1 - Ejercicio 1 *(2 puntos)*

Implementa una clase ***Trabajador*** con los siguientes atributos y métodos:

- ✓ **Atributos privados**: nombre, edad, categoría, antigüedad.
- ✓ **Constantes static públicas.** Determinan los diferentes tipos de categorías y antigüedad de un empleado disponibles en el sistema:
  - ➢ CAT_EMPLEADO. Valor 0
  - ➢ CAT_ENCARGADO. Valor 1
  - ➢ CAT_DIRECTIVO. Valor 2
  - ➢ ANT_NOVATO. Valor 0
  - ➢ ANT_MADURO. Valor 1
  - ➢ ANT_EXPERTO. Valor 2
- ✓ ***Trabajador(String nombre, int edad, int categoria, int antigüedad)***. Constructor de la clase que inicializa los atributos de la misma. Comprobará que la categoría y la antigüedad sean correctos, si no lo son se lanzará la excepción ***IllegalArgumentException***. ***(0.75 puntos)***
- ✓ Métodos públicos **consulta/cambia** de cada uno de los atributos. Se debe comprobar la validez de los atributos y lanzar la excepción correspondiente si hay algún valor incorrecto. ***(0.5 puntos)***
- ✓ ***public double calcularSueldo()***. Devuelve el sueldo del empleado calculado a partir de su antigüedad y categoría profesional. La forma de calcular el sueldo del empleado será de acuerdo a la siguiente tabla ***(0.75 puntos)***:

Sueldo base	607 €
Empleado **Encargado** **Directivo**	+15% sueldo base +35% sueldo base +60% sueldo base
Novato **Maduro** **Experto**	+150 € +300 € +600 €

RESPUESTA

```
public class Trabajador {

    // constantes públicas estáticas
    public static final int CAT_EMPLEADO = 0;
    public static final int CAT_ENCARGADO = 1;
    public static final int CAT_DIRECTIVO = 2;
    public static final int ANT_NOVATO = 0;
    public static final int ANT_MADURO = 1;
    public static final int ANT_EXPERTO = 2;
    //atributos privados
    private String nombre;
    private int edad;
    private int categoria;
    private int antigüedad;

    /**
     * Crea el objeto Trabajador, si existe algún error en la categoría o la
     * antigüedad se lanza la excepción IllegalArgumentException. Como el
     * enunciado no lo expecifica, no es necesario comprobar el resto de parámetros
     * @param nombre
     * @param edad
     * @param categoria
     * @param antigüedad
     */
    public Trabajador(String nombre, int edad, int categoria, int antigüedad) {
        if (comprobarCategoria(categoria) && comprobarAntigüedad(antigüedad)) {
            this.nombre = nombre;
            this.edad = edad;
```

```
            this.categoria = categoria;
            this.antigüedad = antigüedad;
        }
        else{
           throw new IllegalArgumentException("Categoría o antigüedad incorrectos");
        }
    }

    public int getAntigüedad() {
        return antigüedad;
    }

    public void setAntigüedad(int antigüedad) {
        if(comprobarAntigüedad(antigüedad))
            this.antigüedad = antigüedad;
        else{
            throw new IllegalArgumentException("Antigüedad incorrecta");
        }
    }

    public int getCategoria() {
        return categoria;
    }

    public void setCategoria(int categoria) {
        if(comprobarCategoria(categoria))
            this.categoria = categoria;
        else
            throw new IllegalArgumentException("Categoria incorrecta");
    }

    public int getEdad() {
        return edad;
    }

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

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public double calcularSueldo() {
        double sueldo=0;
        switch(this.categoria){
            case CAT_EMPLEADO:
                sueldo+=sueldo*0.15;
                break;
            case CAT_ENCARGADO:
                sueldo+=sueldo*0.35;
                break;
            case CAT_DIRECTIVO:
                sueldo+=sueldo*0.60;
                break;
        }
        switch(this.antigüedad){
            case ANT_NOVATO:
                sueldo+=150;
                break;
            case ANT_MADURO:
                sueldo+=300;
                break;
            case ANT_EXPERTO:
                sueldo+=600;
                break;
        }
        return sueldo;
    }
    private boolean comprobarCategoria(int categoria){
        boolean check=true;
        if(categoria<CAT_EMPLEADO || categoria>CAT_DIRECTIVO)
            check=false;
```

```
        return check;
    }
    private boolean comprobarAntigüedad(int antigüedad){
        boolean check=true;
        if(antigüedad<ANT_NOVATO || antigüedad>ANT_EXPERTO)
            check=false;
        return check;
    }
}
```

Ejercicio 2 (1.5 puntos)

- ✓ Implementa dos funciones para obtener, la parte entera y la parte decimal de un número en punto flotante (***double***). La definición de las funciones es como sigue:
- ✓ ***int getParteEntera(double numero)*** ***(0.5 puntos)***
- ✓ ***int getParteDecimal(double numero)*** ***(0.5 puntos)***
- ✓ Internamente, estas dos funciones convierten el número ***double*** a cadena de caracteres para obtener la parte correspondiente.
- ✓ Una vez obtenido el valor buscado, la cadena se convertirá al tipo de retorno indicado.

