

Entornos de Desarrollo

Bloque 2

Tema 1: Estructura básica de un programa en Java. I/O. Tipos básicos de datos. Operadores.

Soluciones

Ejercicios propuestos:

2.1.6.1. Crea un programa llamado *FormattedDate* con una clase con el mismo nombre dentro. El programa pedirá al usuario que introduzca el día, el mes y el año de su nacimiento (todos los valores son enteros). Entonces, se sacará por pantalla la fecha de nacimiento con el siguiente formato *d/m/a*. Por ejemplo, si el usuario teclea día = 7, mes =11, año =1990, el programa sacará *7/11/1990*.

```
import java.util.Scanner;

public class FormattedDate
{
    public static void main(String[] args)
    {
        int day, month, year;
        Scanner sc;

        sc = new Scanner(System.in);

        System.out.println("Enter you day of birth");
        day = sc.nextInt();
        System.out.println("Enter you month of birth");
        month = sc.nextInt();
        System.out.println("Enter you year of birth");
        year = sc.nextInt();

        System.out.println(day + "/" + month + "/" + year);

        sc.close();
    }
}
```

2.1.6.2. Crea un programa llamado *GramOunceConverter* que transforme de gramos a libras. El programa pedirá al usuario que introduzca un peso en gramos (un número entero), y se mostrará el correspondiente peso en libras (número real). 1 libra = 28,3495 gramos.

```
import java.util.Scanner;

public class GramOunceConverter
{
    // Constant to convert from grams to ounces (1 ounce = 28.3495 grams)
    static final float GRAMS_OUNCES = 28.3495f;

    public static void main(String[] args)
    {
        int grams;
        double ounces;
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter grams:");
        grams = sc.nextInt();
        ounces = grams / GRAMS_OUNCES;
        // Format the output with 2 decimal digits
        System.out.printf("%d grams are %.2f ounces", grams, ounces);
    }
}
```

2.1.6.3. Crea un programa llamado *NumbersStrings*. Este programa debe pedir al usuario que introduzca 4 números que guardará en 4 variables *String*. Entonces, el programa concatenará el primer par de números formando un número entero. Y concatenará el segundo par de números para formar otro entero. Después de esto se sumarán los dos números resultantes y se sacará el resultado de la suma. Por ejemplo si se introduce 23,11,45 y 122, entonces el programa creará el 2311 y el 45112. Y la suma resultará 47423.

```
import java.util.Scanner;

public class NumbersStrings
{
    public static void main(String[] args)
    {
        String numberString1, numberString2,
            numberString3, numberString4;

        int number1, number2, result;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter 4 numbers:");
        numberString1 = sc.nextLine();
        numberString2 = sc.nextLine();
        numberString3 = sc.nextLine();
        numberString4 = sc.nextLine();

        // Join first two numbers
        number1 = Integer.parseInt(numberString1 + numberString2);
        // Join last two numbers
        number2 = Integer.parseInt(numberString3 + numberString4);
        // Add both resulting numbers
        result = number1 + number2;

        System.out.println("Result = " + result);
    }
}
```

2.1.6.4. Crea un programa llamado *CircleArea* que defina una constante float llamada **PI** con el valor **3.14159**. El programa pedirá al usuario que introduzca el radio de un círculo y se sacará por pantalla el área de dicho círculo (**PI*** radio *radio). Este área se mostrará por pantalla con 2 decimales.

```
import java.util.Scanner;

public class CircleArea
{
    // Constant to store a value of PI
    static final float PI = 3.14159f;

    public static void main(String[] args)
    {
        float radius, area;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter circle radius:");
        radius = sc.nextFloat();

        area = PI * radius * radius;

        System.out.printf("Area = %.2f", area);

    }
}
```