

## EIF207 Estructuras de Datos

### Práctica #3

Prof. M.Sc. Georges E. Alfaro S.

Implemente la solución de los siguientes ejercicios en **Java**, empleando el IDE **NetBeans**.

#### PARTE ÚNICA.

---

##### Ejercicio #1

Construya una función:

```
public void ordenar(Stack<T> s1)
```

que ordene ascendentemente los elementos una pila (*stack*) utilizando solamente una segunda pila temporal. No deberá modificar la implementación de la pila. Puede utilizar variables no estructuradas del tipo base de la pila.

Solamente puede utilizar los métodos básicos de la lista (sin acceder directamente a los atributos de la clase):

- isEmpty()
- top()
- push()
- pop()

Explique cuál es el tiempo de ejecución de la función.

Pruebe el método con el siguiente conjunto de prueba. Muestre en cada paso de ejecución el contenido de ambas pilas.

```
Integer[] v = {23, 56, 46, 41, 56, 30, 47, 23};
```

Recuerde que la clase Stack en la biblioteca de colecciones vista en clase muestra los elementos con el tope de la pila al inicio. El elemento de **menor** valor quedará en el **tope** de la pila.

##### Ejercicio #2

Utilizando solamente los métodos básicos push() y pop(), construya un método que extraiga el elemento en la n-ésima posición de la pila.

Los métodos deberán ser los mismos sin importar la implementación particular de la estructura (memoria contigua o enlazada). Es decir, no puede utilizar directamente los campos o atributos de la clase, solamente la interfaz (métodos públicos) de la clase.

Puede utilizar una segunda pila para resolver el ejercicio y cualquier método de la clase (por ejemplo, para preguntar cuántos elementos tiene la pila o averiguar si la pila está vacía).

**Suponga que la posición superior de la pila (top()) corresponde con la posición 0.** Verifique que la posición solicitada existe en la pila (lanzando una excepción en caso contrario).

### Ejercicio #3

Implemente un método para calcular la función de Padovan:

$$P(n) = \begin{cases} 1 & \text{si } n = 0 \\ 0 & \text{si } n = 1 \\ 0 & \text{si } n = 2 \\ P(n-2) + P(n-3) & \text{si } n > 2 \end{cases}$$

<https://oeis.org/A000931>

Calcule los primeros 129 valores de la secuencia ([P(0)···P(128)]). El ejercicio no se considerará correcto si no calcula todos los valores correctos.

Nota: en algunas referencias, los valores iniciales de la secuencia son diferentes.

### Ejercicio #4

Escriba dos funciones:

```
public static String dec2Bin(long n);  
public static long bin2Dec(String b);
```

Las funciones sirven para convertir un número entero en su representación binaria y viceversa. El número binario se representa por medio de una cadena (*string*) de dígitos binarios. Implemente ambas funciones **recursivamente**, sin utilizar iteraciones.