Ingeniería del Software 2 – Segundo Ejercicio Teoría

Tercer Problema

Autores:

- **Aitor Jorge Jiménez.**
- **Óscar Veloso Delgado.**

Índice

1. Código.

Main.java

Main.java

```java
package C03_Testing_P3;

public class Main{

    public static void main(String[] args) {

        Adecuacion_Funcional adecuacionFun = new Adecuacion_Funcional(0,0,0,80);
        mantenibilidad mantenibilidad = new mantenibilidad(0,0,0,0,0,0,90);
        Certificado certificadofin = new Certificado(adecuacionFun, mantenibilidad);
        System.out.println(certificadofin.certificadofin());
    }
}
```

Adeucacion_Funcional.java

Adecuacion_Funcional.java

```java
package C03_Testing_P3;

public class Adecuacion_Funcional {

    int adecuacionFun;

    int completitudFun;
    int correccionFun;
    int pertinenciaFun;

    int rango;

    public Adecuacion_Funcional(int completitudFun, int correccionFun, int pertinenciaFun, int rango) {

        this.completitudFun = completitudFun;
        this.correccionFun = correccionFun;
        this.pertinenciaFun = pertinenciaFun;
        this.rango = rango;
    }

    // Metodos necesarios para adecuacion

    public int medicionCompletitudFun(int rango) {

        if(rango>=0 && rango<10) {

            completitudFun= 0;
        }else if(rango>=10 && rango<35) {

            completitudFun= 1;
        }else if(rango>=35 && rango<50) {

            completitudFun= 2;

        }else if(rango>=50 && rango<70) {

            completitudFun= 2;
        }else if(rango>=70 && rango<90) {

            completitudFun= 3;
        }else if(rango>=90 && rango<100) {
            completitudFun= 4;
        }

        return completitudFun;
    }

    public int medicionCorreccionFun(int rango) {

        if(rango>=0 && rango<10) {
            correccionFun= 0;
```

```
        }
        else if(rango>=10 && rango<35) {
            correccionFun= 1;
        }
        else if(rango>=35 && rango<50) {
            correccionFun= 2;
        }
        else if(rango>=50 && rango<70) {
            correccionFun= 2;
        }
        else if(rango>=70 && rango<90) {
            correccionFun= 3;
        }
        else if(rango>=90 && rango<100) {
            correccionFun= 4;
        }

        return correccionFun;
    }

    public int medicionPertinencia(int rango) {

        if(rango>=0 && rango<10) {

            pertinenciaFun= 0;
        }
        else if(rango>=10 && rango<35) {

            pertinenciaFun= 1;
        }
        else if(rango>=35 && rango<50) {

            pertinenciaFun= 2;
        }
        else if(rango>=50 && rango<70) {
            pertinenciaFun= 2;
        }
        else if(rango>=70 && rango<90) {
            pertinenciaFun= 3;
        }
        else if(rango>=90 && rango<100) {
            pertinenciaFun= 4;
        }

        return pertinenciaFun;
    }


    public int AdecuacionFuncional(int a, int b,int c) {


        int min= 10;
        int num= 0;
        int[] mediciones = (new int[] {a,b,c});

        for(int i =0; i<mediciones.length; i++) {

            num= mediciones[i];

            if (num<min) {
                min = num;

            }
        }
        adecuacionFun = min;
        return adecuacionFun;
    }
}
```

Mantenibilidad.java

mantenibilidad.java

```
package C03_Testing_P3;

public class mantenibilidad {

   int mantenibilidad;

   int modularidad;
   int reusabilidad;
   int analizabilidad;
   int capacidadMod;
   int capacidadPro;

   int rango;

   public mantenibilidad(int modularidad, int reusabilidad,int analizabilidad, int capacidadMod,
           int capacidadPro, int calidadFinal, int rango) {

       this.modularidad = modularidad;
       this.reusabilidad = reusabilidad;
       this.analizabilidad = analizabilidad;
       this.capacidadMod = capacidadMod;
       this.capacidadPro = capacidadPro;
       this.rango = rango;
   }

