

IES Virrey Morcillo. (Nombre Ciclo Formativo)

Ejercicio casos de prueba caja negra

Módulo: ENDE

Nombre del Alumno: Santiago Alarcón

Fecha: 22/2/2021

Contenido

Criterios de Evaluación 1

Material 1

Objetivos 1

1. Introducción 1

- 2. Tareas 2
- 3. Conclusión 2
- 4. Links 2

Criterios de Evaluación

- CE.x.x) ...

Material

Haremos uso de los siguientes recursos para realizar las tareas:

- Ordenador
- Conexión de Internet
- Procesador de Textos

Objetivos

- Conocer ...
- Repasar ...

1. Introducción

Enunciado 1: Un programa toma como entrada de datos a un fichero cuyo formato de registro es un objeto Empleado con los siguientes atributos:

Numero-empleado // Nombre-empleado // Meses-Trabajo // Directivo

donde:

- Numero-empleado es un campo de números enteros positivos de 3 dígitos (excluido el 000).
- Nombre-empleado es un campo alfanumérico de 10 caracteres.
- Meses-Trabajo es un campo que indica el número de meses que lleva trabajando el empleado;
es un entero positivo (incluye el 000) de 3 dígitos.
- Directivo es un campo de un solo carácter que puede ser «+» para indicar que el empleado es un directivo y «-» para indicar que no lo es.
El programa asigna una prima (que se imprime en un listado) a cada empleado según las normas siguientes:
 - - P1 a los directivos con, al menos, 12 meses de antigüedad → prima 1000
 - - P2 a los no directivos con, al menos, 12 meses de antigüedad → prima 75
 - - P3 a los directivos sin un mínimo de 12 meses de antigüedad → prima 500
 - - P4 a los no directivos sin un mínimo de 12 meses de antigüedad → prima 0

Enunciado 2:

Se ha ideado el siguiente algoritmo pseudocódigo para cumplir la especificación del problema anterior.

Algoritmo CalculaPrima

Begin

Print ("Ilmo.Sr.Director General:\n");

Read (objeto_EMP);

Prima = 0;

If (meses_EMP >= 12) Then

 If (directivo_EMP = "+") then

 Prima=1000;

 else Prima=75;

 else

 If (directivo_EMP = "+") then Prima=500;

 else;

```
Print (numEmpleado_EMP, nombreEmpleado_EMP, Prima);
Print ("S.e.u.o.");
```

End

El fichero de entrada que contiene los datos de objetos EMP y los nombres de los campos son más o menos iguales.

Suponiendo que en la anterior prueba de clases de equivalencia el programa se ejecuta con estos dos registros siguientes (de casos válidos), consecutivamente, en el orden dado:

Registro 1: 123 Fernández 009 +

Registro 2: 456 Fernando 013

Enunciado 3:

Genera un proyecto JAVA en Eclipse, creando el objeto empleado, incluyendo sus atributos,

constructores, métodos get y set.

2. Tareas

Tarea 1 sobre Enunciado 1) Crear una tabla de clases de equivalencia (las clases deberán ser numeradas) en la que se indiquen las siguientes columnas en cada fila:

- Condición de entrada que se analiza
- Clases válidas y
- Clases no válidas que se generan para la condición
- Regla heurística que se aplica para la generación de las clases de la fila

CONDICION DE ENTRADA	CLASES DE EQUIVALENCIA	CLASES VÁLIDAS	COD	CLASES NO VÁLIDAS	COD
NumeroEmpleado	RANGO	1CLASE VALIDA		2CLASES NO VALIDAS	
		001-999	V1	VALORES => 1000	NV1
				VALORES <= 000	NV2
Nombre-empleado	RANGO	CLASE VALIDA		CLASES NO VALIDAS	
		LONGITUD 1 A 10	V2	LONGITUD = 0	NV3

