Clean Systems

Admin

- Next Quiz
- Exam discussion
- Anything else?
- Images from chapter 11 of clean code by Bob Martin
- Read chapter 11 of clean code for this slide set.

Build an Aircraft Carrier

- How many nations in the world have at least one aircraft carrier?

Build an Aircraft Carrier

- How many nations in the world have at least one aircraft carrier?
  - 9 (Brazil, China, France, India, Italy, Russia, Spain, Thailand, US)
- How many countries have ever completely built an aircraft carrier from start to finish?

Build an Aircraft Carrier

- How many nations in the world have at least one aircraft carrier? (accurate as of 2016)
  - 9 (Brazil, China, France, India, Italy, Russia, Spain, Thailand, US)
- How many countries have ever completely built an aircraft carrier from start to finish?
  - France, Italy, Japan, USSR, Great Britain, United States (Maybe China, built, and in Sea Trials as of 2018.)
- Why so few?

Why so few?

- Aircraft carriers are a huge undertaking
  - Big investment sure
  - But major logistical issues too
    - Designing a floating nuclear power city
    - Dealing with the logistics of manning and maintaining it
    - Facilities for docking
    - Fleets of planes

From ships to software

- 1997 USS Yorktown
  - Crew member enters 0 in database field
  - Divide by zero error crashes all windows machines on network – dead in water
- 2006 F-22 squadron goes on overseas deployment for the first time
  - Hawaii to Okinawa Japan
  - Crosses international dateline
  - All computer systems 'dump' (crash)
  - No navigation – no nothing.
  - Followed the refueling tanker plane to Japan

Software Systems

- Often extremely complex
- Aircraft carrier is apt metaphor
- One person can't do it all
- Not even Notch.

Separation of Concerns

- Separate starting the system from running it.
  - Construction is a different use case than use of a system
  - Make the startup routine create what is need by the app
    - That way there is no path the leaves something uninitialized
    -

Separating construction

- Separate construction from use
  - Notice that all of the knows-a relationships go out of the 'main'/setup routine

