

ISO2-2023-BC01-Testing-P1

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

Como nuestro programa se centra en escribir la clase Persona, el pseudocódigo que se mostrará a continuación, será los métodos necesarios para que los requisitos del enunciado se cumplan con respecto a la clase Persona. Dentro de estos métodos tenemos el constructor de la clase, así como un getter para obtener el nombre de la persona, como los métodos para comprobar si es mayor de edad, si es europeo o no y si se puede matricularse en un programa de doctorado porque tenga una titulación igual o superior a un master.

```
Public class Persona () {  
    String nombre;  
    String apellidos;  
    int nacimiento;  
    String nacionalidad;  
    String titulacion;  
    String certificado_ingles;  
    Int numero_telefono;  
    String correo_electronico;  
  
    Public Persona(String nombre,String apellidos, int nacimiento, String nacionalidad, String  
titulación, String certificado_ingles, int numero_telefono, String correo_electronico) {  
        //constructor  
  
        this.nombre=nombre;  
  
        this.apellidos=apellidos;  
  
        this.nacimiento=nacimiento;  
  
        this.nacionalidad=nacionalidad;  
  
        this.titulacion=titulacion;  
  
        this.certificado_ingles=certificado_ingles;  
  
        this.numero_telefono=numero_telefono;  
  
        this.correo_electronico=correo_electronico;  
  
    }  
  
    //getters  
  
    obtenerNombre(): String  
        devolver this.nombre
```

```

//Métodos

Public boolean es_mayor_de_edad() { throw MenorEdadException,NumeroNegativo

    Si año_actual – nacimiento >= 18

        Return true

    Else

        Throws new MenorEdadException

}

Public boolean es_europeo(){ throw NoEuropeoException

    Lista_paises_europeos = {}

    Si Lista_paises_europeos contiene obtenerNacionalidad()

        Return true

    Else{

        Throws new NoEuropeoException

    }

}

Public boolean apuntarse_doctorado() {

    If es_mayor_de_edad() && titulación >= “master”

        return True

    Else

        return False

}

} // fin de la clase

```

NOTA: Hemos decidido que la variable nacimiento sea un entero, que muestra su año de nacimiento en vez de un String con formato día/mes/año para simplificar el problema de obtener su edad, que sería restando el año actual menos el año de su nacimiento, de esta forma asumimos que, aunque no haya cumplido empíricamente la mayoría de edad, al finalizar el año sí que sabemos que es mayor de edad en el caso particular de que en el año actual cumpla los 18 años.

También cabe destacar, que, al ser pseudocódigo, se entiende las jerarquías en el mundo académico donde un título de “Bachiller” objetivamente hablando es menor que un “Master” o “Doctorado”, posiblemente, a la hora de hacer la implementación real, podemos optar por poner un enum en vez de una cadena, pero a modo de simplificar el problema, hemos decidido ponerlo como String.

También hemos considerado que solo vamos a tener métodos que, si vamos a necesitar, no ponemos todos los getters ni setters porque a la hora de evaluar la cobertura de la clase Persona, resultará que hay muchas líneas de código que no habrán pasado las pruebas, debido a que no usamos ningún setter y solo el getter del nombre de la persona.

También tenemos en consideración que para que se cumpla el método “apuntarse_doctorado”, no es necesario realmente evaluar la propia edad de la Persona, porque se podría dar el caso (muy poco probable) de que la persona sea superdotada y que se pueda apuntar a un doctorado siendo menor de edad, porque tiene ya el master, por ejemplo. Claramente en una circunstancia normal, para optar a un doctorado, tienes que ser mínimo mayor de edad para acceder a estudios superiores como un grado y mínimo 24/25 años (depende de los años que dure la carrera) para hacer un master y 25-27 años (dependiendo de la duración del master). Por esta razón, al no tener en cuenta la edad de la persona, no discriminamos casos especiales que remotamente pueden ser posibles.

Sin embargo, para poder aplicar los conocimientos de la asignatura, hemos considerado que un requisito para apuntarse a un doctorado es que sea mayor de edad, así podemos hacer lo apartados 7 y 8 sin que sean triviales.