				LONGITUD > 10	NV4
	MIEMBROS DE UN CONJUNTO	CLASE VALIDA		CLASE NO VALIDA	
		CARACTERES ALFANUMÉRICOS	V3	NÚMEROS	NV5
Meses-trabajo	RANGO	1CLASE VALIDA		1CLASE NO VALIDA	
		(000-999)	V4	VALORES => 1000	NV6
				VALORES < 000	NV7
Directivo	LÓGICA	1CLASE VALIDA		1CLASE NO VALIDA	
		VALOR= +	V5	VALOR = -	NV8

Tarea 2 sobre Enunciado 1) Generar los casos de prueba (especificando la entrada en todos los casos y la salida esperada sólo en los casos válidos) para las clases creadas usando la técnica de particiones de equivalencia, indicando en cada caso las clases que cubre. Enunciar la regla se aplica para derivar los casos a partir de las clases de equivalencia.

CASO DE PRUEBA	CLASES DE EQUIVALENCIA	CONCICIONES DE ENTRADA				RESULTADO ESPERADO
		Número-Empleado	Nombre-Empleado	Meses-Trabajo	Directivo	
CP1	V1,V2,V3,V4,V5	1	PEDRO SANS	0	+	PRIMA = 500
CP2	V1,V2,V3,V4,V5	2	ANGEL	12	+	PRIMA = 1000
CP3	V1,V2,V3,V4,NV8	4	ANA	14	-	PRIMA = 75
CP4	V1,V2,V3,V4,NV8	5	SILVIA	8	-	PRIMA = 0
CP5	V1, NV1, NV3, NV6, NV8	1	#	1000	-	ERROR1
CP6	V1, NV3, NV6, NV8	6	#	1001	-	ERROR2
CP7	V1,V2,NV3,NV6,NV8	2	ANA	1000	-	ERROR3
CP8	V1,V2,V3,NV7,V5	10	ANGEL	-5	+	ERROR4

Tarea 3) sobre Enunciado 2) Comprobar si se cumple la cobertura de sentencias indicando, en su caso, cuáles de ellas no se ejecutan, empleando la numeración de líneas del código que se ofrece. En caso de que no se cumpla, añadir el mínimo número de registros adicionales para que se cumpla la cobertura.

Suponiendo que en la anterior prueba de clases de equivalencia el programa se ejecuta con estos dos registros siguientes (de casos válidos), consecutivamente, en el orden dado:

o **Registro 1: 123 Fernández 009 +**

//METODO CONSTRUCTOR calculaPRIMA

```
void CalculaPRIMA(){  
    begin  
    read (objeto_EMP);  
    prima = 0;  
    if (Meses_trabajo >= 12){  
        if(Directivo == "+") prima = 1000;  
        else prima = 75;  
    }else{  
        if(Directivo == "+") prima = 1000;  
        else;  
    } //fin else  
  
    System.out.println("La prima de este empleado le corresponde: " +  
prima);  
    System.out.println("S.e.o.u.")  
  
}
```

Por tanto, no se ejecutan las sentencias 5, 6, 7 ni 10

o **Registro 2: 456 Fernando 013 -**

//METODO CONSTRUCTOR calculaPRIMA

```
void CalculaPRIMA(){  
    begin  
    read (objeto_EMP);  
    prima = 0;  
    if (Meses_trabajo >= 12){  
        if(Directivo == "+") prima = 1000;  
        else prima = 75;  
    }else{  
        if(Directivo == "+") prima = 1000;  
        else;  
    } //fin else  
  
    System.out.println("La prima de este empleado le corresponde: " +  
prima);  
    System.out.println("S.e.o.u.")  
  
}
```

Por lo tanto no se ejecutan las sentencias 6, 8, 9 ni 10.

Para ser más concretos en ambos casos Nombre_empleado ni Numero_empleado están cubiertos por ninguna sentencia dentro de este método pero como dice el enunciado están cubiertos por sentencias dentro del objeto_EMP.

Tarea 4) Comprobar si se cumple la cobertura de decisiones, creando una tabla donde se marque los valores que adopta cada decisión, identificada con la numeración de líneas de código que se ofrece (por ej., la decisión "meses->=12" será la decisión 7). En caso de no cumplirse la cobertura, añadir el mínimo número de registros adicionales para que se cumpla. Los casos a añadir deben definir la entrada y la salida esperada al ejecutarse junto a los dos casos de caja negra (registros 1 y 2). Emplear el formato de salida que se incluye en el código del programa.

