**Prueba técnica Lifeware**

Se detallarán tres pruebas de programación que deben ser realizadas para optar a un puesto de desarrollador dentro de la empresa Lifeware Spa. Las pruebas deben ser realizadas en el lenguaje de programación C# con la herramienta Visual Studio 2015 en ambiente Windows 10.

Se proveerá de tres distintas bibliotecas para la realización de las pruebas, una para cada una. Se pide desarrollar código sólido, estable y mantenible respecto a futuros cambios. Además, de la documentación respectiva.

**Prueba 1**

Imagine que usted está trabajando en un proyecto de desarrollo de software que está recién comenzando. El proyecto consta de un software que realiza muchísimas operaciones matemáticas complejas con ayuda de una pieza de código que representa ciertas calculadoras.

El software del proyecto consta principalmente de dos partes: calculadoras y algoritmos matemáticos. A usted se le pide desarrollar sólo una parte de este software: las calculadoras.

Desarrollará tres calculadores distintas que futuramente llegarán a ser usadas por los algoritmos matemáticos. Para probar que las calculadoras funcionan correctamente (y como la parte de algoritmos matemáticos no existe aún) deberá crear una aplicación de consola que demuestre el funcionamiento de las calculadoras ejecutando los métodos programados e imprimiendo sus resultados por consola. Los argumentos de los métodos pueden ser pedidos por teclado. También se le pide documentar todo lo realizado y finalmente explicar como trabajar con su código.

La aplicación de consola decidirá qué tipo de calculadora utilizar al momento de iniciarse. Cuando se inicie ejecutará un método de una biblioteca que le indicará el tipo de calculadora a utilizar. Seguirá usando éste tipo de calculadora hasta que el programa sea cerrado. Cuando el software se inicie nuevamente la biblioteca le volverá a indicar que tipo de calculadora utilizar.

El tipo de calculadora a utilizar se leerá desde el método GetType() de la clase Lifeware de la biblioteca Test01Lib.

A continuación se procede a detallar las calculadoras que usted debe programar.

La primera calculadora (llamada "Casio") es una calculadora normal que debe permitir:

* Recibir dos números flotantes, sumarlos y entregar su resultado en flotante.
* Recibir dos números flotantes, restarlos y entregar su resultado en flotante.
* Recibir un entero "n", y entregar el n-ésimo número de la serie de Fibonacci . [basta con que funcione para n entre 1 y 20]
* Recibir un entero "n", y entregar el "n-ésimo" decimal de Pi. [basta con que funcione para n entre 1 y 20]

La segunda calculadora (llamada "Kazio") es una calculadora un poco distinta que debe permitir:

* Recibir dos números flotantes, sumarlos, a ese resultado sumarle 42 y entregar el resultado final en flotante.
* Recibir dos números flotantes, restarlos, a ese resultado restarle 42 y entregar el resultado final en flotante.
* Recibir un entero "n", y entregar el n-ésimo número de la serie de Fibonacci dividido por 2 en entero. [basta con que funcione para n entre 1 y 20]
* Recibir un entero "n", y entregar el "n-ésimo + 2" decimal de Pi en entero. [basta con que funcione para n entre 1 y 20]

La tercera calculadora (llamada "FourModule") es una calculadora que trabaja con aritmética modular, debe permitir:

* Recibir dos números flotantes, obtener el módulo 4 de cada uno, estos resultados sumarlos y entregar el módulo 4 del resultado final en flotante.
* Recibir dos números flotantes, obtener el módulo 4 de cada uno, estos resultados restarlos y entregar el módulo 4 del resultado final en flotante.
* Recibir un entero "n", obtener el módulo 4 de "n", obtener el número de la serie Fibonacci cuya posición es el valor obtenido anteriormente, entregar el módulo 4 del resultado final en entero. [basta con que funcione para n entre 1 y 20]
* Recibir un entero "n", obtener el módulo 4 de "n", obtener el decimal de PI cuya posición es el valor obtenido anteriormente, entregar el módulo 4 del resultado final en entero. [basta con que funcione para n entre 1 y 20]

Ahora, imagine que una vez terminada las calculadoras usted es movido a otro proyecto. Un compañero suyo deberá continuar programando ahora los algoritmos matemáticos que aún no son abordados. Usted debe dejar documentado como su compañero o futuros compañeros deben trabajar con su pieza de código en los algoritmos matemáticos.

También explique que se debería modificar o añadir en ambas partes (calculadoras y algoritmos matemáticos) si se quieren añadir periódicamente cada cuatro meses 3 calculadoras extras al sistema.

**Prueba 2**

Ahora usted ha sido movido a un proyecto de captura de video. El proyecto consta de un software que permitirá hacer grabaciones de video con tres distintos dispositivos. Dos de estos dispositivos tienen sus bibliotecas externas con las que usted deberá trabajar y uno de ellos tiene una biblioteca interna que usted deberá desarrollar.

Usted debe programar la pieza de código que permita al software utilizar los tres dispositivos para realizar grabaciones. Esto quiere decir, que se pueda iniciar y detener grabaciones con los distintos dispositivos. Para probar que se pueden utilizar los dispositivos correctamente (y como la parte interfaz gráfica aún no está realizada) deberá crear una aplicación de consola que demuestre el funcionamiento de los dispositivos ejecutando los métodos programados e imprimiendo sus resultados por consola. También se le pide documentar todo lo realizado y finalmente explicar como trabajar con su código.