También supongamos que “año actual” siempre sea el año real donde estemos, empezando por el 2023.

2) Identificar las variables que se deben tener en cuenta para probar los métodos de interés.

Como hemos visto en los métodos “es_mayor_de_edad”, “es_europeo” y “apuntarse_doctorado”, no todas las variables son relevantes a la hora de probar nuestros métodos, ya que no todas toman lugar en ellas. Observando los métodos mencionados anteriormente, las variables que si influye son:

Fecha de nacimiento: Int

Nacionalidad: String

Titulación: String

3. 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.

Variable	Clase equivalencia	Valores	Valores ligeros	Valores pesados	Conjetura de errores
Nacimiento	(-infinito,0) [0, 2006] (2006,infinito)	-2000, 2005, 2007	0, 2006	-1, 1	6000000000 $2^{31} + 1$ $-2^{31} - 1$
Nacionalidad	Europeo	“Alemana” “Italiana” “Rusa”	No tiene porque no	No tiene porque no	444

	No europeo	"Chilena" "Japonesa" "Egipcia" "Australiana"	existe ningún valor límite	existe ningún valor límite	
Titulación	Apto NoApto	"Graduado Escolar" "Instituto" "Bachiller" "Grado Universitario" "Doctorado"	"Master"		105

NOTAS:

- Las clases de equivalencia "**Apto**" y "**NoApto**" hace referencia si son aptos para apuntarse a un doctorado, ya que es necesario que la persona cumpla con unos requisitos mínimos.
- En la variable Nacionalidad hay que tener en cuenta que esta al ser países europeos o no europeos, significa que esta puede tomar como valor todos los países del mundo.

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

Teniendo en cuenta que el número máximo posible de casos de pruebas a partir de combinatorio resulta en el producto cartesiano de las variables involucradas:

- Suponiendo solo los datos propuestos, el número que nos da es
 $| \text{Nacimiento} | * | \text{Nacionalidad} | * | \text{Titulación} | = 10 * 8 * 7 = \mathbf{560 \text{ posibles casos.}}$
- Suponiendo **todos** los posibles valores que pueden tomar las variables, como tenemos una variable de tipo entero, donde puede tomar desde el valor -2^{31} hasta el valor 2^{31} por si queremos hacer todas las combinaciones, podemos suponer que hay **infinitas** de ellas debido al gran número de posibles valores que puede tener un entero, por lo que el número máximo sería **infinito**.

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

CP1: (-2000, Alemana, Graduado Escolar)

CP2:(2005, Italiana, Instituto)

CP3:(2007, Rusa, Bachiller)

CP4:(0, Chilena, Grado Universitario)

CP5:(2006, Japonesa, Master)

CP6:(-1, Egipcia, Doctorado)

CP7:(1, Australiana, Instituto)

CP8:(6000000000, Japonesa, Bachiller)

CP9:($2^{31} + 1$, Rusa, Master)

CP10:($-2^{31} - 1$, Alemana, Doctorado)

6. Defina conjuntos de pruebas para alcanzar cobertura pairwise usando el algoritmo explicado en clase. Se pueden comprobar los resultados con el programa PICT

Test: { (-2000, Alemana , Graduado Escolar), (-2000, Italiana, Instituto), (-2000, Rusa, Bachiller), (-2000, Chilena, Grado Universitario), (-2000, Japonesa, Master), (-2000, 444, Doctorado), (-2000, Australiana, Doctorado), (-2000, egipcia, 105)

(2005, Italiana, Graduado Escolar), (2005, Rusa, Instituto), (2005, Chilena, Bachiller), (2005, 444, Grado Universitario), (2005, Egipcia, Master), (2005, Australiana, Doctorado), (2005, Alemana, Doctorado) (2005, japonesa, 105)

(2007, Rusa, Graduado Escolar), (2007, Chilena, Instituto), (2007, Japonesa, Bachiller), (2007, Egipcia, Grado Universitario), (2007, Australiana, Master), (2007, Alemana, Doctorado), (2007, Italiana, Doctorado), (2007, 444, 105)

