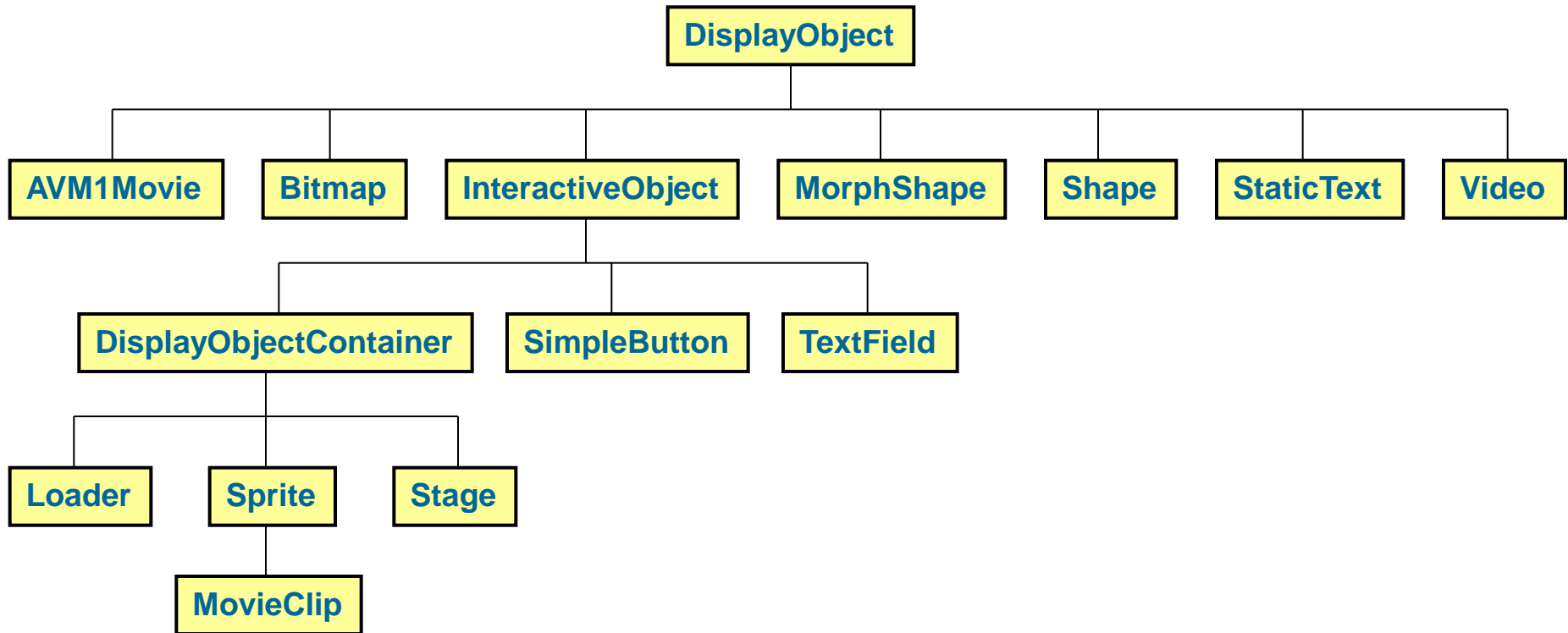


PROGRAMMAZIONE VISUALE

LA CLASSE DISPLAYOBJECT



DISPLAY LIST

- La Display list è la struttura ad albero che contiene tutti gli elementi visuali di un filmato Flash.
- La Display List determina quali oggetti vengono visualizzati e in che ordine
- Action script può intervenire sulla Display List e quindi intervenire su cosa viene visualizzato in un filmato Flash attraverso le classi che discendono da DisplayObjectContainer.

DISPLAY LIST

- La classe **Loader** consente di gestire il caricamento in un filmato Flash di risorse esterne presenti su disco (file swf o immagini)
- Le classi Sprite e MovieClip consentono di aggiungere, togliere cambiare l'ordine di visualizzazione di oggetti grafici creati run time, caricati utilizzando la classe loader, o presenti in libreria

DisplayObjectContainer

- Le classi derivate **Sprite** e **MovieClip** possono contenere e gestire la visualizzazione di qualsiasi oggetto grafico discendente da **DisplayObject**:
 - Oggetti semplici come **TextField** o **Shape**
 - Oggetti **Loader** che contengono contenuti caricati da disco
 - Discendenti di **Sprite** e **MovieClip** che a loro volta possono contenere altri oggetti.

