

Тема 1. Введение

Автор:

А. С. Власовских

Лицензия:

Creative Commons Attribution-Noncommercial-Share Alike 3.0

Организационные вопросы

- Занятия
 - Лекции: 1 пара в неделю
 - Самостоятельно: статьи, задания
- Отчётность
 - Экзамен, домашние задания
- Сайт и блог курса
 - <http://kspt.ftk.spbstu.ru/2010/course/lang>
 - Объявления, литература, слайды, задания
- Требуемый софт
 - По мере изучения, open source
 - ОС желательно Linux

Задачи курса

- Парадигмы и концепции программирования
 - Процедурное, ООП, ФП, параллельное
- Знакомство с рядом языков программирования
 - Python, Scheme, Erlang, Haskell и др.
- Практика программирования

Группы тем

- ООП

- Парадигма ООП, передача сообщений, ввод-вывод, изменяемое состояние, распределённые системы

- ФП

- Парадигма ФП, функции высших порядков, списки, абстрактные типы данных, функциональная чистота, λ -исчисление

- Мета

- Классы языков, синтаксис, семантика, парсеры, системы типов, статический анализ

- Практика

- Формализация задачи, тестирование, идиомы ООП и ФП, языки DSL

Связь с другими курсами

○ Зависит от курсов

- Теория и технология программирования
 - C, C++, Java, алгоритмы, практика
- Транслирующие системы
 - Синтаксис, семантика, компиляторы
- Дискретная математика
 - Множества, отношения, моноиды, группы
- Информатика
 - Конечные автоматы

○ Связан с курсами

- Промышленные технологии проектирования ПО
 - Требования, тестирование
- ПО распределённых вычислительных систем
 - Распределённое и параллельное программирование

Изучение концепций языков программирования

- Больше возможностей для выражения идей
 - Новые парадигмы и средства
- Более обоснованный выбор подходящего языка
 - 1 LOC на shell или 10K LOC на C
- Способности к изучению новых языков
 - Языки продолжают появляться
- Способности к разработке новых языков
 - В т. ч. языков описания данных, языков DSL

Задачи программирования

- Задачи
 - Преобразование информации
 - Передача информации
 - Управление
- Программирование — формализация этих задач с целью автоматизировать их решение на компьютере

Качества задач

- Неформальность — формальность
 - Чувство смысла *vs.* смысл в форме
- Декларативность — императивность
 - Что есть *vs.* как сделать
- Абстракция — конкретика
 - Минимум деталей *vs.* все детали
- Статика — динамика
 - Максимум утверждений *vs.* никаких утверждений

Пример формальной задачи (1)

- Грамматика BNF для формальной системы

$$\text{term} ::= 0 \mid (\text{term} + \text{term})$$

-

$$\text{wff} ::= \text{term} = \text{term} \mid (\text{wff} \rightarrow \text{wff})$$

- Примеры термов: 0 , $((0 + 0) + 0)$

- Примеры формул: $0 = 0$, $(0 = 0 \rightarrow (0 + 0) = 0)$

- Доказательство — последовательность формул wff , таких что для каждой формулы Q верно:

- Q — аксиома или применение схемы аксиом

- Перед Q есть формулы P и $(P \rightarrow Q)$ (силлогизм, *modus ponens*, MP), т. е. $\frac{P, (P \rightarrow Q)}{Q}$

- Теорема (истинная формула) — строка доказательства

- Схемы аксиом

- Пусть r , s и t — термы

- Ax1. $(t = r \rightarrow (t = s \rightarrow r = s))$

-

- Ax2. $(t + 0) = t$

Пример формальной задачи (2)

○ Схема теорем $t = t$

○ Доказательство

1. $(t + 0) = t$ (Ax2)

