A programming environment supporting a prototype-based introduction to OOP

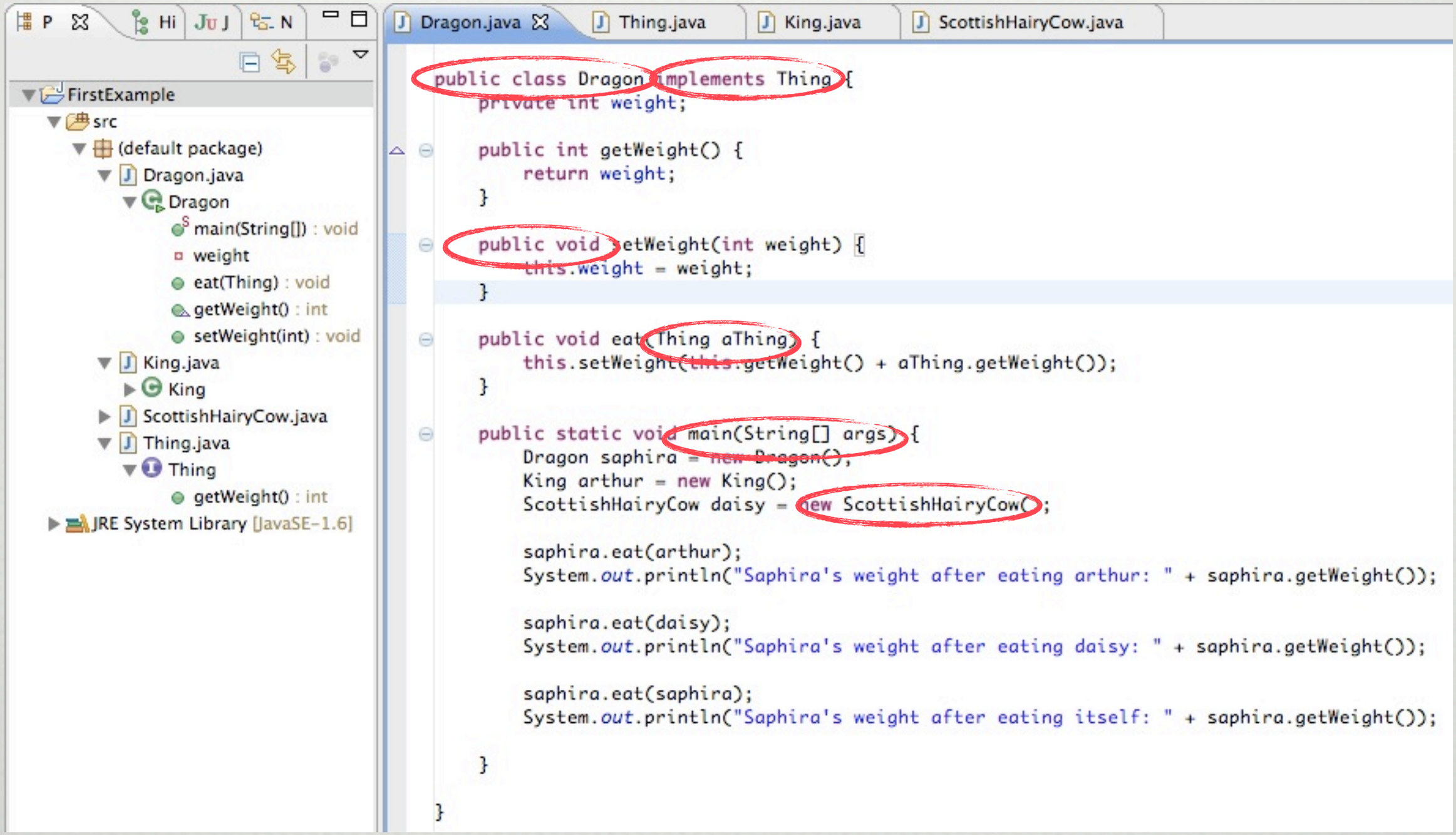
CONTEXT

- ☐ First OOP course
- ☐ Previous knowledge in structured programming (PASCAL)
- ☐ Tight schedule

