

# Programación orientada a objetos

---

EJERCICIOS PRÁCTICOS II

# Pirámide

---

- Reto:
  - Realizar un programa que muestre por pantalla la siguiente pirámide:  
1  
22  
333  
4444  
55555

# Pirámide - resuelto

---

```
private static void piramide() {  
    System.out.println("-----PIRÁMIDE-----");  
    for (int i = 1; i <= 5; i++) {  
        for (int j = 1; j <= i; j++) {  
            System.out.print(i);  
        }  
        System.out.println();  
    }  
}
```

# Cuenta atrás

---

- Reto:
  - Hacer un fragmento de código que cree un *array* unidimensional de 50 posiciones y lo inicialice con los siguientes valores, para después imprimirlo:  
50, 49, 48, 47, ...3, 2, 1

# Cuenta atrás - resuelto

---

```
private static void cuentaAtras() {  
    System.out.println("-----CUENTA ATRÁS-----");  
    int[] array = new int[50];  
  
    for (int i = 0; i <= 49; i++) {  
        array[i] = 50 - i;  
    }  
  
    for (int i = 0; i <= 49; i++) {  
        System.out.print(array[i] + " ");  
    }  
  
    System.out.println();  
}
```

# Clase punto

---

- Reto:
  - Construir la clase Punto.
  - Un punto posee unas coordenadas en un plano bidimensional ( $x$  e  $y$ ).
  - Un punto se puede inicializar de dos modos:
    - Sin argumentos → el punto estará en el origen de coordenadas.
    - Con dos argumentos → coordenadas del punto.
  - Un punto tendrá el siguiente comportamiento:
    - Se podrá trasladar:
      - Indicando el desplazamiento en ambas coordenadas.
      - Indicando el desplazamiento en  $x$ , quedando  $y$  inalterada.
    - El punto devolverá el valor de su coordenada  $x$ .
    - El punto devolverá una copia de sí mismo.
    - El punto podrá informar de su posición mediante un mensaje de consola

# Clase punto - resolución (I)

---

```
//Clase Punto
public class Punto {
    //Atributos de la clase
    private int x, y;

    //Constructor por defecto
    public Punto() {
        x = 0;
        y = 0;
    }

    //Constructor con dos argumentos
    public Punto(int coordenadaX, int coordenadaY) {
        x = coordenadaX;
        y = coordenadaY;
    }
}
```

# Clase punto - resolución (II)

---

*//Método que desplaza un punto, modificando los atributos*

```
public void desplazar(int desplazamientoX, int desplazamientoY) {  
    x += desplazamientoX; //x = x+desplazamientoX;  
    y += desplazamientoY; //y = y+desplazamientoY;  
}
```

*//Método que desplaza un punto, modificando uno de sus atributos*

```
public void desplazar(int desplazamientoX) {  
    x += desplazamientoX; //x = x+desplazamientoX;  
}
```

```
/*  
 * Método que imprime por pantalla la posición del punto.  
 * Nótese que no tiene argumentos, trabaja con el propio punto, el argumento implícito.  
 */  
public void posicion() {  
    System.out.println("El valor de la coordenada x es: " + x + "\nEl valor de la coordenada y es: " +  
y);  
}
```



# Clase punto - resolución (III)

---

```
//Método que devuelve la coordenada X del punto.
public int coordenadaX() {
    return x;
}

/*
 * Método que devuelve una copia del punto.
 * El código comentado es equivalente al programado
 */
public Punto copia() {

    Punto c = new Punto(x, y);
    return c;
    //return new Punto (x,y);
}
```

# Clase punto - resolución (IV)

---

```
public static void main(String[] args) {  
  
    Punto puntoUno = new Punto();  
    puntoUno.posicion();  
    Punto puntoDos = new Punto(3,7);  
    puntoDos.posicion();  
  
    System.out.println("\n");  
  
    puntoUno.desplazar(7);  
    puntoDos.desplazar(2,4);  
    puntoUno.posicion();  
    puntoDos.posicion();  
  
    int coordenadaX = puntoUno.coordenadaX();  
    System.out.println("\nLa coordenada x del punto uno es: " + coordenadaX);  
  
    System.out.println("\n");  
  
    Punto copia = puntoUno.copia();  
    copia.posicion();  
    puntoUno.posicion();  
}
```

