

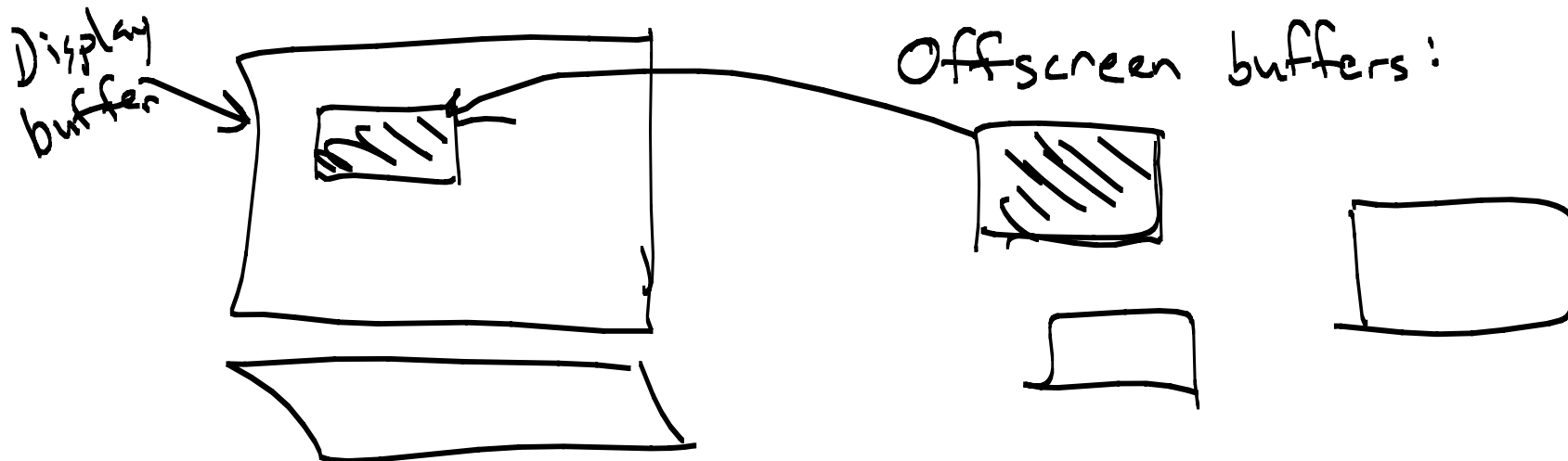
Custom Components, Testing, Inner Classes

Note Title

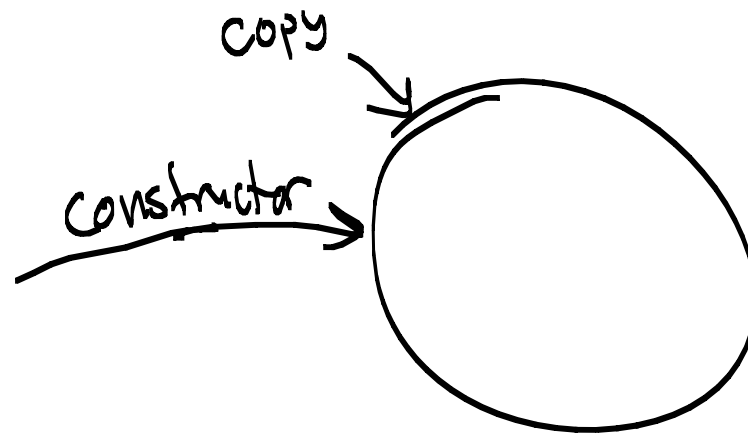
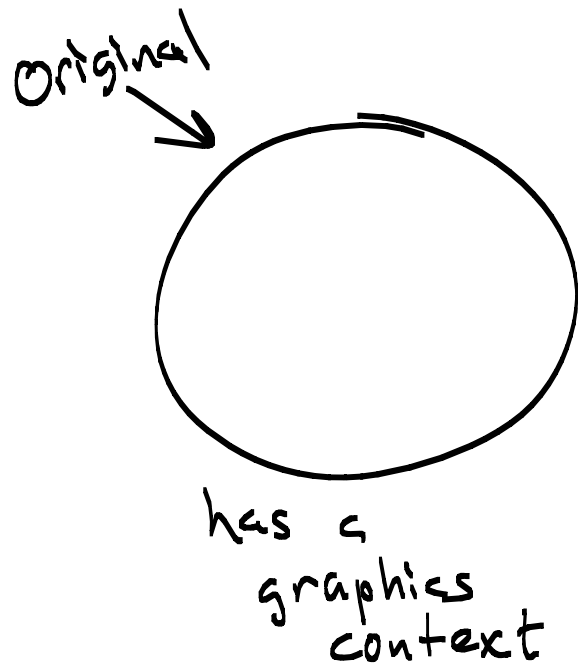
2/22/2007

