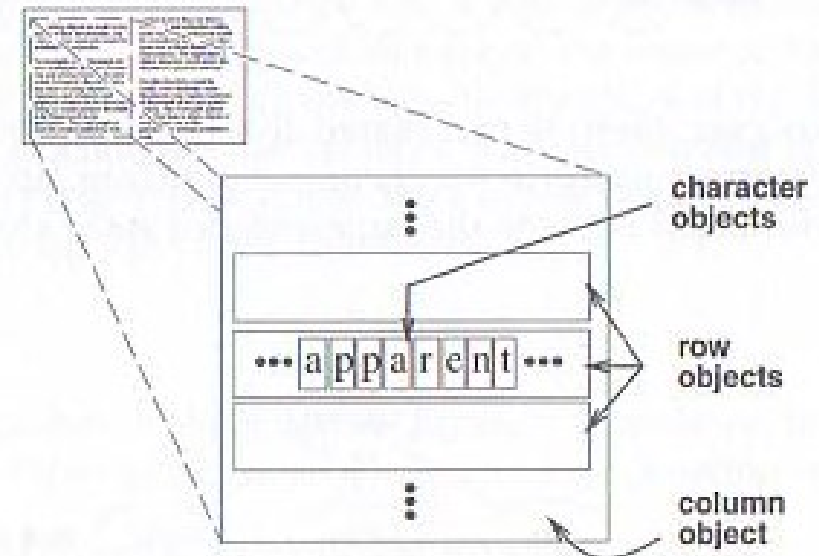
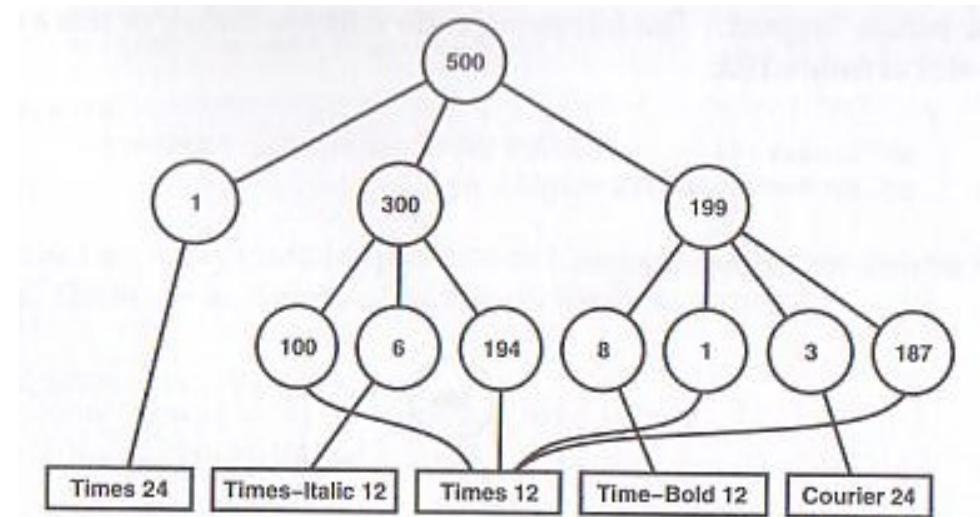
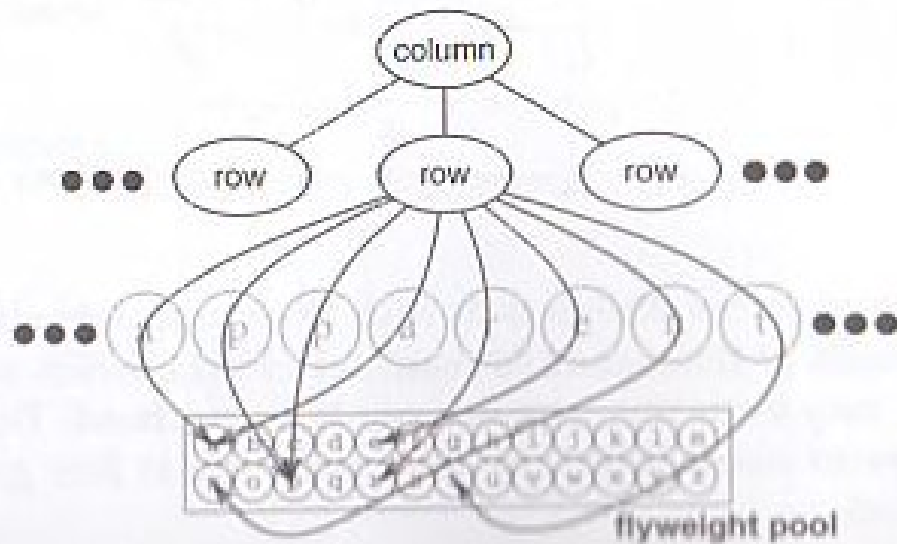
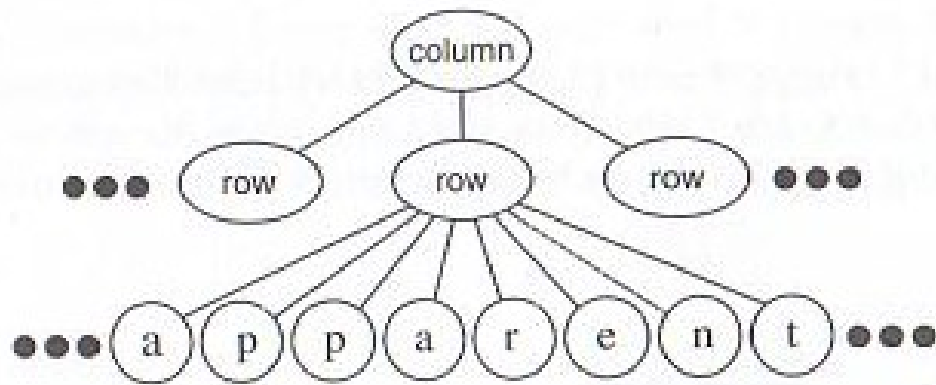


