1- Pruebas unitarias de la clase Seguro: 2 métodos, partición equivalente, AVL.

-Constructor:

-Precio(): según lo que se indica.

2- Implementar la clase SeguroTest con sus métodos. Test Driven Development.

3- Implementar la clase Seguro.

4- Ejecución de la clase de prueba y corrección de fallos. Caja Negra.

5- Si todas las pruebas de caja negra son correctas, se comprueba la cobertura (Caja Blanca).

6- Pruebas de integración de la GUI con la clase Seguro usando Fest.

7- Pruebas unitarias de la clase ListaOrdenada e identificar en qué métodos están los fallos.

8- Corregir esos fallos.

PASO 1:

**Método precio():**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parámetros** | **Clases válidas** | **Valores** | **Clases no válidas** | **Valores** |
| Nivel de cobertura | 1. Todo riesgo.  2. Terceros + lunas.  3. Terceros simple. | Todo riesgo  Terceros + lunas  Terceros simple | 11. !={Todo riesgo, terceros + lunas, terceros simple} | Terceros ampliados |
| Potencia | 4. [90, 110]  5. (110, ∞) | 90  100  110  ------  200 | 12. < 90  13. !Número | 0  AAA |
| Fecha último siniestro | 6. [hoy – 1 año, hoy]  7. [hoy – 3 años, hoy – 1 año)  8. (-∞, hoy – 3 años) | Hoy – 1 año  Hoy – 6 meses  Hoy  ------  Hoy – 3 años  Hoy – 2 años  ------  Hoy – 5 años | 14. > hoy  15. !fecha | Cualquier fecha no válida (fecha con mes 13)  Mañana |
| Grado minusvalía | 9. True.  10. False. | True  False | - | - |

Casos de prueba válidos:

(Todo riesgo, 90, hoy – 1 año, True): 937,5

(Terceros + lunas, 100, hoy – 6 meses, False): 830

(Terceros simple, 110, hoy, True): 465

(Todo riesgo, 200, hoy – 3 años, False): 1250

(Terceros + lunas, 200, hoy – 2 años, False): 770

(Terceros simple, 90, hoy – 5 años, False): 420

Casos de prueba no válidos:

(Terceros ampliados, 110, hoy, True): DatoIncorrectoException

(Todo riesgo, 0, hoy – 1 año, False): DatoIncorrectoException

(Terceros + lunas, AAA, hoy – 6 meses, True): NumberFormatException

(Terceros simple, 200, ’03/13/2021’, False): DatoIncorrectoException

(Todo riesgo, 110, mañana, False): DatoIncorrectoException

**Método Seguro():**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parámetros** | **Clases válidas** | **Valores** | **Clases no válidas** | **Valores** |
| Potencia | 1. [90, 110]  2. (110, ∞) | 90  100  110  ------  200 | 7. < 90  8. !Número | 0  AAA |
| Cliente | 3. No null | - | 9. Null | - |
| Cobertura | 4. Todo riesgo.  5. Terceros + lunas.  6. Terceros simple. | Todo riesgo  Terceros + lunas  Terceros simple | 10. !={Todo riesgo, terceros + lunas, terceros simple} | Terceros ampliados |

Casos de prueba válidos:

(90, No null, Todo riesgo)

(100, No null, Terceros + lunas)

(110, No null, Terceros simple)

(200, No null, Todo riesgo)

Casos de prueba no válidos:

(0, No null, Todo riesgo): DatoIncorrectoException

(AAA, No null, Todo riesgo): NumberFormatException

(90, Null, Terceros + lunas): NullPointerException

(100,No null, Terceros ampliados): DatoIncorrectoException