

Analisi: 05Reparti_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 produzioni 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. Dovrà essere mostrata la produzione di ogni ora, i pezzi prodotti dalla linea e infine la quantità prodotta da tutte le linee.

Classe: Linee

Attributi della classe:

Attributo	Tipo	Descrizione e utilizzo
nLinea	Variabile intera senza segno a 16 bit.	Indicherà il numero della linea di produzione.
prodMax	Variabile a virgola mobile.	Indicherà la produzione oraria massima.
prodH	Variabile a virgola mobile.	Indicherà la produzione di ogni ora cambiando quindi ad ogni ora.
nOrePassate	Variabile intera senza segno a 16 bit.	Indicherà le ore passate.
pezziProd	Variabile intera senza segno a 32 bit.	Indicherà la quantità di pezzi prodotti in una giornata di lavoro.

Costruttori della classe:

Costruttore:

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

Metodi della classe:

Metodo eventi casuali:

Firma: `private void RandomEvt()`

- **Descrizione:** Questo metodo simulerà i possibili eventi che avvengono rallentando 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 Vivi:

- **Firma:** `public 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. Infine verrà incrementata la variabile **nOrePassate** e quando supererà il valore 9 verrà stampato la quantità totale prodotta dalla linea.