TALLER 3: ARREGLOS

Maria C. Torres Madroñero
Profesora asociada
Departamento de Ciencias de la Computación y la Información
CONTRIBUCIONES: ALEJANDRO SALAZAR MEJÍA (BECARIO 2023) – CARLOS SEBASTIAN ZAMORA ROSERO (BECARIO 2024) – MARIA ALEJANDRA MUNOZ GONZALEZ (BECARIA 2024)

Este taller es una herramienta de estudio y no constituye una actividad evaluativa del curso.

INSTRUCCIONES: Se recomienda desarrollar cada problema mediante la presentación de un pseudocódigo, y en el caso de diseño de clase mediante diagramas de clase. Sin embargo, si el estudiante puede solucionar cada problema empleando un lenguaje de programación de alto nivel.

ARREGLOS

1- Dado un arreglo A[ ] de enteros. Presente un algoritmo que invierta los elementos del arreglo. Por ejemplo, dado el arreglo A = [5 10 27 30], la salida es [30 27 10 5].

2- Dado un arreglo A[ ] de enteros. Presente un algoritmo que encuentre un elemento del arreglo que representa un pico, es decir, que sus vecinos son menores. Para los elementos en los extremos, solo se considera un vecino. Por ejemplo. Para el arreglo A = [3 10 25 11 9], la salida es 25 dado que sus vecinos son el 10 y 11 que son menores a 25.

3- Dado un arreglo A[ ] de enteros. Presente un algoritmo de ordenamiento, donde los elementos queden organizados de mayor a menor. Por ejemplo, para el arreglo A = [8 5 7 0 3 9], la salida es [9 8 7 5 3 0].

4- Dado un arreglo A[ ] de enteros y un número K menor a la longitud del arreglo A, encontrar los K elementos más pequeños del arreglo. Por ejemplo, A = [7 10 5 3 15 12] y K = 3, la salida [3 5 7].

5- Dado un arreglo A[ ] de enteros ordenado y un número x, determine el número de ocurrencias de x en el arreglo A. Por ejemplo, dado el arreglo A = [1 1 1 2 2 3 3 3 3 5 5], y x=3, la salida es 4.

6- Dado un arreglo A[ ] de enteros positivos y un entero sum, determine el subarreglo que sumando sus elementos resulta igual a sum. Por ejemplo, dado el arreglo A = [1 4 20 3 10 5] y sum = 33, la salida será [20 3 10].

7- Dado un arreglo A[ ], encuentre el subarreglo continuo cuya suma es la mas alta posible. Por ejemplo, para el arreglo A[-2 -3 4 -1 -2 1 5 -1], el subarreglo con la suma mas alta de sus elementos es A = [4 -1 -2 1 5] con una suma igual a 7.

8- Dado un arreglo A[ ] de enteros, encuentra todos los pares de elementos cuya suma es igual a un número dado S. Por ejemplo, para el arreglo A = [1,2,3,4,5] y S = 5, las parejas serían: [(1,4), (2,3)].

9- Dado un arreglo A[ ] de enteros, encuentra el segundo elemento más grande sin ordenar el arreglo. Por ejemplo, para el arreglo A = [10,20,4,45,99], el segundo elemento más grande es 45.

10- Longitud del Subarreglo Más Largo con Suma Igual a K. Dado un arreglo de enteros A[ ] y un número entero K, encuentra la longitud del subarreglo continuo más largo cuya suma es exactamente K. Si no existe tal subarreglo, devuelve 0.

Ejemplo de entrada:

A = [10,5,2,7,1,9]

K = 15

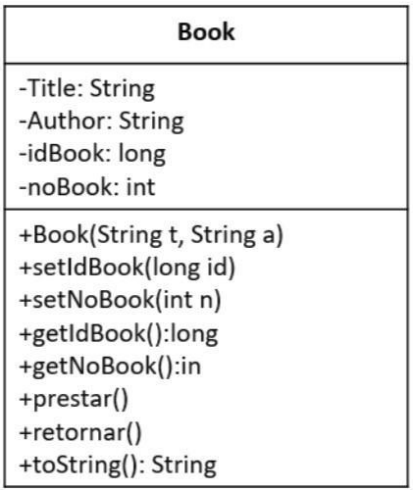
Ejemplo de salida:

4

En este caso, el subarreglo continuo más largo que suma 15 es [5,2,7,1], que tiene una longitud de 4.

ARREGLOS DE OBJETOS

1 – Supongamos que tenemos implementada la clase Book presentada en el siguiente diagrama de clases. La Tabla 1 describe cada atributo y método de la clase Book.



Atributos	Métodos

• Title: almacena el título del libro • Author: almacena el nombre completo del autor • idBook: número de identificación asignado por la biblioteca • noBook: número de libros disponibles para préstamos	• Book: constructor que recibe el título y nombre del autor • setIdBook: asigna el Id al libro • setNoBook: inicializa la variable noBook • getIdBook: retorna el idBook • getNoBook: retorna el número de libros disponibles • prestar: verifica que existan libros disponibles, si es así, actualiza el número de libros disponibles para préstamo (noBook) y retorna verdadero; si no hay libros disponibles, retorna falso. • retornar: devuelve un libro, incrementando la variable noBook • toString: devuelve una cadena con el título, autor, id y número de libros disponibles

