



# Abstract Data Types

## Types de Données Abstraits

# Abstract Data Types (ADTs)

- An abstract data type (ADT) is an abstraction of a data structure
- An ADT specifies:
  - Data stored
  - Operations on the data
  - Error conditions associated with operations
- Example: ADT modeling a simple stock trading system
  - The data stored are buy/sell orders
  - The operations supported are
    - ◆ order **buy**(stock, shares, price)
    - ◆ order **sell**(stock, shares, price)
    - ◆ void **cancel**(order)
  - Error conditions:
    - ◆ Buy/sell a nonexistent stock
    - ◆ Cancel a nonexistent order

# ADT

- ❑ Les ADT simplifient et encouragent la décomposition et la modularité
- ❑ Les ADT permettent de découpler implantation et utilisation
  - Par exemple, une fois l'interface d'un ADT connu, plusieurs implantations différentes sont possibles
  - Changer l'implantation d'un ADT n'affecte pas les applications qui l'utilisent

# ADT List...

- Une *List* spécifie :
  - Une collection ordonnée d'éléments
  - Des opérations pour : créer une *List*, obtenir le nombre d'éléments qui la constitue, convertir en chaîne de caractères la liste des éléments, ajouter, retirer et retrouver des éléments, etc.
  - La détection des erreurs provoquées lorsqu'on tente de retirer un élément introuvable (e.g. qui n'est pas dans la liste ou dont l'indice n'existe pas)

# ADT List...

```
#ADT List "interface"
```

```
class List:
```

```
    def __init__( self ):
        pass
```

```
    #return the number of elements in List
    def __len__( self ):
        pass
```

# ADT List...

```
#convert a List into a string:  
# elements listed between brackets  
# separated by commas  
# size and capacity of the data structure  
# indicated when relevant
```

```
def __str__( self ):  
    pass
```

```
#add element at the end of list  
def append( self, element ):  
    pass
```

# ADT List

```
#remove the kth element  
def remove( self, k ):  
    pass
```

```
#find and return the rank of  
#element if in list, False otherwise  
def find( self, element ):  
    pass
```