Graphics Context — an object that will paint for you

Buffers — 2D raster of pixels



Goal: Custom component that paints a copy of a given component



same-sized
Create a buffer
copy component
into the buffer

painting the copy:
painting the buffer
at the right size

Testing hierarchies of classes

Applicable to BlackBox Testing — use specification to generate test cases

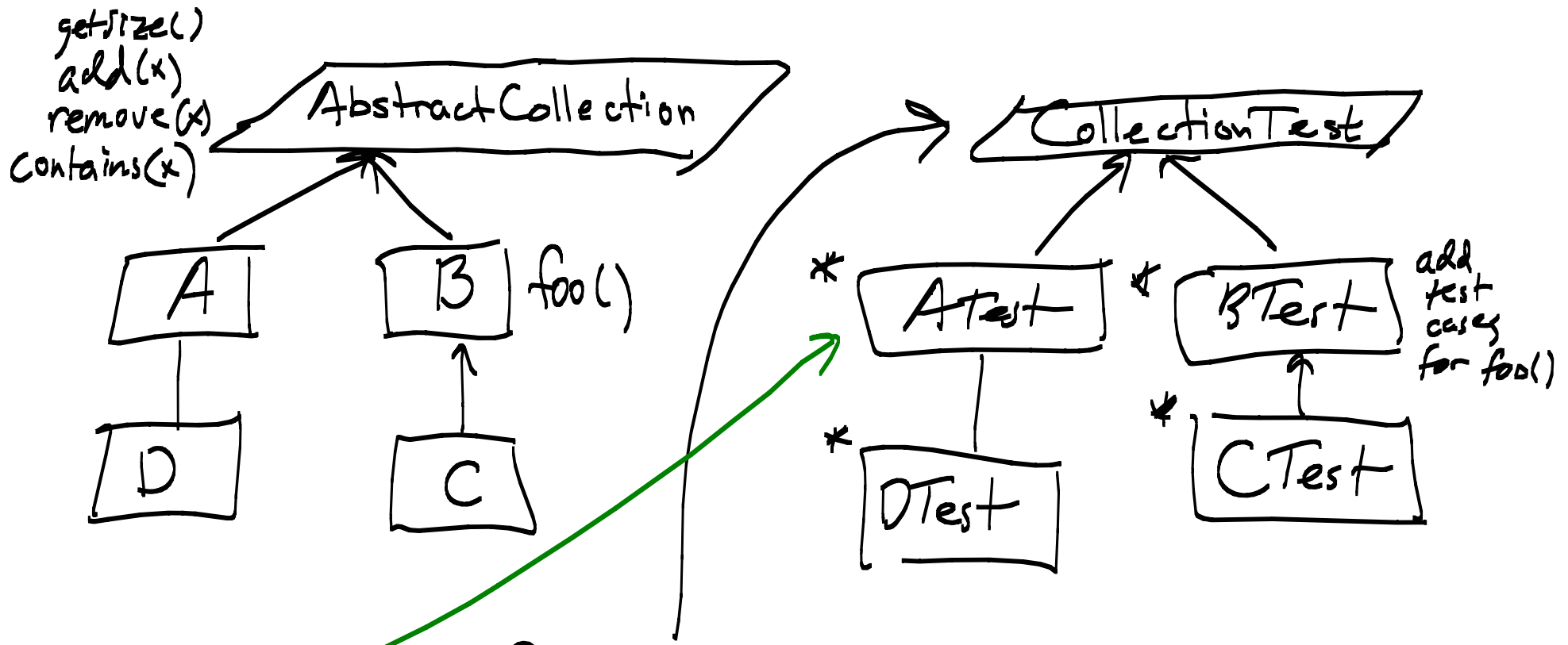
GlassBox Testing — use the code to drive test cases

