

CC1002 Introducción a la Programación

Auxiliar 7

Prof. Benjamin Bustos
Auxs. Cristobal Sepúlveda, Franco Sepúlveda

Fecha: 11 de octubre de 2017

Resolver los problemas usando Python, utilizando estrictamente la Receta de Diseño entregada a lo largo del curso. Use nombres apropiados para funciones y variables, y realice testing cada vez que sea posible.

1. Carrera de Ciclistas

En la ciudad de Santiago se realiza una cicletada con increíbles premios para los tres primeros lugares, por lo que muchas personas compiten dando su máximo esfuerzo. Suponga que usted es el juez de la cicletada y recibe un registro con todos los tiempos de llegada por persona, deseando saber quienes son los ganadores. Dado que su tiempo es muypreciado, usted quiere encontrarlos en una única revisión de los datos. Para esto, usaremos el concepto de *mutación*.

1. Defina tres variables de estado, las cuales mutarán a lo largo del problema, para guardar los tres mejores tiempos de llegada.
2. Programe la función `tresMejores`, la cual recibirá un tiempo en segundos, y lo asignará dentro de los tres mejores tiempos actuales guardados, si corresponde.
3. Termine el resto del programa, el cual usará la función anterior indefinidamente hasta que el usuario ingrese cero como tiempo de llegada. En ese momento, el programa imprime en pantalla los tres mejores tiempos y termina.

2. Desorden de Libros

Viendo el desorden que tiene en su pieza, su mamá le pide ordenar una pila gigante de libros que tiene ahí. Para hacer esta tarea más fácil, decide utilizar una interfaz programada por usted mismo para realizar una administración eficiente de los libros. Para esto, siga los siguientes pasos:

1. Cree primero la estructura `libro`, la cual contendrá un título y un autor, ambos del tipo `String`.
2. Piense ahora cómo puede utilizar el concepto de *mutación* sobre una estructura para poder representar la pila completa, utilizando a su vez una nueva estructura llamada `nodo`, la cual contendrá un `libro` y otro `nodo` en sus atributos. Cree esta variable.
3. Dado que los libros están en una pila, y usted es astuto, sabe que si saca un libro de abajo sin sacar los de arriba se le caerán todos encima. Bajo esta lógica, cree otra variable mutable para ocupar como pila auxiliar, donde usted podrá, en caso de necesitarlo, ir colocando los libros que se vayan sacando de la primera pila.
4. Implemente la función `agregarLibro`, que recibe como parámetro un `libro` y lo agrega a su pila mutable.
5. Implemente la función `buscarLibro`, que recibe como parámetro un `String` con el título del libro y devuelve `True` en caso de encontrarlo en la pila original, o `False` en caso contrario. Asegúrese, posteriormente a encontrar el libro o darse cuenta que no está, de devolver todos los libros a la pila original desde la pila auxiliar, de modo que la pila original vuelva al estado inicial.
6. Implemente la función `eliminarLibro`, que recibe como parámetro un `String` con el título del libro buscado, y que tiene como efecto eliminar dicho libro de la pila en caso de encontrarlo.