UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INFORMÁTICA

GRUPO BC.05

TRABAJO TEÓRICO PROBLEMA 3

Trabajo realizado por:

Andrés López Salgado

Tomás Abarca Cerro

Asignatura: Ingeniería del Software II

Grupo: BC.05

Titulación: Grado en Ingeniería Informática

1. Código correspondiente al método

```java
public class Cliente {
    private int edad;
    private boolean estudiante, independiente;

    public Cliente(int edad, boolean estudiante, boolean independiente) {
        super();
        this.edad = edad;
        this.estudiante = estudiante;
        this.independiente = independiente;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }

    public boolean isEstudiante() {
        return estudiante;
    }

    public void setEstudiante(boolean estudiante) {
        this.estudiante = estudiante;
    }

    public boolean isIndependiente() {
        return independiente;
    }

    public void setIndependiente(boolean independiente) {
        this.independiente = independiente;
    }
}
```

```java
public class Main {

    public static void main(String[] args) {

        Scanner escaner = new Scanner(System.in);
        Boolean independiente = null;
        Boolean estudiante = null;
        int edad = -1;

        while (edad < 0 || edad > 120) {
            System.out.println("Introduzca la edad del cliente entre 0 y 120");
            try {
                edad = escaner.nextInt();
            } catch (Exception e) {
                System.out.println("Valor numerico entero no detectado.");
                // Limpiar el escaner para que no haga un bucle continuo de excepción.
                escaner.nextLine();

            }
        }
        while (estudiante == null) {
            System.out.println("Introduzca si el cliente es estudiante (true) o trabaja (false).");
            try {
                estudiante = escaner.nextBoolean();
            } catch (Exception e) {
                System.out.println("Valor booleano no detectado.");
                // Limpiar el escaner para que no haga un bucle continuo de excepción.
                escaner.nextLine();

            }
        }
        while (independiente == null) {
            System.out.println(
                    "Introduzca si el cliente es independiente (true) o reside en el domicilio de sus padres (false).");
            try {
                independiente = escaner.nextBoolean();
            } catch (Exception e) {
                System.out.println("Valor booleano no detectado.");
                // Limpiar el escaner para que no haga un bucle continuo de excepción.
                escaner.nextLine();

            }
        }
        Cliente c1 = new Cliente(edad, estudiante, independiente);

        if (c1.getEdad() < 18 && c1.isEstudiante() && !c1.isIndependiente()) {
            System.out.println("La cuenta ideal para este cliente es la 'Cuenta Confort'");

        } else if (edad < 25 && edad >= 18) {

            if (c1.isEstudiante() && c1.isIndependiente())
                System.out.println("La cuenta ideal para este cliente es la 'Cuenta Vamos que tú puedes'");

            else if (!c1.isEstudiante() && !c1.isIndependiente())
                System.out.println("La cuenta ideal para este cliente es la 'Cuenta Ahorra ahora que puedes'");

            else if (!c1.isEstudiante() && c1.isIndependiente())
                System.out.println("La cuenta ideal para este cliente es la 'Cuenta Saltando del Nido'");

        } else if (edad >= 25) {

            if (!c1.isEstudiante() && !c1.isIndependiente())
                System.out.println("La cuenta ideal para este cliente es la 'Cuenta Independizate que va siendo hora'");

            else if (c1.isEstudiante() && c1.isIndependiente())
                System.out.println("La cuenta ideal para este cliente es la 'Cuenta Bienvenido a la Vida Adulta'");

        } else
            System.out.println("No se le puede ofrecer una tipo de cuenta de acuerdo a sus necesidades");
    }

}
```

2. Identificar las variables que se deben tener en cuenta

Se deben tener en cuenta las siguientes variables:

- Edad = {0,120} = Indica la edad del usuario, que no podrá ser negativa y ponemos una limitación en la edad máxima a 120 años.
- esEstudiante = {true, false} = Indica si el usuario es estudiante.
- esIndependiente = {true, false} = Indica si el usuario vive de forma independiente.

3. Indicar los conjuntos de valores de prueba para cada variable

PARÁMETROS	VALORES OBJETIVOS	CLASES DE EQUIVALENCIA	VALORES POR PARTICION	VALOR LIMITE	CONJETURA DE ERRORES	VALORES TOTALES
EDAD	(0,120)	(-inf,0),[0,18), [18,25),[25,120)	-25, 15, 20, 70	1,0,1, 17,18,19, 24,25,26, 119,120,121	150	-25, 15, 20, 70, 1,0,1, 17,18,19, 24,25,26, 119,120,121,150
ESESTUDIANTE	(true,false)	(true,false)				(true,false)
ESINDEPENDIENTE	(true,false)	(true,false)				(true,false)

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

La combinación de los tres parámetros 17*2*2 nos daría 52 casos de prueba

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

Para cumplir each use tenemos que fijarnos que parámetro tiene más valores totales, para ello nos fijamos en los tres parámetros comentados anteriormente consiguiendo así el número de casos de prueba para poder realizar each use, utilizaremos el parámetro edad y esEstudiante:

(-1,true)

(0, false)

(1, false)

(17, false)

(18, false)

(19, false)

(24, false)

(25, false)

(26, false)

(119, false)

(120, false)

(121, false)

(150, false)

6. Defina conjuntos de pruebas para alcanzar cobertura pairwaise usando el algoritmo explicado en clase. Se pueden comprobar los resultados con el programa PICT2

El pairwaise se define como todos los pares los casos de prueba deben visitar, al menos una vez, todos los pares de valores de dos parámetros cualesquiera .En nuestro caso hemos decidido usar los parámetros día y mes para hacer todas las combinaciones posibles entre ellos dos.