Además el programa principal (***main***) pedirá al usuario que introduzca un número por teclado y posteriormente mostrará un menú por pantalla en el que se pregunte si desea obtener la parte entera o decimal del número introducido. El programa principal ***main*** se ejecutará hasta que el usuario introduzca la opción adecuada para salir ***(0.5 puntos).***

RESPUESTA

```
import java.util.Scanner;

/**
 *
 * @author jose
 */
public class Ejercicio2 {

    public static int getParteEntera(double numero){
        String snumero = String.valueOf(numero);
        int posicion_coma = snumero.indexOf('.');
        int parte_entera = Integer.parseInt(snumero.substring(0, posicion_coma));
        return parte_entera;
    }
    public static int getParteDecimal(double numero){
        String snumero = String.valueOf(numero);
        int posicion_coma = snumero.indexOf('.');
        int parte_decimal = Integer.parseInt(snumero.substring(posicion_coma+1));
        return parte_decimal;
    }
    public static void main(String [] args){
        Scanner entrada_teclado = new Scanner(System.in);
        int opcion_menu=0;
        double numero=0;
        do{
            //lectura de un valor Double válido
            boolean numero_valido=false;
            do{
              System.out.println("Introduzca un número double válido: ");
              if(entrada_teclado.hasNextDouble()){
                  numero = entrada_teclado.nextDouble();
                  numero_valido=true;
              }
              else{
                  System.out.println("Número erroneo introducido. Vuelva a intentarlo");
                  entrada_teclado.next();
              }
            }while(!numero_valido);
            //mostramos el menú de opciones
            System.out.println("¿Qué desea hacer con el número?");
```

```
            System.out.println("1.Obtener la parte entera.");
            System.out.println("2.Obtener la parte decimal");
            System.out.println("3.Salir del programa");
            //leemos una opción válida
            numero_valido=false;
            do{
              System.out.println("Introduzca opción: ");
              if(entrada_teclado.hasNextInt()){
                  opcion_menu = entrada_teclado.nextInt();
                  if(opcion_menu>=1 && opcion_menu<=3)
                      numero_valido=true;
                  else
                      System.out.println("Debe introducir un valor entre 1 y 3");
              }
              else{
                  System.out.println("Número erroneo introducido. Vuelva a intentarlo");
                  entrada_teclado.next();
              }
            }while(!numero_valido);
            //Según opción introducida hacemos la acción deseada
            switch(opcion_menu){
                case 1:
                    System.out.println("La      parte      entera      de      "+numero+"      es
"+getParteEntera(numero));
                    break;
                case 2:
                    System.out.println("La      parte      decimal      de      "+numero+"      es
"+getParteDecimal(numero));
                    break;

            }
        }while(opcion_menu!=3);
    }
}
```

Ejercicio 3 *(2 puntos)*

Realizaremos un programa que calcule el valor de los **coeficientes binomiales**. Los coeficientes binomiales son ampliamente utilizados en distintas ramas de las matemáticas, como por ejemplo la estadística. Se representan como un binomio (dos valores) entre paréntesis (similar a una fracción pero sin la barra horizontal). La fórmula para calcular el coeficiente binomial de dos valores **n** y **k**, lo representaremos como **(n,k)** es la siguiente:

$$(n, k) = \frac{n!}{k!\,(n-k)!}$$

donde **!** representa el factorial y **n** y **k** mayores o iguales que **0**.

Para realizar este ejercicio sigue los siguientes pasos:

a) Se implementará una función que calcule el factorial de un número. El número se pasará por parámetro y retornará el factorial calculado. La declaración de la función será como sigue:

(0.75 puntos)

public static int factorial(int n)

El factorial de un número **n**, **n!** Se calcula de la siguiente manera:

- 1, si n=0.
- n*n-1....*1, si n>0.

Por ejemplo, el factorial de **3**, `3! = 3*2*1 = 6`.

b) Se implementará una función que calcule el coeficiente binomial. Los valores de **n** y **k** se pasarán por parámetro y la función devolverá el coeficiente calculado. Se utilizará la función ***factorial()*** para realizar este cálculo. La declaración de la función será como sigue ***(0.5 puntos)***:

public static int coeficienteBinomial(int n, int k)

c) El programa principal (***main***) deberá pedir al usuario que introduzca los valores de **n** y **k** (mayor o igual que 0). Se calculará el valor del binomio llamando a la función **coeficienteBinomial()** y se mostrarán los resultado obtenidos por pantalla. ***(0.75 puntos)***.

