

Analisi: 07Rain_x

Analisi del problema:

Il problema chiede di creare un programma che simuli, tramite i costrutti del multithreading, l'andamento delle quantità giornaliere di pioggia in una località, determinata dall'utente. La pioggia giornaliera dipenderà dal mese e dalla quantità media mensile di pioggia che cade nella località, dato che verrà inserito dall'utente. Durante l'esecuzione del programma sarà visibile la pioggia caduta ogni giorno, ed infine la quantità media tra tutti i giorni dell'anno. Sarà possibile eseguire più volte l'applicazione.

Classe: RainDay

Attributi della classe:

Attributo	Tipo	Descrizione e utilizzo
rainInDay	Vettore di numeri a virgola mobile.	Contiene la quantità di pioggia per ogni giorno dell'anno.
maxRain	Vettore di interi.	Indicherà la quantità massima di pioggia nei giorni di un mese.
location	Stringa.	Contiene il nome della località.
rainMed	Variabile per numero con virgola mobile.	Indicherà la quantità di pioggia media caduta nella località in un mese.
iOC1	Oggetto InOutCtrl .	Puntatore dell'oggetto InOutCtrl passato al costruttore della classe.
del	Delegate assist .	Delegato che punterà al metodo Display della classe. Non ritorna nulla e chiede due parametri nella firma.

Costruttore della classe:

- **Firma:** `public RainDay(InOutCtrl iOC1)`
- **Descrizione:** il costruttore inizierà l'oggetto impostando il nome della località, la pioggia media, l'oggetto `InOutCtrl` e il delegato.

Metodi della classe:

Metodi Get e Set:

- **Firme:**
 - `public void SetRainMed(double rainMed)`
 - `public double GetRainMed()`
 - `public void SetLocation(string location)`
 - `public string Getlocation()`
- **Descrizione:** i metodi Get e Set permettono di modificare(Set) e di recuperare(Get) i dati contenuti negli attributi dell'oggetto.

Metodo Rain:

- **Firma:** `public void Rain()`
- **Descrizione:** Questo metodo simulerà la pioggia caduta per ogni giorno, stampando nel form la quantità piovuta ed infine la media di pioggia giornaliera in un anno chiamando il metodo **Invoke** dell'oggetto **iOC1** e passandogli il delegato e i parametri necessari al metodo puntato **Display**.

Metodo Display:

- **Firma del metodo:** `private void Display(int i, double cOut)`
- **Descrizione:** questo metodo sarà puntato dal delegato **del**. Renderà la label di output visibile nell'oggetto **iOC1** e ne imposterà il testo passando i valori simulati dalla classe.

Classe: Form1

Attributi della classe:

Nome Attributo	Tipo	Descrizione e utilizzo
threads	Array di una dimensione di thread.	Vettore che conterrà 10 istanze della classe thread .
rain	Array di una dimensione della classe RainDay .	Vettore che conterrà 10 istanze della classe RainDay .
running	Booleano.	Indicherà se l'applicazione è in esecuzione o in attesa di eventi.
del	Delegate MyDelegate .	Delegato che punterà ai metodi ChBtnStrt e ChIOCltxtBxesRdOnly della classe. Non ritorna nulla e non richiede parametri.
iOCL	Vettore di oggetti InOutCtrl .	Vettore che conterrà i 10 controlli utenti che verranno inizializzati nel form dal costruttore della classe.

Metodi della classe:

Metodo Run:

- **Firma:** private void **Run()**
- **Descrizione:** Questo metodo farà partire i thread e aspetterà che terminino la loro esecuzione. Questo metodo verrà eseguito da un thread.

Metodo SetIOCl:

- **Firma:** private void **SetIOCl()**
- **Descrizione:** Questo metodo verrà eseguito dal costruttore ed inizierà i controlli nel form.

Metodi Ch:

- **Firma:**
 - private void **ChBtnStrt()**
 - private void **ChIOCltxtBxesRdOnly()**
- **Descrizione:** Questi metodi cambieranno gli attributi booleani dei controlli e verranno richiamati per mezzo del delegato.

Eventi della classe:

Evento click bottone start:

- **Firma del metodo:** private void **btnStrt_Click(object sender, EventArgs e)**
- **Descrizione:** Questo evento istanzia gli oggetti della classe **RainDay** e i thread passando al costruttore i valori inseriti nei controlli del form. Infine verrà creato il thread che eseguirà il metodo **Run**.

Evento click bottone Auto:

- **Firma del metodo:** private void **btnAut_Click(object sender, EventArgs e)**
- **Descrizione:** Questo evento imposterà automaticamente dei valori nelle text box dei controlli.

Evento chiusura form:

- **Firma del metodo:** private void **Form1_FormClosing(object sender, EventArgs e)**
- **Descrizione:** Questo evento terminerà i thread in esecuzione.

Classe: InOutCtrl

Attributi della classe:

Nome Attributo	Tipo	Descrizione e utilizzo
running	booleano.	Indicherà se l'applicazione è in esecuzione o in attesa di eventi.
startWidth	Variabile statica intera.	Variabile contenente la larghezza dello user control.

Metodi della classe:

Metodo Ch_Label_Visible:

- **Firma:** public void **ChLblOutVisible()**
- **Descrizione:** Questo metodo cambia l'attributo della visibilità della label di output.

Metodi Get e Set:

- **Firme:**
 - public void **SetLblLoc**(string txt)
 - public void **SetLblOut**(string txt)
 - public void **SetTxtBxRain**(string txt)
 - public void **SetTxtBxLoc**(string txt)
 - public string **GetTxtBxLoc**()
 - public bool **GetLblOutVisible**()
 - public string **GetTxtBxRain**()
 - public void **SetRunning**(bool running)
 - public bool **GetRunning**()
 - public void **SetTxtBxesReadOnly**(bool tof)
- **Descrizione:** i metodi Get e Set permettono di modificare(Set) e di recuperare(Get) i dati contenuti negli attributi dell'oggetto.