

Ejercicio teórico Problema 1 – Año Bisiesto

Identificar las unidades de prueba.

Unidad Principal:

- Método comprobarBisiesto () de la clase Fecha

```
public boolean comprobarBisiesto() throws AnioNegativoException, NoExisteBisiestoException {
    boolean esBisiesto = false;
    if (anio < 0)
        throw new AnioNegativoException("Error. El año introducido no puede ser negativo.");

    if (anio < 1582)
        throw new NoExisteBisiestoException("Error. El año introducido no es válido para el calendario gregoriano.");

    if (anio % 400 == 0)
        esBisiesto = true;
    else if (anio % 100 == 0)
        esBisiesto = false;
    else
        esBisiesto = (anio % 4 == 0);

    return esBisiesto;
}
```

Unidades auxiliares:

Validación del constructor Fecha (int año).

Excepciones Importantes:

- AnioNegativoException
- NoExisteBisiestoException

Identificar las variables que se deben tener en cuenta para probar los métodos de interés incluidos en las unidades de prueba.

Los valores principales del problema son los atributos de la clase Fecha, formada por:

- Anio (int)
- Mes (int)
- Dia (int)

Valores de prueba.

Parámetros	Clases De Equivalencia	Valores Seleccionados
------------	------------------------	-----------------------

Anio (int)	($-\infty, 0$), [0,1582),[1582, ∞)	-10,0,1,1582,1020,2020,"(", "ADF"
Mes (int)	($-\infty, 0$],[1,12],[13, ∞)	-2, 5, 20,"","a", 0, 1, 12, 13
Dia(int)	($-\infty, 0$], [1,28], [29,30), [30,31), [31, ∞)	-2, 10, 35, 0, 1, 28, 29, 30, 31,""

Los parámetros elegidos son anio de tipo int, el mes y el día de tipo int también. El parámetro anio cuenta con valores posibles infinitos, pero los valores importantes son desde el 1582 (primer año bisiesto del calendario gregoriano), y el máximo año actual bisiesto (que a la hora de escribir este trabajo es el 2024). Los demás posibles valores son valores que se pueden introducir en el sistema y no serán aceptados. En el caso de la variable mes y día son casos muy similares, el mes puede tener valores validos del 1 al 12, y los demás no serán aceptados, en el caso del día tenemos que tener en cuenta la variabilidad de los meses, teniendo 28,29,30,31 como valores posibles válidos. En el sistema debemos tener en cuenta que el usuario puede introducir valores ajenos al tipo de dato correcto, por eso hemos elegido algunos valores seleccionados que no cuadran con el tipo de dato del parámetro, como comillas, letras o contenido vacío. También es importante elegir los valores límites (los datos al filo del paréntesis) así como a los datos adyacentes a estos, se hace para controlar los posibles errores al introducir valores importantes y críticos que varían el sistema de manera notable.

Calcular el número máximo posible de casos de pruebas que se podrían generar a partir de los valores de pruebas (combinatoria).

Para calcular el número máximo de casos de prueba posibles, debemos multiplicar el número de valores seleccionados para cada parámetro, quedando de esta forma:

- Valores seleccionados para *día*: 10
- Valores seleccionados para mes: 9
- Valores seleccionados para *anio*: 8

Por tanto, el resultado del número máximo de casos de prueba sería el de multiplicar $10 \times 9 \times 8 = 720$ casos de prueba.

Defina un conjunto de casos de pruebas para cumplir con each use (cada valor una vez).

Para definir el conjunto de casos de prueba con each use, utilizaremos como base el parámetro con más valores seleccionados, en este caso es *día*, con 10 valores, y los combinaremos con los valores de los demás parámetros para probar todos.

Conjunto de casos de prueba each use

día	mes	anio
-2	-2	-10
10	5	0
35	20	1
0	”	1582
1	“a”	1020
29	0	2020
29	1	“(”
30	12	“ADF”
31	13	2020
”	12	1020

Defina conjuntos de pruebas para alcanzar cobertura pairwise usando el algoritmo explicado en clase.

El método Pairwise, también conocido como combinación por pares sirve para reducir el número de casos de prueba que se deben visitar de visitar, al menos una vez, todos los pares de los parámetros cualesquiera. Para saber los casos de prueba totales que vamos a tener cogemos los parámetros que más valores seleccionados tenemos que son los días (10 valores) y el mes (9 valores).

Casos de prueba = 10 valores x 9 valores = 90 casos de prueba

(ver tabla de pairwise al final del documento)

Cobertura de decisiones.

Tablas de verdad:

If día < 1 || día>diasPorMes

A	B	AvB
V	V	V
V	F	V
F	V	V
F	F	F

Se eligen las fila 2 y la 4 porque en el caso de la 4 es la única que hace el resultado falso, y en el caso de la fila 2 porque hace el resultado verdadero y además las variables tienen valores dispares.

If mes==2

A	A
V	V
F	F

Se eligen la fila 1 y 2 simplemente porque son las únicas filas donde el resultado es verdadero y falso respectivamente

If mes<1 || mes>12

A	B	AvB
V	V	V
V	F	V
F	V	V
F	F	F

Se eligen las fila 2 y la 4 porque en el caso de la 4 es la única que hace el resultado falso, y en el caso de la fila 2 porque hace el resultado verdadero y además las variables tienen valores dispares.

