1
과목소개

목표 및 사용언어

- 목표 : 프로그램 설계의 이해 및 숙달
  - 제어구조 (control structure)
  - 자료구조 (data structure)
  - **부품구조 (component structure)**

- Java를 쓰는 이유
  - 부품 구조 기반으로 설계된 언어이기 때문에
  - 널리 쓰는 언어이기 때문에

교재 및 슬라이드

- 교재
  - *Programming Principles in Java: Architectures and Interfaces*, David A. Schmidt (Kansas State University)
  - http://people.cis.ksu.edu/~schmidt/CIS200/home.html
  - 강의 홈페이지에서도 내려 받기 가능
  - 부교재: 이충기, 문제 해결을 위한 자바 프로그래밍, 혹은 황기태, 김효수, 명품 Java Programming
- 강의 홈페이지
  - http://psl.hanyang.ac.kr/courses/clu1091_2022f/
  - 강의 슬라이드 내려 받기 가능

교재 및 슬라이드

- 실습
  - 원격으로 진행
  - 개념 숙지 및 문제풀이
  - 문제풀이: repl.it - 웹에서 개발 및 과제물 제출 가능
  - 질의응답: LMS 등
  - 강의 홈페이지 참조

강의 주제

- 과목 소개
- 간단한 Java 응용 프로그램
- 산술연산과 변수
- 입출력과 상태
- 부품구조
- 제어구조
- 반복 패턴
- 자료구조
- 인터페이스를 활용한 프로그래밍
- 문서와 파일 처리

평가 및 연락처

- 평가
  - 기말고사 60%, 실습 30%, 출석 10%

- 연락처
  - 교수

    이우석 (woosuk@hanyang.ac.kr, x1031, 3공403호)
  - 조교

    조한결 (saijhg8@gmail.com, 3공 418-2호)

컴퓨터와 프로그래밍

컴퓨터 (Computer)?

- 컴퓨터란?
  - 명령을 수행하는 것
  - 광의의 의미로 사람도 컴퓨터다.

- 컴퓨터의 일반적인 구성
  - 프로세서 (processor): 명령을 처리하는 것
  - 주 저장소: 명령과 자료를 저장하는 것
  - 그 외에
    - 부 저장소: 대용량 자료를 저장하는 것
    - 입출력 장치: 명령 수행 도중 사용자와 상호작용하도록 하는 것

프로그램 (Program)

- 프로그램 또는 코드 (code)
  - 컴퓨터가 수행할 수 있도록 충분히 구체적으로 작성된 명령 나열
  - 이를 작성하는 행위를 프로그래밍 (programming) 또는 코딩(coding) 이라고 한다.

```
sum = 0;
for(i=1; i<=100; i++) {
    sum += i;
}
```

- 알고리즘 (algorithm)
  - 목적을 달성하기 위한 명령의 나열
  - 프로그램에 비해 덜 구체적

i를 1부터 100까지 증가시키면서 모두 더한다.

프로그래밍 언어

- 프로그램을 작성하기 위해 사용하는 언어

- 예전에는 기계어 사용
  - 기계가 빠른 속도로 수행 가능
  - 사람이 작성하기에는 매우 어려움

- 현재는 고급 언어 사용
  - 사람의 생각을 쉽게 프로그램으로 표현 가능
  - Fortran, Cobol, Lisp, Basic, Algol, Prolog, ML, C++, Java 등