```

```java
public int medicionModularidad(int rango) {

        if(rango>=0 && rango<10) {
            modularidad= 0;
        }
        if(rango>=10 && rango<35) {
            modularidad= 1;
        }
        if(rango>=35 && rango<50) {
            modularidad= 2;
        }
        if(rango>=50 && rango<70) {
            modularidad= 2;
        }
        if(rango>=70 && rango<90) {
            modularidad= 3;
        }
        if(rango>=90 && rango<100) {
            modularidad= 4;
        }

        return modularidad;
    }

public int medicionReusabilidad(int rango) {

    if(rango>=0 && rango<10) {
        reusabilidad= 0;
    }
    if(rango>=10 && rango<35) {
        reusabilidad= 1;
    }
    if(rango>=35 && rango<50) {
        reusabilidad= 2;
    }
    if(rango>=50 && rango<70) {
        reusabilidad= 2;
    }
    if(rango>=70 && rango<90) {
        reusabilidad= 3;
    }
    if(rango>=90 && rango<100) {
        reusabilidad= 5;
    }

    return reusabilidad;
}
public int medicionAnalizabilidad(int rango) {

    if(rango>=0 && rango<10) {
        analizabilidad= 0;
    }
    if(rango>=10 && rango<35) {
        analizabilidad= 0;
    }
    if(rango>=35 && rango<50) {
        analizabilidad= 1;
    }
    if(rango>=50 && rango<70) {
        analizabilidad= 2;
    }
    if(rango>=70 && rango<90) {
        analizabilidad= 3;
    }
    if(rango>=90 && rango<100) {
        analizabilidad= 5;
    }

    return analizabilidad;
}
public int medicionCapadidadMod(int rango) {

    if(rango>=0 && rango<10) {
        capacidadMod= 0;
    }
    if(rango>=10 && rango<35) {
        capacidadMod= 1;
    }
    if(rango>=35 && rango<50) {
        capacidadMod= 2;
    }
    if(rango>=50 && rango<70) {
        capacidadMod= 3;
    }
    if(rango>=70 && rango<90) {
        capacidadMod= 4;
    }
    if(rango>=90 && rango<100) {
        capacidadMod= 5;
    }

    return capacidadMod;
}
public int medicionCapadidadPro(int rango) {

    if(rango>=0 && rango<10) {
        capacidadPro= 0;
    }
```

```
    if(rango>=10 && rango<35) {
        capacidadPro= 1;
    }
    if(rango>=35 && rango<50) {
        capacidadPro= 1;
    }
    if(rango>=50 && rango<70) {
        capacidadPro= 2;
    }
    if(rango>=70 && rango<90) {
        capacidadPro= 4;
    }
    if(rango>=90 && rango<100) {
        capacidadPro= 4;
    }

    return capacidadMod;
}


public int Mantenibilidad(int modularidad, int reusabilidad, int analizabilidad, int capacidadMod, int capacidadPro) {
    int min= 10;
    int num= 0;
    int[] mediciones = (new int[] {modularidad, reusabilidad, analizabilidad, capacidadMod, capacidadPro});

    for(int i =0; i<mediciones.length; i++) {
        num= mediciones[i];
        if (num<min) {
            min = num;
        }
    }
    mantenibilidad = min;
    return mantenibilidad;
}
}
```

Certificado.Java

Certificado.java

```
package C03_Testing_P3;

public class Certificado{


    mantenibilidad mantenibilidad;
    Adecuacion_Funcional adecuacionFun;

    public Certificado(Adecuacion_Funcional adecuacionFun, mantenibilidad mantenibilidad) {
        this.adecuacionFun = adecuacionFun;
        this.mantenibilidad = mantenibilidad;

    }
    public String certificadofin() {
        int certificadoFinal=0;
        String msgCertificado= "";

        int adecuacion = adecuacionFun.AdecuacionFuncional(adecuacionFun.medicionCompletitudFun(adecuacionFun.rango)
                ,adecuacionFun.medicionCorreccionFun(adecuacionFun.rango), adecuacionFun.medicionPertinencia(adecuacionFun.rango));

        int mantenibilidadFinal = mantenibilidad.Mantenibilidad(mantenibilidad.medicionModularidad(mantenibilidad.rango), mantenibilidad.medicionReusabilidad(mantenibilidad.rango),
                mantenibilidad.medicionAnalizabilidad(mantenibilidad.rango), mantenibilidad.medicionCapadidadMod(mantenibilidad.rango), mantenibilidad.medicionCapadidadPro(mantenibilidad.rango));

        if (adecuacion == 1) {

            if (mantenibilidadFinal ==1) {
                certificadoFinal = 1;
            }else if (mantenibilidadFinal ==2) {
                certificadoFinal = 1;
            }else if (mantenibilidadFinal ==3) {
                certificadoFinal = 1;
            }else if (mantenibilidadFinal ==4) {
                certificadoFinal = 1;
            }else if (mantenibilidadFinal ==5) {
                certificadoFinal = 1;
            }
        }else if (adecuacion == 2) {

            if(mantenibilidadFinal==1) {
                certificadoFinal = 1;
            }else if (mantenibilidadFinal ==2) {
                certificadoFinal = 2;
            }else if (mantenibilidadFinal ==3) {
                certificadoFinal = 2;
            }else if (mantenibilidadFinal ==4) {
                certificadoFinal = 2;
            }else if (mantenibilidadFinal ==5) {
                certificadoFinal = 2;
            }
        }else if (adecuacion==3) {

