

*"How would you make sure that the person telling your story did NOT enter in a number?"*

In this variation, we've added conditional logic to test for valid input values. Keeping this simple, we raise 'psuedo' exception messages in Message Boxes, rather than the more formal method of using try...catch w/Java exception handers.

Also we introduce the first RegEx expression in the '\_adverb.matches("[\d]\*")) argument.

Be sure to point out the the use of 'If' statements and you may want to draw out the program control of flow mapping.

We also refactored to include convenience methods, i.e. 'askForValueAgain( )....'

## **Part 3 – Variation: AdLibs with .rtf file**

A simple analogy to explain MVC is this:

*The Model is the food, the View is the plate and the Controller is the chef.*

This is lead as a ‘follow the leader’ style exercise. Open Word or something to scribble in. Make a grid, add columns for feature, value; ask the kids:

*which line(s) of code in the original AdLibs recipe create the model, which create the view and which create the controller?*

Answers:

1. The Model or Data model is the concatenated string ("currentStory")
2. The View is the MessageBox in the recipe
3. The Controller is the .showMessage ( ) method in the recipe

In the Variation, several new concepts are introduced:

* to update the Model, create the **Words class** which holds the data Model - you will have to add a new class for this data
* to update the Parser object, which acts as a controller - use the .parseRtfFile (templateFilename, data); method
* to update the View object, which is the viewer - use the .displayRtfFile (model); method
* These objects re-purpose the basic Model which the kids already created in the core recipe.

The View object uses uses a the template file 'view.rtf' and which contains string replacement parameters, designated by the syntax {parameterName} as shown below. This template file (view.rtf) is compiled as a resource in the TKP.jar file.

|  |
| --- |
| Today I woke {adverb}. Then I {edVerb} my {bodyPart}. |

Variation results:

1. The Model (originally the concatenated string) -> becomes the Words class
2. The View (MessageBox) -> becomes an RTF file via the Viewer class, using the display RTF file method (using the template file and passing the parameter values [strings] via the template placeholders
3. The Controller (.showMessage ( ) ) -> becomes a Parser, using the parse RTF method

NOTE: In order to do this lesson, students will have to have a word processing application that is associated to \*.rtf files, for example Microsoft Word, WordPad, TextEdit, etc...

### **Alternate Variations**

For classes that would be better served by simpler variations than the MVC version listed above, teachers may wish to use the guidance provided below.

New column: “Simple changes”

Add one more column: “Cool changes”

Can change anything, but needed to prepare before we mess around.

Could add one or more Integers, i.e. number + 'number of bodyParts' to explore concatenation with variables of Integer type being converted and concatenated to variables of String type such as below:

|  |
| --- |
| String myString = " body parts"; Integer myNumber = 5; result = myNumber + myString; System.out.println(result); |

## **Part 4 – Quiz: AdLibsQuiz**

concepts tested include the following:

Question 1 & 2) String concatenation (variations of such)

Question 3) MVC - implementing a custom parser -

hint: use Parser.parse(,)

Question 4) MVC - implementing a custom template

hint: set the template variable = "{templateName}"

## **Part 5 - Deep Dive: DeepDive06ModelViewController**

concepts reinforced include the following:

1) string concatenation - simple as well as multi-line concatenation

2) type conversion - specifically numbers (int, etc...) to string for concatenation

3) HashMap - using simple types, Strings and integers

## **Part 6 – Recipe: OneFishTwoFish**

New recipe still in development. Uses the Java string manipulation Scanner class.

Uses many methods of the Scanner object - see HINTS in the main recipe English comments.

It also uses an Iterator object in the foreach loop.

## **Part 7 – Worksheet:**

-Print the worksheet '[6\_AdLibsWorksheet.docx](https://drive.google.com/open?id=1CMhK7dnQs4R0QPfAmp2GPhLUgi3WYwkmw6zwmcVmQig)'