

Week 3

Object Interaction 2

Creating cooperating objects

suggested reading:

Textbook, Ch. 3

Objects creating objects

```
public class ClockDisplay
{
    private NumberDisplay hours;
    private NumberDisplay minutes;
    private String displayString;

    public ClockDisplay()
    {
        hours = new NumberDisplay(24) ;
        minutes = new NumberDisplay(60) ;
        ...
    }
}
```

Objects creating objects

in class ClockDisplay:

```
hours = new NumberDisplay(24) ;
```



actual parameter

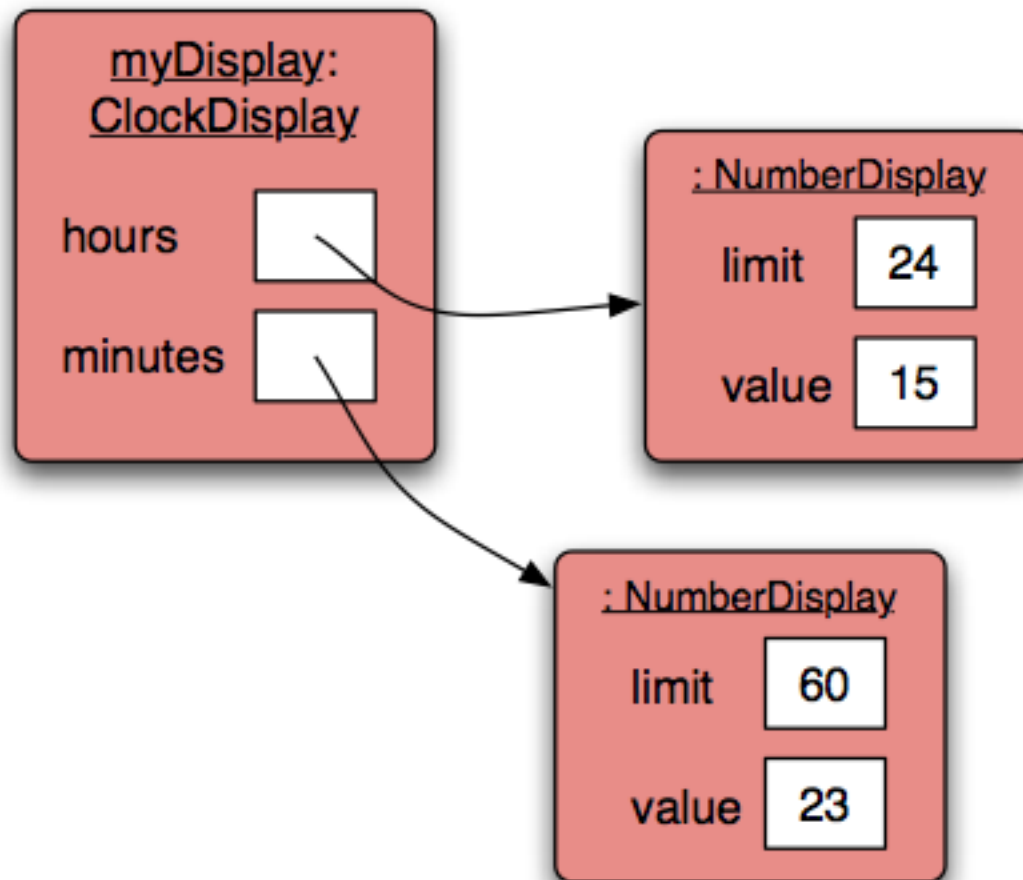
in class NumberDisplay:

```
public NumberDisplay(int rolloverLimit) ;
```



formal parameter

ClockDisplay object diagram



Method calling

```
public void timeTick()  
{  
    minutes.increment();  
    if(minutes.getValue() == 0) {  
        // it just rolled over!  
        hours.increment();  
    }  
    updateDisplay();  
}
```

External method call

- external method calls

`minutes.increment();`

object . methodName (parameter-list)

Internal method call

- internal method calls

`updateDisplay() ;`

- No variable name is required.
- **this**
 - could be used as a reference to the invoking object, but not used for method calls.

