

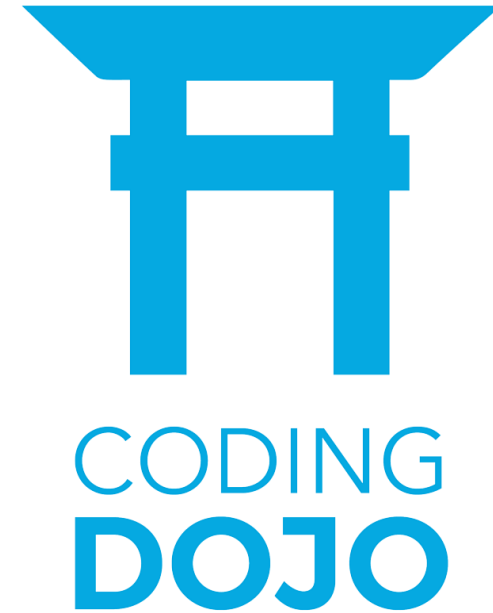
Coding Dojo

Kata String Calculator

Antes de empezar (¿Qué es coding dojo?)

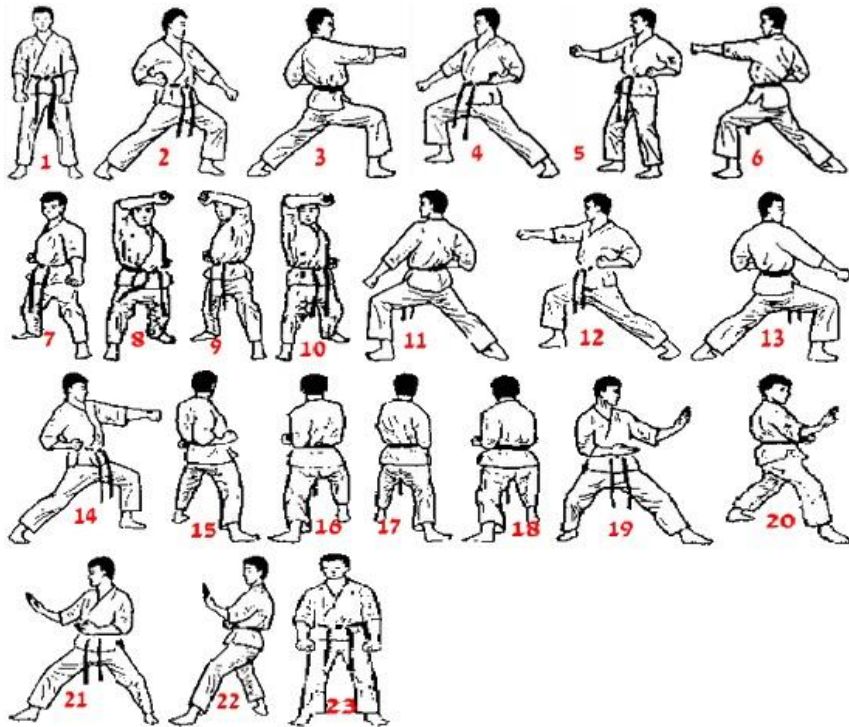


Dojo es el lugar donde se reúne la gente para practicar y entrenar artes marciales.

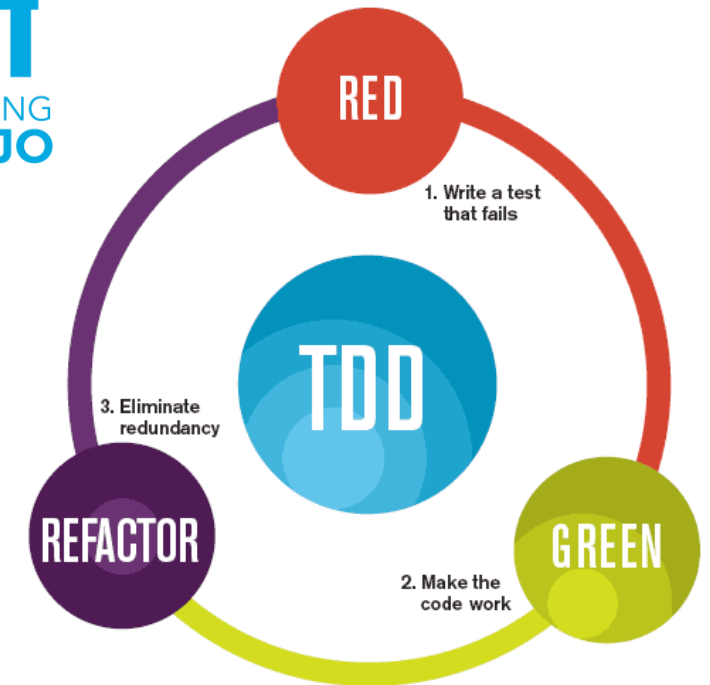


Lugar donde nos reunimos para practicar y entrenar buenas formas de programación.

