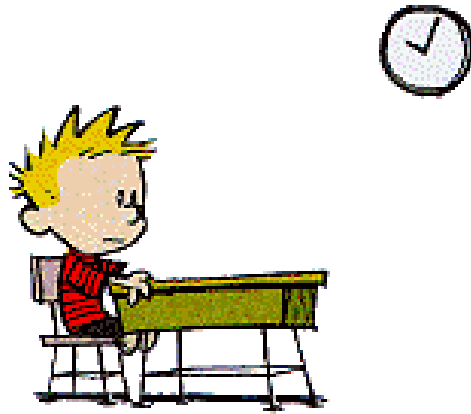


LIVE PROGRAMMING



ovvero: non importa se la scuola è chiusa, vi tocca studiare lo stesso!!!

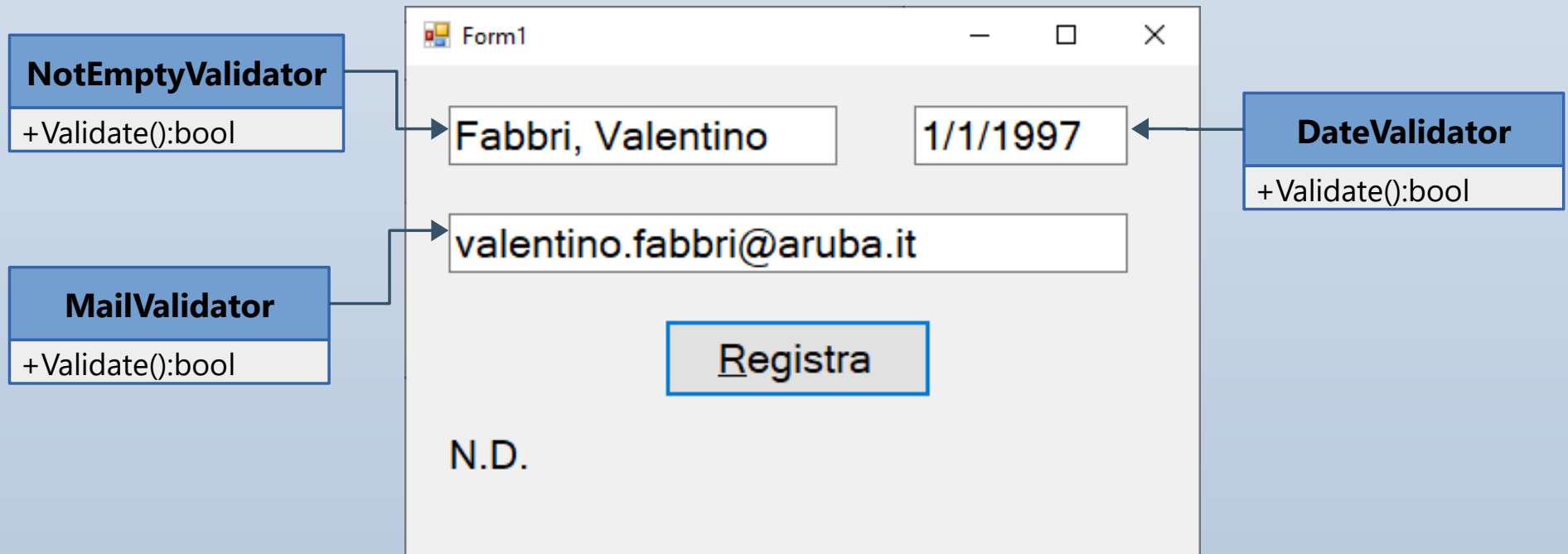
TIPI ASTRATTI

- Definire un tipo astratto mediante un'*interfaccia*
- Limiti delle *interfacce* (in alcuni scenari) (parte II)
- Condividere parte dell'implementazione tra i tipi concreti:
classi astratte (parte II)

VALIDAZIONE DELL'INPUT

- Implementare degli oggetti che semplifichino la validazione dell'input in una applicazione *windows.forms*
- Verificare: input vuoto, date non valide, indirizzi mail, etc
- Concetto di *validatore*: classe che incapsula la funzione di verificare se un valore soddisfa delle condizioni