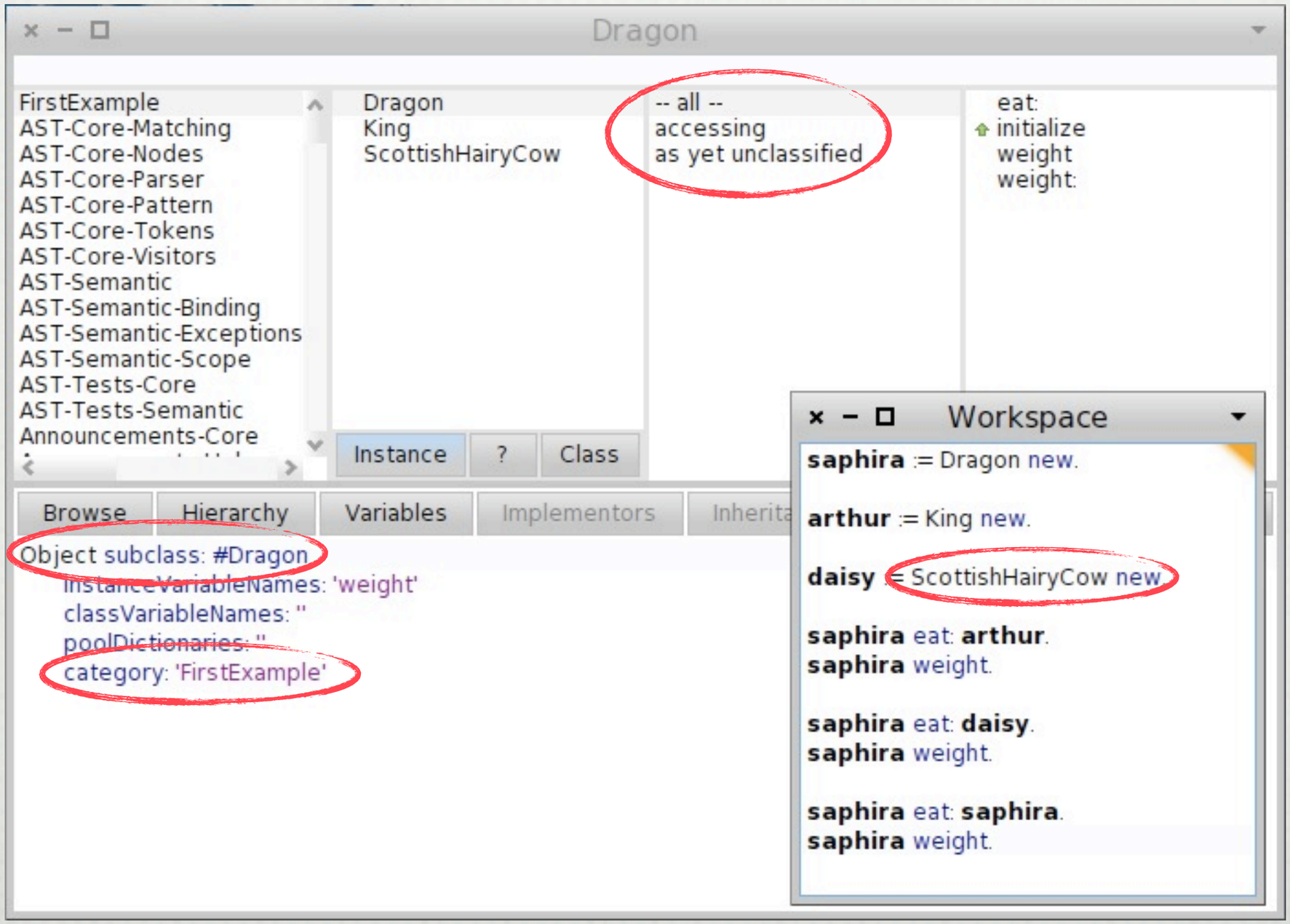
PROBLEM

- Paradigm shift
- A lot of new "stuff" to learn in order to get a first working program