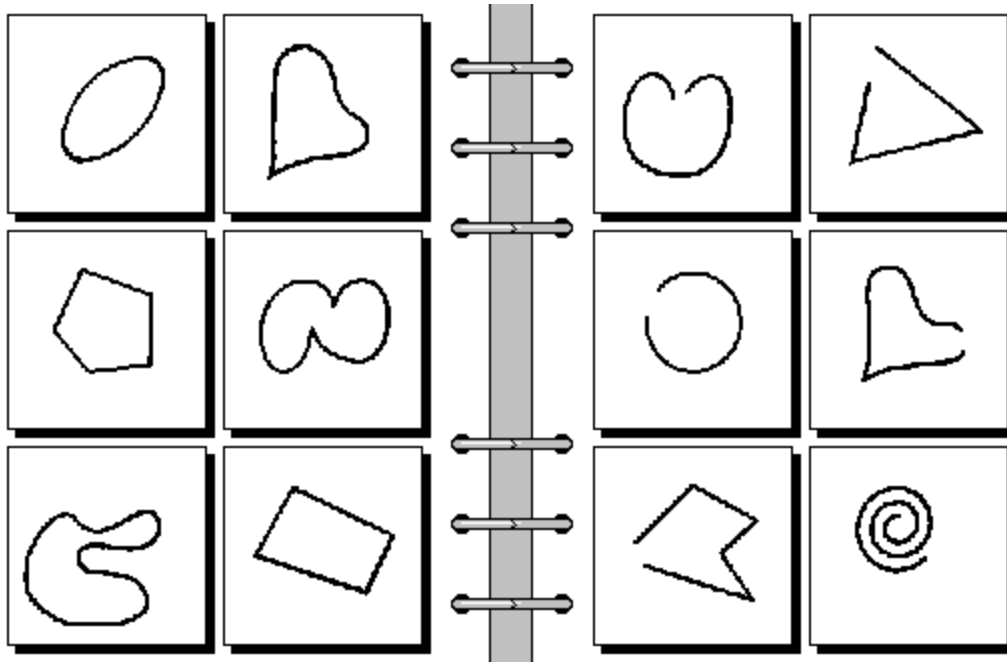
2. $((t + 0) = t \rightarrow ((t + 0) = t \rightarrow t = t))$ (Ax1)

■

3. $((t + 0) = t \rightarrow t = t)$ (MP)

4. $t = t$ (MP)

Пример неформальной задачи



- Задача Бонгарда #15 (источник: <http://www.foundalis.com/res/bps/bongard/p015.htm>)

Декларативное и императивное знание

- Декларативное

- \sqrt{x} — это такое y , что $y^2 = x$ и $y \geq 0$

- Императивное

- Чтобы найти приближение \sqrt{x} :
 - Выбрать начальное приближение g
 - Улучшить приближение при помощи усреднения g и $\frac{x}{g}$
 - Продолжать улучшать, пока приближение не будет достаточно хорошим

Статика и динамика

○ Статические свойства (утверждения)

- `join :: String -> [String] -> String`

- Функция объединения строк через разделитель `join`

- `::` — имеет тип

- $$\frac{s :: \text{String}, ss :: [\text{String}]}{\text{join } s \text{ } ss :: \text{String}}$$

- $$\frac{s :: \text{String}, ss :: [\text{String}]}{\text{length } (\text{join } s \text{ } ss) \geq \text{sum } (\text{map } \text{length } ss)}$$

- Длина объединённой строки больше или равна сумме длин объединяемых строк

○ Динамические свойства

- Значение функции `join` от некоторых аргументов

```
join "\\n" ["a", "b", "c"]  
join ", " ["foo", "bar"] == "foo, bar"
```

Программирование и другие дисциплины

○ Математика

- Изначально формальные задачи, преимущественно декларативные и статические, лёгкая проверка корректности

○ Программирование

- Требуется формальность и императивность, результат получается динамически, очень большие проекты, нужен контроль сложности (абстракция, статика, декларативность)

○ Инженерные дисциплины

- Слабая степень автоматизации, не требуется или невозможна полная формализация, неидеальные материалы, много конкретики

Контроль сложности

○ Причины сложности

- Исходная неформальная задача очень далека от машинного языка
- Результат формализации задачи — миллионы машинных инструкций
- Рекурсия и итерация, потенциально бесконечные трассы выполнения