```

```
            if(mantenibilidadFinal==1) {
                certificadoFinal =2;
            }else if (mantenibilidadFinal ==2) {
                certificadoFinal = 2;
            }else if (mantenibilidadFinal ==3) {
                certificadoFinal = 3;
            }else if (mantenibilidadFinal ==4) {
                certificadoFinal = 3;
            }else if (mantenibilidadFinal ==5) {
                certificadoFinal = 3;
            }
        }else if (adecuacion == 4) {

            if (mantenibilidadFinal ==1) {
                certificadoFinal =3;
            }else if (mantenibilidadFinal==2) {
                certificadoFinal = 3;
            }else if (mantenibilidadFinal==3) {
                certificadoFinal = 3;
            }else if (mantenibilidadFinal==4) {
                certificadoFinal = 3;
            }else if (mantenibilidadFinal ==5) {
                certificadoFinal = 4;
            }
        }else if(adecuacion==5) {

            if (mantenibilidadFinal==1) {
                certificadoFinal = 3;
            }else if (mantenibilidadFinal==2) {
                certificadoFinal = 3;
            }else if (mantenibilidadFinal ==3) {
                certificadoFinal = 4;
            }else if (mantenibilidadFinal ==4) {
                certificadoFinal = 4;
            }else if (mantenibilidadFinal==5) {
                certificadoFinal = 5;
            }
        }

        if (certificadoFinal >= 3) {

            msgCertificado= "El software se ha certificado con una calidad de "+certificadoFinal;
        }else {
            msgCertificado = "No se ha podido certificar el software" +certificadoFinal;
        }

        return msgCertificado;
    }
}

```

2. Variables

Variable	Tipo	Rango
adecuacionFun	int	(-32768,32767)
completitudFun	int	(-32768,32767)
correccionFun	int	(-32768,32767)
pertinenciaFun	int	(-32768,32767)
rango	int	(-32768,32767)
mantenibilidad	int	(-32768,32767)
modularidad	int	(-32768,32767)
reusabilidad	int	(-32768,32767)
analizabilidad	int	(-32768,32767)
capacidadMod	int	(-32768,32767)
capacidadPro	int	(-32768,32767)

3. Casos de prueba máximos.

- Rango Mediciones Adecuación = 5
- Variantes completitud Funcional = 5
- Variantes corrección funcional = 5
- Variantes pertinencia Funcional= 5

- Rango mediciones mantenibilidad =5
- Variantes modularidad = 5
- Variantes reusabilidad = 5
- Variante analizabilidad = 5
- Variante Capacidad de ser modificado = 5
- Variante Capacidad de ser probado = 5

- Variantes finales de A.Funcional = 5
- Variantes finales de Mantenibilidad = 5

- Total = (5x5x5x5) x (5x5x5x5x5x5) x (5x5) = 244.140.625 casos de prueba diferentes.

4. Casos de prueba para cumplimentar each use.

- Certificadofin()