1. Presente el diagrama de clases para BookCollection, una clase que permite administrar una colección de libros. La clase BookCollection debe usar un arreglo de libros. Incluya el constructor, un método que permita agregar un libro a la colección, un método para eliminar un libro dado el idBook, un método que busca un libro específico dado el idBook y retorna su posición en el arreglo, un método que preste un libro dado el idBook, y un método para retornar un libro.
2. Presente el algoritmo para agregar un nuevo libro de la clase BookCollection
3. Presente el algoritmo para el método de eliminar un libro dada el idBook en la clase BookCollection
4. Presente el algoritmo que permita buscar un libro en BookCollection dado un idBook, el método debe retornar la posición del libro en el arreglo
5. Presente el algoritmo que permita prestar un libro de la colección dad un idBook
6. Presente el algoritmo que permita devolver un libro a la colección

2- Vamos a asumir que se tiene la implementación de las clases presentadas en el diagrama de clases de la Figura 2. La clase CuentaBancaria incluye atributos para almacenar el nombre de un cliente, su número de cuenta y el saldo actual de la cuenta. Los métodos que se incluyen en la clase CuentaBancaria son: un constructor que recibe el nombre (n) y el número de la cuenta (c), los respectivos “gettings” que permiten acceder al nombre del cliente, la cuenta y el saldo, así como dos métodos que permiten consignar dinero a la cuenta o retirar dinero de la misma. Las clases CuentaAhorros, CuentaCorriente, y CreditoRotativo son diferentes tipos de cuentas bancarias. La cuenta de ahorros genera un saldo a favor de acuerdo con el interés mensual, el saldo se actualiza a través del método actualizarSaldo().

Por su parte, la cuenta corriente cobra una cuota de manejo cada mes, descontando este valor al saldo de la cuenta a través del método actualizarSaldo(). Por último, el crédito rotativo cobra una cuota mensual que es calculada a partir del saldo e interés mensual; el pago de la cuota se realiza a través del método consignar, los retiros se pueden realizar siempre que el saldo sea menor al cupo del crédito (cp en el constructor).

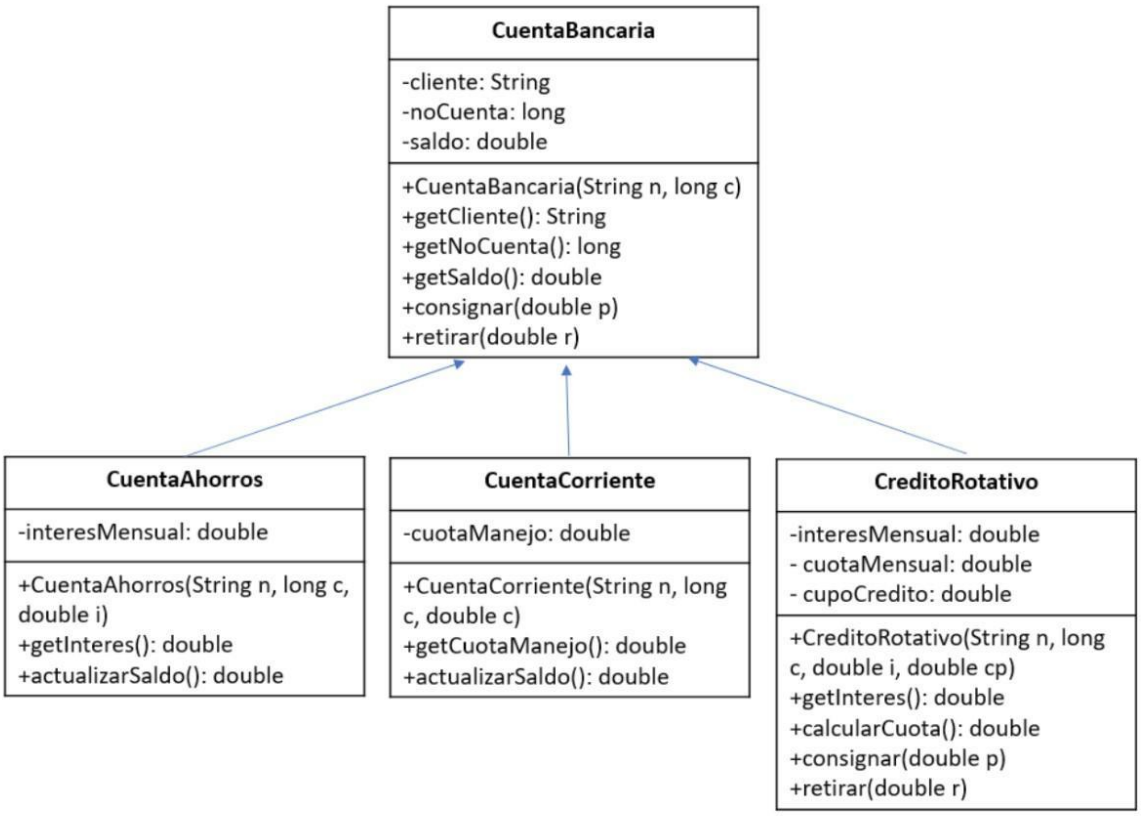


Figura 2: Diagrama de clases para la clase CuentaBancaria.

Considere una clase Clientes() que permite administrar una colección de 100 cuentas bancarias de un banco. Presente los algoritmos para realizar las siguientes operaciones en la colección:

1. Retornar el nombre de cliente con el saldo de la cuenta más alto en la colección
2. Calcular el valor promedio de los saldos de las cuentas en la colección
3. Dado el número de cuenta un valor a consignar, realizar la operación de consignar
4. Suponga que la colección no se encuentra organizada, presente el algoritmo para organizar la colección por el número de cuenta de menor a mayor

ANÁLISIS DE COMPLEJIDAD

1– Determine el tiempo de cómputo en el peor y mejor de los casos en notación asintótica para los problemas de la parte I Arreglos.

2– Determine el tiempo en el peor de los casos para sus algoritmos de los numerales 2 a 6 del problema 1 de la sección Arreglos de Objetos.