Práctica pruebas de software

Desarrolla una batería de pruebas e impleméntala en JUnit para la clase

*ComprobarFecha.java*

que desarrollaste en el módulo Programación en la Primera evaluación.

Las especificaciones para realizar el programa eran:

*Realiza un programa que pida una fecha (día, mes y año) y compruebe, utilizando únicamente lo visto hasta ahora (tipos de datos primitivos, operadores y estructuras de control) si componen una fecha correcta.*

*Para que una fecha sea correcta,*

***Los tres deben de ser enteros → UN VALOR ENUMERADO***

* *El año introducido tiene que ser distinto de 0. → Condición lógica (o es 0 o es !=0)*
* *El mes introducido tiene que estar en el rango 1..12 → Rango de valores*
* *El día introducido:*
  + *en los meses de 30 días, debe estar en el rango 1..30 → Rango de valores*
  + *en los meses de 31 días, debe estar en el rango 1..31 → Rango de valores*
  + *en febrero, debe estar en el rango 1..28, excepto si el año es bisiesto, en cuyo caso el rango es 1..29. → Rango de valores*

**Debes hacer lo siguiente:**

1. **Crear la tabla resultado de clase de equivalencia. Añadir, si procede, reglas para la técnica de los valores límite.**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLA RESULTADO DE CLASES DE EQUIVALENCIA Y VALORES LÍMITE | | | |
| Condición de entrada | Clases de equivalencia válidas | Clases de equivalencia no válidas | Valores límite |
| Día entero | (V1) Día es int | (NV1) día es una letra  (NV2) día es un decimal | Los tres números deben ser enteros → un valor emumerado |
| Mes entero | (V2) Mes es int | (NV3) mes es una letra  (NV4) mes es un decimal |
| Año entero | (V3) Año es int | (NV5) año es una letra  (NV6) año es un decimal |
| *“El año introducido tiene que ser distinto de 0”* | (V4) año !=0 | (NV7) año == 0 | *Condición lógica (o es 0 o es !=0)* |
| “*El mes introducido tiene que estar en el rango 1..12”* | (V5)12<=mes>=1 | (NV8) mes<1  (NV9) mes>12 | (VL1)mes=12  (VL2)mes=13  (VL3)mes=11  (VL4)mes=1  (VL5)mes=0  (VL6)mes=2 |
| “*en los meses de 30 días, debe estar en el rango 1..30*” | (V6)Mes=4,6, 9 u 11; cualquier año válido, 1<=dia<=30 | Con mes= 4,6, 9 u 11  (NV10)dia<1  (NV11)dia>30 | Con mes=4,6, 9 u 11:  (VL7)día=1  (VL8)mes=0  (VL9)mes=2  (VL10)mes=29  (VL11)mes=30  (VL12)mes=31 |
| *“en los meses de 31 días, debe estar en el rango 1..31 ”* | (V7)Mes=1,3,5,7,8,10 u 12; cualquier año válido, 1<=dia<=31 | Con mes= 1,3,5,7,8,10 u 12  (NV12)dia<1  (NV13)dia>31 | Con mes= 1,3,5,7,8,10 u 12  (VL13)día=1  (VL14)día=0  (VL15)día=2  (VL16)día=30  (VL17)día=31  (VL18)día=29 |
| “*en febrero, debe estar en el rango 1..28, excepto si el año es bisiesto, en cuyo caso el rango es 1..29.”* | (V8) Año bisiesto, mes= 2, 1<=día<=29  ---------------------------  (V9) Año no bisiesto, mes=2, 1<=día<=28 | (NV14) Año bisiesto, mes=2,día <1  (NV15) Año bisiesto,  mes=2,dia>29  ------------------------------  (NV16) Año no bisiesto, mes=2,día <1  (NV17) Año no bisiesto,  mes=2,dia>28 | Con mes=2  Año bisiesto:  (VL19)día=1  (VL20)día=0  (VL21)día=2  (VL22)día=30  (VL23)día=28  (VL24)día=29  ----------------------------------------  Año no bisiesto:  (VL25)día=1  (VL26)día=0  (VL27)día=2  (VL28)día=28  (VL29)día=27  (VL30)día=29 |

1. **A partir de la tabla anterior, haz una tabla con la derivación de casos de prueba.**

|  |  |  |
| --- | --- | --- |
| DERIVACIÓN DE CASOS DE PRUEBA | | |
| Valor(es) de entrada | Clases de equivalencia cubiertas | Resultado |
| Dia=30, mes=4, año=1 | V1,V2,V3,V4,V5,V6 | “Fecha válida” |
| Dia=31, mes=1, año=1 | V1,V2,V3,V4,V5,V7 | “Fecha válida” |
| Dia=29, mes=2, año=2024 | V1,V2,V3,V4,V5,V8 | “Fecha válida” |
| Dia=28, mes=2, año=2022 | V1,V2,V3,V4,V5,V9 | “Fecha válida” |
| Día = a, mes=4,año=1 | NV1 | “La fecha es incorrecta”  N/A (porque el método no se ejecuta si no le pones un int) |
| Día = 1,23 mes=4,año=1 | NV2 | “La fecha es incorrecta”  N/A (porque el método no se ejecuta si no le pones un int) |
| Día = 1,mes=a,año=1 | NV3 | “La fecha es incorrecta”  N/A (porque el método no se ejecuta si no le pones un int) |
| Día = 1,mes=1,23,año=1 | NV4 | “La fecha es incorrecta”  N/A (porque el método no se ejecuta si no le pones un int) |
| Día = 1,mes=4,año=a | NV5 | “La fecha es incorrecta”  N/A (porque el método no se ejecuta si no le pones un int) |
| Día = 1,mes=4,año=1,23 | NV6 | “La fecha es incorrecta”  N/A (porque el método no se ejecuta si no le pones un int) |
| Dia=1,mes= 4,año=0 | NV7 | “La fecha es incorrecta” |
| Día = 1 mes=0,año=1 | NV8 | “La fecha es incorrecta” |
| Día = 1 mes=13,año=1 | NV9 | “La fecha es incorrecta” |
| Día = 0 mes=4,año=1 | NV10 | “La fecha es incorrecta” |
| Día = 31 mes=4,año=1 | NV11 | “La fecha es incorrecta” |
| Día = 0 mes=1,año=1 | NV12 | “La fecha es incorrecta” |
| Día = 32 mes=1,año=1 | NV13 | “La fecha es incorrecta” |
| Día = 0 mes=2,año=2024 | NV14 | “La fecha es incorrecta” |
| Día = 30 mes=2,año=2024 | NV15 | “La fecha es incorrecta” |
| Día = 0 mes=2,año=2022 | NV16 | “La fecha es incorrecta” |
| Día = 29 mes=2,año=2022 | NV17 | “La fecha es incorrecta” |

1. **Desarrolla el conjunto de pruebas para el fichero: ComprobarFecha1.java en JUnit. (Refactoriza el código en método/s que pueda/n ser probado/s según hemos visto en clase).**
2. **En caso de que haya pruebas no superadas, depura el código para ver dónde se encuentran los errores.**
3. **Averigua la complejidad ciclomática del ficheros ComprobarFecha1.java original, y añade aquí tus conclusiones.**

|  |
| --- |
| Formato de entrega: proyecto en el que se desarrolla la práctica en IntelliJ IDEA en formato de archivo comprimido titulado con vuestro nombre + este documento con los apartados desarrollados, con explicaciones y capturas de pantalla. |