Plan for Today

- Template pattern
- Interpreter pattern
- Observer pattern

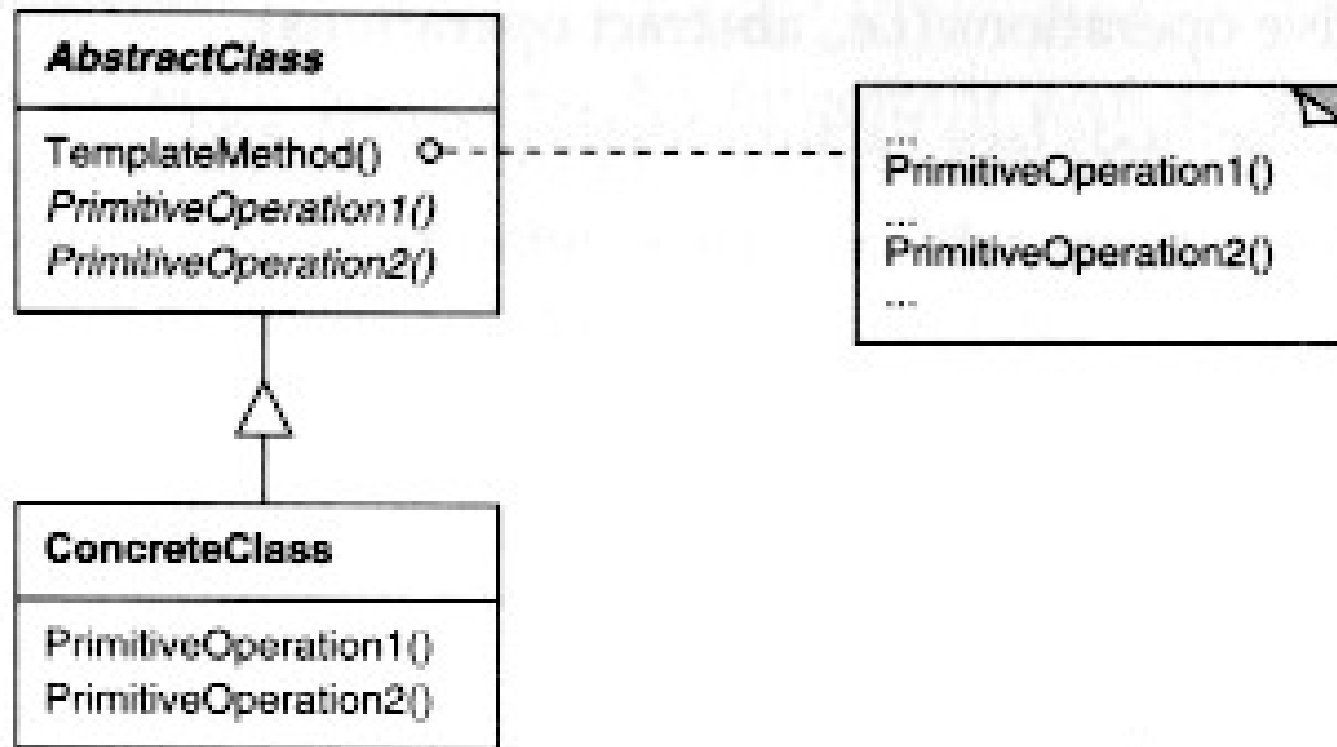
Lecture 52 – 4 December

Task 7 Questions?




