

Analisi: 09Zoo_x

Analisi del problema:

Il problema chiede, di creare un programma che permetta la visualizzazione di una raccolta di animali, presenti in uno zoo. Dovrà essere possibile l'inserimento delle varie proprietà di ogni animale e la loro visualizzazione in una struttura dati.

Classe: Animale

Attributi della classe:

Nome Attributo	Tipo	Descrizione e utilizzo
<i>animal</i>	stringa.	Attributo che contiene il tipo di animale.
<i>name</i>	stringa.	Attributo che contiene il nome dell'animale.
<i>years</i>	Intero a 32bit.	Attributo che indica l'età in anni dell'animale.

Costruttore della classe:

- **Firma:** `public Animale(int years, string name)`
- **Descrizione:** Il costruttore della classe servirà ad inizializzare i valori per ogni singolo oggetto. I dati da passare verranno recuperati da elementi appartenenti al form.

Metodi della classe:

Metodi Get e Set:

- **Firme:**
 - `public void SetYears(Int years)`
 - `public int GetYears()`
 - `public void SetName(string name)`
 - `public string GetName()`
 - `public void SetAnimal(string animal)`
 - `public string GetAnimal()`
- **Descrizione:** i metodi Get e Set permettono di modificare(Set) e di recuperare(Get) i dati presenti negli attributi dell'oggetto.

Metodo Verso:

- **Firme:** `public virtual string Verso()`
- **Descrizione:** il metodo Verso ritornerà il nome del verso dell'animale. Questo metodo sarà contrassegnato come virtual, ciò permetterà l'override del metodo nelle classi derivate.

Classe: Capra ← Animale

Costruttore della classe:

- **Firma:** public Capra(int years, string name) : base(years, name)
- **Descrizione:** Il costruttore della classe servirà ad inizializzare i valori per ogni singolo oggetto. I dati da passare verranno recuperati da elementi appartenenti al form.

Metodo Verso:

- **Firma:** public override string Verso()
- **Descrizione:** il metodo Verso ritornerà il nome del verso dell'animale. Questo metodo sovrascrive il metodo della classe base **Animale** .

Classe: Scimmia ← Animale

Costruttore della classe:

- **Firma:** public Scimmia(int years, string name) : base(years, name)
- **Descrizione:** Il costruttore della classe servirà ad inizializzare i valori per ogni singolo oggetto. I dati da passare verranno recuperati da elementi appartenenti al form.

Metodo Verso:

- **Firma:** public override string Verso()
- **Descrizione:** il metodo Verso ritornerà il nome del verso dell'animale. Questo metodo sovrascrive il metodo della classe base **Animale** .

Classe: Lupo ← Animale

Costruttore della classe:

- **Firma:** public Lupo(int years, string name) : base(years, name)
- **Descrizione:** Il costruttore della classe servirà ad inizializzare i valori per ogni singolo oggetto. I dati da passare verranno recuperati da elementi appartenenti al form.

Metodo Verso:

- **Firma:** public override string Verso()
- **Descrizione:** il metodo Verso ritornerà il nome del verso dell'animale. Questo metodo sovrascrive il metodo della classe base **Animale** .

Classe: Tigre ← Animale

Costruttore della classe:

- **Firma:** public Tigre(int years, string name) : base(years, name)
- **Descrizione:** Il costruttore della classe servirà ad inizializzare i valori per ogni singolo oggetto. I dati da passare verranno recuperati da elementi appartenenti al form.

Metodo Verso:

- **Firma:** public override string Verso()
- **Descrizione:** il metodo Verso ritornerà il nome del verso dell'animale. Questo metodo sovrascrive il metodo della classe base **Animale** .

Classe: Elefante ← Animale

Costruttore della classe:

- **Firma:** `public Elefante(int years, string name) : base(years, name)`
- **Descrizione:** Il costruttore della classe servirà ad inizializzare i valori per ogni singolo oggetto. I dati da passare verranno recuperati da elementi appartenenti al form.

Metodo Verso:

- **Firma:** `public override string Verso()`
- **Descrizione:** il metodo Verso ritornerà il nome del verso dell'animale. Questo metodo sovrascrive il metodo della classe base **Animale**.

Classe: Form1

Attributi della classe:

Nome Attributo	Tipo	Descrizione e utilizzo
<i>animals</i>	Vettore di oggetti Animale.	Vettore contenente gli oggetti delle classi derivate dalla classe base Animale.

Metodi della classe:

Metodo per la visualizzazione:

- **Firma:** `private void Display()`
- **Descrizione:** questo metodo controllerà ogni elemento del vettore **animals**, se l'elemento contiene un oggetto, ne verranno visualizzati, in un **DataGridView**, gli attributi. Infine verrà richiamato il metodo **Conta**.

Evento click bottone inserisci assenza:

- **Firma del metodo:** `private void button1_Click(object sender, EventArgs e)`
- **Descrizione:** questo evento permette l'aggiunta degli oggetti che rappresentano gli animali nel vettore **animals** passando ai costruttori i valori immessi nelle caselle (TextBox e ComboBox) del form. Questo evento richiamerà le funzioni **CtrlStrng** e **Display**.

Metodo conta:

- **Firma:** `private void Conta()`
- **Descrizione:** questo metodo conterà gli animali facendo, infine, visualizzare i valori in una **label**. Verranno contati gli animali in totale e per tipo.

Metodo di controllo caselle:

- **Firma:** `private bool CtrlStrng()`
- **Descrizione:** questo metodo controllerà che le caselle non siano vuote. Se una casella, non conterrà valori, la funzione ritornerà il valore booleano **false** non permettendo all'evento **button1_Click** di creare gli oggetti e inserirli nel vettore.