RESPUESTA

```
import java.util.Scanner;

/**
 *
 * @author jose
 */
public class Ejercicio3 {

    public static int factorial(int n) {
        int fact = 1;
        int contador = 2;
        while (contador <= n) {
            fact *= contador++;
        }
        return fact;
    }

    public static int coeficienteBinomial(int n, int k) {
        return factorial(n) / (factorial(k) * factorial(n - k));
    }

    public static void main(String[] args) {
        Scanner entrada_teclado = new Scanner(System.in);
        boolean numero_valido = false;
        int n = 0, k = 0;
        do {
            System.out.println("Introduzca N: ");
            if (entrada_teclado.hasNextInt()) {
                n = entrada_teclado.nextInt();
                if (n>0) {
                    numero_valido = true;
                } else {
                    System.out.println("Debe introducir un valor mayor que 0");
                }
            } else {
                System.out.println("Número erroneo introducido. Vuelva a intentarlo");
                entrada_teclado.next();
            }
        } while (!numero_valido);
        do {
            System.out.println("Introduzca K: ");
            if (entrada_teclado.hasNextInt()) {
                k = entrada_teclado.nextInt();
                if (n>0) {
                    numero_valido = true;
                } else {
                    System.out.println("Debe introducir un valor mayor que 0");
                }
            } else {
                System.out.println("Número erroneo introducido. Vuelva a intentarlo");
                entrada_teclado.next();
            }
        } while (!numero_valido);
        System.out.println("El coeficiente binomial (n,k)="+coeficienteBinomial(n,k));
    }
}
```

Ejercicio 4 (0.75 puntos)

Implementa un algoritmo en el que dado un entero n > 1 leído por teclado, calcule e imprima los elementos correspondientes a la **Conjetura de Ullman** (en honor al matemático *S. Ullman*). La conjetura consiste en lo siguiente:

- ✓ Empieza con cualquier entero positivo.
- ✓ Si es par, se divide entre 2; si es impar se multiplica por 3 se agrega 1.
- ✓ Se itera hasta obtener el número 1.

Al final se obtendrá el número 1, independientemente del entero inicial. Por ejemplo, cuando el entero inicial es 26, la secuencia será:

26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

RESPUESTA

```
import java.util.Scanner;

/**
 *
 * @author jose
 */
public class Ejercicio4 {
    public static void main(String [] args){
        Scanner entrada_teclado = new Scanner(System.in);
        boolean numero_valido = false;
        int n = 0;
        do {
            System.out.println("Introduzca un número positivo: ");
            if (entrada_teclado.hasNextInt()) {
                n = entrada_teclado.nextInt();
                if (n>0) {
                    numero_valido = true;
                } else {
                    System.out.println("Debe introducir un valor mayor que 0");
                }
            } else {
                System.out.println("Número erroneo introducido. Vuelva a intentarlo");
                entrada_teclado.next();
            }
        } while (!numero_valido);
        //algoritmo para la conjetura
        System.out.print(n+", ");
        while(n!=1){
            if(n%2==0)
                n/=2;
            else
                n=n*3+1;
           System.out.print(n+", ");
        }
    }
}
```

Ejercicio 5 (0.75 puntos)

Implemente una función que sirva para cifrar un texto con el conocido método de César. El criptosistema consiste en el desplazamiento de 3 caracteres en la posición del carácter a cifrar, es decir, la A se sustituye por la D, la B por la E, …, la X por la A, la Y por la B y la Z por la C. Por simplicidad, supondremos que el texto a cifrar solo contiene caracteres alfabéticos. Por tanto el ejercicio consiste en implementar la siguiente función:

public String cifradoCesar(String cadenaACifrar)

La función recibe como parámetro la cadena a cifrar y devuelve un objeto String con la cadena cifrada mediante el sistema de Cesar.

```
public class Ejercicio5 {
    public static String cifradoCesar(String cadenaACifrar){
        String abc="ABCDEFGHIJKLMNÑOPQRSTUVWXYZ";
        int pos_c;
        String cadena_cifrada="";
        for(int i=0;i<cadenaACifrar.length();++i){
            //obtenemos la posición del caracter a cifrar en el abcedario
```

```
            pos_c=abc.indexOf(cadenaACifrar.charAt(i));
            //obtenemos el caracter tres posiciones avanzado, cuidando el extremo
            cadena_cifrada+=abc.charAt((pos_c+3)%abc.length());
        }
        return cadena_cifrada;
    }
    public static void main(String [] args){
        String abc="ABCDEFGHIJKLMNÑOPQRSTUVWXYZ";
        System.out.println("Cifrado Cesar: "+cifradoCesar(abc));
    }
}
```