Template Pattern

- Defines skeleton of algorithm in operation, deferring some steps to subclasses. Lets subclasses redefine certain steps of algorithm without changing its structure



Template Pattern

- Informal contract for staging calls to delegated implementations
 - analogous to Command pattern at code level
- Higher-level plug-and-play; e.g., introduction + body + conclusion
 - typical
 - contractually specify behavior on implementation
 - the plug-and-play *what* (at low “primitive” conceptual access level)
 - template
 - augment typical contract with (runtime) agreement on usage of behavior
 - *how* the *what* will be called (at higher “composite” conceptual access level):
 1. `_open`
 2. `_initialize`
 3. `_start`
 4. `_run`
 5. `_stop`
 6. `_close`

possibly some sharable elements; reduces code duplication

Template Pattern

- Similar to macro:

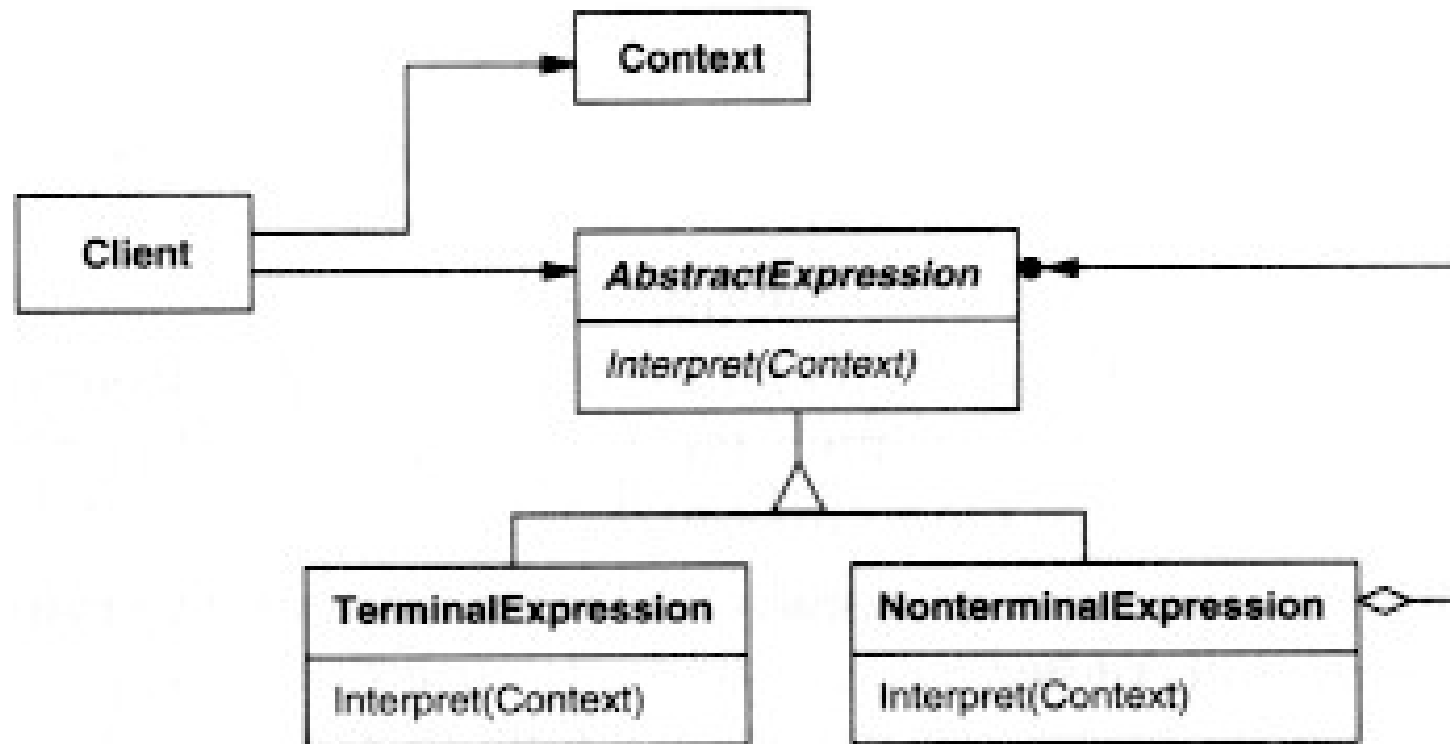
```
interface I_Viewer {
    // macros
    void _doOpenFile();
    void _doCloseFile();
    void _doExit();

    // callbacks / hooks
    void _handleAboutToOpen();
    void _handleOpening(double percentDone);
    void _handleDoneOpening();
}

public class Viewer implements I_Viewer {
    public void doOpenFile() {
        I_Document document = Document._requestDocument(this); // filename?
        document._doOpen();    // minimize handling of Document data
        document._doRead();
        document._doRender();
        document._doClose();
    }
    ...
    public void _handleAboutToOpen() { ... } // optional reactions
    public void _handleOpening(double percentDone) { ... }
    public void _handleDoneOpening() { ... }
}
```