프로그램 설계

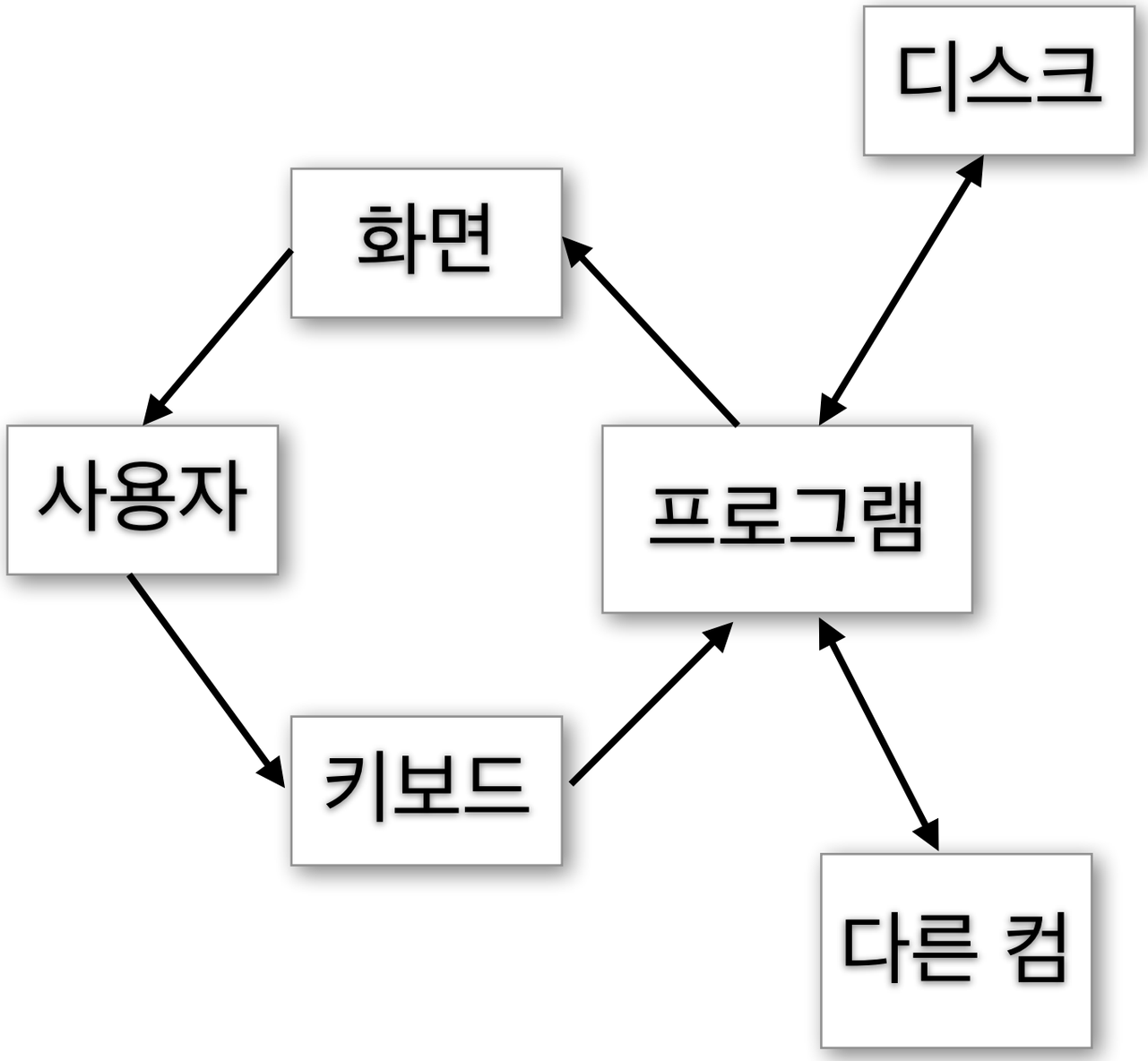
- 집짓기는 설계와 공사를 분리한다.
  - 건축가 (architect): 집을 어떻게 지을지 전체적인 구성을 설계하는 사람
  - 빌더 (builder): 집을 실제로 짓는 사람

- 프로그래머는 일반적으로 두 작업을 다 하지만, 좋은 프로그램 작성을 위해서는 설계와 구현을 분리해야 한다.
  - 설계 (design): 소프트웨어 구조 (software architecture) 구축
  - 구현 (implementation): 프로그램 또는 코드 작성

객체 지향 설계 (Object-Oriented Design)

- 프로그램을 설계하는 한 방법
- 기본
  - 모든 것은 객체(object)이다.
  - 객체는 자신이 해야 할 일, 즉 메소드(method)를 갖는다.

