



USMP

FACULTAD DE
INGENIERÍA Y ARQUITECTURA

Semestre Académico 2022-I



ASIGNATURA

Algoritmo y Estructura de Datos I

Docente: Ing. Juan Puerta Arce

**Escuela Profesional de
Ingeniería de Computación y Sistemas**

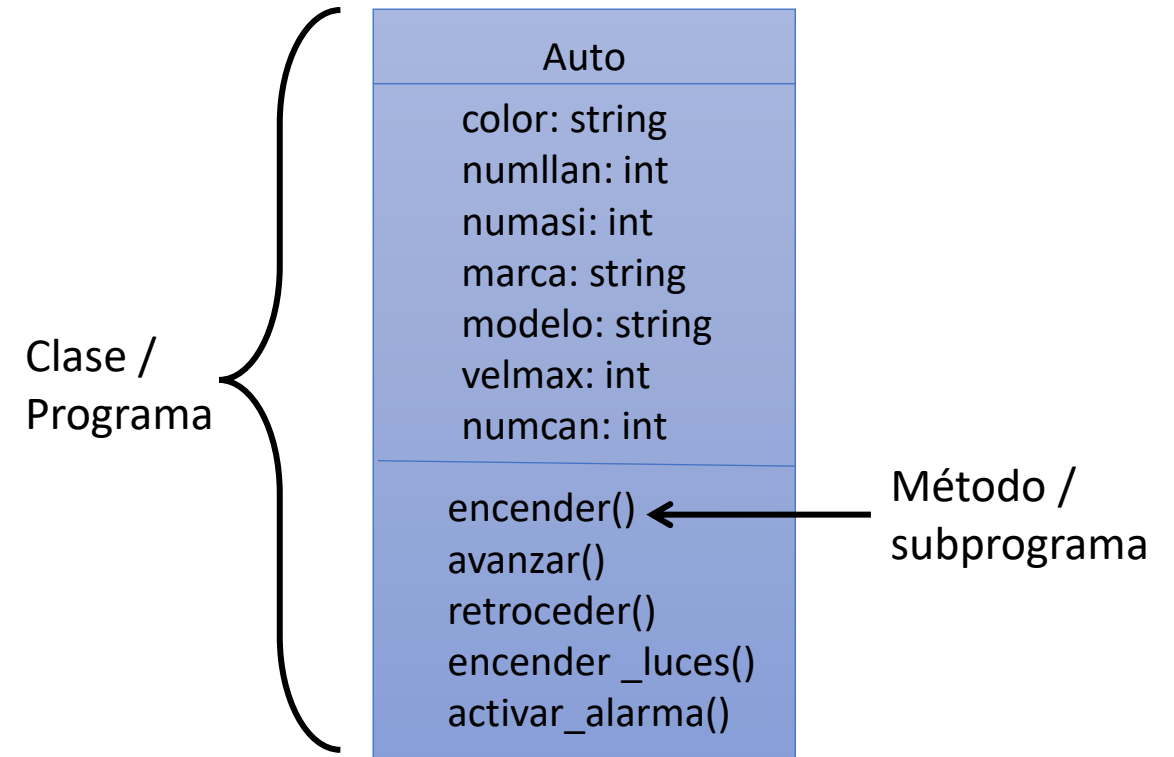
Semana 2

- Primera sesión
 - Métodos: Definición y clasificación. Llamada de métodos. Métodos definidos por usuario. Métodos que no retornan valor y no reciben parámetros.
- Segunda sesión
 - Ejercicio aplicativo.

Métodos

- Definición

- Es una porción de programa que realiza una determinada tarea.
- Según la POO es un subprograma que realiza una acción específica y forma parte de un programa que lo contiene (una clase).



Consideraciones al usar métodos

- Un método para que se ejecute debe ser “llamado”.
- Un método no puede estar creado dentro de otro método.
- El orden de ejecución de un método esta determinado por la lógica de programación (es decir lo define el programador).
- Cuando un método termina su ejecución, el flujo del programa regresa a la línea de llamada del método.



Beneficios al usar métodos

- Los métodos se pueden ejecutar (reutilizar) las veces que sean necesarias.
- Permite ahorrar código (líneas de programa)
- El programa es mucho mas ordenado.
- Se pueden usar la cantidad de métodos que sean necesarios.