Interpreter Pattern

- Defines representation based on language/grammar along with decoder that uses representation to interpret sentences in language



```

evaluate("define sensor radar          SENSOR_RADAR1      with field of view 30 power 21 sensitivity 1.5");
evaluate("define sensor thermal        SENSOR_THERMAL1    with field of view 31 sensitivity 2.5");
evaluate("define sensor acoustic       SENSOR_ACOUSTIC1    with sensitivity 3.5");
evaluate("define sensor sonar active   SENSOR_SONARACTIVE1 with power 32 sensitivity 4.5");
evaluate("define sensor sonar passive  SENSOR_SONARPASSIVE1 with sensitivity 5.5");
evaluate("define sensor depth          SENSOR_DEPTH1      with trigger depth -200");
evaluate("define sensor distance       SENSOR_DISTANCE1    with trigger distance 20");
evaluate("define sensor time           SENSOR_TIME1        with trigger time 23.1");

evaluate("define sensor radar          FUZE_RADAR1        with field of view 32 power 23 sensitivity 10.5");
evaluate("define sensor thermal        FUZE_THERMAL1      with field of view 33 sensitivity 10.5");
evaluate("define sensor acoustic       FUZE_ACOUSTIC1      with sensitivity 15.5");
evaluate("define sensor sonar active   FUZE_SONARACTIVE1  with power 34 sensitivity 15");
evaluate("define sensor sonar passive  FUZE_SONARPASSIVE1 with sensitivity 14");
evaluate("define sensor depth          FUZE_DEPTH1        with trigger depth -250");
evaluate("define sensor distance       FUZE_DISTANCE1     with trigger distance 25");
evaluate("define sensor time           FUZE_TIME1         with trigger time 25.1");

evaluate("define munition bomb          MUNITION_BOMB1");
evaluate("define munition shell         MUNITION_SHELL1");

evaluate("define munition depth_charge  MUNITION_DEPTHCHARGE1 with fuze FUZE_ACOUSTIC1");
evaluate("define munition depth_charge  MUNITION_DEPTHCHARGE2 with fuze FUZE_SONARACTIVE1");
evaluate("define munition depth_charge  MUNITION_DEPTHCHARGE3 with fuze FUZE_SONARPASSIVE1");
evaluate("define munition depth_charge  MUNITION_DEPTHCHARGE4 with fuze FUZE_DEPTH1");
evaluate("define munition depth_charge  MUNITION_DEPTHCHARGE5 with fuze FUZE_TIME1");

evaluate("define munition torpedo       MUNITION_TORPEDO1  with sensor SENSOR_ACOUSTIC1 fuze FUZE_ACOUSTIC1 arming time 1.1");
evaluate("define munition torpedo       MUNITION_TORPEDO2  with sensor SENSOR_ACOUSTIC1 fuze FUZE_SONARACTIVE1 arming time 1.2");
evaluate("define munition torpedo       MUNITION_TORPEDO3  with sensor SENSOR_ACOUSTIC1 fuze FUZE_SONARPASSIVE1 arming time 1.3");