* Leverage existing test cases

by extending them —

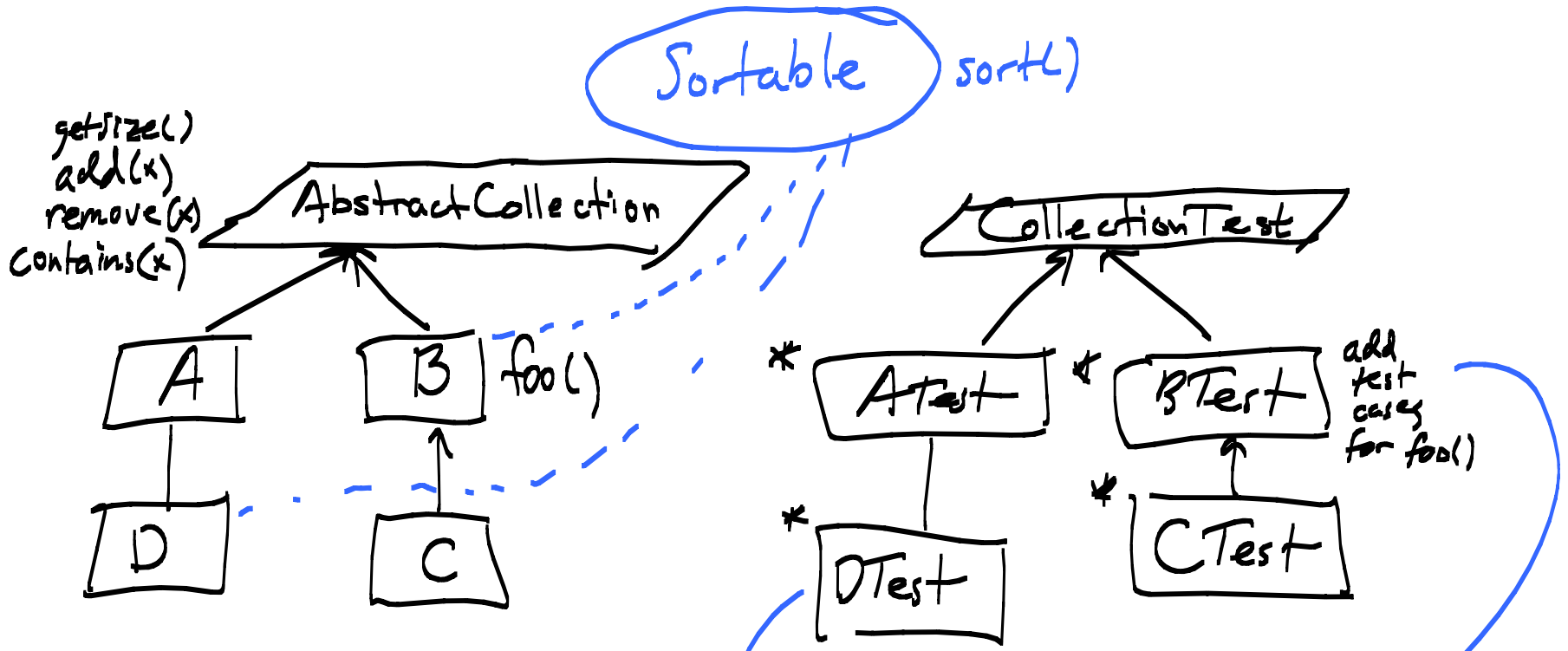
parallels class hierarchy of classes under test

"code coverage"



- ① factory method to create test instance
`public abstract AbstractCollection getTestInstance(),`
- ② several tests for the various methods of
`AbstractCollection`

```
public A getTestInstance() { return new A(); }
```



```
@Test
public void testSortable() {
    // D also
    sortTest(getTestInstance());
}
```

Since B implements Sortable, call "canned" tests for the sortable interface

Inner Classes

- ① static nested class (not really inner classes)
treated like top-level classes, but the name is
OuterClassName.Foo ⇒ No special privileges
Use static members of containing class
- ② Member class — not static
declared inside another class
each has an implicit reference to a containing instance
the compiler adds a hidden parameter to all constructors +
a hidden instance variable
- ③ local class
class defined inside a method
- ④ anonymous class
just like local classes, but they have no name!
declared inside of a statement

```
class MyCollection {  
    static DUMMY_TAIL = ...  
    Object[] a = ...
```

→ static class MyIterator { nested class
 use DUMMY_TAIL (static)
 can't use instance variables
 int currentLocation; Type: MyCollection.MyIterator
 can't use a (instance variable)
}

```
class MemberIterator {  
    can use a — have an implicit reference to  
                    an instance of MyCollection  
    MyCollection.this ≡ the containing object
```

If want to create an instance of a member class from outside the code of the containing class

==

In Foo:

MyCollection coll = ...

MyCollection.MemberIterator it =

coll.new MemberIterator(i);

↖ implicit parameter

↗
RARE!

Local Class:

```
Iterator iterator() {  
    [final int i = 0;  
    [final Foo f = new Foo(...);  
    class MyIterator {  
        ;  
    }  
    → return new MyIterator();  
}
```

can see `i` (because it's final)
copied into instances of this class

Anonymous class:

```
Iterator iterator() {  
    return new Iterator() {  
        public boolean hasNext() { ... }  
        ;  
    }  
}
```