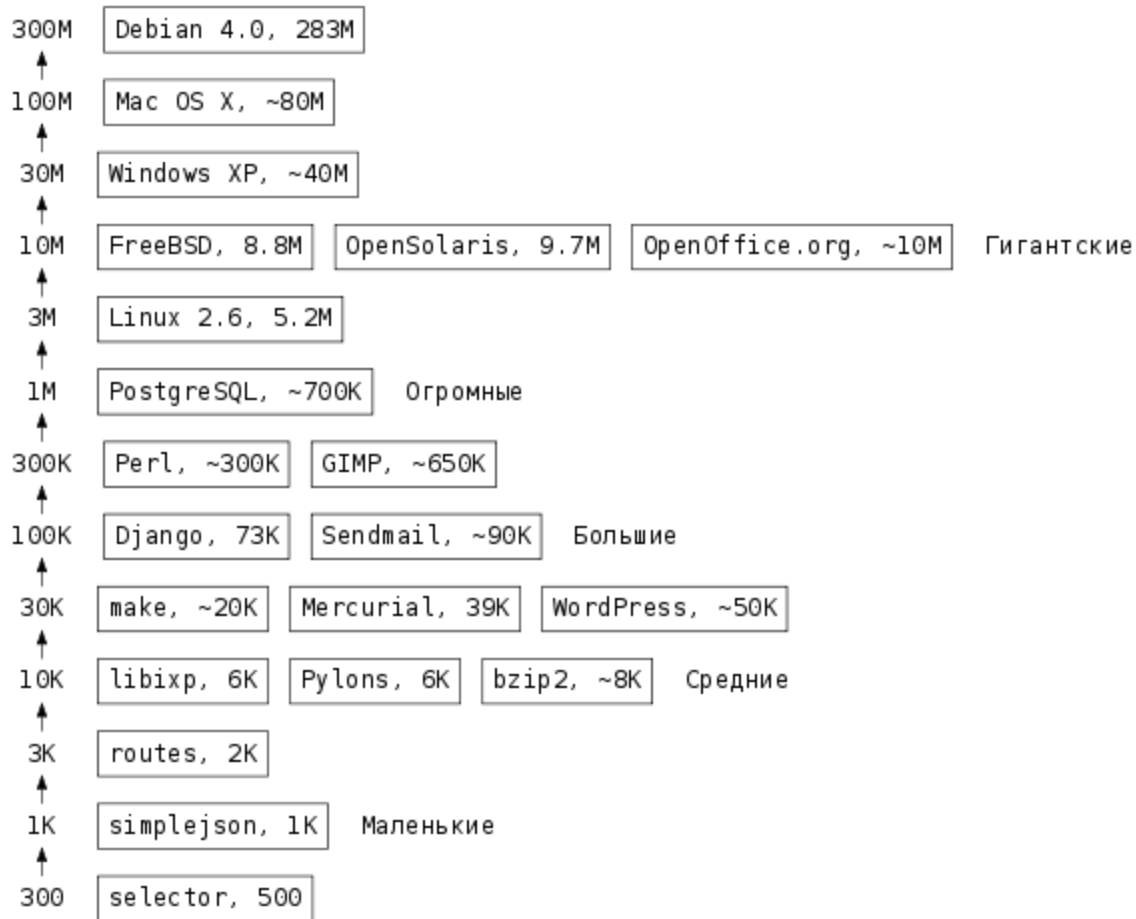
○ Принципы контроля

- Абстракция
 - Меньше деталей, только существенное, ближе к языку задачи
- Декларативность
 - Проще понять группу фактов, чем следить за последовательностью действий
- Статика
 - Выразить часть задачи в виде утверждений, ограничить и упорядочить

○ Средства контроля

- Языки программирования
- Способы и парадигмы программирования
- Взаимодействующие программы

LOC как мера сложности



Литература (1)

- Megill N. *Metamath. A Computer Language for Pure Mathematics* [Electronic Resource]. — <http://us.metamath.org/downloads/metamath.pdf>
 - S. 2.2. Your First Formal System
 - Ch. 3. Abstract Mathematics Revealed
- Себеста Р. У. *Основные концепции языков программирования*. — Вильямс, 2001. — <urn:isbn:5-8459-0192-8>
 - Разд. 3.3. Формальные методы описания синтаксиса
- Реймонд Э. *Искусство программирования для Unix = The Art of Unix Programming*. — Вильямс, 2005. — <urn:isbn:5-8459-00791-8>, <http://www.faqs.org/docs/artu/>
 - Гл. 13. Сложность: просто, как только возможно, но не проще
- *Index of Bongard Problems* [Electronic Resource] / Н. Foundalis. — <http://www.foundalis.com/res/bps/bpidx.htm>

Литература (2)

- Spinellis D. *The Way We Program* [Electronic Resource]. — <http://www.spinellis.gr/blog/20080626/>
 - Статистика LOC разных проектов, соотношение кода и комментариев