EXAMPLE IN JAVA



EXAMPLE IN SMALLTALK



MOTIVATION

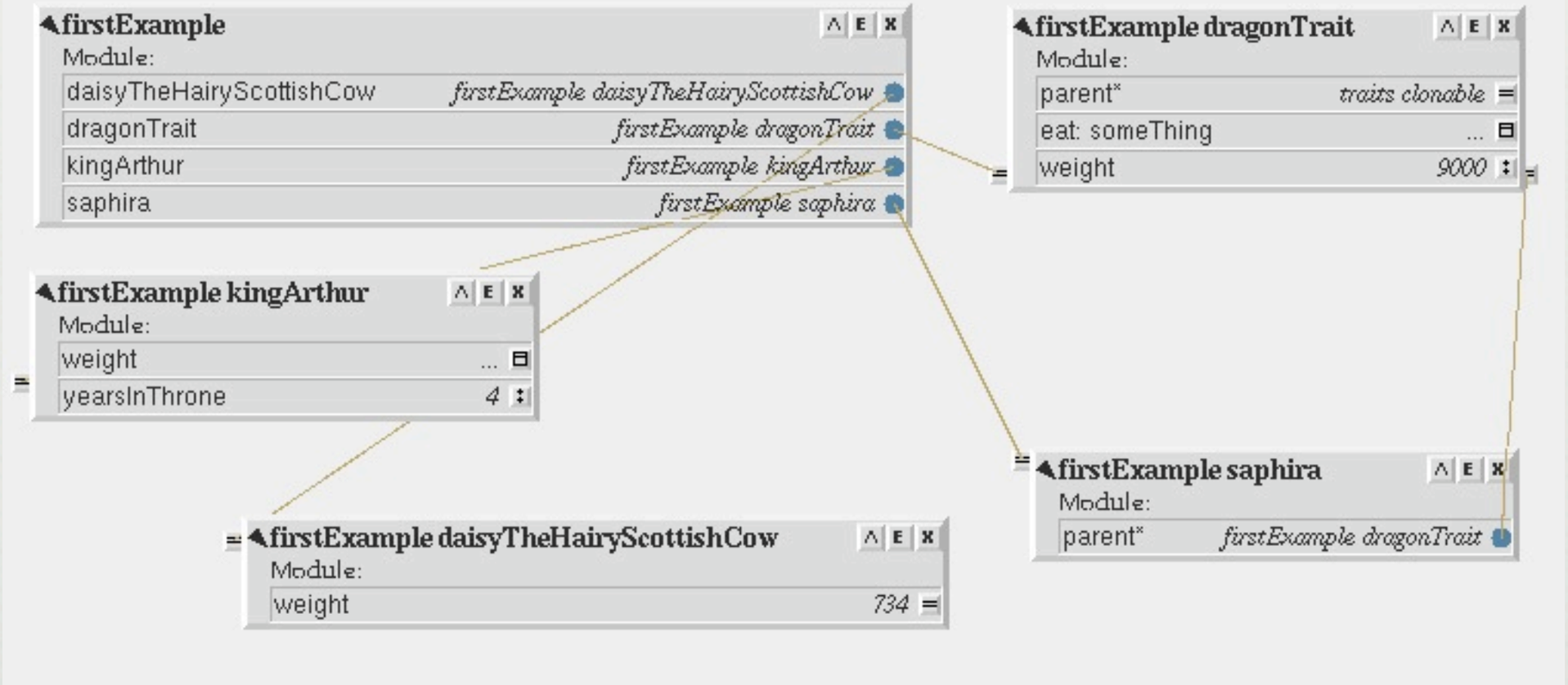
Focus on:

- ☐ Messages
- ☐ Objects
- ☐ Rich interaction between them
- ☐ Polymorphism

SOLUTION

- [ ] Start with a custom programming environment
- [ ] Delay the introduction of classes and inheritance

WHAT ABOUT SELF?