(0, Chilena, Graduado Escolar), (0, Japonesa, Instituto), (0, Egipcia, Bachiller), (0, Australiana, Grado Universitario), (0, Alemana, Master), (0, Italiana, Doctorado), (0, Rusa, Doctorado), (0, 444, 105)

(2006, Japonesa, Graduado Escolar), (2006, Egipcia, Instituto), (2006, Australiana, Bachiller), (2006, Alemana, Grado Universitario), (2006, Italiana, Master), (2006, 444, Doctorado), (2006, Chilena, Doctorado), (2006, Rusa, 105)

(-1, Egipcia, Graduado Escolar), (-1, Australiana, Instituto), (-1, Alemana, Bachiller), (-1, Italiana, Grado Universitario), (-1, Rusa, Master), (-1, 444, Doctorado), (-1, Japonesa, Doctorado), (-1, chilena, 105)

(1, Australiana, Graduado Escolar), (1, Alemana, Instituto), (1, 444, Bachiller), (1, Rusa, Grado Universitario), (1, Chilena, Master), (1, Japonesa, Doctorado), (1, Egipcia, Doctorado), (1, italiana, 105)

(6000000000, chilena, Graduado Escolar), (6000000000, 444, Instituto), (6000000000, egipcia, Bachiller), (6000000000, australiana, Grado Universitario), (6000000000, Alemana, Master), (6000000000, Italiana, Doctorado), (6000000000, Rusa, Doctorado), (6000000000, japonesa, 115)

($2^{31} + 1$, 444, Graduado Escolar), ($2^{31} + 1$, Alemana, Instituto), ($2^{31} + 1$, Italiana, Bachiller), ($2^{31} + 1$, Rusa, Grado Universitario), ($2^{31} + 1$, Chilena, Master), ($2^{31} + 1$, Japonesa, Doctorado), ($2^{31} + 1$, Egipcia, Doctorado), ($2^{31} + 1$, Australiana, 105)

($-2^{31} - 1$, Italiana, Graduado Escolar), ($-2^{31} - 1$, Rusa, Instituto), ($-2^{31} - 1$, Chilena, Bachiller), ($-2^{31} - 1$, Japonesa, Grado Universitario), ($-2^{31} - 1$, Egipcia, Master), ($-2^{31} - 1$, Australiana, Doctorado), ($-2^{31} - 1$, 444, Doctorado), ($-2^{31} - 1$, Alemana, 105) }

En los test para conseguir el pairwise lo que hemos hecho ha sido ir combinando los valores de la tabla mayor con todos los posibles valores de las tablas más pequeñas.

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

Para alcanzar cobertura de decisiones, tenemos que evaluar cada decisión tanto a **true** como a **false**.

- **es_mayor_de_edad()**

Metodo es_mayor_de_edad()
A
T
F

Siendo A la condicion “año_actual – obtenerNacimiento() >= 18”. Como esta decisión casualmente también solo tiene una condición, solo deberíamos cubrir que dicha condicion se evalúe tanto a **True** como a **False**.

Año_actual	nacimiento	A
2023	2003	True
2023	2015	False

Creamos los casos de prueba:

{(2003, “español”, “Bachiller”), (2015, “español”, “Graduado Escolar”)}

- **es_europeo()**

Metodo es_europeo()
A
T
F

Siendo A la condición “Lista_paises_europeos contiene obtenerNacionalidad()”. Al igual que el método anterior, esta decisión solo tiene una condición, por lo que tenemos que asegurar que para esa condición se evalúa como T y F.

nacionalidad	A
“española”	True

"chilena"	False
-----------	-------

Creamos los casos de prueba:

{(2003, "española", "Instituto"), (2010, "chilena", "Graduado Escolar")}

• **apuntarse_doctorado()**

Metodo apuntarse_doctorado()		
A	B	A and B
F	F	F
F	T	F
T	F	F
T	T	T

Siendo A la condición "año_actual – nacimiento >= 18" y B la condición "titulación >= "master". En esta ocasión, la decisión está formada por 2 condiciones, por lo que la tabla de verdad resulta ser un poco más grande.

Para los casos de prueba, como hay que asegurarse de que la decisión tome el valor true y false, vamos a escoger la casuística 3 y 4 para generar los casos de pruebas.

