CS 535 Object-Oriented Programming & Design
Fall Semester, 2008
Doc 18 Some Heuristics
Nov 25 2008

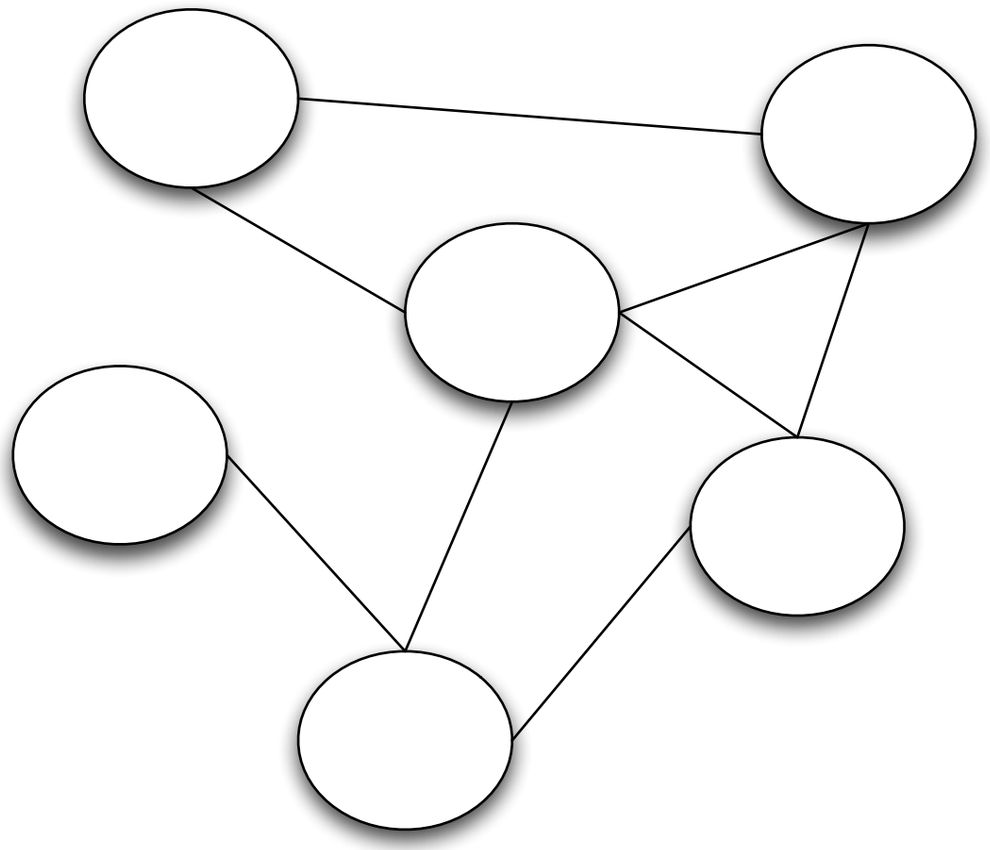
Reference

Object-Oriented Design Heuristics, Riel

Reading

Object-Oriented Design Heuristics, Riel
Chapters 2 & 3 this week
Chapter 4 - Dec 2-4
Chapter 5 - Dec 9-11

OO Program



Building Blocks

OrderedCollection
String
Dictionary
Characters
Streams
Trolls
etc.

"Main"

Adventure open

What does "main" in a program do? Think of a GUI application. Where is the main there?