Adecuación	Mantenibilidad	Certificado
1	0	Null
2	2	False
4	4	True

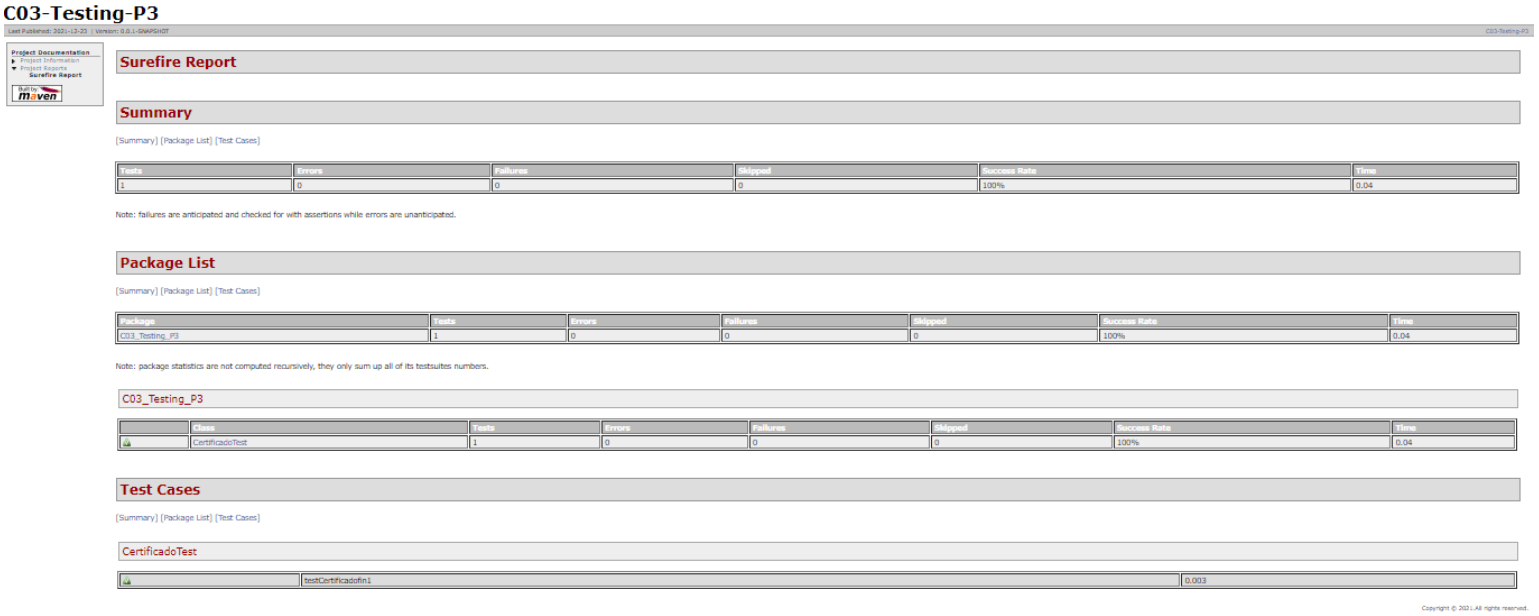
5. Casos de prueba para alcanzar cobertura pairwaise.

Se ha cubierto el 88,6%, ya que con el valor de la tabla de adecuación no se podrá acceder al valor 5 de esta debido a que su valor mínimo siempre va a ser 4.

RANGO Adecuación	RANGO Mantenibilidad	ADECUACION	MANTENIBILIDAD	CERTIFICADO
[0,10)- 0	**[0-10) - 0**	0	0	**False**
[10-35) - 22	**[10-35) - 15**	1	0	**False**
[35-50) - 40	**[35–50) - 40**	1	1	**False**
[50–70) - 55	**[50–70) - 60**	2	2	**False**
[70-90) - 80	**[70-90) - 85**	3	3	**True**
[90-100) - 91	**[90-100) - 95**	4	4	**True**
[35-50) - 40	**[70-90) - 70**	1	3	**False**

6. Comentar resultados apartados 4 y 5

Una vez realizado el Testing en Eclipse con Junit, hemos obtenido una cobertura del 88,6%, es decir, hemos cubierto 952 instrucciones de las 1074 que contiene. Se ha generado el siguiente informe con surefire:



C03-Testing-P3

Project Documentation
Project Information
Project Reports
Surefire Report

Surefire Report

Summary

[Summary] [Package List] [Test Cases]

Tests	Errors	Failures	Skipped	Success Rate	Time
1	0	0	0	100%	0.04

Note: failures are anticipated and checked for with assertions while errors are unanticipated.

Package List

[Summary] [Package List] [Test Cases]

Package	Tests	Errors	Failures	Skipped	Success Rate	Time
C03_Testing_P3	1	0	0	0	100%	0.04

Note: package statistics are not computed recursively, they only sum up all of its testsuites numbers.

C03_Testing_P3

	Class	Tests	Errors	Failures	Skipped	Success Rate	Time
	CertificadoTest	1	0	0	0	100%	0.04

Test Cases

[Summary] [Package List] [Test Cases]

CertificadoTest

	testCertificadofin1	0.003

Copyright © 2021. All rights reserved.