



Soluciones Ejercicios Tema 2

Germán Moltó

gmolto@dsic.upv.es

Estructuras de Datos y Algoritmos

Escuela Técnica Superior de Ingeniería Informática

Universidad Politécnica de Valencia

1

BonoMetro con Excepciones (I)

```
public class BonoMetro{
    private int saldo;
    private final static int SALDO_INICIAL = 10;
    public BonoMetro(int saldo){ this.saldo = saldo; }
    public BonoMetro(){
        this(SALDO_INICIAL);
    }
    public int getSaldo(){ return this.saldo;}
    public void setSaldo(int saldo){this.saldo = saldo;}
    public String toString(){return "BonoMetro de " + this.saldo + " viajes.";}
    public void fichar() throws SaldoAgotadoException{
        if (saldo > 0) saldo--;
        else throw new SaldoAgotadoException("Ya no quedan viajes disponibles");
    }
}
2
```

BonoMetro con Excepciones (II)

```
public class SaldoAgotadoException extends Exception{
    public SaldoAgotadoException(){
        super();
    }
    public SaldoAgotadoException(String msg){
        super(msg);
    }
}
```

3

BonoMetro con Excepciones (III)

```
public class TestBonoMetro{
    public static void main(String args[]){
        BonoMetro b = new BonoMetro(3);
        for (int i = 1 ; i <= 4 ; i++){
            System.out.println("Fichando el viaje #" + i);
            try{
                b.fichar();
            }catch(SaldoAgotadoException ex){
                System.out.println("Problema:" + ex.getMessage());
            }
        }
    }
}
```

4

Solución: Transferencia de Ficheros

```
public class TestCopyViaFTP
{
    public static void main(String args[]){
        String hostName = "fileserv.dsic.upv.es";
        String localFilePath = "/tmp/data";
        boolean ok = false;
        int nfallos = 0;
        while (!ok && (nfallos < 3))
            try {
                CopyViaFTP.copyTo(hostName, localFilePath);
                ok = true;
            }catch(UnableToTransferException ex) {
                nfallos++;
                System.err.println(nfallos + " intentos. Error: " + ex);
            }
        }
}
```



Módulo de Autorización (I)

```
public static void grantAccess(String username, String password) throws
AccessDeniedException{
    boolean granted = true;
    try{
        AuthModule.check(username,password);
        System.out.println("Successfully authorized!");
    }catch(InvalidUserException e1){
        granted = false; System.out.println("Unknown user name:" + username);
    } catch(InvalidPasswordException e2){
        granted = false; System.out.println("Invalid password:" + password);
    }catch(ExpirationDeadlineException e3){ //No es obligatorio capturarla
        System.out.println("Warning:Your password will expire in a few days.");
    }
    if (!granted) throw new AccessDeniedException();
}
```



Excepción en Base de Datos

```
public class TestBaseDeDatos {
    public static void main(String args[]){
        int contador = 0; boolean hayError = false;
        BaseDeDatos bd = new BaseDeDatos("MiBD");
        do {
            try{
                bd.inicializar(); hayError = false;
            }catch(ImposibleAbrirBD e){
                System.out.println("Imposible abrir la Base de Datos");
                contador++; hayError = true;
            }
        }while( contador < 3 & hayError );}}
}
```



Solución Ejercicio Excepciones

► Solución I (sencilla pero puede ser desaconsejable)

```
public static void main(String [] args){
    try{
        String leido = leer();
        int denominador = Integer.parseInt(leido);
        int cociente = 42/denominador;
        System.out.println("El cociente de 42/" + denominador + " es: " + cociente);
        actualizaInicio(denominador);
    }catch(Exception ex) {
        System.err.println("Ocurrió un error:" + ex.getMessage());
    }
}
```



Solución Ejercicio Excepciones (II)

- ▶ Solución 2 (menos sencilla, pero mayor control)

```
public static void main(String [] args){
    int denominador, cociente;
    String leído = leer();
    try{
        denominador = Integer.parseInt(leído);
        cociente = 42/denominador;
    }catch(NumberFormatException ex){
        System.err.println("El valor introducido no es un numero:" + ex.getMessage());
    }catch(ArithmeticException ex) {
        System.err.println("Excepcion aritmética, posible división por 0:" + ex.getMessage());
    }
    System.out.println("El cociente de 42/"+denominador+" es: "+cociente);
    try{
        actualizaInicia(denominador);
    }catch(ArrayIndexOutOfBoundsException ex){
        System.err.println("Error al escribir fuera del vector:" + ex.getMessage());
    }catch(IllegalArgumentException ex) {
        System.err.println("Argumento más especificado al método inicia:" + ex.getMessage());
    }
} ▶ 9
```

Solución: Carnet por Puntos (1/3)

1. Definir excepción:

```
public class RetiradaInmediataCarnet extends Exception {
    public RetiradaInmediataCarnet(String msg){
        super(msg);
    }
    public RetiradaInmediataCarnet(){ super(); }
}
```

▶ 10

Solución: Carnet por Puntos (2/3)

2. Modificar quitarPuntos

```
public final void quitarPuntos ( int penalizacion ) throws
    RetiradaInmediataCarnet
{
    puntos -= penalizacion;
    if ( this.puntos <= 0)
        throw new RetiradaInmediataCarnet("Agotados los 12
    puntos: saldo actual = " + puntos);
}
```

▶ 11

Solución: Carnet por Puntos (3/3)

3. Modificar el método multar

```
public static void multar(CarnetDeConducir c, Scanner teclado)
{
    System.out.println("Introduce la penalización:");
    int penalizacion = teclado.nextInt();
    try{
        c.quitarPuntos(penalizacion);
    } catch (RetiradaInmediataCarnet e) {
        System.out.println(e);
    }
}
```

▶ 12