The Fragile Base Class Problem
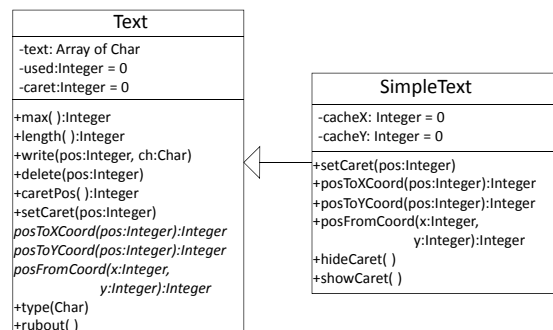
55:182/22C:182

Spring, 2011

Inheritance--A Simple Classification

- Subclassing
  - inheritance of implementation fragments/code from a superclass.
- Interface Inheritance
  - inheritance of contract fragments/interfaces.

The Complexities of Subclassing

- Methods of a class may freely invoke each other.
- Subclasses may override inherited methods.
- Subclass methods may call methods of superclasses, including overridden superclass methods.
- This is actually a form of "callback" from subclass to superclass.
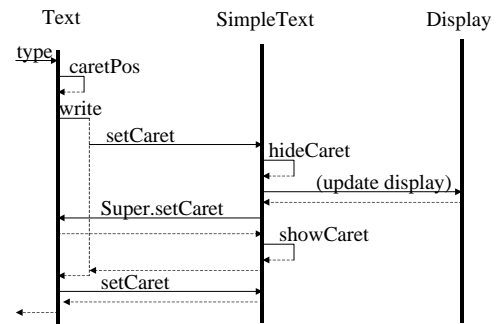

Inheritance Issues Example

Text

-text: Array of Char
-used:Integer = 0
-caret:Integer = 0

+max( ):Integer
+length( ):Integer
+write(pos:Integer, ch:Char)
+delete(pos:Integer)
+caretPos( ):Integer
+setCaret(pos:Integer)
posToXCoord(pos:Integer):Integer
posToYCoord(pos:Integer):Integer
posFromCoord(x:Integer,
            y:Integer):Integer
+type(Char)
+rubout( )

SimpleText

-cacheX: Integer = 0
-cacheY: Integer = 0

+setCaret(pos:Integer)
+posToXCoord(pos:Integer):Integer
+posToYCoord(pos:Integer):Integer
+posFromCoord(x:Integer,
            y:Integer):Integer
+hideCaret( )
+showCaret( )

Example--Continued

```
abstract class Text {                  class SimpleText extends Text {
         .                                      .
         .                                      .
         .                                      .
 private int caret = 0;                  void setCaret(int pos) {
         .                                int old = caretPos( );
         .                                if (old != pos) {
         .                                 hideCaret( );
 void setCaret(int pos)  {                super.setCaret(pos);
  caret = pos;                            showCaret( );
 }                                       }
         .                                      .
         .                                      .
         .                                      .
}                                      }
```

Example--Continued

Interaction diagram resulting from call to method **type** of Text class:

Example--Continued

A new version of Text class that "breaks" the subclass SimpleText:

```
abstract class Text  {
         .
         .
         .
 void write (int pos, char ch) {
  int i;
  for ( i = used; i > pos; i--)
    text[i] = text[i-1];
  used = used + 1;
  if (caret >= pos)
    caret = caret +1;
  text[pos] = ch;
 }
         ...
```

Inheritance Issues--The Fragile Base Class Problem

- There is an implicit interface between a class and its ancestor classes (superclasses).
  - Syntactic aspect--Does a class need to be recompiled due to purely syntactic changes among it superclasses?
  - Semantic Aspect--How dependent is a subclass upon changes in the implementation of its superclasses?

Dealing With Class-Subclass Dependencies

- Specialization Interface
  - Interface between a class and its subclasses
  - For Java and C++, the specialization interface consists of the public and protected interface of the superclass.
- Various methods have been proposed to control behavior across a specialization interface, but these are largely of theoretical interest.