Creamos los casos de prueba:

A	B	A & B	nacimiento	titulación
T	F	F	2003	"Bachiller"
T	T	T	2003	"Master"

Por lo tanto, un buen conjunto de casos de prueba sería:

{(2003, "española", "Master"), (2015, "tailandesa", "Bachiller")}

Con estos casos de pruebas, abarcamos todas las decisiones, ya que con el primer caso de prueba evaluamos todas las decisiones a True, mientras que el segundo caso de prueba evaluamos todas las decisiones como False.

8. Para los trozos de código que incluyan decisiones, proponga conjunto de casos de prueba para alcanzar cobertura MC/DC.

Los métodos que incluyen decisiones son los métodos **es_mayor_de_edad()**, **es_europeo()** y **apuntarse_doctorado()** respectivamente, sin embargo, tal y como hemos abordado estos métodos con sus supuestos, resulta que solo hay 1 condición en las decisiones de los metodos **es_mayor_de_edad()** y **es_europeo()**, por lo que la condición dominante siempre va a ser la misma, pero para el método **apuntarse_doctorado()** se podría hacer casos de prueba ya que tiene 2 condiciones en la decisión:

A and B (método apuntarse_doctorado())			
Condiciones		Decisión	
A	B	A and B	Condición Dominante
F	F	F	A, B
F	T	F	A
T	F	F	B
T	T	T	A, B

Para satisfacer que todas las evaluaciones de la decisión posibles y que al menos cada condición sea la dominante al menos una vez, hemos escogido las 3 casuísticas sombreadas en amarillo, ya que así aseguramos que la decisión sea evaluada tanto a true como a false y que cada condición sea la dominante en al menos 1 caso.

A	B	A & B	nacimiento	titulación
F	T	F	2015	"Doctorado"
T	F	F	1978	"Graduado Escolar"
T	T	T	1998	"Master"

Un buen conjunto de pruebas sería:

{(2015, "japonesa", "Doctorado"), (1978, "española", "Graduado Escolar"), (1998, "italiana", "Master")}

Realmente en la variable nacionalidad, podríamos poner la que quisiéramos, ya que no es relevante en la decisión.

9. Comente los resultados del número de los casos de pruebas conseguidos en los apartados 4, 5 y 6 ¿qué podría decirse de la cobertura alcanzada?

Como se puede ver en los valores propuestos, el número de casos de pruebas depende de cuantos valores hemos propuesto para abordar el problema, por lo que, si hay más valores, el número de pruebas a realizar sería mayor. Como las pruebas cuestan dinero, es crucial encontrar que con el menor número de pruebas para alcanzar todas las posibilidades del comportamiento del programa (MINMAX).

Con nuestros valores propuestos, el número de casos que tendríamos que hacer en el **apartado 4** sería de **560 casos** ya que estamos cubriendo todas las combinaciones para los valores propuestos. Para el **apartado 5** nos sale la cantidad de **10 casos** para cubrir each use. Finalmente, para el

apartado 6 necesitaríamos **80 casos** para cubrir pairwise, por lo que la suma total de casos asciende a **650 casos**.

Evidentemente, este número de casos podría ser relativamente elevado acorde con el punto de que hay que intentar hacer el mínimo número de casos para abordar todas las posibilidades.

Se podría solo proponer los valores justos para abordar todas las clases de equivalencias propuestas tanto en la variable de Nacionalidad como en Titulación, ya que sabemos que si 2 valores pertenecen a la misma clase de equivalencia se van a comportar de la misma manera.

Con respecto a la cobertura, creando los casos de prueba atendiendo a los valores propuesto, que han sido previamente seleccionados para abordar todas las clases de equivalencia propuestas, así también como algunos valores que podrían generar un comportamiento erróneo del programa en el apartado de “Conjeturas”, la cobertura que se alcanzará será muy alta, ya que al proponer valores buenos atendiendo a cada clase de equivalencia creada, sumada a los casos de prueba propuestos en los apartados 4,5,6 posiblemente se comprobaría en su totalidad el comportamiento del programa.