DisplayObjectContainer

Sprite

MovieClip

Le calssi derivate da Sprite:

- Rispondono agli eventi del mouse e della tastiera
- Possono contenere altri oggetti grafici
- Hanno un solo frame

Le calssi derivate da MovieClip:

- Rispondono agli eventi del mouse e della tastiera
- Possono contenere altri oggetti grafici
- Hanno la timeline e quindi più frame

CHILD LIST

- Le classi **Sprite** e **MovieClip** hanno metodi specifici per gestire la propria child list. Cioè l'elenco degli oggetti grafici che contengono.
- **addChild**(child:DisplayObject) aggiunge un elemento alla child listt
 - Ad ogni elemento viene assegnato un indice. Gli elementi vengono visualizzati nell'ordine in cui sono stati aggiunti (l'ultimo risulta in primo piano)

CHILD LIST

- **addChildAt**(child:DisplayObject, index:int) aggiunge un elemento in un punto determinato della child list
- **getChildAt**(index:int):DisplayObject restituisce l'oggetto grafico che si trova all'indice specificato.
- **removeChild**(child:DisplayObject) elimina l'oggetto specificato.

DOCUMENT CLASS

- La **Document Class** è la classe che associa al filmato flash principale
- In l'istanza della classe questo caso è il filmato stesso e viene creata automaticamente in fase di compilazione.
- **Se la Document Class non è una sottoclasse di Sprite o di MovieClip la compilazione verrà interrotta da un errore.**

ESEMPIO

1. Dichiarazione di una classe facendola discendere da Sprite o da MovieClip:

```
package {  
    import flash.display.Sprite;  
    .....  
    public class Orologio extends Sprite {  
        .....  
    }  
}
```

ESEMPIO

2. Definizione di una o più proprietà che contengano gli oggetti grafici da aggiungere alla child list:

```
.....  
import flash.text.TextField  
public class Orologio extends Sprite {  
    private var orologio_txt:TextField;  
    .....  
}
```

ESEMPIO

3. Creazione degli oggetti grafici da aggiungere alla child list:

```
public class Orologio extends Sprite {  
    private var orologio_txt:TextField;  
    .....  
    public function Orologio () {  
        orologio_txt = new TextField();  
        .....  
    }  
    .....  
}
```

ESEMPIO

4. Impostazione delle proprietà degli oggetti creati:

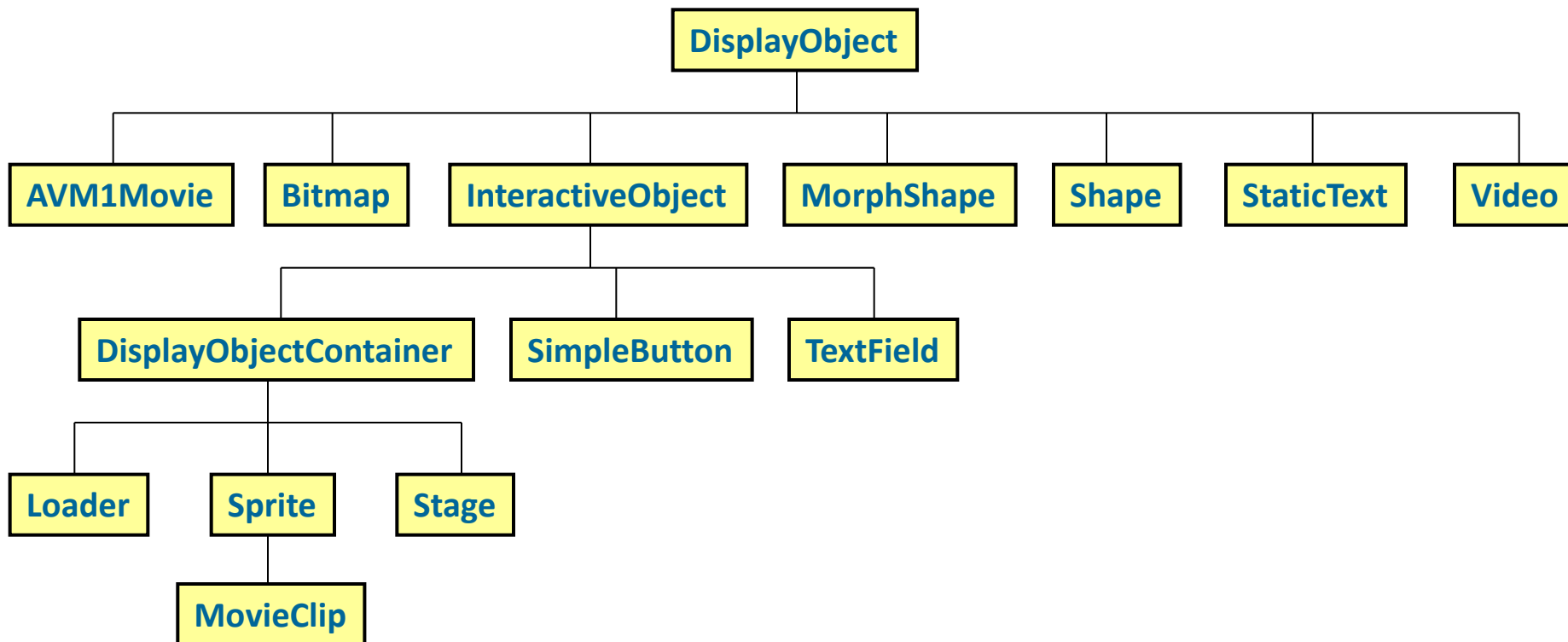
```
.....  
public function Orologio () {  
    orologio_txt = new TextField();  
    orologio_txt.text = "00:00:00" ;  
    orologio_txt.autoSize=  
        TextFieldAutoSize.LEFT;  
}  
.....
```

