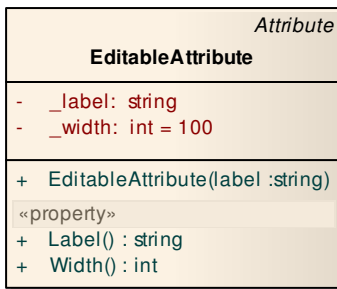
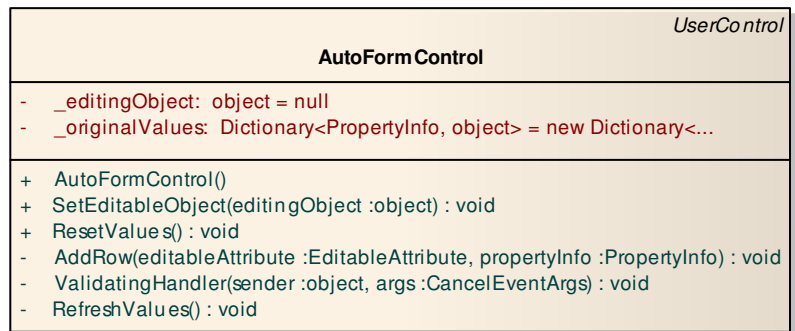


Laboratorio 4



Passo 1 – Creare un nuovo progetto di tipo *Windows Forms Application* e di nome **Lab4**. Eliminare dal progetto la classe **Form1** e la classe **Program**. Scaricare il file “**Lab4Start.zip**” dal sito del corso e inserire i file “.cs” in esso contenuti nel progetto **Lab4**. Aggiungere al progetto la classe **EditableAttribute** (come da diagramma UML) –l’attributo **Editable** deve essere applicabile solo alle **proprietà** di una classe. Compilare il progetto.

Passo 2 – Aggiungere al progetto una sottoclasse di **UserControl** di nome **AutoFormControl** (come da diagramma UML – nel diagramma non compaiono *field* e metodi generati dal *designer*). Il metodo **SetEditableObject** deve accettare come parametro un oggetto qualsiasi **editingObject** e, utilizzando l’introspezione, deve aggiu-



gnere all’**AutoFormControl** una coppia di controlli (**Label**, **TextBox**) per ogni proprietà di **editingObject** alla quale risulti associato l’attributo **Editable**. Per posizionare opportunamente i controlli, si utilizzi un **TableLayoutPanel** (di nome `_tableLayoutPanel`) con due colonne; la prima colonna dovrà contenere i controlli **Label**, la seconda colonna dovrà contenere i corrispondenti controlli **TextBox**. Organizzare il metodo **SetEditableObject** (che deve essere invocabile più volte) nel modo seguente:

1. Eliminare da `_tableLayoutPanel` eventuali controlli residui.
2. Eliminare da `_originalValues` eventuali valori originali residui; tale collezione consente di effettuare un *rollback* dei valori inseriti.
3. Memorizzare nel *field* `_editingObject` l’oggetto da editare.
4. Per ogni proprietà pubblica di **editingObject** alla quale è stato associato l’attributo **Editable**:
 - a. se la proprietà è *write-only*, sollevare un’eccezione
 - b. se la proprietà è *writable*, memorizzare il valore originale di tale proprietà in `_originalValues`, utilizzando come chiave il descrittore della proprietà stessa
 - c. mediante il metodo **AddRow**, creare e aggiungere a `_tableLayoutPanel` prima una **Label** dove:
 - il testo da visualizzare è fornito dal valore della proprietà **Label** dell’attributo associato alla proprietà corrente di **editingObject**;
 - la proprietà **AutoSize** è impostata a **true** in modo che il testo venga visualizzato correttamente;quindi una **TextBox** dove:
 - l’altezza è pari al valore della proprietà **PreferredHeight** del controllo stesso e la larghezza pari al valore della proprietà **Width** dell’attributo associato alla proprietà corrente di **editingObject**;
 - se la proprietà corrente di **editingObject** è *read-only*, occorre disabilitare la **TextBox** (utilizzare la proprietà **Enabled**)
 - alla proprietà **Tag** va assegnato il riferimento al descrittore della proprietà corrente di **editingObject**
 - all’evento **Validating** va collegato il gestore **ValidatingHandler**.

Laboratorio 4

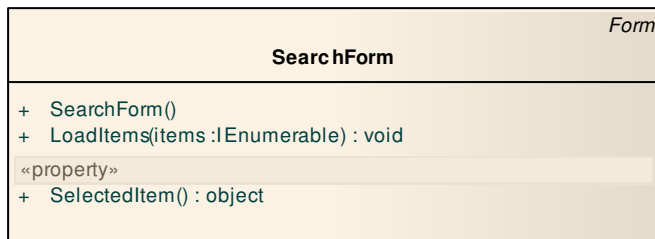
5. Invocare il metodo **RefreshValues**.

Il metodo **RefreshValues** deve inserire in ogni controllo di tipo **TextBox** il valore della corrispondente proprietà di **editingObject**. A tal fine, si selezionino i controlli di tipo **TextBox** da **_tableLayoutPanel** e si usi l'istanza di **PropertyInfo** memorizzata precedentemente nella proprietà **Tag** di tali controlli. Il metodo **ResetValues** deve effettuare il *rollback* dei valori inseriti reimpostando i valori originali contenuti in **_originalValues**.

Ogni valore inserito dall'utente in una **TextBox** deve essere validato e, se corretto, assegnato alla corrispondente proprietà di **editingObject**; tale operazione deve essere effettuata nel metodo **ValidatingHandler** da associare all'evento **Validating** delle **TextBox**. Il metodo deve:

1. Recuperare dalla **TextBox** (oggetto **sender** dell'evento) la **PropertyInfo** associata.
2. Recuperare dalla **PropertyInfo** il tipo della proprietà.
3. Assegnare il valore contenuto nella **TextBox** alla proprietà dell'oggetto in editing (si utilizzi il metodo **Convert.ChangeType**) – in caso di eccezione, occorre segnalare all'utente il messaggio dell'eccezione mediante un **ErrorProvider** (**_errorProvider**, da aggiungere all'**AutoFormControl** mediante *designer*) e cancellare l'evento (**CancelEventArgs.Cancel = true**), in modo che la **TextBox** contenente il valore errato mantenga il *focus*.
4. A ogni assegnamento di un valore a una delle proprietà di **editingObject**, invocare il metodo **RefreshValues** (alcune proprietà di **editingObject** potrebbero essere calcolate basandosi sui valori di altre proprietà).

Eliminare i commenti dal file **AutoForm.cs** e testare il codice.



Passo 3 – Aggiungere al progetto la *form* **SearchForm** (come da diagramma UML – nel diagramma non compaiono *field* e metodi generati dal *designer*). La *form* deve consentire di selezionare un oggetto da una lista di oggetti; a tal fine, deve contenere un controllo di tipo **ComboBox** e due bottoni (OK

e Annulla) per confermare o annullare l'operazione di selezione. Il metodo **LoadItems** deve inserire nella **ComboBox** (proprietà **Items**) tutti gli oggetti contenuti in **items**. La proprietà **SelectedItem** deve restituire il valore della proprietà **SelectedItem** della **ComboBox** che corrisponde al valore scelto dall'utente.

Eliminare i commenti dal file **MainForm.cs** e testare il codice.