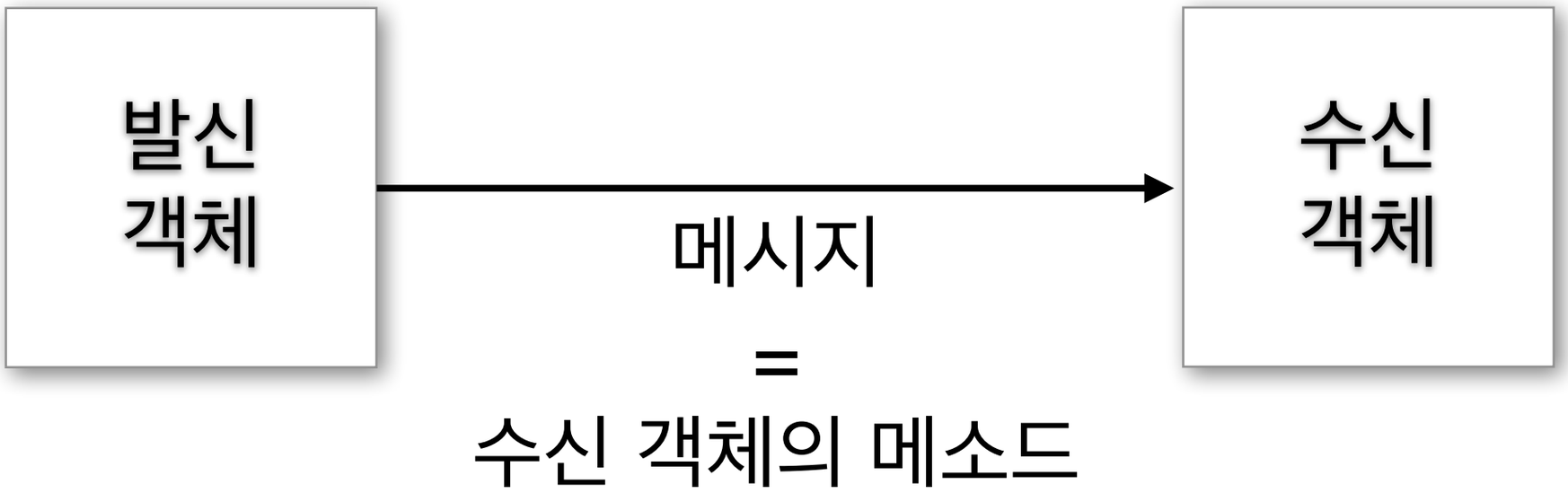
프로그램 수행을 객체로



- 객체: 키보드
  - 메소드: 문자 입력
- 객체: 화면
  - 메소드: 화면과 그림 출력
- 객체: 사용자
  - 메소드: 키보드로 문자열 입력, 화면에서 결과를 봄

계산 = 메시지 전달의 연속

- 객체 간에 메시지(message)를 전달하여 소통하는 것을 통해 계산이 수행된다.



예

- <사용자>는 어떤 수의 제곱 값을 알고 싶어 <키보드>에 그 수에 해당하는 키를 입력한다.
- <키보드>는 눌려진 키를 수로 변환하여 <프로그램 수행>에 수를 전달한다.
- <프로그램 수행>은 수의 제곱 값을 구해 <화면>에 전달한다.
- <화면>은 수를 화면에 기호로 표시하여 <사용자>의 눈에 전달한다.