# Ejercicios extra...

---

# Matrículas

---

- Reto:
  - Realizar un programa que dado el número de créditos en primera, segunda, tercera y cuarta matrícula o sucesivas, imprima el importe a pagar de las tasas universitarias de la Universidad.
- Pista:
  - Para el cálculo de las tasas universitarias se definirán cuatro constantes que indicarán el precio en euros de:
    - un crédito en primera matrícula (26,81 euros),
    - segunda (47,61 euros),
    - tercera (89,28 euros)
    - o sucesivas (119,04 euros).
  - En cuanto al número de créditos de cada tipo de matrícula, se almacenarán en variables, asignando los valores que se deseen.

# Matrículas - resuelto

---

```
private static void precios() {  
    System.out.println("-----PRECIOS-----");  
    final double PRIMERA = 26.81;  
    final double SEGUNDA = 47.61;  
    final double TERCERA = 89.28;  
    final double CUARTA = 119.04;  
  
    int uno = 30;  
    int dos = 10;  
    int tres = 12;  
    int cuatro = 10;  
  
    double total = PRIMERA * uno + SEGUNDA * dos + TERCERA * tres + CUARTA * cuatro;  
    System.out.print(total + "\n");  
}
```

# Matrículas con *arrays*

---

- Reto:
  - Repetir el ejercicio anterior, metiendo esta vez los precios (la información que antes aparecía en constantes) en un *array* y calcular la matrícula a pagar accediendo a la posición correspondiente.

# Matrículas con *arrays* - resuelto

---

```
private static void preciosArray() {  
    System.out.println("-----PRECIOS ARRAY-----");  
    double[] precios = new double[4];  
  
    precios[0] = 26.81;  
    precios[1] = 47.61;  
    precios[2] = 89.28;  
    precios[3] = 119.04;  
  
    int uno = 30;  
    int dos = 10;  
    int tres = 12;  
    int cuatro = 10;  
  
    double total2 = precios[0] * uno + precios[1] * dos + precios[2] * tres + precios[3] * cuatro;  
    System.out.print(total2+ "\n");  
}
```

# Matriz

---

- Reto:
  - Hacer un fragmento de código que cree e imprima un *array* bidimensional de 5x6 y lo inicialice a los siguiente valores, para después imprimirlo:

```
1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
19 20 21 22 23 24
25 26 27 28 29 30
```



# Matriz - resuelto

---

```
private static void matriz() {  
    System.out.println("-----MATRIZ-----");  
    int[][] matriz = new int[5][6];  
  
    for (int i = 0; i < 5; i++) {  
        for (int j = 0; j < 6; j++) {  
            matriz[i][j] = j + 1 + 6 * i;  
            System.out.print(matriz[i][j] + " ");  
        }  
        System.out.println();  
    }  
}
```

# Primos

---

- Reto:
  - Imprimir por pantalla todos los números primos comprendidos entre el 3 y el 100.
- Reto dos:
  - Imprimir por pantalla todos los números primos comprendidos entre el 3 y un número aleatorio.

# Primos - resuelto

---

```
private static void primos() {  
    System.out.println("-----PRIMOS-----");  
    boolean isPrimo;  
  
    System.out.println("Números primos comprendidos entre 3 y 100: ");  
  
    for (int numero = 3; numero < 100; numero += 2) {  
        isPrimo = true;  
        for (int i = 3; i < numero / 2 && isPrimo; i++) {  
            if (numero % i == 0)  
                isPrimo = false;  
        }  
        if (isPrimo) {  
            System.out.println(numero + " - ");  
        }  
    }  
}
```