ESEMPIO

5. Aggiunta degli oggetti creati alla child list nell'ordine desiderato

```
.....  
public function Orologio () {  
    .....  
    addChild(orologio_txt);  
    .....  
}  
.....
```

LE CLASSI VISUALI IN AS3



UNA SUPERFICIE SU CUI DISEGNARE

I METODI PER DISEGNARE

- Ogni Shape, Sprite e MovieClip ha una proprietà **graphics** che è una istanza della classe **Graphics** creata automaticamente con la creazione dell'oggetto.
- Usando i metodi offerti da **Graphics** è possibile disegnare linee e riempimenti e figure.
- I disegni vengono fatti su una superficie trasparente come su un lucido.
- Si possono usare indifferentemente uno delle tre classi. La classe Shape è di gran lunga la più efficiente.
- Per visualizzare una shape è necessario aggiungerla a una child list di uno Sprite o di una MovieClip

DISEGNARE UN LINEA

- Questa classe disegna una linea

```
package {
    import flash.display.Shape;
    import flash.display.Graphics;
    import flash.display.Sprite;

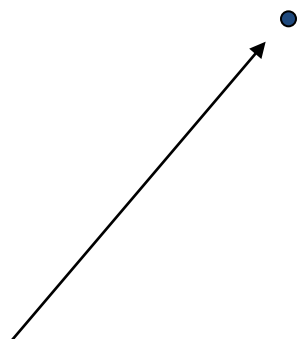
    public class Disegno extends Sprite {
        public function Disegno() {
            var linea:Shape=new Shape() ;

            linea.graphics.lineStyle(2,0xFF0000);
            linea.graphics.moveTo(100,100);
            linea.graphics.lineTo(200,200);
            addChild(linea)
        }
    }
}
```

DISEGNARE UN LINEA

```
linea.graphics.moveTo(100,100);
```

(x=100, y=100)



Il pennino viene spostato alle coordinate x=100, y=100

DISEGNARE UN LINEA

```
linea.graphics.lineTo(200,200);
```

(x=100, y=100)



(x=200, y=200)

