**PRUEBA TÉCNICA DESARROLLADORES**

Resuelva esta prueba con sus propios conocimientos, sin buscar respuesta en internet, no esperamos que todas las respuestas sean correctas, esto nos permite identificar mejor sus capacidades y plantear un plan de carrera

1. Explique en sus propias palabras y de un ejemplo de los siguientes conceptos / principios de software

|  |  |  |
| --- | --- | --- |
|  | Explicación | Ejemplo |
| Bajo acoplamiento / Alta cohesión | Los componentes dependen el uno del otro | Supongamos que tenemos una aplicación de ventas, debería haber un modulo para ventas y otro modulo para pagos para que no dependan entre si |
| Thread safety | Lo definiría como el encargado de evitar que se generen errores, cuando hay multiples hilos accediento a un mismo recurso | Supongamos que hacemos multiples inserciones en una base de datos, la falta de thread haría que hubieran fallos, ya que no se sincronizan las inserciones y no esperan a que se termine una para empezar la otra |
| Acoplamiento | Los componentes no dependen altamente entre si | Supongamos que tenemos una aplicación de ventas, tenemos un modulo de compras o pedidos, pero este modulo debe conectarse al modulo de inventario, para validar la disponibilidad de algún producto |
| Polimorfismo | Es la capacidad que tiene un objeto de comportarse de distintas maneras | Tenemos una clase padre que sea animal el cual tiene un método que se llama Habilidad()  Tenemos un animal que puede volar  Entonces podría sobreescribir el método Habilidad para que me devuelva que el animal vuela,  Por otro lado, tenemos un pez, sobreescribiriamos el método para que me devuelva que su habilidad es nadar |

1. Escriba un fragmento de código que solucione el siguiente problema, puede utilizar cualquier lenguaje de programación

Problema:

rotar a la derecha m veces los elementos de un arreglo, donde m >= 0 y el arreglo tiene tamaño n, donde n >= 0. Note que no se pierden elementos en el arreglo.

|  |  |  |
| --- | --- | --- |
| **Ejemplo** | **Valor de m** | **Resultado** |
| arr[1,2,3,4] | m = 1 | arr[4,1,2,3] |
| arr[1,2,3,4] | m = 2 | arr[3,4,1,2] |
| arr[1,2,3,4] | m = 3 | arr[2,3,4,1] |
| arr[1,2,3,4] | m = 4 | arr[1,2,3,4] |

Solución:

using System;

class Program

{

public const string Rotatedarrangement = "arr[";

static void Main(string[] args)

{

int[] arr = { 1, 2, 3, 4 };

int m = 2;

RotateRight(arr, m);

Console.WriteLine($"{Rotatedarrangement}{string.Join(",", arr)}]");

}

static void RotateRight(int[] arr, int m)

{

int n = arr.Length;

m = m % n;

if (n == 0 || m == 0)

{

return;

}

int[] temp = new int[n];

for (int i = 0; i < n; i++)

{

temp[(i + m) % n] = arr[i];

}

Array.Copy(temp, arr, n);

}

}

# Caso Bluesoft Bank

Bluesoft Bank es un banco tradicional que se encarga de guardar el dinero de sus ahorradores, ofrece dos tipos de cuenta; ahorros para personas naturales y corrientes para empresas. Adicionalmente para cada cuenta se pueden hacer consignaciones y retiros.  
  
Adicionalmente tiene que soportar algunos requerimientos para sus ahorradores:

* Consultar el saldo de la cuenta
* Consultar los movimientos más recientes
* Generar extractos mensuales

Reglas de negocio:

* Una cuenta no puede tener un saldo negativo.
* El saldo de la cuenta siempre debe ser consistente frente a dos operaciones concurrentes (consignación, retiro)

También se deben generar reportes en tiempo real como:

* Listado de clientes con el número de transacciones para un mes es particular, organizado descendentemente (primero el cliente con mayor # de transacciones en el mes)
* Clientes que retiran dinero fuera de la ciudad de origen de la cuenta con el valor total de los retiros realizados superior a $1.000.000.

**En base a lo anterior, por favor dar respuesta a los siguientes puntos:**

* Cree un diagrama de clases que modelo el problema, identifica los elementos principales y sus relaciones.
* Qué arquitectura y tecnologías usaría para resolver el caso Bluesoft Bank

**Rta:** crearía una arquitectura de micro servicios para dividir el aplicativo en partes mas pequeñas, para la lógica de negocio utilizaría .net core y para el frontend utilizaría angular o vue, después crearía un servivio api RESTful para conectar los microservicios con la aplicación del cliente

* Subir la implementación del caso a un repositorio público y compartir enlace