Partes de un método

Partes de un método

Firma (declaración)

<definición de acceso> **<definición de retorno>** **<nombre del método>** (**<definición de parámetros>**)

{ llave de inicio



.....

.....

} llave de fin

Cuerpo (implementación)

2

3

4

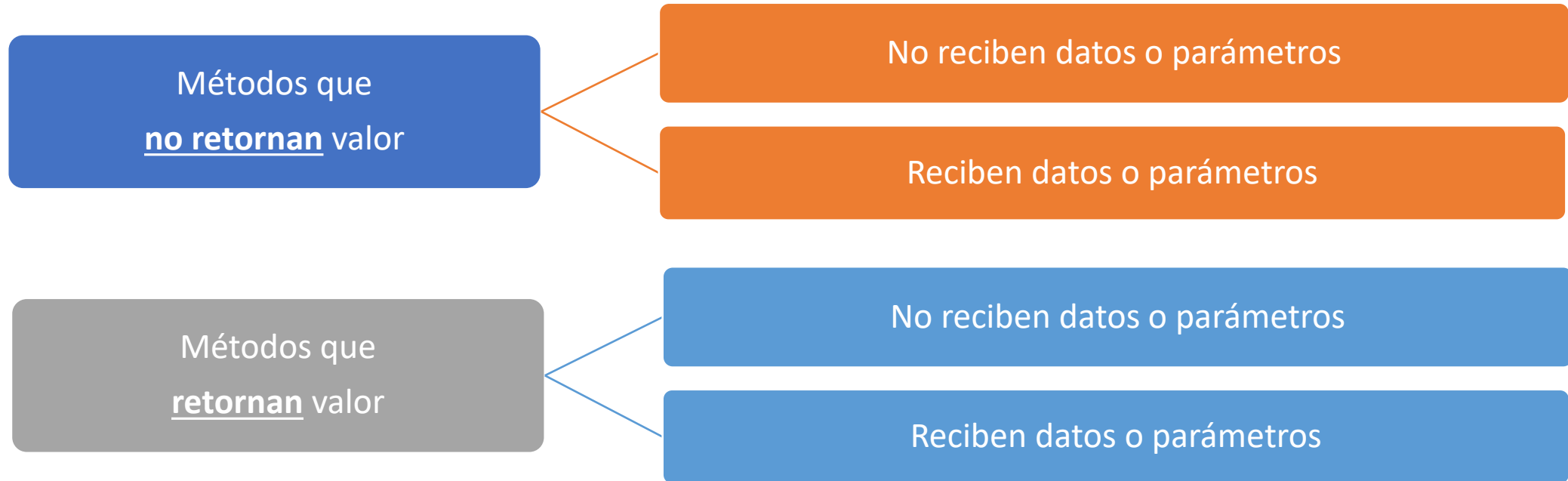
Se establece el nivel de acceso: public, protected o private

Se establece si el método va retornar algún resultado o no

Se establece un nombre representativo de la actividad del método

Se establece si el método requiere datos adicionales para su ejecución o no

Tipos de métodos



```
<definición de acceso> <definición de retorno> <nombre del método> (<definición de parámetros>)  
{  
    .....  
    .....  
}
```


Métodos especiales



Método Principal

```
public static void main(String[] args) {  
    EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            try {  
                Aleatorios frame = new Aleatorios();  
                frame.setVisible(true);  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    });  
}
```

```
static void Main()  
{  
    Application.EnableVisualStyles();  
    Application.SetCompatibleTextRenderingDefault(false);  
    Application.Run(new Aleatorios());  
}
```

- El método main (principal) es el primer método que se ejecuta.
- En la programación visual crea un objeto de la clase.



Método Constructor

```
class Aleatorios {  
  
    public Aleatorios ( ) {  
        ...  
    }  
}
```

- Es el método que no tiene declaración de retorno.
- Su nombre se escribe igual que el nombre de la clase, incluida la letra mayúscula.
- Una de sus funciones principales es la de inicializar las variables globales.

