

aahh 10 11

ACCADEMIA DI BELLE ARTI DI URBINO

SISTEMI INTERATTIVI DUE

# CREARE UN OROLOGIO

# COMPITO IN CLASSE

- Disegnate un campo di testo dinamico sullo schermo e dategli il nome di istanza **orologio\_txt**.
- Costruite una funzione **aggiornaOrologio** che
  - Dichiarate una variabile **adesso** come variabile di tipo **Date** e assegna alla variabile un'istanza della classe **Date**.
  - Estrae ora, minuti e secondi dalla variabile **adesso** e li scrive nel campo di testo **orologio\_txt**
- Nota bene:
  - **function** `aggiornaOrologio():void`
  - `adesso.hours`, `adesso.minutes` e `adesso.seconds` tutte di tipo **Number**.
- Richiamate la funzione per scrivere l'ora. Esempio:



09:32:58

## SECONDA FASE

- Creare un gestore di eventi per l'evento **Event.ENTER\_FRAME**. Ad esempio:
  - **function** cambioDiFrame(e:**Event**)
- Nel corpo della funzione inserire solo il comando **aggiornaOrologio()**;
- Associare il gestore all'oggetto **orologio\_txt** per l'evento **Event.ENTER\_FRAME**.

11:59:29

aahh 10 11

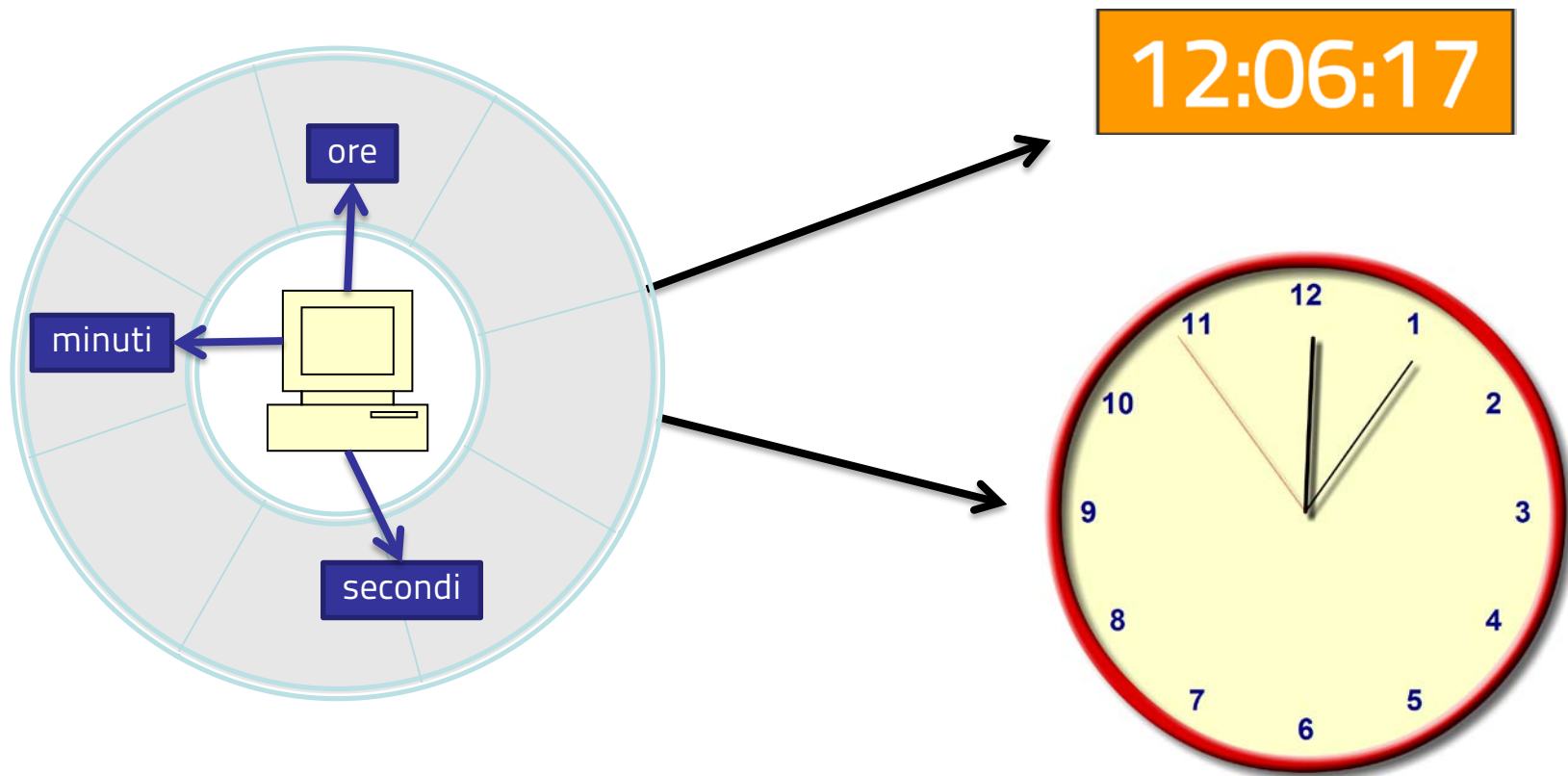
ACCADEMIA DI BELLE ARTI DI URBINO

SISTEMI INTERATTIVI DUE

---

# OROLOGIO GENERICO

# STRUTTURA



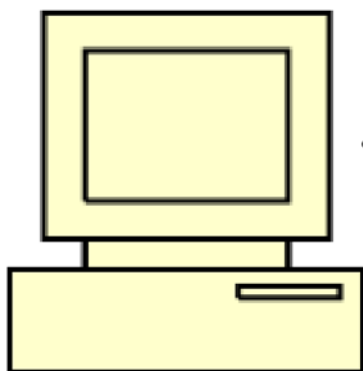
estrazione ora

visualizzazione

# OROLOGIO GENERICO

- Creo una classe astratta da cui derivare orologi specifici
- Compito della classe sarà semplicemente tenere aggiornate (una volta al secondo) tre proprietà che conterranno rispettivamente ora, minuti e secondi.
- Useremo una nuova classe la classe Timer

# TIMER



ore

minuti

secondi

# LA CLASSE TIMER

- La classe timer è una classe fornita con ActionScript 3 che genera un evento a intervalli di tempo prestabilito
- E' utile quando ho bisogno di generare eventi ripetuti nel tempo indipendenti dal ritmo scandito dai frame
- In questo caso basta aggiornare l'orologio una volta la secondo.



# OROLOGIO GENERICO

- Importo solo le classi che mi consentono di definire lo sprite e il timer.

```
package {  
    import flash.display.Sprite;  
    import flash.utils.Timer;  
    import flash.events.TimerEvent;  
  
    .....  
}
```

# OROLOGIO GENERICO

- Definisco le proprietà che il mio timer dovrà aggiornare

