Basic concepts – element type

EqualityComparable

A type is EqualityComparable if objects of that type can be compared for equality

We denote this type: TE

- Element Type that can be compared for equality
- has a default value;

Alternative name: TElement

Notation: ε_{TE} – the default element from the set

TE as an ADT:

$\mathcal{D}_{TE}$= {e | e –element}

operations:

Subalg. assignment(e1,e2)

Desc.: assign value from one element to another

Prec.: e1 $\in \mathcal{D}_{TE}$ *here "$\in$" and "=" are the math. signs*

Post.: e2 = e1 *with mathematical significance*

Funct. isEqual (e1,e2)

Desc.: test if that 2 values are equal

Prec.: e1 $\in \mathcal{D}_{TE}$, e2 $\in \mathcal{D}_{TE}$

Post.: isEqual = true if e1=e2

false if e1<>e2

Subalg. initDefault(e)

Desc.: initialize with the default element

Prec.: -

Post.: e= ε_{TE}

Notations:

assignment	operator:	:= or ←
equalityTest	operator:	=

Remarks:

- we consider TE a general (abstract) Type (that can have specialization)
- basic data types are a kind of TE
- Specification of the type of an element:

  e $\in \mathcal{D}_{TE}$

  e: TE

  We can read/print an element by using *read*/*print (*pseudocode*)*

LessThanComparable

A type is LessThanComparable if it is ordered.

- it must be possible to compare two elements of that type
  by using operators < , = and <
  - <, > - is a partial ordering relation
  - = is a equivalence relation
  - any 2 elements are in one of the 3 relations: <,=,>

Example:

<, =, > - mathematical relations for numbers

<, =, > - given by the "dictionary" order of strings (of chars)

Remark:

Only `operator <=` *is fundamental;*
the other inequality operators are essentially syntactic sugar.

We denote this type: TCE (Type: Comparable Element)
It is a kind of TE, that is it has all operations of TE and *some* more
Alternative name: TLessThanComparable

TCE as an ADT:

$\mathcal{D}_{TCE}$= {e | e – less than comparable element}

operations:

@all operations of TE

Funct. compare(e1,e2)

Desc.: compare 2 elements

Prec.: e1 $\in \mathcal{D}_{TE}$, e2 $\in \mathcal{D}_{TE}$

Post.: compare =		
	-1	if e1 < e2
	0	if e1 = e2
	1	if e1 > e2

Note:

sometimes, we will use operators: <,=,>,<=, >=
instead of calling function *compare*