

Coding Dojo 5

Kata Potter

TDD - Algorithm



devonfw

Kata Potter (Enunciado)



Harry Potter es una colección de 7 novelas fantásticas escritas por la autora británica J.K. Rowling, en la que se describen las aventuras del joven aprendiz de mago Harry Potter y sus amigos.

La librería para la que trabajamos ha decidido, en plenas rebajas, poner en venta los libros con una oferta especial:

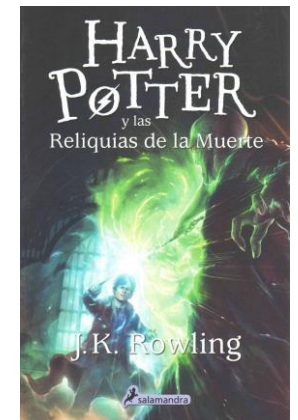
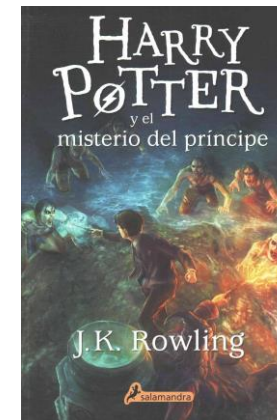
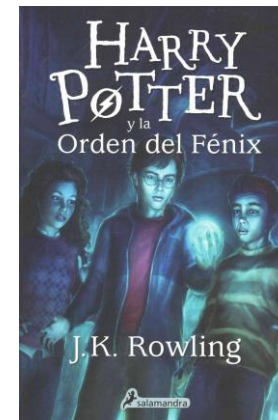
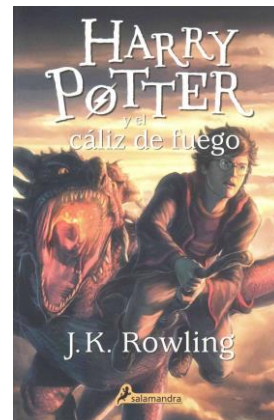
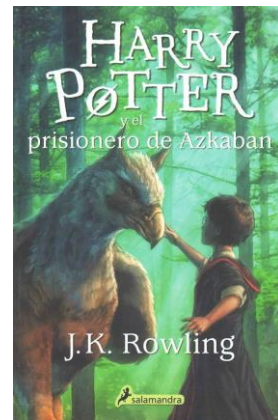
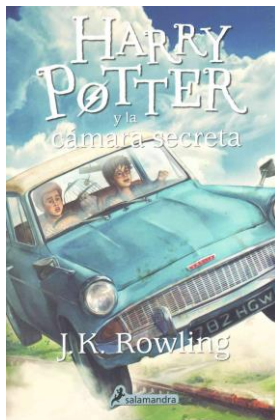
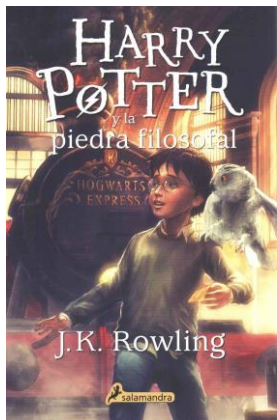
- Cada libro cuesta 8 €
- Si compras x libros diferentes tienes un de descuento sobre esos x libros
 - 2 libros \rightarrow 5% de descuento
 - 3 libros \rightarrow 10% de descuento
 - 4 libros \rightarrow 15% de descuento
 - 5 libros \rightarrow 20% de descuento
 - 6 libros \rightarrow 30% de descuento
 - Toda la colección 45% de descuento

Kata Potter (Enunciado)



Tras la locura desatada por semejante oferta, la librería se está llenando de adolescentes alocados comprando libros en grandes cantidades y las cajas no dan abasto porque pierden mucho tiempo realizando cálculos.

Por lo tanto han decidido contratar nuestros servicios, para crear un módulo de código fuente que pueda aportar la solución a este problema y devuelve el precio de compra para cada combinación que haga cualquier cliente.



Kata Potter (Consejos)



- Se debe intentar realizar el test de uno en uno.
 - Desde lo más sencillo a lo más complejo. Test recomendados:
 - Probar diferentes libros de uno en uno, el coste debe ser 8€
 - Probar diferentes combinaciones de 2 libros diferentes, el coste debe ser $2 * 8 * 0,95\%$
 - Probar el resto de combinaciones de libros diferentes, el coste debe tener el descuento correspondiente
 - Probar agrupaciones de combinaciones
- Se debe intentar implementar el código mínimo necesario para cada test.
 - Aunque se conozca el problema completo, hay que evitar el “*Speculative generalization*”
- En cada implementación de un test, se debe refactorizar el código.
 - ¿me siento cómodo con este código?
 - ¿cambiaría la implementación a objetos?
 - ¿cambiaría la implementación con herencia?
 - ¿es la mejor estructura de datos que se podría utilizar?



devonfw



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.