If anio<0

A	A
V	V
F	F

Se eligen la fila 1 y 2 simplemente porque son las únicas filas donde el resultado es verdadero y falso respectivamente

If anio<1582

A	A
V	V
F	F

Se eligen la fila 1 y 2 simplemente porque son las únicas filas donde el resultado es verdadero y falso respectivamente

If anio%400==0

A	A
V	V
F	F

Se eligen la fila 1 y 2 simplemente porque son las únicas filas donde el resultado es verdadero y falso respectivamente

If anio%100==0

A	A
V	V
F	F

Se eligen la fila 1 y 2 simplemente porque son las únicas filas donde el resultado es verdadero y falso respectivamente

Tabla de decisiones:

Dia	Mes	Anio
0		
“;”		
	2	
	7	
	0	
	“;”	
		-3
		2000
		1240
		1590
		400
		3
		100
		5

Cobertura MC/DC.

Tablas de verdad:

If día < 1 || día>diasPorMes

A	B	AvB	Condición dominante
V	V	V	A, B
V	F	V	A
F	V	V	B
F	F	F	A,B

En este caso elegimos las filas 2,3 y 4. La razón de la 4 es simplemente porque es la única que da falso. En el caso de la 2 y 3, ambas dan verdadero, pero en el caso de la fila 2 la variable que condiciona el resultado es la A, mientras que en la tercera es la B, por ello elegimos ambas filas para probar las decisiones verdaderas.

If mes==2

A	A	Condición dominante
V	V	A
F	F	A

En este caso al haber una única variable simplemente cogemos las filas donde los resultados den verdadero y falso respectivamente

If mes < 1 || mes>12

A	B	AvB	Condición dominante
V	V	V	A,B
V	F	V	A
F	V	V	B
F	F	F	A,B

En este caso elegimos las filas 2,3 y 4. La razón de la 4 es simplemente porque es la única que da falso. En el caso de la 2 y 3, ambas dan verdadero, pero en el caso de la fila 2 la variable que condiciona el resultado es la A, mientras que en la tercera es la B, por ello elegimos ambas filas para probar las decisiones verdaderas.

If anio<0

A	A	Condición dominante
V	V	A
F	F	A

En este caso al haber una única variable simplemente cogemos las filas donde los resultados den verdadero y falso respectivamente

If anio<1582

A	A	Condición dominante

V	V	A
F	F	A

En este caso al haber una única variable simplemente cogemos las filas donde los resultados den verdadero y falso respectivamente

If anio%400==0

A	A	Condición dominante
V	V	A
F	F	A

En este caso al haber una única variable simplemente cogemos las filas donde los resultados den verdadero y falso respectivamente

If anio%100==0

A	A	Condición dominante
V	V	A
F	F	A

En este caso al haber una única variable simplemente cogemos las filas donde los resultados den verdadero y falso respectivamente

Tabla de decisiones

Dia	Mes	Anio
0		
40		
“;”		
	2	
	7	
	0	
	14	
	“,”	
		-2
		4
		2000
		1590
		400
		97
		100
		37

Comentarios finales sobre cobertura.

Con todas estas pruebas y coberturas hemos conseguido pasar por todos los posibles valores que pueden ir tomando los datos a lo largo del programa y así no haya errores lógicos ni sintácticos que afecten al funcionamiento del programa.

	Anio	Mes	Dia	
1	-10	5	10	
2	-10	20	35	
3	-10	","	0	
4	-10	"a"	1	
5	-10	0	28	
6	-10	1	29	
7	-10	12	30	
8	-10	13	31	
9	0	5	35	
10	0	20	0	
11	0	","	1	
12	0	"a"	28	
13	0	0	29	
14	0	1	30	
15	0	12	31	
16	0	13	""	
17	0	-2	-2	
18	1	20	1	
19	1	","	28	
20	1	"a"	29	
21	1	0	30	
22	1	1	31	
23	1	12	""	

24	1	13	-2	
25	1	-2	10	
26	1	-2	35	
27	1	5	0	
28	1582	" , "	29	
29	1582	"a"	30	
30	1582	0	31	
31	1582	1	""	
32	1582	12	-2	
33	1582	13	10	
34	1582	-2	35	
35	1582	-2	0	
36	1582	5	1	
37	1582	20	28	
38	1020	"a"	31	
39	1020	0	""	
40	1020	1	-2	
41	1020	12	10	
42	1020	13	35	
43	1020	-2	0	
44	1020	-2	1	
45	1020	5	28	
46	1020	20	29	
47	1020	" , "	30	
48	2020	0	-2	
49	2020	1	10	
50	2020	12	35	
51	2020	13	0	
52	2020	-2	1	
53	2020	-2	28	
54	2020	5	29	
55	2020	20	30	
56	2020	" , "	31	

57	2020	"a"	""	
58	("	1	35	
59	("	12	0	
60	("	13	1	
61	("	-2	28	
62	("	-2	29	
63	("	5	30	
64	("	20	31	
65	("	,"	""	
66	("	"a"	-2	
67	("	0	10	
68	"ADF"	12	1	
69	"ADF"	13	28	
70	"ADF"	-2	29	
71	"ADF"	-2	30	
72	"ADF"	5	31	
73	"ADF"	20	""	
74	"ADF"	,"	-2	
75	"ADF"	"a"	10	
76	"ADF"	0	35	
77	"ADF"	1	0	
78	-10	13	29	
79	-10	-2	31	
80	-10	5	""	
81	-10	20	-2	
82	-10	,"	10	
83	-10	"a"	35	
84	-10	0	0	
85	-10	1	1	
86	-10	12	28	
87	0	-2	""	
88	0	5	-2	
89	0	20	10	

90	0	","	35	
91	0	"a"	0	
92	0	0	1	
93	0	1	28	
94	0	12	29	
95	0	13	30	