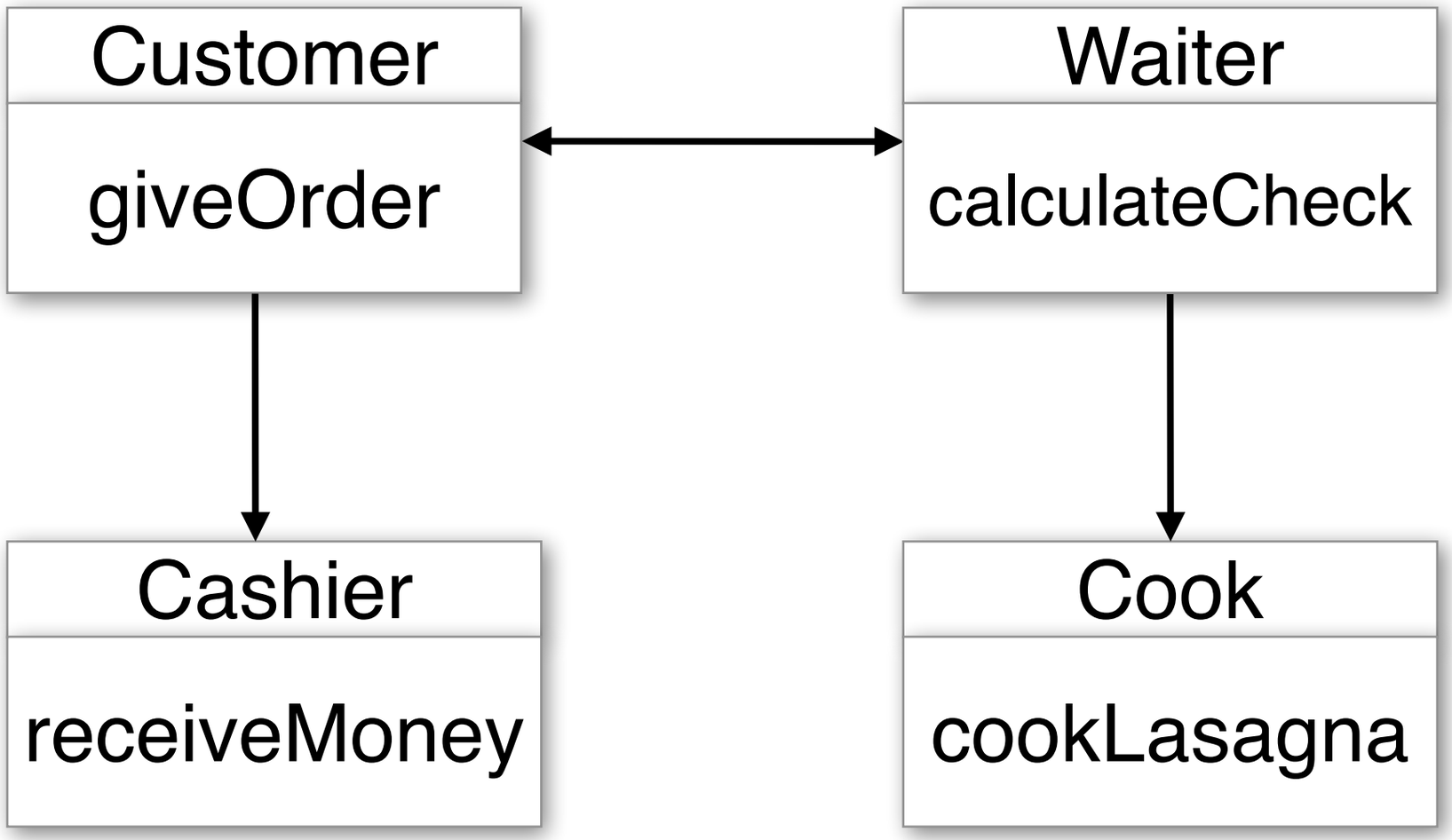
클래스 구조도 (Class Diagram)

- 객체 지향 설계 방법론에서의 설계도

- 클래스 (class)
  - 객체를 생성하기 위한 틀
  - 메소드를 갖고 있다.
  - 객체 vs 클래스: <화면> 클래스로부터 <화면1>, <화면2> 객체를 생성한다.