evaluate("define munition torpedo       MUNITION_TORPEDO4  with sensor SENSOR_ACOUSTIC1 fuze FUZE_DISTANCE1 arming time 1.4");
evaluate("define munition torpedo       MUNITION_TORPEDO5  with sensor SENSOR_ACOUSTIC1 fuze FUZE_TIME1 arming time 1.5");

evaluate("define munition torpedo       MUNITION_TORPEDO6  with sensor SENSOR_SONARACTIVE1 fuze FUZE_ACOUSTIC1 arming time 2.1");
evaluate("define munition torpedo       MUNITION_TORPEDO7  with sensor SENSOR_SONARACTIVE1 fuze FUZE_SONARACTIVE1 arming time 2.2");
evaluate("define munition torpedo       MUNITION_TORPEDO8  with sensor SENSOR_SONARACTIVE1 fuze FUZE_SONARPASSIVE1 arming time 2.3");
evaluate("define munition torpedo       MUNITION_TORPEDO9  with sensor SENSOR_SONARACTIVE1 fuze FUZE_DISTANCE1 arming time 2.4");
evaluate("define munition torpedo       MUNITION_TORPEDO10 with sensor SENSOR_SONARACTIVE1 fuze FUZE_TIME1 arming time 2.5");

evaluate("define munition torpedo       MUNITION_TORPEDO11 with sensor SENSOR_SONARPASSIVE1 fuze FUZE_ACOUSTIC1 arming time 3.1");
evaluate("define munition torpedo       MUNITION_TORPEDO12 with sensor SENSOR_SONARPASSIVE1 fuze FUZE_SONARACTIVE1 arming time 3.2");
evaluate("define munition torpedo       MUNITION_TORPEDO13 with sensor SENSOR_SONARPASSIVE1 fuze FUZE_SONARPASSIVE1 arming time 3.3");
evaluate("define munition torpedo       MUNITION_TORPEDO14 with sensor SENSOR_SONARPASSIVE1 fuze FUZE_DISTANCE1 arming time 3.4");
evaluate("define munition torpedo       MUNITION_TORPEDO15 with sensor SENSOR_SONARPASSIVE1 fuze FUZE_TIME1 arming time 3.5");

evaluate("define munition missile       MUNITION_MISSILE1  with sensor SENSOR_RADAR1 fuze FUZE_RADAR1 arming distance 1");
evaluate("define munition missile       MUNITION_MISSILE2  with sensor SENSOR_RADAR1 fuze FUZE_THERMAL1 arming distance 2");

evaluate("define munition missile       MUNITION_MISSILE3  with sensor SENSOR_THERMAL1 fuze FUZE_RADAR1 arming distance 3");
evaluate("define munition missile       MUNITION_MISSILE4  with sensor SENSOR_THERMAL1 fuze FUZE_THERMAL1 arming distance 4");

evaluate("define airplane ACTOR_AIRPLANE1 with munition (MUNITION_BOMB1 MUNITION_SHELL1 MUNITION_DEPTHCHARGE1 MUNITION_DEPTHCHARGE2 "+
"  MUNITION_DEPTHCHARGE3 MUNITION_DEPTHCHARGE4 MUNITION_DEPTHCHARGE5 MUNITION_TORPEDO1 MUNITION_TORPEDO2 MUNITION_TORPEDO3 "+
"  MUNITION_TORPEDO4 "+ MUNITION_TORPEDO5 MUNITION_TORPEDO6 MUNITION_TORPEDO7 MUNITION_TORPEDO8 MUNITION_TORPEDO9 MUNITION_TORPEDO10 "+
"  MUNITION_TORPEDO11 MUNITION_TORPEDO12 MUNITION_TORPEDO13 MUNITION_TORPEDO14 MUNITION_TORPEDO15 MUNITION_MISSILE1 MUNITION_MISSILE2 "+