Decisión	Variable prima	objeto_EMP
1	La variable no ha sido inicializada	No se ha llamado al objeto
2	La variable no ha sido inicializada	No se ha llamado al objeto
3	La variable no ha sido inicializada	No se ha llamado al objeto
4	La variable no ha sido inicializada	(numEmpleado_EMP, nombreEmpleado_EMP, meses_EMP, directivo_EMP)
5	0	(numEmpleado_EMP, nombreEmpleado_EMP, meses_EMP, directivo_EMP)
6	0	(numEmpleado_EMP, nombreEmpleado_EMP, meses_EMP, directivo_EMP)
7	1000	(numEmpleado_EMP, nombreEmpleado_EMP, meses_EMP, directivo_EMP)
8	75	(numEmpleado_EMP, nombreEmpleado_EMP, meses_EMP, directivo_EMP)
9	0	(numEmpleado_EMP, nombreEmpleado_EMP, meses_EMP, directivo_EMP)
10	500	(numEmpleado_EMP, nombreEmpleado_EMP, meses_EMP, directivo_EMP)
11	0	(numEmpleado_EMP, nombreEmpleado_EMP, meses_EMP, directivo_EMP)
12	Prima	(numEmpleado_EMP, nombreEmpleado_EMP, meses_EMP, directivo_EMP)

//METODO CONSTRUCTOR calculaPRIMA

```

public int CalculaPRIMA(){
    Print("Ilmo.Sr.Director General:\n");
    Read (objeto_EMP);
    int prima = 0;

    if (Meses_trabajo >= 12){
        if(Directivo == "+") prima = 1000;
        else prima = 75;
    }else{
        if(Directivo == "+") prima = 1000;
        else;
    } //fin else

    System.out.println("La prima que corresponde al empleado " + numEmpleado_EMP
+ " de nombre " + nombreEmpleado_EMP + " es: " + prima);
    System.out.println("S.e.o.u.")
    return prima;

}

```

```

void objeto_EMP (int ne, nemp, int mt, char d){
    Scanner teclado = new Scanner(System.in);
    System.out.println("Il. Sr Director General:");
    System.out.println("Dame el código de empleado");
    This.numEmpleado_EMP = teclado.nextInt();
    System.out.println ("Dame el nombre de empleado");
    This.nombreEmpleado_EMP = teclado.nextLine();
    System.out.println("Dame los meses trabajados");
    This.meses_EMP= teclado.nextInt();
    System.out.println("Di si es directivo (+) o no lo es (-)");
    This.directivo_EMP = teclado.next().charAt(0);
} //fin objeto EMP

```

~

//Añado aquí los casos de prueba, mediante el uso de JUnit para la cobertura de la tarea 6, así como el código anterior para el proyecto JAVA de la tarea 5,

```

class pruebas{
    TestcalculaPRIMA(){
        objeto_EMP registro1 = new objeto_EMP (123, "Fernández", 009, "+");
        Int r = registro1.calculaPRIMA();
        assertEquals(1000, r);
    }

    TestcalculaPRIMA(){
        objeto_EMP registro1 = new objeto_EMP (456, "Fernando", 012, "-");
        Int r = registro1.calculaPRIMA();
        assertEquals(75, r);
    }
}

```


Se pide:

Tarea 5) Crea en el proyecto JAVA, la codificación en código Java del anterior pseudocódigo para cumplir

la especificación del problema anterior.

Tarea 6) Genera una prueba de test unitario y comprueba si se cumple la cobertura de los casos de prueba.

Tarea 7) Crea un documento PDF mostrando la resolución de las tareas y explicando todo el proceso que has seguido. El documento PDF debe llamarse: CasoCajaNegra-NombreAlumno.

3. Conclusión

4. Links

Indicamos los sitios web visitados para poder realizar las actividades.