Método que no retorna resultado
y no recibe parámetros



USMP

FACULTAD DE
INGENIERÍA Y ARQUITECTURA

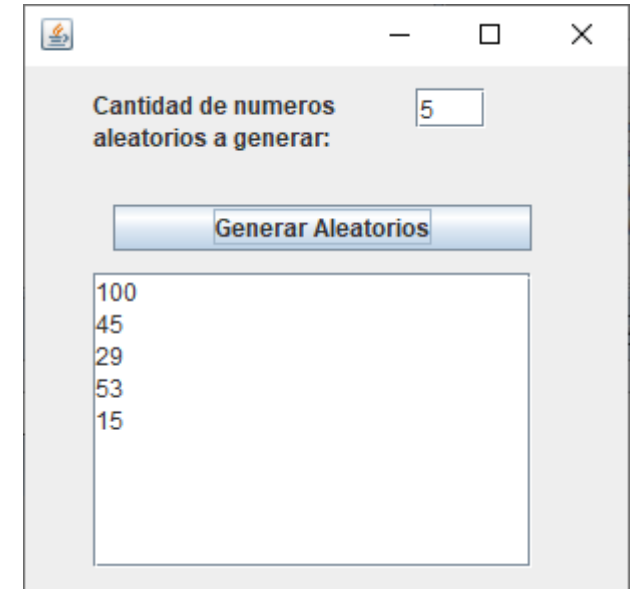
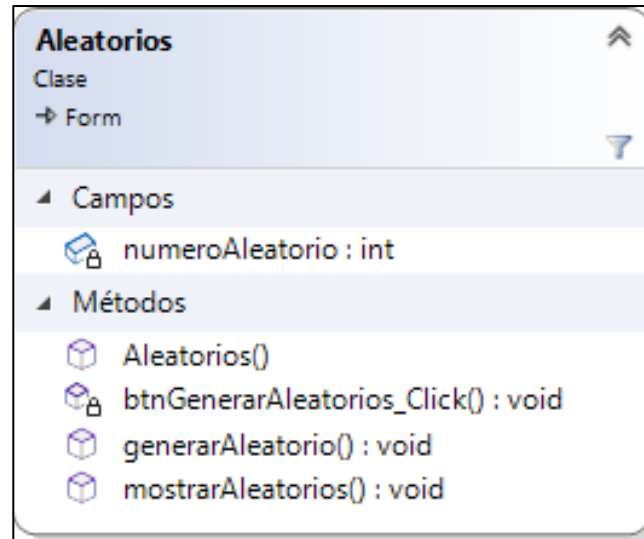
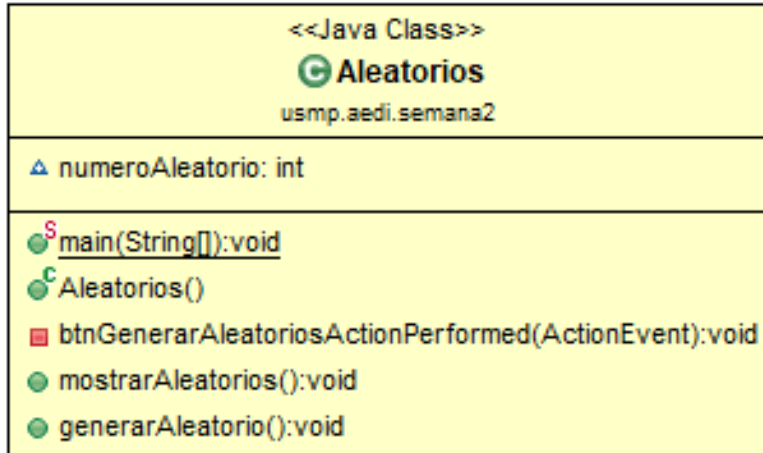
Consideraciones

- Es el más fácil de implementar y de usar.
- Hace uso extensivo de variables globales.

Estructura



Ejemplo



Se requiere crear un programa que permita mostrar una cantidad N de números aleatorios entre 1 y 100. Tener presente el diagrama de clase y el diseño de pantalla.

Flujo de la ejecución del programa.

C#

```
public class Aleatorios extends JFrame {
```

```
    int numeroAleatorio;
```

```
    private void btnGenerarAleatoriosActionPerformed(ActionEvent arg0) {
```

```
        mostrarAleatorios();
```

```
    }
```

```
    public void mostrarAleatorios() {
```

```
        int cantidad;
```

```
        int i=1;
```

```
        cantidad = int.Parse(txtCantidadAleatorios.Text);
```

```
        rthResultado.Clear();
```

```
        while (i<=cantidad){
```

```
            generarAleatorio();
```

```
            rthResultado.AppendText(numeroAleatorio + "\n");
```

```
            i++;
```

```
        }
```

```
    }
```

```
    public void generarAleatorio(){
```

```
        Random rand = new Random();
```

```
        numeroAleatorio = rand.Next(1, 101);
```

```
    }
```

```
}
```