"  MUNITION_MISSILE3 MUNITION_MISSILE4)");

evaluate("create actor MY_AIRPLANE1 from ACTOR_AIRPLANE1 at 49*39'30#/117*25'33#/0 with course 300 speed 100");
evaluate("create actor MY_SHIP1 from ACTOR_SHIP1 at 49*39'31#/117*25'34#/0 with course 301 speed 101");
evaluate("create actor MY_SUBMARINE1 from ACTOR_SUBMARINE1 at 49*39'32#/117*25'35#/0 with course 302 speed 102");

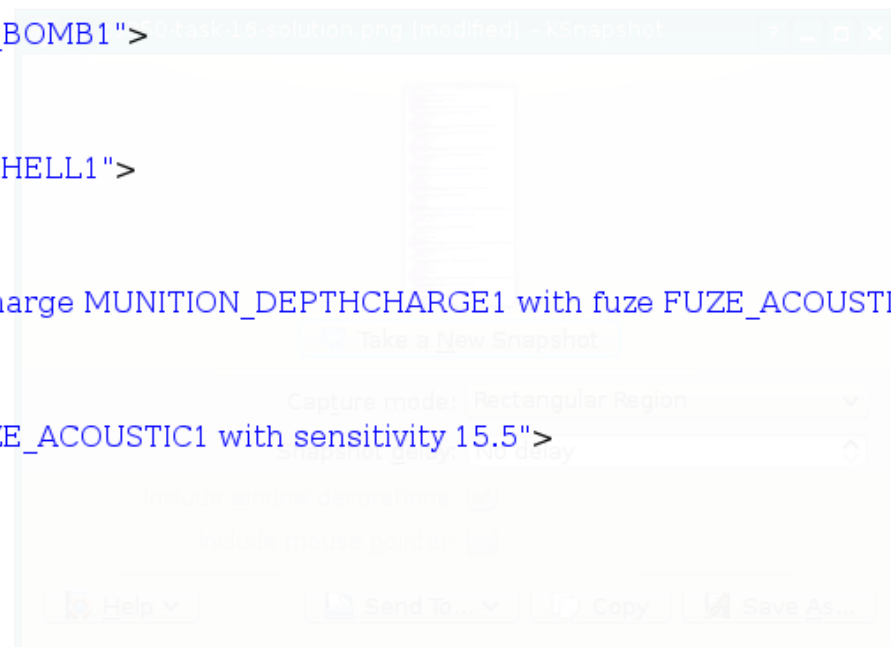
```



```

- <CarrierActorAirplane text="define airplane ACTOR_AIRPLANE1 with munition (MUNITION_BOMB1 MUNITION_SHELL1
MUNITION_DEPTHCHARGE1 MUNITION_DEPTHCHARGE2 MUNITION_DEPTHCHARGE3 MUNITION_DEPTHCHARGE4
MUNITION_DEPTHCHARGE5 MUNITION_TORPEDO1 MUNITION_TORPEDO2 MUNITION_TORPEDO3 MUNITION_TORPEDO4
MUNITION_TORPEDO5 MUNITION_TORPEDO6 MUNITION_TORPEDO7 MUNITION_TORPEDO8 MUNITION_TORPEDO9
MUNITION_TORPEDO10 MUNITION_TORPEDO11 MUNITION_TORPEDO12 MUNITION_TORPEDO13 MUNITION_TORPEDO14
MUNITION_TORPEDO15 MUNITION_MISSILE1 MUNITION_MISSILE2 MUNITION_MISSILE3 MUNITION_MISSILE4)">
  <actor-id> ACTOR_AIRPLANE1 </actor-id>
- <munitions>
  - <CarrierMunitionBomb text="define munition bomb MUNITION_BOMB1">
    <munition-id> MUNITION_BOMB1 </munition-id>
    <munition-bomb/>
  </CarrierMunitionBomb>
  - <CarrierMunitionShell text="define munition shell MUNITION_SHELL1">
    <munition-id> MUNITION_SHELL1 </munition-id>
    <munition-shell/>
  </CarrierMunitionShell>
  - <CarrierMunitionDepthCharge text="define munition depth_charge MUNITION_DEPTHCHARGE1 with fuze FUZE_ACOUSTIC1">
    <munition-id> MUNITION_DEPTHCHARGE1 </munition-id>
    - <munition-depth-charge>
      - <fuze>
        - <CarrierSensorAcoustic text="define sensor acoustic FUZE_ACOUSTIC1 with sensitivity 15.5">
          <sensor-id> FUZE_ACOUSTIC1 </sensor-id>
          - <sensor-acoustic>
            <sensitivity> 15.5 </sensitivity>
          </sensor-acoustic>
        </CarrierSensorAcoustic>
      </fuze>
    </munition-depth-charge>
  </CarrierMunitionDepthCharge>
  - <CarrierMunitionDepthCharge text="define munition depth_charge MUNITION_DEPTHCHARGE2 with fuze FUZE_SONARACTIVE1">
    <munition-id> MUNITION_DEPTHCHARGE2 </munition-id>
    - <munition-depth-charge>
      - <fuze>
        - <CarrierSensorSonarActive text="define sensor sonar active FUZE_SONARACTIVE1 with power 34 sensitivity 15">
          <sensor-id> FUZE_SONARACTIVE1 </sensor-id>
          - <sensor-sonar-active>
            <power> 34.0 </power>
            <sensitivity> 15.0 </sensitivity>
          </sensor-sonar-active>
        </CarrierSensorSonarActive>
      </fuze>
    </munition-depth-charge>
  </CarrierMunitionDepthCharge>

