

Proyecto 2 evaluación

Ignacio Vacas Matencio:

Repositorio git hub:

<https://github.com/VACASh/Proyecto-paractico-2-evaluaci-n-entorno-de-desarrollos.git>

Política de estilos y normas básicas de ordenación de código , comentarios y definición de variables:

- **Nombres de clases.**
 - **Uso de sustantivos**
 - **Estilo caMel**

- **Nombres de campos y variables**
 - **Uso de sustantivos**
 - **Estilo caMel**
 - **Nombres reconocibles y lógicos**
 - **Declaración de variables juntas si tienen algo que ver**

- **Nombres de metodos**
 - **Estilo caMel**
 - **Ordenación dejando un espacio entre sentencias aunque solo tenga una sentencia.**

- **Nombre de namespace**
 - **Uso de estilo Pascal**
 - **Todos tienen el mismo namespace principal y luego un punto con el nombre de cada uno**

Realizacion de pruebas:

En las pruebas de caja negra realizo las pruebas de los limites por abajo y por arriba y por encima de ellos. Así como los casos especiales si los hubiera.

Hago las pruebas de equivalencia después de realizar las pruebas base con caracteres , números negativos, caracteres vacios y limites.

Casos de prueba a continuación:

Prueba de caja negra: Fecha de pruebas 5/03/2017

Botón números Fibonacci: `string textFibo(int numIntroducido)`

ENTRADA	SALIDA	COMENTARIO
""	La cadena de entrada no tiene formato correcto	Inserte un número entero el resto de parámetros son inválidos OK
A	La cadena de entrada no tiene formato correcto	Inserte un número entero el resto de parámetros son inválidos Ok
0	El número Fibonacci hasta tu numero es : 0	Devuelve el 0 que seria el primer número Fibonacci Ok
1 <i>Clase de equivalencia con resto de números positivos</i>	Los números Fibonacci hasta tu numero es :0, 1	Devuelve el 0 y el 1 que seria el primer número Fibonacci y el segundo Ok
2 Caso especial	Los números Fibonacci hasta tu numero es :0, 1,1,2	Es un caso especial porque se suma el 0 + 1 y después el 1+1 OK
-1 <i>Clase de equivalencia con resto de números negativos</i>	El número Fibonacci hasta tu numero es : 0	Devuelve erróneo porque solo lee positivos ERROR
Número con decimales	La cadena de entrada no tiene formato correcto	Inserte un número entero el resto de parámetros son inválidos OK
-2147483648 MISMO CASO QUE -1 <i>Limite por abajao</i>	El número Fibonacci hasta tu numero es : 0	Devuelve erróneo porque solo lee positivos ERROR
-2147483649 <i>Limite +1 por abajao</i>	La cadena de entrada no tiene formato correcto	Al no cumplir el tryparse por estar por encima de los números permitidos por visual estudio nos da el mensaje:Inserte un número entero el resto de parámetros son inválidos OK
2147483647	FALLO POR ALMACENAMIENTO DE STRING SIN MENSAJE	Al almacenarse sucesivamente en string se cuelga y no da mensaje alguno ERROR
2147483648	La cadena de entrada no tiene formato correcto	Al no cumplir el tryparse por estar por encima de los números permitidos por visual estudio nos da el mensaje:Inserte un número entero el resto de parámetros son inválidos OK

Cumplimos todos los casos bases para este método. Deberíamos poner una condición para números negativos y usar algo para los casos de almacenamiento de string.

Prueba de caja Blanca:

Ya tenemos hechas las pruebas base de negativos y caracteres especiales así como números con decimales.

También cumplimos cuando entra 0 veces y 1 vez.

La condición del if se cumple siempre porque el bucle lo hace hasta que es llega al número introducido y así no pasarlo. **Falta la prueba de n veces que la hacemos con 33.**

ENTRADA	SALIDA	COMENTARIO
33	Los números Fibonacci hasta tu numero es :0, 1,1,2,3,5,8,13,21	Entra n veces en este caso 8 que es los números que añade después de el 0 OK

Prueba de caja negra: : Fecha de pruebas 8/03/2017

```
void torresDeHanoi(int numDiscos, string discOrigen, string discDestino, string discAux, ref string textoMovimientos)
```

ENTRADA	SALIDA	COMENTARIO
""	La cadena de entrada no tiene formato correcto	Inserte un número entero el resto de parámetros son inválidos OK
A	La cadena de entrada no tiene formato correcto	Inserte un número entero el resto de parámetros son inválidos Ok
0	NULO	No hace na puesto que juegas con 0 discos Ok
1 Clase de equivalencia con resto de números positivos	Muevo el disco 1 de origen a destino	Te da el movimiento necesario para llegar al destino Ok
16 Caso especial	Muestra todos los movimientos	Despues de 1 minuto en un ordenador con 16gb consigue mostrar movimientos OK
-1 Clase de equivalencia con resto de números negativos	NULO	No devuelve nada porque no se puede jugar con -1 disco Deberíamos arreglarlo para que mostrara un mensaje ERROR
Número con decimales	La cadena de entrada no tiene formato correcto	Inserte un número entero el resto de parámetros son inválidos OK
-2147483648 MISMO CASO QUE -1 Limite por abajao	NULO	No devuelve nada porque no se puede jugar con -n disco Deberíamos arreglarlo para que mostrara un mensaje ERROR
-2147483649 Limite +1 por abajo	La cadena de entrada no tiene formato correcto	Al no cumplir el tryparse por estar por encima de los números permitidos por visual estudio nos da el mensaje:Inserte un número entero el resto de parámetros son inválidos OK
2147483647	Excepción no controlada del tipo 'System.StackOverflowException' en proyecto2evaluacion.exe	Se cuelga por trabajar con recursividad con un número muy grande ERROR
2147483648	La cadena de entrada no tiene formato correcto	Al no cumplir el tryparse por estar por encima de los números permitidos por visual estudio nos da el mensaje:Inserte un número entero el resto de parámetros son inválidos OK
17 Caso especial	Se queda parado	No consigue pasar de 16 movimientos se queda congelado. ERROR

Prueba de caja Blanca:

No hay casos que no se cumplan con las pruebas de caja negra

Con los casos especiales de caracteres negativos decimales y con la condición de 0 y más de 0 ya los cubrimos todos los casos de caja blanca.