Antes de empezar (¿Qué es coding dojo?)



Ejecutan Katas para aprender los movimientos y las técnicas.



Usamos TDD para el desarrollo
Ejecución por consenso

Antes de empezar (¿Qué es Refactor?)

“Cambios en el código para hacerlo más fácil de entender y más barato de modificar, sin alterar su comportamiento observable”

- Martin Fowler -

Antes de empezar (¿Qué es Refactor?)

“Cambios en el código para hacerlo
más fácil de entender y más barato
de modificar, sin alterar su
comportamiento observable”

- Martin Fowler -

Antes de empezar (¿Por qué debe ser más fácil de entender?)

El código fuente es el único legado que dejamos, incluso para uno mismo en un futuro.

Hay que escribir código como si se tratara de un libro.

Cuanto más fácil sea de leer, más fácil será de reescribir o corregir, menos tiempo costará y por tanto más barato será de mantener.

Antes de empezar (¿Cómo sabemos que está “mal escrito”?)

Basic smells

COMMENTS

MAGIC NUMBER

LONG METHOD

DUPLICATE METHOD

LARGE CLASS

LONG PARAMETER LIST

Refactor Actions

EXTRACT CLASS

INTRODUCE EXPLAINING VARIABLE

EXTRACT METHOD

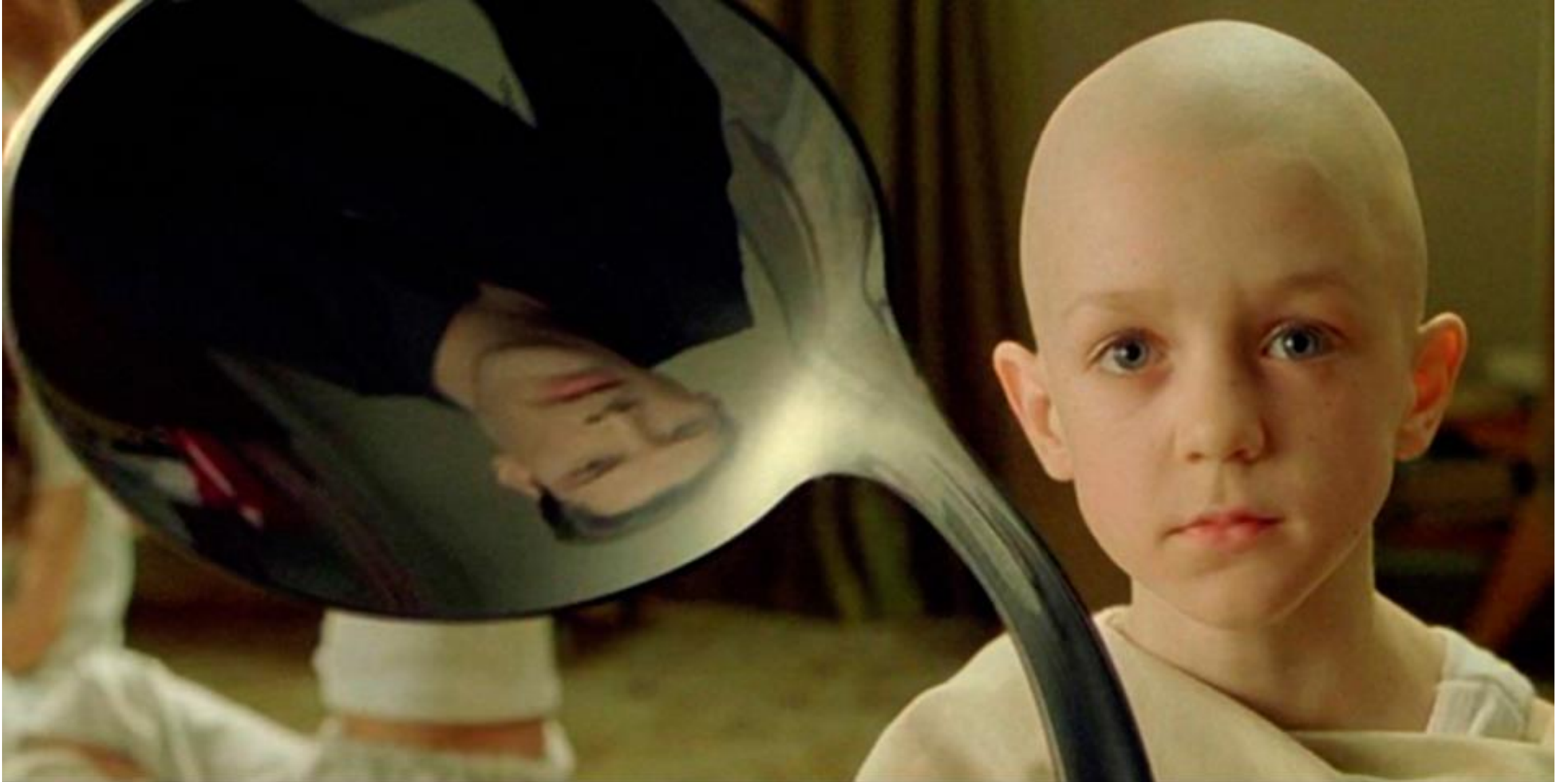
SELF ENCAPSULATE FIELD

INLINE METHODS

ENCAPSULATE COLLECTIONS

<https://refactoring.guru/refactoring/smells>

Antes de empezar (¿Y luego?)



Antes de empezar (¿Y luego?)

Design smells

SWITCHS

PRIMITIVE OBSESSION

MESSAGE CHAINS

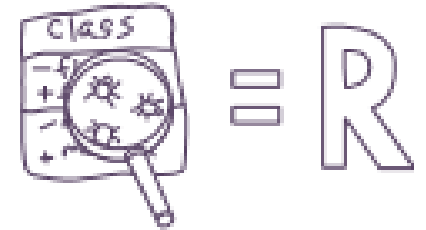
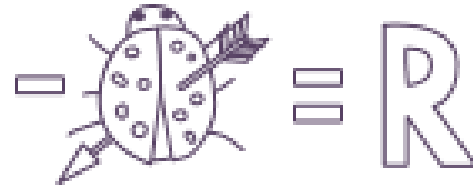
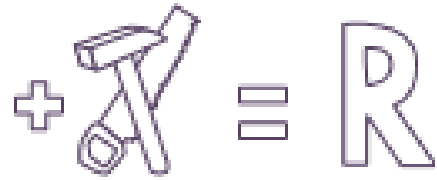
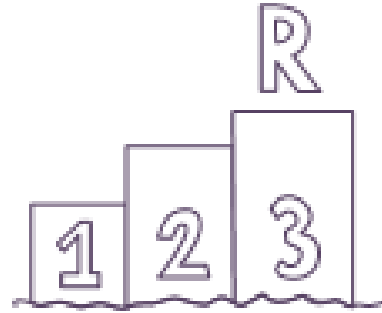
SPECULATIVE GENERALIZATION

DATA CLUMPS

FEATURE ENVY

<https://refactoring.guru/refactoring/smells>

Antes de empezar (¿Cuándo refactorizamos?)



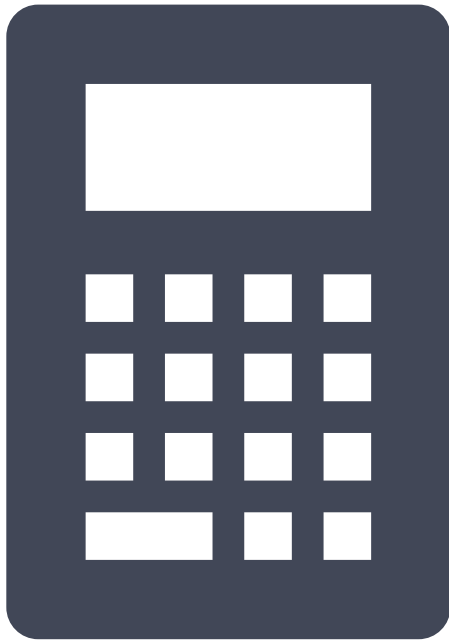
La regla del Boy Scout: “Siempre deja el lugar de acampada más limpio que como lo encontraste”.

<https://refactoring.guru/refactoring/smells>

Práctica

Kata String Calculator

String Calculator (Enunciado)



La famosa empresa de calculadoras “Casio” ha contratado los servicios de nuestra compañía. Quieren lanzar al mercado un nuevo concepto de calculadora basada en proceso de texto.