```



```

"define munition bomb MUNITION_BOMB1"
"define airplane ACTOR_AIRPLANE1 with munition (MUNITION_BOMB1)"

"create actor MY_AIRPLANE1 from ACTOR_AIRPLANE1 at 49*39'30#/117*25'33#/1000 with
  course 300 speed 100"
"set MY_AIRPLANE1 load munition MUNITION_BOMB1"
"set MY_AIRPLANE1 deploy munition MY_AIRPLANE1.MUNITION_BOMB1"

<CarrierActorAirplane text="define airplane ACTOR_AIRPLANE1 with munition (MUNITION_BOMB1)">
  <actor-id>
    ACTOR_AIRPLANE1
  </actor-id>
  <munitions>
    <CarrierMunitionBomb text="define munition bomb MUNITION_BOMB1">
      <munition-id>
        MUNITION_BOMB1
      </munition-id>
      <munition-bomb/>
    </CarrierMunitionBomb>
  </munitions>
</CarrierActorAirplane>

```

```

"define sensor acoustic FUZE_ACOUSTIC1 with sensitivity 15.5"
"define munition depth_charge MUNITION_DEPTHCHARGE1 with fuze FUZE_ACOUSTIC1"
"define airplane ACTOR_AIRPLANE1 with munition (MUNITION_DEPTHCHARGE1)"

"create actor MY_AIRPLANE1 from ACTOR_AIRPLANE1 at 49*39'30#/117*25'33#/1000
with course 300 speed 100"
"set MY_AIRPLANE1 load munition MUNITION_DEPTHCHARGE1"
"set MY_AIRPLANE1 deploy munition MY_AIRPLANE1.MUNITION_DEPTHCHARGE1"

```

```

<CarrierActorAirplane text="define airplane ACTOR_AIRPLANE1 with munition (MUNITION_DEPTHCHARGE1)">
  <actor-id>
    ACTOR_AIRPLANE1
  </actor-id>
  <munitions>
    <CarrierMunitionDepthCharge text="define munition depth_charge MUNITION_DEPTHCHARGE1 with fuze FUZE_ACOUSTIC1">
      <munition-id>
        MUNITION_DEPTHCHARGE1
      </munition-id>
      <munition-depth-charge>
        <fuze>
          <CarrierSensorAcoustic text="define sensor acoustic FUZE_ACOUSTIC1 with sensitivity 15.5">
            <sensor-id>
              FUZE_ACOUSTIC1
            </sensor-id>
            <sensor-acoustic>
              <sensitivity>
                15.5
              </sensitivity>
            </sensor-acoustic>
          </CarrierSensorAcoustic>
        </fuze>
      </munition-depth-charge>
    </CarrierMunitionDepthCharge>
  </munitions>
</CarrierActorAirplane>

```

```

"define sensor acoustic SENSOR_ACOUSTIC1 with sensitivity 3.5"
"define sensor acoustic FUZE_ACOUSTIC1 with sensitivity 15.5"
"define munition torpedo MUNITION_TORPEDO1 with sensor SENSOR_ACOUSTIC1
  fuze FUZE_ACOUSTIC1 arming time 1.1"
"define airplane ACTOR_AIRPLANE1 with munition (MUNITION_TORPEDO1)"
"create actor MY_AIRPLANE1 from ACTOR_AIRPLANE1 at 49*39'30#/117*25'33#/1000
  with course 300 speed 100"
"set MY_AIRPLANE1 load munition MUNITION_TORPEDO1"
"set MY_AIRPLANE1 deploy munition MY_AIRPLANE1.MUNITION_TORPEDO1"

```

```

<CarrierActorAirplane text="define airplane ACTOR_AIRPLANE1 with munition (MUNITION_TORPEDO1)">
  <actor-id>
    ACTOR_AIRPLANE1
  </actor-id>
  <munitions>
    <CarrierMunitionTorpedo text="define munition torpedo MUNITION_TORPEDO1 with sensor SENSOR_ACOUSTIC1 fuze FUZE_ACOUSTIC1 arming time 1.1">
      <munition-id>
        MUNITION_TORPEDO1
      </munition-id>
      <munition-torpedo>
        <sensor>
          <CarrierSensorAcoustic text="define sensor acoustic SENSOR_ACOUSTIC1 with sensitivity 3.5">
            <sensor-id>
              SENSOR_ACOUSTIC1
            </sensor-id>
            <sensor-acoustic>
              <sensitivity>
                3.5
              </sensitivity>
            </sensor-acoustic>
          </CarrierSensorAcoustic>
        </sensor>
        <fuze>
          <CarrierSensorAcoustic text="define sensor acoustic FUZE_ACOUSTIC1 with sensitivity 15.5">
            <sensor-id>
              FUZE_ACOUSTIC1
            </sensor-id>
            <sensor-acoustic>
              <sensitivity>
                15.5
              </sensitivity>
            </sensor-acoustic>
          </CarrierSensorAcoustic>
        </fuze>
      </munition-torpedo>
    </CarrierMunitionTorpedo>
  </munitions>
</CarrierActorAirplane>

```

Interpreter Pattern

- Old task
 - spaceship controller language:

Command	Field 2	Field 3	Field 4	Field 5	Description
A	<i>id</i>	<i>x</i>	<i>y</i>	<i>angle</i>	adds spaceship <i>id</i> at position <i>x,y</i> with orientation <i>angle</i>
D	<i>id</i>				removes spaceship <i>id</i>
S	<i>id</i>	<i>state</i>			brings spaceship <i>id</i> online for state B or takes it offline for state T
L	<i>id</i>				rotates spaceship <i>id</i> counterclockwise one step
R	<i>id</i>				rotates spaceship <i>id</i> clockwise one step
T	<i>id</i>				thrusts spaceship <i>id</i> forward one unit
I	<i>count</i>				idles <i>count</i> ticks

```
controller.execute("A dog 2 3 45");  
controller.execute("A cat 5 6 -90");
```

```
controller.execute("R dog");  
controller.execute("L cat");  
controller.execute("T dog");  
controller.execute("T cat");
```

```
controller.execute("D dog");  
controller.execute("D cat");
```

Interpreter Pattern

- Textbook
 - read readable parts of pages 243-255
 - skim over syntax trees
 - CS481 Compilers and Lexical Analysis
 - simpler implementations possible

Interpreter Pattern

- Conventional behavioral control
 - method calls
 - direct (“wired”)
 - indirect (“wireless”)
 - one-to-one mapping: one method per unit of functionality
- Interpreter pattern
 - complex “portal” adapter
 - many-to-one mapping: one method for all (most) units of functionality
 - miniature command language
 - conventional: `lamp.setBrightness(5);`
 - interpreter: `lamp.execute(“set brightness 5”);`

decouples network protocol



pros: very flexible and extensible; compile-time validation

cons: additional complexity and overhead; run-time validation

Interpreter Pattern

- Highly useful plug-and-play pattern
 - powerful
 - flexible
 - adaptable
 - extensible
- Consider contemporary games
 - command-line interface
 - bots, cheats, codes, cons, console enhancements, DLLs, drivers, expansion packs, exploits, hacks, maps, mods, patches, rigging, tampering, tweaks, upgrades, wads, looking at my screen
- Decoupled control
 - command-abstraction language
 - widely varying input devices
 - reproducibility in testing

Interpreter Pattern

- Implementation
 - customary order
 1. syntax: what a command *says*
 2. semantics: what a command *means*
 - opposite order more appropriate here
 - semantics first: what do we want to convey?
 - syntax last: how should we convey it?
 - simple and easy
 - complex and hard (see book discussion)

Interpreter Pattern

- Later lab uses flexible string-based representation (without arguments):

```
public static final String COMMAND_TURN_ON           = "turn-on";
public static final String COMMAND_TURN_OFF         = "turn-off";
public static final String COMMAND_REPORT_ON        = "report-on";
public static final String COMMAND_REPORT_ON_RESULT = "report-on-result";
...
public void _receive(Message message) throws NetworkError {
    super._receive(message);
    String command = message.getCommand();
    if (command.equals(COMMAND_TURN_ON)) {
        _basicLamp._turnOn();
    }
    else if (command.equals(COMMAND_TURN_OFF)) {
        _basicLamp._turnOff();
    }
    else if (command.equals(COMMAND_REPORT_ON)) {
        _networkController._transmit(
            new Message(COMMAND_REPORT_ON_RESULT, (""+_basicLamp._isOn())));
    }
}
```

Interpreter Pattern

- Could include basic arguments
 - **turn (on|off)**
 - turn on
 - turn off
 - **set (brightness|color) <integer_percent>**
 - set brightness 0
 - set brightness 50
 - set brightness 100
 - set color 5
 - set color 75

where **<integer_percent>** ∈ 0 ∪ 100

```
String s = "set brightness 50";
String[] fields = s.split(" "); // fields delimited by space
for (String field : fields) {
    System.out.println(field);    // remember Integer.parseInt()?
}
```

any issues?

Interpreter Pattern

- Grammatical considerations
 - fixed vs. freeform
 - fixed: split fine for rigidly defined grammars (no human involved)
 - freeform: StringTokenizer better

```
StringTokenizer t = new StringTokenizer(s);  
while (t.hasMoreTokens())  
{  
    System.out.println(t.nextToken());  
}
```

Interpreter Pattern

- Could include complex, programmatic arguments
 - **set (brightness|color) <function>**
 - **set brightness (linear-step 0 50 10)**

where **linear-step** function :

<start_value> is initial value

<end_value> is final value

<step_per_second> is transition from initial to final value

where

<start_value> □ 0 □ 100

<end_value> □ 0 □ 100

<step_per_second> □ 0.0 □ 100.0