Edad	esEstudiante
-1	false
-1	true
0	true
0	false
1	false
1	true
24	true
24	false
25	false
25	true
26	true
26	false
119	false
119	true
120	true
120	false
121	false
121	true
17	true
17	false
18	false
18	true
19	true
19	false
150	false
150	true

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

Método para introducir la edad

- Edad< 0 → A
- Edad > 120 → B

```
while (edad < 0 || edad > 120) {

        edad = escaner.nextInt();

                }
```

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Método para introducir si es estudiante

- esEstudiante = true(1) → C

```
while (estudiante == null) {

        estudiante = escaner.nextBoolean();

}
```

C	C
0	0
1	1

Método para introducir si es independiente

- esIndependiente = true(1) → D

```
while (independiente == null) {

        independiente = escaner.nextBoolean();

}
```

D	D
0	0
1	1

Método para calcular el tipo de cuenta

- edad>18 → E
- esEstudiante → C
- esIndependiente → D
- edad<25 → G
- edad>=25 → F

```
if (c1.getEdad() < 18 && c1.isEstudiante() && !c1.isIndependiente()) {
        System.out.println("La cuenta ideal para este cliente es la 'Cuenta Confort'");
                }
else if (edad < 25 && edad >= 18) {
}
else if (edad >= 25){
}
```

E	**C**	**D**	**E AND C AND D**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

G	E	G OR E
0	0	0
0	1	1
1	0	1
1	1	1

F	F
0	0
1	1

Simplificando todos los casos de prueba posibles podemos a escoger los siguientes para tener una cobertura casi completa

{-25,true,false}

{0,false,false}

{15,true,true}

{18,false,true}

{20,false, false}

{25,true,true}

{70,false,true}

{150,true,false}

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

Método para introducir la edad

- Edad< 0 → A
- Edad > 120 → B

```
while (edad < 0 || edad > 120) {

        edad = escaner.nextInt();

                }
```

A	B	A OR B	**CONDICIÓN DOMINANTE**
0	0	0	A,B
0	1	1	B
1	0	1	A
1	1	1	A,B

Método para introducir si es estudiante

- esEstudiante = true(1) → C

```
while (estudiante == null) {

        estudiante = escaner.nextBoolean();

}
```

C	C	**CONDICIÓN DOMINANTE**
0	0	C
1	1	C

Método para introducir si es independiente

- esIndependiente = true(1) → D

```
while (independiente == null) {

        independiente = escaner.nextBoolean();
```

}

D	D	CONDICIÓN DOMINANTE
0	0	D
1	1	D

Método para calcular el tipo de cuenta

- edad>18 → E
- esEstudiante → C
- esIndependiente → D
- edad<25 → G
- edad>=25 → F

```
if (c1.getEdad() < 18 && c1.isEstudiante() && !c1.isIndependiente()) {
        System.out.println("La cuenta ideal para este cliente es la 'Cuenta Confort'");
            }
else if (edad < 25 && edad >= 18) { }
else if (edad >= 25){ }
```

E	C	D	E AND C AND D	CONDICIÓN DOMINANTE
0	0	0	0	E,C,D
0	0	1	0	E,C
0	1	0	0	E,D
0	1	1	0	E
1	0	0	0	C,D
1	0	1	0	C
1	1	0	0	D
1	1	1	1	E,C,D
0	0	0	0	E,C,D
0	0	1	0	E,C
0	1	0	0	E,C
0	1	1	0	C,D
1	0	0	0	C,D
1	0	1	0	C
1	1	0	0	D
1	1	1	1	E,C,D

G	E	G OR E	CONDICIÓN DOMINANTE
0	0	0	G,E
0	1	1	E
1	0	1	G
1	1	1	G,E

F	F	**CONDICIÓN DOMINANTE**
0	0	F
1	1	F

{0,true, true}

{120,true, false}

{10,false,false}

(12,true,true}

{30,false,true}

(22,true,false}

(25,true,true}

9. 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?

En el apartado 4 en el que tenemos todos los posibles casos de prueba vamos a obtener una cobertura del 100%, que son todas las posibilidades que puede existir en el sistema.

En el caso 5 no alcanzaremos una cobertura tan satisfactoria como la del apartado 4, cubrimos el 50% de casos posibles, pudiendo ser satisfactoria esta cobertura.

Por último, del ejercicio 6 podemos concluir que va a haber muy poca cobertura, ya que solo vamos a coger todas las posibilidades, pero únicamente de entre dos variables cualquiera del sistema, por lo que, la cobertura podría ser insuficiente.