El problema es que “Casio” es un cliente un poco difícil, no tiene claros los requisitos y quiere resultados inmediatos, sin perder funcionalidad ya implementada.

Al frente del proyecto ha puesto a los mejores ingenieros que, con ayuda de TDD, intentarán dar respuesta a los requisitos cambiantes sin romper ninguna funcionalidad previa.

¿Como podemos ayudar a Casio?

String Calculator (Enunciado)

Consejos:

- Intentaremos no leer el enunciado completo.
 - Iremos leyendo poco a poco.
- Haremos solo una tarea a la vez.
 - Tenemos que aprender a trabajar de forma incremental.
- Para esta kata solo vamos a testear las entradas correctas.
 - Por agilidad en la session.
- Llegaremos hasta donde nos de tiempo.
 - No se trata de terminarlo, sino de aprender durante el proceso.
- Programaremos entre todos.
 - No existe una solución correcta y una incorrecta, se puede hacer de muchas formas.

No te olvides de pasar los tests a cada cambio que se efectue del código, para verificar que funciona todo correctamente y que no se ha roto nada más.

String Calculator (Enunciado)

Requisitos:

1. Crea una String Calculator con el método: `int add(String input)`

- El parámetro del método puede contener 0, 1 o 2 números y devolverá su suma (para un string vacío devolverá 0). Por ejemplo: "" o "1" o "1,2"
- Comienza por un test simple para un string vacío y luego para 1 y 2 números.
- Recuerda resolver el problema de la manera más simple posible para que te fuerce a escribir las pruebas que aún no se te habían ocurrido.
- Recuerda refactorizar después de conseguir pasar cada test.

2. Permite al metodo "add" manejar cualquier cantidad de números.

3. Permite al método "add" manejar saltos de línea entre números en lugar de usar comas.

- La siguiente entrada es correcta: "1\n2,3" (el resultado será 6)
- La siguiente entrada NO es correcta: "1,\n" (no hace falta que la pruebes, es simplemente para clarificar)

4. Soporta diferentes delimitadores

- Para cambiar un delimitador, el comienzo del string debe contener una línea separada que sea como esta: "//[delimitador]\n[números...]". Por ejemplo: "//;\n1;2" debe dar como resultado 3 donde el delimitador por defecto es ",".
- La primera línea es opcional. Todos los escenarios existentes hasta ahora, deben estar soportados.

String Calculator (Enunciado)

Requisitos (II):

5. Llamar al método “add” con números negativos deberá lanzar una excepción con el texto "negativos no soportados" y el número negativo que ha sido pasado. Si hay múltiples números negativos, muestra todos ellos en el mensaje de la excepción.
6. Los numeros mayores de 1000 deben ser ignorados.
 - Por ejemplo "2,1001" dará como resultado 2.
7. Los delimitadores pueden ser de cualquier longitud con el siguiente formato: "//[delimiter]\n".
 - Por ejemplo: "//[;;;]\n1;;;2;;;3" debe dar como resultado 6.
8. Permite múltiples delimitadores de la siguiente manera: "//[delim1][delim2]\n".
 - Por ejemplo: "//[#][%]\n1#2%3" debe dar como resultado 6.
9. Asegúrate de que puedes manejar delimitadores de cualquier longitud mayor de un caracter.

A refactorizar!

Kata String Calculator



People matter, results count.

This message contains information that may be privileged or confidential and is the property of the Capgemini Group.

Copyright © 2017 Capgemini. All rights reserved.

Rightshore® is a trademark belonging to Capgemini.

About Capgemini

With more than 190,000 people, Capgemini is present in over 40 countries and celebrates its 50th Anniversary year in 2017. A global leader in consulting, technology and outsourcing services, the Group reported 2016 global revenues of EUR 12.5 billion. Together with its clients, Capgemini creates and delivers business, technology and digital solutions that fit their needs, enabling them to achieve innovation and competitiveness. A deeply multicultural organization, Capgemini has developed its own way of working, [the Collaborative Business Experience™](#), and draws on [Rightshore®](#), its worldwide delivery model.

Learn more about us at

www.capgemini.com

This message is intended only for the person to whom it is addressed. If you are not the intended recipient, you are not authorized to read, print, retain, copy, disseminate, distribute, or use this message or any part thereof. If you receive this message in error, please notify the sender immediately and delete all copies of this message.