Java Utility Classes

- Java utilities
  - most core to Java, not extras
  - java.util package
- Type wrappers
  - java.lang package
    - automatically imported
  - lets the primitive types be treated like objects
    - int -> Integer
    - boolean -> Boolean
    - double -> Double
    - etc.

Java Utility Classes

- constructors
  - Integer ({int, String})
- some methods
  - double doubleValue ()
  - equals
  - static int parseInt (String {, int})
  - valueOf (String {, int})
  - hashCode
    - most classes should provide this method
    - used when storing class into a HashTable

Java Utility Classes

- Vector
  - dynamic array of *Objects*
    - grows & shrinks as required
    - essential for dealing with users…
  - essentially a type wrapper for arrays
    - provides extended functionality
      - e.g. enumerations
  - type unsafe
    - can add T1 and retrieve as T99…
      - can lead to runtime abort
    - creating T1Vector (extends Vector) solves problem

Java Utility Classes

```
class T1Vector extends Vector
  {
  public void set (T1 item, int n)
    {
    setElementAt (item, n);
    }

  public T1 get (int n)
    {
    return ((T1) elementAt (n));
    }
  }
```

asserts that the dynamic type retrieved is a T1, not a generic Object (the static type)

Java Utility Classes

- constructors
  - Vector (int {, int})
- some methods
  - int capacity ()
  - void {set, insert}ElementAt (Object, int)
  - void removeElementAt (int)
  - int indexOf (Object)
  - Object elementAt (int)
  - void trimToSize ()
  - Enumeration elements ()
    - allows all elements to be accessed in some (arbitrary) order

Java Utility Classes

- Enumeration
  - interface
    - 2 methods
      - boolean hasMoreElements ()
      - Object nextElement ()
  - traverse data structure
    - no need to make implementation public
  - only go ‘forward’, not ‘backward’
    - but order not defined
  - can’t be reset

```
Vector v = …;
Enumeration e = v.elements ();
while (e.hasMoreElements ())
  {
  AnObj a = (AnObj) e.nextElement ();
  … use a
  }
```

Java Utility Classes

- HashTable
  - associative array; extends java.util.Dictionary abstract class
  - some methods
    - Object put (Object, Object)
      - associate key with value
    - Object get (Object)
      - retrieve value given key
    - remove ()
    - keys (), elements ()
      - allow enumeration of the table
    - isEmpty ()
    - containskey (Object)

Java Utility Classes

- Properties
  - extends HashTable
    - allows I/O of key/value pairs
  - useful for specifying configuration info.
    - user programs

```
Properties defaults = new Properties ();
defaults.put ("FONT", "Courier"); defaults.put ("SIZE", "10");

defaults.save (defaultsFileOutputStream, "application defaults");

# application defaults
# Sat May 03 20:05:33 1997
FONT=Courier
SIZE=10
```

    - System.getPropert{y, ies) ()

Java Utility Classes

- Date
  - mostly deprecated; now use Calendar
    - *"Unfortunately, the API for these functions was not amenable to internationalization."*
  - Date ()
    - constructor—current time&date
    - several others available
  - getYear ()
    - years since 1900
  - parse ()
    - flexible: accepts many formats
    - millisecs since January 1, 1970, 00:00:00 GMT

Java Utility Classes

- Calendar
  - slightly more complex than expected
    - accounts for locale/timezone/calendar differences
  - abstract, so can't be instantiated
    - many static constants/members
    - getInstance ()
      - 'factory' method
    - get () / set () methods
      - HOUR, DAY_OF_WEEK, AM, WEEK_OF_MONTH, etc.
      - UNDECIMBER
        - lunar calendar's 13th month
    - date arithmetic
      - before (), after (), add (), etc.

Java Utility Classes

- output handled by java.text.SimpleDateFormat

```
// prints 1996.07.10 AD at 15:08:56 PDT
SimpleDateFormat formatter =
  new SimpleDateFormat ("yyyy.MM.dd G 'at' hh:mm:ss z");
System.out.println (formatter.format(new Date ()));
```

- input handled by parse () method
  - converts String
- various different calendars potentially catered for
  - GregorianCalendar extends Calendar
    - isLeapYear ()
    - getGregorianChange ()
      - 15th August 1582

Java Utility Classes

- Some other utilities
  - Random
  - Stack
  - Bitset
  - Observable class & Observer interface
    - implements Model, View, Controller concept
      - used heavily by ‘swing’
    - observable objects notify observers (via their update method) when they change state
    - {add, delete}Observer (Observer)

Java Utility Classes