

# Corso di Linguaggi di Programmazione

## Lezione 17

Alberto Ceselli  
`alberto.ceselli@unimi.it`

Dipartimento di Informatica  
Università degli Studi di Milano

07 Maggio 2013

## Riassumendo:

### Encapsulation:

- catturare più errori in fase di compilazione – analisi semantica (anche compilazioni separate!)
- aumentare la facilità di manutenzione del software (locality)
- fornire strumenti per condividere packages e librerie in modo “sicuro” (protetto da accessi esterni, soprattutto accidentali)

### Modularità:

- Avere ADT in files diversi consente unità di compilazione separate
- Avere interfaccia ed implementazione in files diversi consente di avere le specifiche descritte a parte

# ADT parametrici





# ADT param. in C

- stessi problemi visti in precedenza
- la printf non è generica



## ADT param. in Ada

- Stack ristretto a contenere valori interi
- Lo stack può contenere al massimo 100 elementi
- CODICE ESEMPIO
- interfaccia ed implementazione possono essere in files diversi.
- `Float_Stack` e `Int_Stack` generano *due versioni diverse* del codice sorgente *a compile time*.



# ADT param. in C++

- CODICE ESEMPIO
- interfaccia ed implementazione *nello stesso file*
- `template` per definire il parametro di tipo
- Il tipo è poi utilizzato all'interno della struct
- Funziona in modo simile ad una macro, *sostituendo* il codice del template nel codice del client *a compile time* (e per questo deve essere tutto nell'header, e di conseguenza viene trattato *inline*)

# ADT param. in Java

- da Java 5.0
- CODICE ESEMPIO
- la virtual machine è la stessa: poca efficienza
- permette una gestione fine dei tipi generici (es. wildcards '?' per i tipi ...)



# ADT param. in C sharp

- da .NET versione 2.0
- CODICE ESEMPIO
- gestito in modo efficiente: codice viene “duplicato” a *run-time* solo in caso di necessità

## Esercizio:

- Realizzare un ADT “Buffer FIFO” (interfaccia, implementazione e codice client).
- Realizzare un’implementazione alternativa nel secondo linguaggio: come cambiano interfaccia e codice client?
- Realizzare un ADT “Buffer LIFO”: come cambiano interfaccia e codice client?
- Rendere gli ADT realizzati *parametrici* rispetto al tipo dei dati da memorizzare nel buffer.