Internal method

```
/**
 * Update the internal string that
 * represents the display.
 */
private void updateDisplay()
{
    displayString =
        hours.getDisplayValue() + ":" +
        minutes.getDisplayValue();
}
```


Method calls

- ‘Internal’ means ‘this object’.
- ‘External’ means ‘any other object’, regardless of its type.
- NB: A method call on another object of the same type would be an external call.

null

- `null` is a special value in Java
- Object fields are initialized to `null` by default.
- You can test for and assign `null`:

```
private NumberDisplay hours;
```

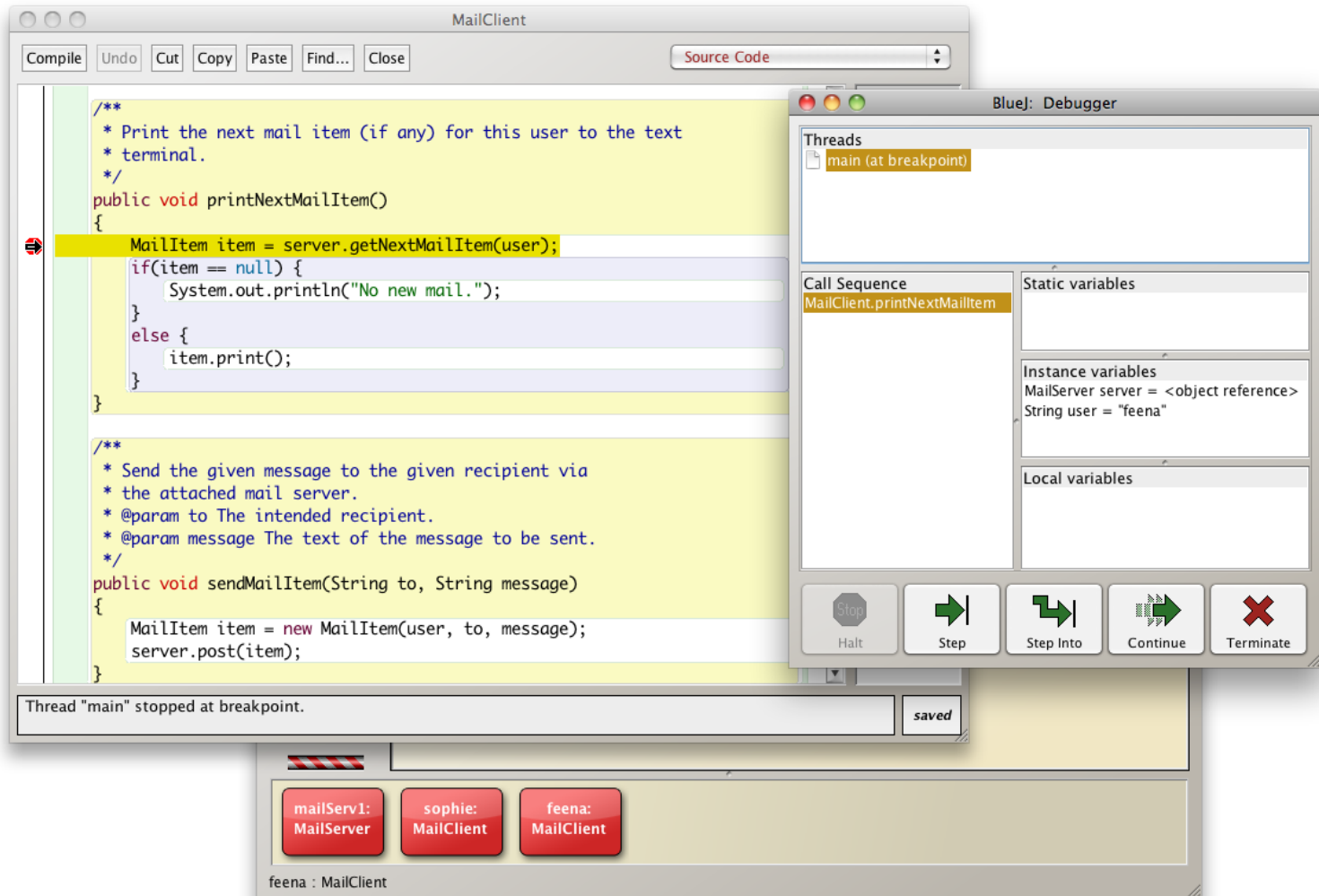
```
if(hours != null) { ... }
```

```
hours = null;
```

The debugger

- Useful for gaining insights into program behavior ...
- ... whether or not there is a program error.
- Set breakpoints.
- Examine variables.
- Step through code.

The debugger



Concept summary

- object creation
- overloading
- internal/external method calls
- debugger