Finding Building Blocks

Detailed scenarios

Step through the operations

Concrete abstractions easier to find

Cards are Useful

Building Block = Class

2.8 A class should capture one and only one key abstraction

Keep related data and behavior in one place

This is the most important idea in OO

Corollary

To perform an operation send a message to the object that contains the data

Collections and Items

Class Potions

Instance variables

locations - OrderedCollection

numberOfUses - OrderedCollection

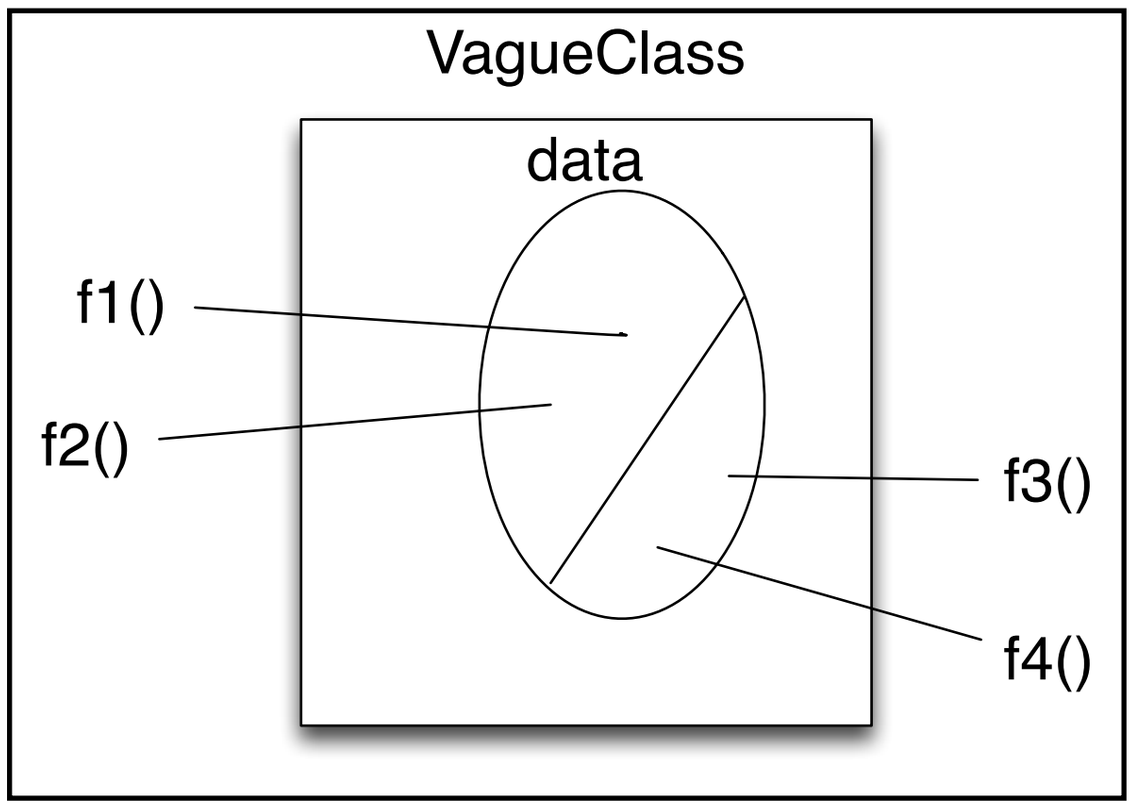
verses

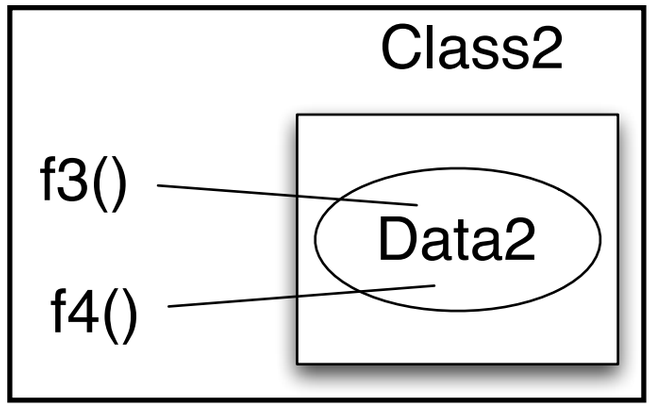
Class Potion

Instance variables

numberOfUses - Integer

Spin off nonrelated information into another class





God Class

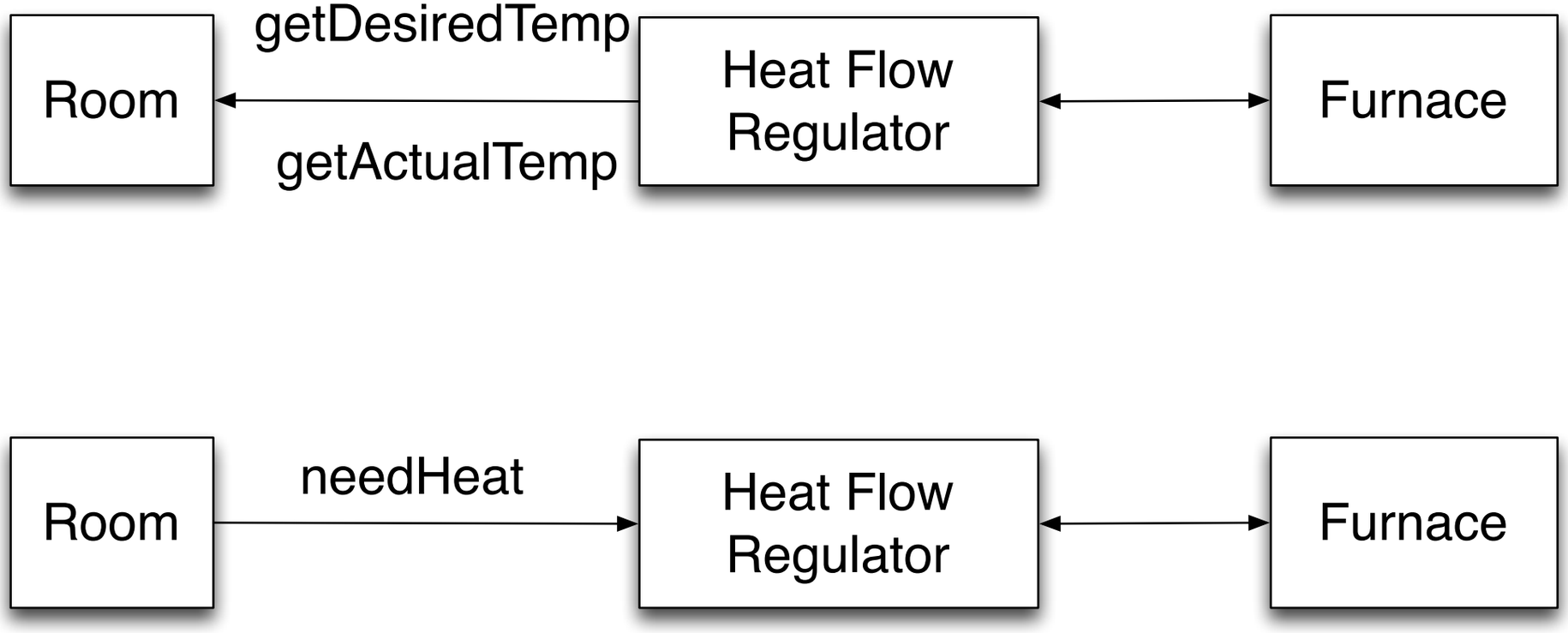
God object is an object that knows too much or does too much

Behavioral Form

Replaces the main
Does too much

Definition is from Wikipedia

Example


Room
getDesiredTemp
getActualTemp
Heat Flow Regulator
Furnace
Room
needHeat
Heat Flow Regulator
Furnace

Heuristics

Distribute system intelligence horizontally as uniform as possible

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

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

Beware of classes that have too much noncommunicating behavoir

Using GUIs

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

So interface needa to access data in the model

Proliferation of Classes Problem

Eliminate irrelevant classes from your design

Do not turn an operation into a class

Should the cow send the milk the uncow yourself message?
Should the milk send the cow the unmilk yourself message?

Role of Agent Classes

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