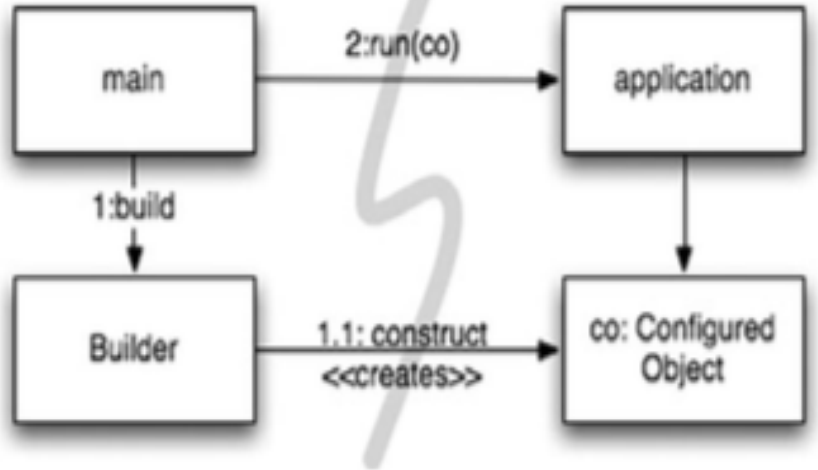

Figure 11-1

Separating construction in `main()`

Factory approach

- This is a perfect situation to use a factory to reduce coupling:
-

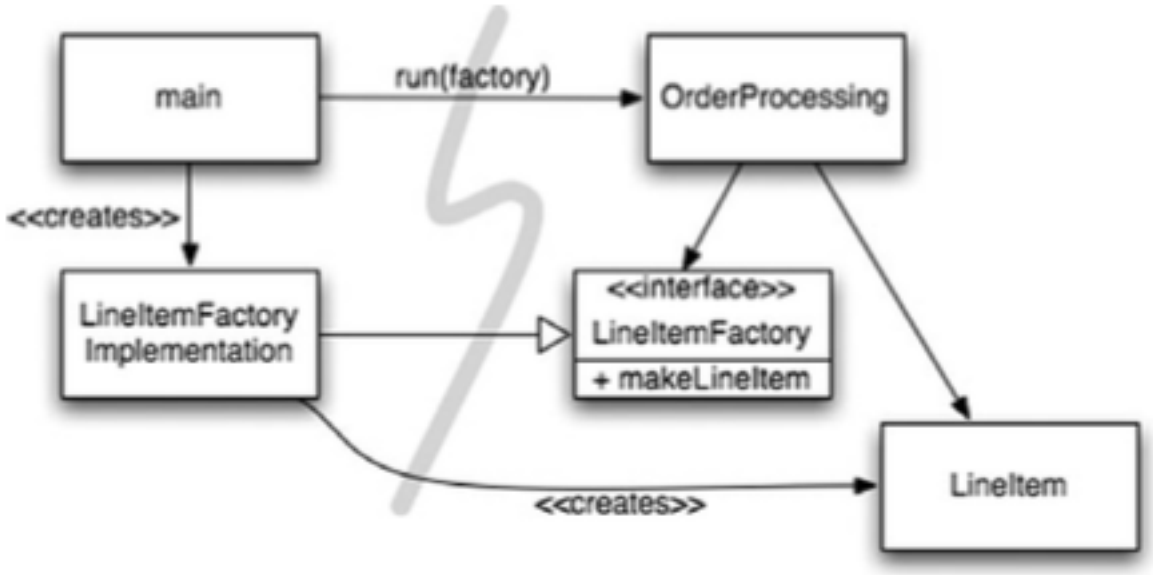

Figure 11-2
Separation construction with factory

Scaling up

- Ruby on rails in 2008
  - For toy sites it was a genre changing moment
  - It made full stack development an out of the box experience
  - Then?

Scaling up

- Ruby on rails in 2008
  - For toy sites it was a genre changing moment
  - It made full stack development an out of the box experience
  - Then?
  - It didn't scale – beyond a few thousand connections an hour the whole thing fell apart.
- Ruby on rails today
  - After several years of development works well for small and medium businesses
  - 30k-ish connections an hour

Scaling up

- Software grows
  - Build for today
  - Build to be easy to maintain/change
  - But not for what you think tomorrow will bring.
  - In early 2006 no one thought that an underpowered Nintendo Wii would be the biggest selling game console of 2007 and 2008
  - In 2009 no one would have predicted that MS windows would be a minority operating system today.

Crosscutting Concerns

- CrossCutting Concerns
  - Semi jargon term for an issue that has to be addressed across a software system
  - Separation of concerns not really possible with these
  - Examples?

Crosscutting Concerns

- CrossCutting Concerns
  - Semi jargon term for an issue that has to be addressed across a software system
  - Separation of concerns not really possible with these
  - Common Examples:
    - Transaction logic (saving/persistence)
    - Security/Authentication
    - Logging
  - Need these in multiple modules.

Aspect Oriented Programming

- Aspect Oriented Programming
  - Designed to allow separation of cross cutting concerns
  - Usually by meta programming
    - Decorators
    - All methods beginning with loggedXXXX get the logging functionality automatically.
  - Code is valid in original language
  - But with new interpreter gains AOP functionality
  - And code isn't cluttered with cross cutting concerns everywhere

Use the right language

- Programming languages
  - What does it take for a language to be Turing complete? (lets consider imperative languages for now)
  -

Use the right language

- Programming languages
  - What does it take for a language to be Turing complete? (lets consider imperative languages for now)
    - Needs conditionals and looping
  - Once a language is Turing Complete what does that mean for us?

Use the right language

- Programming languages
  - What does it take for a language to be Turing complete? (lets consider imperative languages for now)
    - Needs conditionals and looping
  - Once a language is Turing Complete what does that mean for us?
    - That we can write any program that we can write in any language
    - But should we!?!?

Use the right Language

- Languages are made to make some problems easier
  - Use a domain specific language
  - R for statistical stuff
  - Lua for imbedded interpretation
  - C/C++ for fast hard-real time programming
  - Etc.
  -