Viene tracciata una linea retta fino a 200,200

DISEGNARE UN LINEA

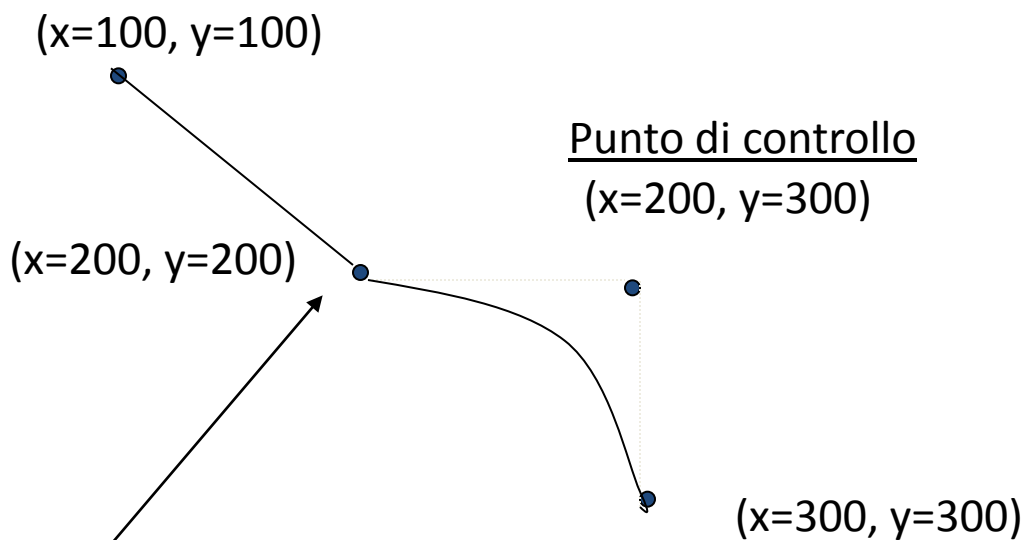
- Questa classe disegna aggiunge un tratto curvo

```
package {
    import flash.display.Shape;
    import flash.display.Graphics;
    import flash.display.Sprite;
    public class Disegno2 extends Sprite {
        public function Disegno2() {
            var linea:Shape=new Shape() ;
            //var curva:Shape = new Shape();

            linea.graphics.lineStyle(2,0xFF0000);
            linea.graphics.moveTo(100,100);
            linea.graphics.lineTo(200,200);
            linea.graphics.curveTo(200,300,300,300);
            addChild(linea)
        }
    }
}
```

DISEGNARE UN LINEA

```
linea.graphics.curveTo(200,300,300,300);
```



Viene tracciata una linea curva fino a (300,300)

DISEGNARE FIGURE

- Disegna un cerchio e un elissi

```
package {
    import flash.display.Shape;
    import flash.display.Graphics;
    import flash.display.Sprite;
    public class Disegno3 extends Sprite {
        public function Disegno3() {
            var cerchio:Shape=new Shape() ;
            var ellissi:Shape = new Shape();
            cerchio.graphics.lineStyle (1,0x00FF00);
            cerchio.graphics.drawCircle(100,100,45);
            addChild(cerchio)
            ellissi.graphics.lineStyle (1,0x0000FF);
            ellissi.graphics.drawEllipse(200,150,45, 100);
            addChild(ellissi)
        }
    }
}
```

DISEGNARE FIGURE

- Disegna un rettangolo

```
package {
    import flash.display.Shape;
    import flash.display.Graphics;
    import flash.display.Sprite;

    public class Disegno4 extends Sprite {
        public function Disegno4() {
            var rettangolo:Shape = new Shape();

            rettangolo.graphics.lineStyle (1,0x0000FF);
            rettangolo.graphics.drawRect(30. 40,45, 100);
            addChild(rettangolo)
        }
    }
}
```


DISEGNARE FIGURE

- Disegna un rettangolo

```
package {
    import flash.display.Shape;
    import flash.display.Graphics;
    import flash.display.Sprite;

    public class Disegno4 extends Sprite {
        public function Disegno4() {
            var rettangolo:Shape = new Shape();

            rettangolo.graphics.lineStyle (1,0x0000FF);
            rettangolo.graphics.drawRect(30. 40,45, 100);
            addChild(rettangolo)
        }
    }
}
```

DISEGNARE FIGURE

- Disegna un rettangolo con riempimento

```
package {
    import flash.display.Shape;
    import flash.display.Graphics;
    import flash.display.Sprite;
    import flash.display.GradientType;
    public class Disegno5 extends Sprite {
        public function Disegno5() {
            var rettangolo:Shape = new Shape();

            rettangolo.graphics.lineStyle (1,0x0000FF);
            rettangolo.graphics.beginFill(0x0000FF);
            rettangolo.graphics.drawRect(30, 40,45, 100);
            addChild(rettangolo)

        }
    }
}
```