VALIDAZIONE DELL'INPUT



PROTOTIPO DI UN VALIDATORE

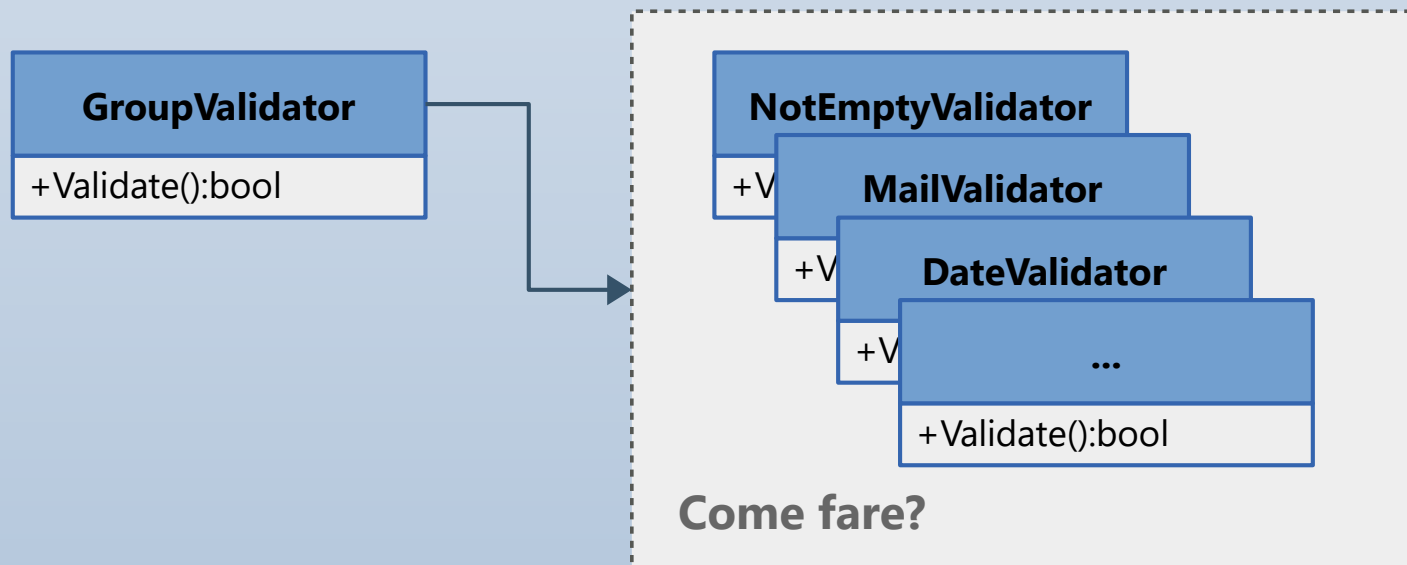
```
public delegate string ValueProvider();  
1 reference  
public class PrototipoValidatore  
{  
    ValueProvider valueProvider;  
    0 references  
    public PrototipoValidatore(ValueProvider valueProvider)  
    {  
        this.valueProvider = valueProvider;  
    }  
  
    0 references  
    public bool Validate()  
    {  
        //fornisce il valore da validare  
        string value = valueProvider();  
        |  
        return true; // TODO: logica di validazione di "value"  
    }  
}
```

VALIDAZIONE DI GRUPPO

Semplificare il codice applicativo: fornire la possibilità di validare più oggetti mediante una sola invocazione

VALIDATORE DI GRUPPO

- Servirebbe un “validatore di gruppo”, che contenga altri validatori
- Come fare? Non è possibile avere una lista di oggetti di tipo diverso!



PROTOTIPO E USO DI VALIDATORE DI GRUPPO

```
public class GroupValidator
{
    List<Validatore> validatori = new List<Validatore>();

    0 references
    public void Add(Validatore v)
    {
        validatori.Add(v);
    }

    0 references
    public bool Validate()
    {
        bool valid = true;
        foreach (var v in validatori)
        {
            valid = valid && v.Validate();
        }
        return valid;
    }
}
```

```
NotEmptyValidator notEmptyValid;
DateValidator dateValid;
MailValidator mailValid;
```

```
GroupValidator groupValid;
```

```
groupValid.Add(notEmptyValid);
groupValid.Add(dateValid);
groupValid.Add(mailValid);
```

```
//bool ok = notEmptyValid.Validate()
// |           && dateValid.Validate()
//           && mailValid.Validate();

bool ok = groupValid.Validate();
```


CONCETTO ASTRATTO DI *VALIDATORE*

- Definire un'*interfaccia*, *IValidator*, che rappresenti qualsiasi tipo di validatore
- Dichiarare i tipi di validatori "conformi" all'*interfaccia* (implementare l'*interfaccia IValidator*)
- Ora è possibile definire una lista di validatori, poiché tutti possono essere considerati appartenenti allo stesso tipo (astratto) *IValidator*

CONCETTO ASTRATTO DI *VALIDATORE*

