

# Analisi: 06FrmRep\_x

## Analisi del problema:

Il problema chiede di creare un programma che simuli la giornata lavorativa di un reparto di produzione di un'industria, sono presenti delle linee di produzione, ognuna ha una produzione massima. Le linee dovranno produrre in maniera concorrente, e dunque verranno simulate con dei thread. La produzione oraria cambierà ogni ora, in base a fattori random e al cambio dei 3 turni, in cui la produzione diminuirà del 30% per poi recuperare l'ora successiva. Dovranno essere mostrati i pezzi prodotti dalla linea e infine la quantità prodotta da tutte le linee.

## Classe: Linee

### Attributi della classe:

Attributo	Tipo	Descrizione e utilizzo
<b>nLinea</b>	Variabile intera senza segno a 16 bit.	Indicherà il numero della linea di produzione.
<b>prodMax</b>	Variabile a virgola mobile.	Indicherà la produzione oraria massima.
<b>prodH</b>	Variabile a virgola mobile.	Indicherà la produzione di ogni ora cambiando quindi ad ogni ora.
<b>pezziProd</b>	Variabile intera senza segno a 32 bit.	Indicherà la quantità di pezzi prodotti in una giornata di lavoro.
<b>del</b>	Delegate <b>MyDelegate</b> .	Delegato che punterà al metodo <b>Produci</b> della classe. Non ritorna nulla e non ha bisogno di parametri.

### Costruttori della classe:

#### Costruttore:

- **Firma:** `public Linee(ushort nLinea, double prodMax, ListView lstVw)`
- **Descrizione:** il costruttore inizierà l'oggetto impostando il numero della linea di produzione, la produzione massima e l'oggetto `ListView`.

### Metodi della classe:

#### Metodo eventi casuali:

- **Firma:** `private void RandomEvt()`
- **Descrizione:** Questo metodo simulerà i possibili eventi che avvengono diminuendo o aumentando la produzione, e quindi modificando l'attributo **prodH**.

#### Metodo per il cambio del turno:

- **Firma:**  
`private void CambioTurno(ref ushort nCambioTurni, ushort nOrePassate)`
- **Descrizione:** Questo metodo controllerà se ci sarà un cambio del turno nella linea di produzione, che avverrà tre volte nella giornata di lavoro ogni tre ore di lavoro. Se il metodo rileverà un cambio di turno diminuirà del 30% la produzione oraria per poi ripristinarla nell'ora successiva.

## Metodo Produci:

- **Firma:** private void **Produci()**
- **Descrizione:** Questo metodo eseguirà la funzione principale dell'oggetto che simulerà la giornata di lavoro, all'interno del metodo è presente un ciclo le cui iterazioni indicano le ore di lavoro. All'interno del ciclo verranno richiamate le funzioni **CambioTurno** e **RandomEvnt**, verrà anche incrementata la variabile **pezziProd** con il numero di pezzi prodotti in un'ora, quest'ultimi verranno stampati nella **lstVw**. Infine verrà incrementata la variabile **nOrePassate** e quando supererà il valore 9 verrà stampato la quantità totale prodotta dalla linea.

## Metodo InvokeMethod:

- **Firma del metodo:** public void **InvokeMethod()**
- **Descrizione:** questo metodo sarà eseguito dai thread per richiamare il metodo **invoke** passandogli il delegate che punta al metodo **Produci**.

## Classe: Form1

### Attributi della classe:

Nome Attributo	Tipo	Descrizione e utilizzo
<b>threads</b>	Array di una dimensione di thread.	Vettore che conterrà tutte le istanze della classe <b>thread</b> .
<b>linea</b>	Array di una dimensione della classe Linee.	Vettore che conterrà tutte le istanze della classe <b>Linee</b> .

### Eventi della classe:

#### Evento click bottone crea linea:

- **Firma del metodo:** private void **btnCr\_Click**(object **sender**, EventArgs **e**)
- **Descrizione:** questo evento permette di istanziare oggetti della classe **Linee**. Controllerà che nella **textBox** sia presente un valore inseribile e lo passerà al costruttore della classe **Linee**. Attiverà il pulsante **btnStrt**.

#### Evento click bottone start:

- **Firma del metodo:** private void **btnStrt\_Click**(object **sender**, EventArgs **e**)
- **Descrizione:** questo evento definisce la grandezza del vettore **threads** e con un ciclo, ne inizializza gli elementi. In un altro ciclo partiranno i thread. Infine verrà stampato il totale prodotto dalle linee.

#### Evento click bottone visualizza totale:

- **Firma del metodo:** private void **btnVis\_Click**(object **sender**, EventArgs **e**)
- **Descrizione:** questo evento permette la visualizzazione della somma dei pezzi prodotti. In oltre rinizializzerà i vettori per permettere di rieseguire la simulazione.