```
package {  
    public class OrologioGenerico extends Sprite {  
        protected var ore:uint;  
        protected var minuti:uint;  
        protected var secondi:uint;  
  
        .....  
    }
```

# OROLOGIO GENERICO

- Come **constructor** definisco una funzione che chiama i metodi necessari a disegnare l'orologio e a inserirvi una valore iniziale:

```
package {  
  public class OrologioGenerico extends Sprite {  
    public function OrologioGenerico () {  
      leggiOra();  
      inizializzaVisualizzazione();  
      visualizzaOra();  
      inizializzaTimer();  
    }  
    .....  
  }  
}
```

# OROLOGIO GENERICO

- Definisco i metodi che inizializzano il timer e lo fanno partire: **leggiOra** aggiorna le variabili sulla base dell'ora fornita dal computer:

```
package {  
    public class OrologioGenerico extends Sprite {  
        .....  
        protected function leggiOra() {  
            var adesso:Date = new Date();  
            ore = adesso.getHours();  
            minuti = adesso.getMinutes();  
            secondi = adesso.getSeconds();  
        }  
        .....  
    }
```

# OROLOGIO GENERICO

- **inizializzaTimer** e **aggiorna** sono rispettivamente il metodo che crea e fa partire il timer e il metodo che viene chiamato ad ogni evento generato dal timer:

```
package {  
  public class OrologioGenerico extends Sprite {  
    .....  
    protected function inizializzaTimer() {  
      var myTimer:Timer = new Timer(1000);  
      myTimer.addEventListener(TimerEvent.TIMER, aggiorna);  
      myTimer.start();  
    }  
  
    private function aggiorna(e:TimerEvent) {  
      leggiOra();  
      visualizzaOra();  
    }  
  }  
}
```

# OROLOGIO GENERICO

- Dichiaro i metodi **inizializzaVisualizzazione** e **visualizzaOra** che lascio vuoti in quanto saranno implementati nelle sub classi derivata da OrologioGenerico.

```
package {  
  public class OrologioGenerico extends Sprite {  
    .....  
    public function inizializzaVisualizzazione() {  
      //metodo da definire nelle classi derivate  
    }  
  
    public function visualizzaOra () {  
      //metodo da definire nelle classi derivate  
    }  
  }  
}
```