- 클래스 구조도
  - 클래스와 클래스간의 소통을 그려놓은 설계도

이태리 요리집의 클래스 구조도

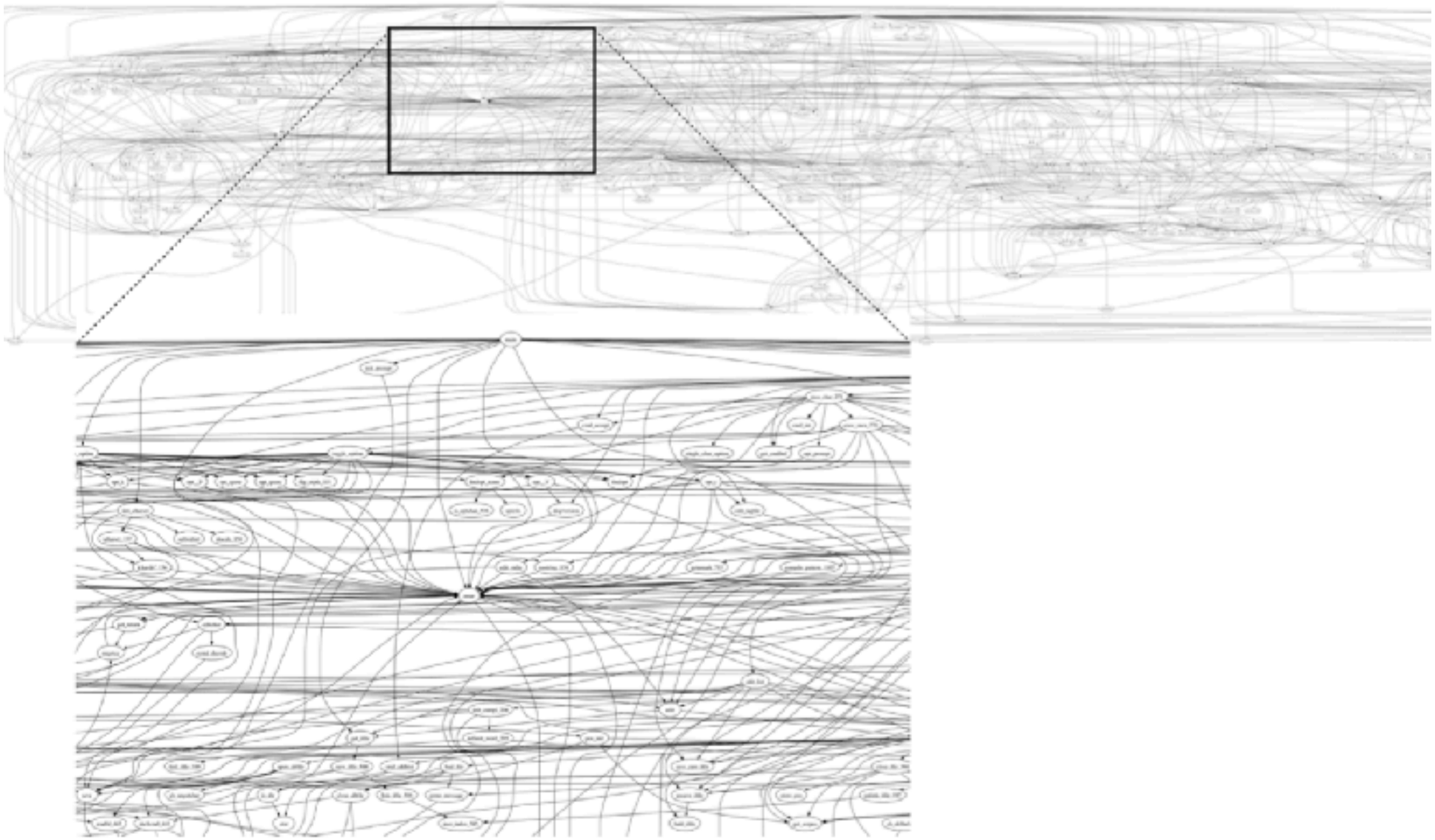


프로그래밍의 어려움

- 프로그램의 규모와 복잡도가 점점 커짐
  - SW 복잡성의 증가속도 >> HW 성능의 성장 속도
  - “SW 는 가스다.” “Software is gas.”

프로그래밍의 어려움

- SW 복잡성의 예: less-382 (23,822 LoC)



중요

- 좋은 프로그래머란
  - 좋은 건축가가 되는 것과 같은 것
  - 체계적인 개발 방법론에 입각하여 복잡도를 다룰 수 있어야 함.

- 좋은 프로그래머가 되려면
  - 표준 예제들을 익히고
  - 조합의 기본 기술들을 익히고
  - 연습하는 것