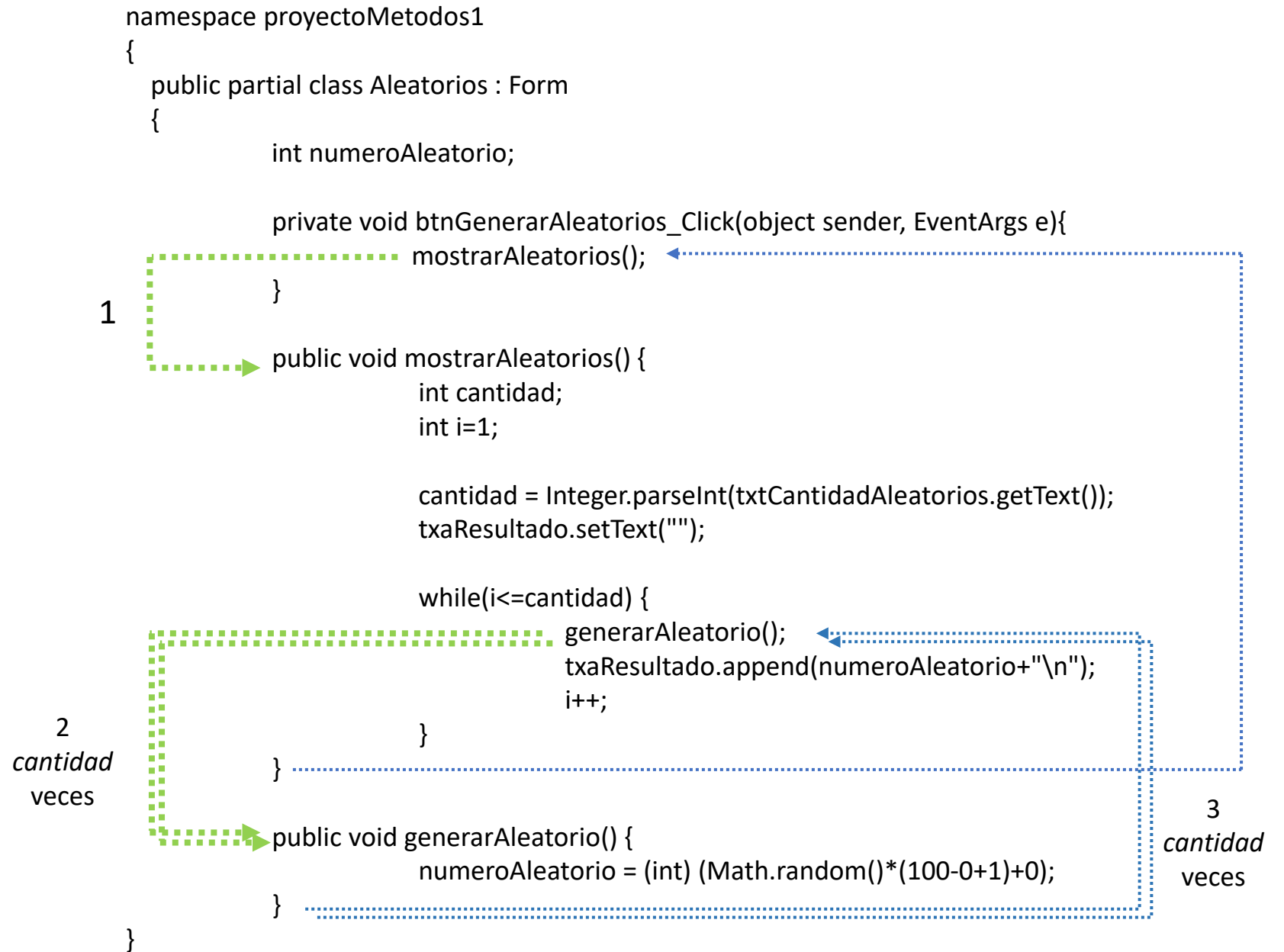
1

2
cantidad
veces

3
cantidad
veces

Flujo de la ejecución del programa.

Java





GRACIAS.



USMP

FACULTAD DE
INGENIERÍA Y ARQUITECTURA

Escuela Profesional de
Ingeniería de Computación y Sistemas

Acreditada por:



ICACIT

