ET.02.03 Primer Problema

1. Escribir, al menos el pseudocódigo correspondiente al método o a los métodos identificados.

* Clase Persona

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Resto de getters y setters…

* Clase Espacio

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

* Clase Meteorología

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Resto de getters y setters

* Clase Principal

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto, Aplicación

Descripción generada automáticamente

1. Identificar las variables que se deben tener en cuenta para probar el método de interés.

* **Temperatura**: entero – (-∞,0) – [0,15] – (15,25] – (25-30] – (30,35] – (35, ∞)
* **Humedad**: entero – [0,15) – [15,60] – (60,100]
* **Lluvia**: boolean - true- false
* **Nieve**: boolean - true- false
* **Niebla**: boolean - true- false
* **capacidadMax**: boolean - true- false
* **restriccionConfi**: boolean - true- false
* **enferma**: boolean - true- false
* **síntomas**: boolean - true- false
* **contacto**: boolean - true- false
* **pasaporte**: boolean - true- false
* **pasoCovid**: boolean - true- false

1. Identificar los valores de pruebas para cada una de las variables anteriores usando las tres técnicas vistas en teoría, especificando para cada una cual es la que ha sido usada.

* Por clases de equivalencia:
* Temperatura: -12, 5, 20, 27, 32, 45
* Humedad: 12, 30, 70
* Por valores límite:
  + Temperatura: 0, 15, 25, 30, 35
  + Humedad: 0, 15, 60, 100
* Por valores límite en variante pesada:
  + Temperatura: -1, 1, 14, 16, 24, 26, 29, 31, 34, 36
  + Humedad: -1, 1, 14, 16, 59, 61, 99, 101

**Temperatura**: 21 valores de prueba obtenidos

**Humedad**: 15 valores de prueba obtenidos

**Lluvia**: 2 valores de prueba obtenidos

**Nieve**: 2 valores de prueba obtenidos

**Niebla**: 2 valores de prueba obtenidos

**capacidadMax**: 2 valores de prueba obtenidos

**restriccionConfi**: 2 valores de prueba obtenidos

**enferma**: 2 valores de prueba obtenidos

**síntomas**: 2 valores de prueba obtenidos

**contacto**: 2 valores de prueba obtenidos

**pasaporte**: 2 valores de prueba obtenidos

**pasoCovid**: 2 valores de prueba obtenidos

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

El número máximo de casos de prueba es la multiplicación del número de casos de prueba de cada una de las variables que influyen en este caso el resultado sería: **322.560**.

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

La variable que tienes más valores de valores de prueba es temperatura con 21 valores, por tanto, con **21 casos de prueba** podremos probar todas las variables al menos una vez.

1. Defina conjuntos de pruebas para alcanzar cobertura pairwaise usando el algoritmo explicado en clase. Se pueden comprobar los resultados con el programa PICT1.

Para alcanzar la cobertura pairwaise se multiplican las dos variables con más valores de prueba, en este caso, **temperatura y humedad**, dándonos un total de **315** combinaciones posibles.

1. Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura de decisiones.

Tenemos 17 decisiones:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A (p.isEnferma()) | B (p.isSintomas()) | C (p.isContacto()) | D (p.isPasaporte()) | E (p.isPasoCovid()) | A or B or C or not D or not E |
| F | F | F | F | F | V |
| V | F | F | F | F | V |
| F | V | F | F | F | V |
| V | V | F | F | F | V |
| F | F | V | F | F | V |
| V | F | V | F | F | V |
| F | V | V | F | F | V |
| V | V | V | F | F | V |
| F | F | F | V | F | V |
| V | F | F | V | F | V |
| F | V | F | V | F | V |
| V | V | F | V | F | V |
| F | F | V | V | F | V |
| V | F | V | V | F | V |
| F | V | V | V | F | V |
| V | V | V | V | F | V |
| F | F | F | F | V | V |
| V | F | F | F | V | V |
| F | V | F | F | V | V |
| V | V | F | F | V | V |
| F | F | V | F | V | V |
| V | F | V | F | V | V |
| F | V | V | F | V | V |
| V | V | V | F | V | V |
| F | F | F | V | V | F |
| V | F | F | V | V | V |
| F | V | F | V | V | V |
| V | V | F | V | V | V |
| F | F | V | V | V | V |
| V | F | V | V | V | V |
| F | V | V | V | V | V |
| V | V | V | V | V | V |

* Primera decisión
* Segunda decisión

|  |  |  |
| --- | --- | --- |
| A (m.getTemperatura() < 0) | B (m.getHumedad() < 15) | A and B |
| V | V | V |
| F | F | F |
| V | F | F |
| F | V | F |

* Tercera decisión

|  |  |  |
| --- | --- | --- |
| A (m.isLluvia()) | B (m.isNieve()) | A or B |
| V | V | V |
| F | F | F |
| V | F | V |
| F | V | V |

