Veevart

Prueba de aptitudes #9

La siguiente es una prueba que tiene como objetivo conocer el nivel de abstracción, lógica, investigación, recursividad y velocidad de aprendizaje del aplicante.

Lenguaje de programación
De libre elección

Objetivo
Su objetivo es realizar el algoritmo de un elevador de personas en un edificio de **29 pisos**. Este debe ser eficiente en cuanto a reducción de tiempo innecesario de desplazamiento respetando su dirección de desplazamiento actual (subiendo o bajando).

Diseñe un simple método que imprima en consola las iteraciones del elevador a medida que este se encuentra en funcionamiento, este debe recibir como parámetros: un **arreglo de pisos** a los cuales el elevador será llamado en un orden definido, un **piso inicial de ejecución** y un mapa de **pisos ingresados**.

El método debe imprimir el **piso actual del elevador**, la **dirección en la que se desplaza**, el **piso en el que se detiene** y el **piso ingresado** cada vez que alguno de estos cambie

Aclaración

- El mapa de **pisos ingresados** hace referencia al piso en el que se ingresa el nuevo piso, es decir, la llave es el piso en el que se ingresa y el valor es el nuevo piso ingresado. Ejemplo: { 8 : 10 }, 8 es el piso en el que se ingresa (llave) y 10 es el nuevo piso (valor).

Ejemplo de impresión en consola

Arreglo de pisos: [5, 29, 13, 10]
Piso inicial de ejecución: 4
Pisos ingresados: {5:2, 29: 10, 13: 1, 10:1}
Sentido: Subiendo

1. Elevador en piso 4
2. Elevador subiendo
3. Elevador en piso 5
4. Elevador se detiene → [29, 13, 10]

5. Piso ingresado 2 → [29, 13, 10, 2]
6. Elevador subiendo
7. Elevador en piso 6, … 7, … 8, … 9
8. Elevador en piso 10
9. Elevador se detiene → [29, 13, 2]
10. Piso ingresado 1 → [29, 13, 1]
11. Elevador subiendo
12. Elevador en piso 11, … 12
13. Elevador en piso 13
14. Elevador se detiene → [29, 2, 1]
15. Elevador subiendo
16. Elevador en piso 14, … 28
17. Elevador en piso 29
18. Elevador se detiene → [2, 1]
19. Piso ingresado 10 → [2, 1, 10]
20. Elevador descendiendo
21. Elevador en piso 28, … 11
22. Elevador en piso 10
23. Elevador se detiene → [2, 1]
24. Elevador descendiendo
25. Elevador en piso 9, … 3
26. Elevador en piso 2
27. Elevador se detiene → [1]
28. Elevador descendiendo
29. Elevador en piso 1
30. Elevador se detiene

Recuerde

1. Analizar el funcionamiento esperado en el mundo real del elevador para diseñar el algoritmo de una forma optima
2. Documentar el código realizado
3. Enviar código realizado (Link a repositorio en github) a josed.angarita@veevart.com
4. Indicar en el readme como ejecutar la aplicación

Bonus

1. Documentar el código realizado.
2. Cree una nueva versión de la aplicación donde la entrada inicial solo incluya el arreglo de pisos y el piso inicial. El usuario debe poder solicitar el ascensor en cualquier momento de ejecución de la aplicación (tenga presente la dirección actual del ascensor para manejar la cola de solicitudes).

3. Cree una versión de la aplicación donde hayan 2 elevadores y manejan la solicitud de pisos de la forma más eficiente (menos recorrido posible).