Object-Oriented Design Heuristics, Riel, Addison-Wesley, 1996, Chapter 3
OO Program
Building Blocks

OrderedCollection
String
Dictionary
Characters
Streams
Ants
etc.
"Main"

AntApplicationModel open

What does "main" in a program do? Think of a GUI application. Where is the main there?
Modify the Ant demo so that the drawing is done by a view rather than by Ant and Colony objects.

this seems less OOP
How OO can go wrong

God class

Proliferation of classes
most of a program's overall functionality is coded into a single "all-knowing" object, which maintains most of the information about the entire program and provides most of the methods for manipulating this data

Wikipedia
Example

Room

Heat Flow Regulator

Furnace

getDesiredTemp

getActualTemp
Example

Room

Heat Flow Regulator

Furnace

getDesiredTemp

getActualTemp
Heuristics

Distribute system intelligence horizontally as uniform as possible

Do not create god classes/objects
Be very suspicious of a class whose name contains Driver, Manager, System

Beware of classes that have many accessor methods defined in their public interface

Beware of classes that have too much noncommunicating behaviour
The Accessor Issue

If data and operations are together why do we need accessors?
Why accessors

Objects sometimes have to collaborate

- Course
  - Prerequisites
- Student
  - Courses
- CourseOffering
Using GUls

Model should not depend on the interface
The interface should depend on the model

So interface needs to access data in the model using accessors

But
Model should not use accessors unless needed
Proliferation of Classes Problem

Eliminate irrelevant classes from your design

Eliminate classes that are outside the system

Do not turn an operation into a class
Role of Agent Classes

During design time many agents are found to be irrelevant and should be removed.