* Cuarta decisión

|  |  |  |
| --- | --- | --- |
| A (m.isLluvia()) | B (m.isNieve()) | not A and not B |
| V | V | F |
| F | F | V |
| V | F | F |
| F | V | F |

* Quinta decisión

|  |  |
| --- | --- |
| A (e.isCapacidadMax()) | not A |
| V | F |
| F | V |

* Sexta decisión

|  |  |  |  |
| --- | --- | --- | --- |
| A (m.getTemperatura() >= 0) | B (m.getTemperatura() <= 15) | C (m.isLluvia()) | (A and B) and not C |
| F | F | F | F |
| V | F | F | F |
| F | V | F | F |
| V | V | F | V |
| F | F | V | F |
| V | F | V | F |
| F | V | V | F |
| V | V | V | F |

* Séptima decisión

|  |  |
| --- | --- |
| A (e.isCapacidadMax()) | not A |
| V | F |
| F | V |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A (m.getTemperatura() > 15) | B (m.getTemperatura() <= 25) | C (m.isLluvia()) | D (m.isNiebla) | E (m.getHumedad() < 60) | A and B and not C and not D and E |
| F | F | F | F | F | F |
| V | F | F | F | F | F |
| F | V | F | F | F | F |
| V | V | F | F | F | F |
| F | F | V | F | F | F |
| V | F | V | F | F | F |
| F | V | V | F | F | F |
| V | V | V | F | F | F |
| F | F | F | V | F | F |
| V | F | F | V | F | F |
| F | V | F | V | F | F |
| V | V | F | V | F | F |
| F | F | V | V | F | F |
| V | F | V | V | F | F |
| F | V | V | V | F | F |
| V | V | V | V | F | F |
| F | F | F | F | V | F |
| V | F | F | F | V | F |
| F | V | F | F | V | F |
| V | V | F | F | V | V |
| F | F | V | F | V | F |
| V | F | V | F | V | F |
| F | V | V | F | V | F |
| V | V | V | F | V | F |
| F | F | F | V | V | F |
| V | F | F | V | V | F |
| F | V | F | V | V | F |
| V | V | F | V | V | F |
| F | F | V | V | V | F |
| V | F | V | V | V | F |
| F | V | V | V | V | F |
| V | V | V | V | V | F |

* Octava decisión
* Novena decisión

|  |  |
| --- | --- |
| A (e.isRestriccionConfi()) | not A |
| V | F |
| F | V |

* Decima decisión

|  |  |  |  |
| --- | --- | --- | --- |
| A (m.getTemperatura() > 25) | B (m.getTemperatura() <= 30) | C (m.isLluvia()) | (A and B) and not C |
| F | F | F | F |
| V | F | F | F |
| F | V | F | F |
| V | V | F | V |
| F | F | V | F |
| V | F | V | F |
| F | V | V | F |
| V | V | V | F |

* Undécima decisión

|  |  |
| --- | --- |
| A (e.isCapacidadMax()) | not A |
| V | F |
| F | V |

* Duodécima decisión

|  |  |  |  |
| --- | --- | --- | --- |
| A (m.getTemperatura() > 30) | B (m.getTemperatura() <= 35) | C (m.isLluvia()) | (A and B) and not C |
| F | F | F | F |
| V | F | F | F |
| F | V | F | F |
| V | V | F | V |
| F | F | V | F |
| V | F | V | F |
| F | V | V | F |
| V | V | V | F |

* Decimotercera decisión

|  |  |
| --- | --- |
| A (e.isCapacidadMax()) | A |
| V | V |
| F | F |

* Decimocuarta decisión

|  |  |
| --- | --- |
| A (e.isCapacidadMax()) | not A |
| V | F |
| F | V |

* Decimoquinta decisión

|  |  |  |
| --- | --- | --- |
| A (m.getTemperatura() > 30 | B (m.isLluvia()) | A and not B |
| V | V | F |
| F | F | F |
| V | F | V |
| F | V | F |

* Decimosexta decisión

|  |  |
| --- | --- |
| A (e.isCapacidadMax()) | A |
| V | V |
| F | F |

* Decimoséptima decisión

|  |  |
| --- | --- |
| A (e.isCapacidadMax()) | not A |
| V | F |
| F | V |

1. Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura MC/DC.
2. Comente los resultados del número de los casos de pruebas conseguidos en los apartados 4, 5 y 6 ¿qué podría decirse algo de la cobertura alcanzada?

Según los resultados obtenidos nos parece mucho más eficiente el criterio pairwaise , ya que comprueba más casos que el each use pero no de una manera tan extensa como el combinacional , los resultados obtenidos han sido: **21**, siguiendo el criterio each use , **315** siguiendo el criterio pairwaise , y **322.560** por el criterio combinacional, si bien combinacional es el que más cobertura ofrece (prueba todos los posibles casos en el programa), no es para nada el más optimo posible a realizar.