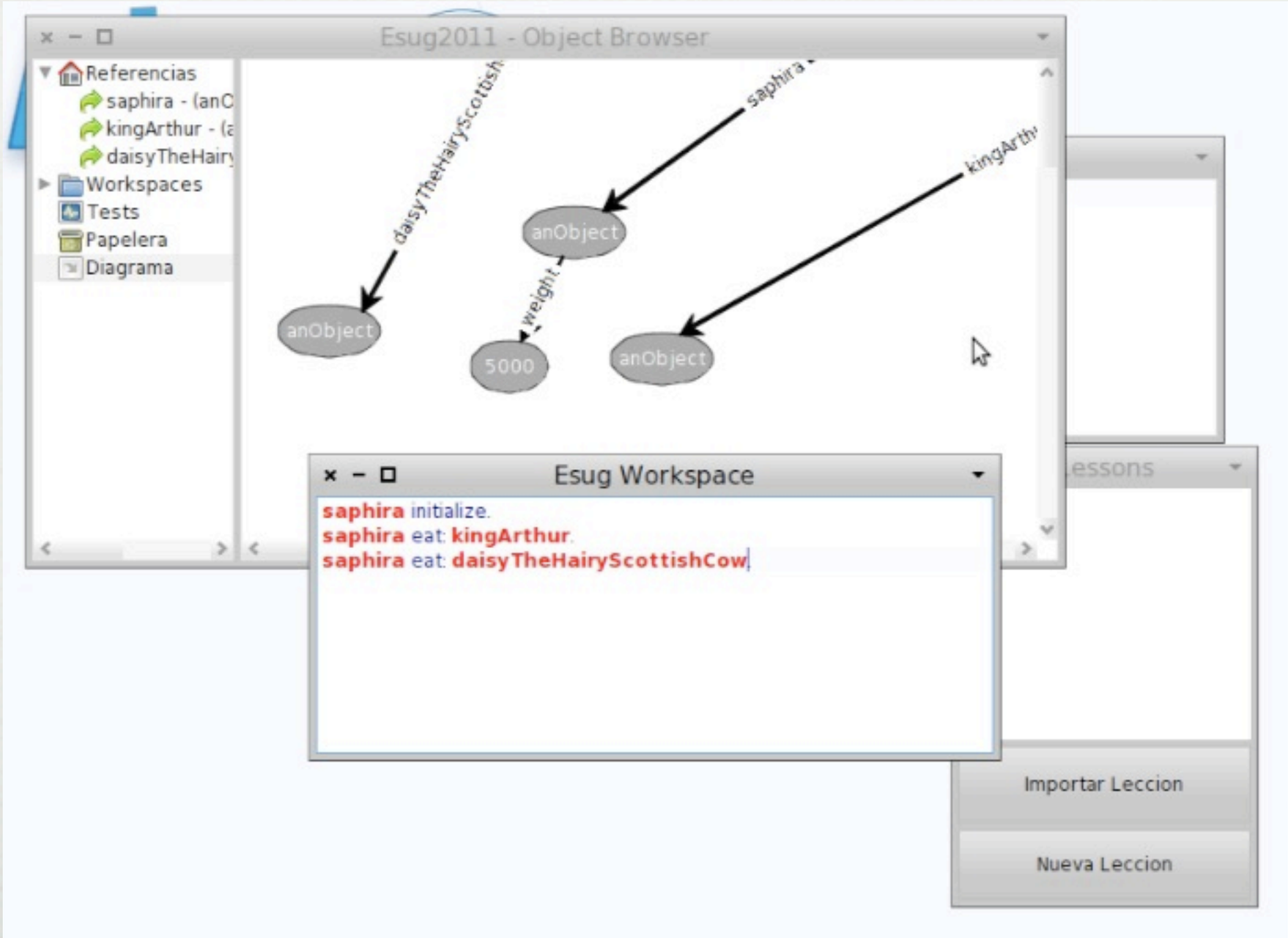
WHAT ABOUT SELF?

- Traits, prototypes, parent slots, etc.
- GUI is not familiar to the students

FEATURES

- ☐ Definition and usage of objects
- ☐ Multiple object environments
- ☐ Live object diagrams
- ☐ Explicit diferentiation between objects and references

EXAMPLE IN LOOP



TOWARDS CLASSES

```
saphira >> eat: aThing
   self weight: self weight + aThing weight.


arthur >> weight
   ^yearsInThrone * 666


daisy >> weight
   ^732
```

TOWARDS CLASSES

```
Dragon >> eat: aThing
   self weight: self weight + aThing weight.


King >> weight
   ^yearsInThrone * 666


ScottishHairyCow >> weight
   ^732
```

EXPERIENCE

Quarter	Pass Rate
2011 Q1	**84,62%**
2010 Q2	68,42%
2010 Q1	69,76%
2009 Q2	80,95%
2008 Q2	66,67%
2008 Q1	74,07%
2007 Q2	73,33%
2006 Q2	75,00%

CONCLUSIONS

- ☐ More time for the important concepts
- ☐ More time for more complex exercises
- ☐ Concrete to abstract learning path

QUESTIONS

?