Para el caso de esta prueba las grabaciones de vídeo son sólo simuladas, realmente no usaremos ningún dispositivo y las bibliotecas externas y propias solo imprimen mensajes por consola indicando acciones.

Se procede a detallar los tres tipos distintos de dispositivos (simulados) a utilizar y cómo se deben utilizar para poder realizar grabaciones de video (simuladas):

El primer dispositivo llamado "Kinect" proviene de la librería Test02Lib y de la clase Kinect:

* Kinect(string version, string mode, bool persistent): Constructor, para poder utilizarlo en el sistema se debe construir de la siguiente forma: la versión debe ser el string "1.5", el mode debe ser el string "video" y persistent debe ser true.
* void Connect(): Permite conectar el Kinect para su posterior uso de grabación. Es necesario para poder configurar.
* void Configure(): Permite configurar el Kinect una vez conectado. Es necesario para poder calibrar.
* void Calibrate(): Permite calibrar el Kinect una vez configurado. Es necesario para poder grabar.
* void StartRecord(): Inicia la grabación de vídeo.
* void StopRecord(): Detiene la grabación de vídeo.
* void Disconnect(): Desconecta el Kinect y libera los recursos.

El segundo dispositivo llamado "Infrared Rgb Cam" proviene de la librería Test02Lib2 y de la clase infrared\_rgb\_cam:

* infrared\_rgb\_cam(): Constructor vacío necesario para instanciar un sensor.
* void connect\_infrared\_rgb\_cam(): Permite conectar el sensor para su posterior uso de grabación. Es necesario para poder grabar.
* void start\_record(): Permite iniciar la grabación con el sensor. Para poder iniciar la grabación en el sistema se debe establecer la propiedad "\_mode" en "rgb".
* void stop\_record(): Detiene la grabación con el sensor.
* void disconnect\_infrared\_rgb\_cam(): Desconecta el sensor y libera los recursos.

El tercer dispositivo llamada "Webcam" no proviene de ninguna biblioteca y ésta debe ser programada por usted. Lo importante que debe hacer está biblioteca es lo siguiente:

* Conectar el sensor: debe tener un método que imprime por consola "Webcam connected".
* Iniciar grabación: debe tener un método que imprime por consola "Webcam recording started".
* Detener grabación: debe tener un método que imprime por consola "Webcam recording stopped".
* Desconectar el sensor: debe tener un método que imprime por consola "Webcam disconnected".

Ahora, imagine que una vez terminada su pieza de código usted es movido a otro proyecto. Un compañero suyo deberá continuar programando el resto del software de grabación de video aún no es abordado. Usted debe dejar documentado como su compañero o futuros compañeros deben trabajar con su pieza de código.

También explique que se debería modificar o añadir en ambas partes (su pieza de código y resto del software de edición de vídeo) si se quieren añadir periódicamente cada cuatro meses 3 sensores extras al sistema.

**Prueba 3**

Usted fue nuevamente movido a otro proyecto. Este proyecto trata sobre el desarrollo de operaciones sobre listas de números para la NASA. Estas operaciones deben realizarse por medio de 3 distintos algoritmos dependiendo como se le indica al software. Durante la ejecución del software se debe poder cambiar de manera simple los algoritmos que se utilizan para realizar las operaciones si es necesario.

Usted debe programar la pieza de código que permita al software realizar distintas operaciones con distintos algoritmos. Para probar el correcto funcionamiento de lo desarrollado (y como no contamos con el posterior código de la NASA) deberá crear una aplicación de consola que demuestre el funcionamiento de lo pedido ejecutando los métodos programados e imprimiendo sus resultados por consola. También se le pide documentar todo lo realizado y finalmente explicar como trabajar con su código.

Usted debe desarrollar los siguientes métodos:

* Obtener primeros n: Entrega los n números mayores ordenadamente de una lista de números enteros no ordenada.
* Obtener últimos n: Entrega los n números menores ordenadamente de una lista de números enteros no ordenada.
* Obtener mediana: Entrega la mediana de una lista de números entero no ordenada.
* Obtener mediana primeros n: Entrega la mediana de los primeros n números mayores de una lista de enteros no ordenada.
* Obtener mediana últimos n: Entrega la mediana de los primeros n números menores de una lista de enteros no ordenada.

La NASA necesita que los métodos anteriormente descritos se puedan realizar con 3 algoritmos específicos de ordenamiento de listas. Éstos métodos son los siguientes: Bubble sort, Quick sort y Bogo sort. Recuerde que durante la ejecución del programa se debe poder usar un algoritmo de ordenamiento u otro dependiendo de la necesidad del momento.

Ahora, imagine que una vez terminada su pieza de código usted se la entrega personalmente al jefe de proyecto de la NASA. Ellos deberán continuar programando el resto del software utilizando sus algoritmos de medianas y ordenamiento. Usted debe dejar documentado (en español) como ellos deben trabajar con su pieza de código.

También explique que se debería modificar o añadir en ambas partes (su pieza de código y la de ellos) si se quieren añadir periódicamente cada cuatro meses 3 métodos de ordenamiento extra.