

Филология программирования

День проблем, 30 декабря 2023

Коновалов А.В.

Проблема

- ▶ Нужна дисциплина, которая рассматривает языки программирования (ЯП), как средства коммуникации и выражения мыслей.
- ▶ Одной (но не единственной) из задач дисциплины будет изучение выразительных средств программирования (средства самих языков, идиомы, паттерны), а также практик использования этих средств.
- ▶ Практическим приложением дисциплины будут как методики проектирования и разработки новых языков программирования, так и систематической разработки практик использования существующих языков, в частности, разработки новых идиом и паттернов.

Учебников и методичек на тему «как реализовать ЯП» масса (в частности, сам докладчик читает курс «Конструирование компиляторов»). Учебники и методички на тему «как спроектировать ЯП» докладчику не попадались.

Язык программирования — это язык (1)

При разговоре о программировании часто используются метафоры, характерные для естественных языков:

- ▶ **определения** (функций, типов, переменных) — **определения** слов в толковых словарях,
- ▶ **стиль** программирования — **стиль** писателя,
- ▶ **идиомы** языка программирования,
- ▶ **трансляция** (буквально: перевод) программы с одного ЯП на другой,
- ▶ **выразительность** языков программирования,
- ▶ собственно, сам **язык** программирования.

Язык программирования — это язык (2)

А всё потому, что программирование — это не только инженерная задача (создать программу, решающую данную задачу = создать устройство, выполняющее данную функцию), но и языковая (максимально точно и подробно на формальном языке описать задачу).

В будущем историки современное налогообложение будут изучать по бухгалтерским программам.

Два смысловых слоя программирования

Два смысловых слоя программирования

Любая программа содержит два слоя информации:

1. То, что понимает компьютер.
2. То, что компьютер игнорирует, но понимает программист:
имена переменных, стиль оформления, комментарии,
идиомы и прочее

Два смысловых слоя программирования

Любая программа содержит два слоя информации:

1. То, что понимает компьютер.
2. То, что компьютер игнорирует, но понимает программист:
имена переменных, стиль оформления, комментарии,
идиомы и прочее

Проводя аналогию с естественными языками, первый слой изучает *лингвистика*, второй — *филология*.

Поэтому рабочее название для предлагаемой дисциплины — *филология программирования*.

Примеры отсутствия одного из слоёв

- ▶ При *трансляции* строится программа на целевом языке, содержащая ту же информацию в первом смысловом слое. Второй слой, как правило, теряется (исключение: сохранение информации для отладчика вроде соответствия адресов инструкций номерам строк исходного кода).
- ▶ *Обфускация* — намеренное уничтожение второго смыслового слоя (однако, обфускация JavaScript ради сокращения объёма — скорее, разновидность трансляции).
- ▶ *Псевдокод* — второй смысловой слой без первого в чистом виде.

Псевдокод

- ▶ Псевдокод — неформальная запись алгоритма, использующая конструкции ЯП, математические обозначения и фразы на естественном языке.
- ▶ Псевдокод невозможно выполнить, т.к. иначе он был бы не псевдокодом, а программой. Поэтому это второй смысловой слой в чистом виде.
- ▶ Функция псевдокода — в чистом виде коммуникация и выражение мыслей.

```
/* Маркировка */
1) Unscanned = множество объектов, достижимых из корневого множества;
2) while (Unscanned ≠ ∅) {
3)   Удалить объект o из Unscanned;
4)   for (Каждый объект o', на который имеется ссылка в o) {
5)     if (o' недостижим) {
6)       Пометить o' как достижимый;
7)       Поместить o' в список Unscanned;
8)     }
9)   }
}
/* Вычисление новых позиций */
8) free = Начальный адрес памяти кучи;
9) for (Каждый блок памяти o в куче, начиная с нижнего конца) {
10)  if (o достижим) {
11)    NewLocation (o) = free;
12)    free = free + sizeof(o);
13)  }
}
/* Изменение ссылок и перенос достижимых объектов */
13) for (Каждый блок памяти o в куче, начиная с нижнего конца) {
14)  if (o достижим) {
15)    for (Каждая ссылка o.r в o)
16)      o.r = NewLocation (o.r);
17)    Копирование o в NewLocation (o);
18)  }
}
18) for (Каждая ссылка r в корневом множестве)
19)   r = NewLocation (r);
```

Рис. 7.26. Сборщик мусора “пометить и сжать”

Скриншот из книги *Ахо, Альфред В., Лам, Моника С., Сети, Рави, Ульман, Джеффри Д. Компиляторы: принципы, технологии и инструментарий, 2-е изд. : Пер. с англ. — М. : ООО «И.Д. Вильямс», 2008. — 1184 с. : ил.*

Второй слой и филология программирования

Первый смысловой слой хорошо, давно и плодотворно изучается: теория формальных языков, теория и практика написания компиляторов, формальные семантики и т.д.

Филология программирования должна изучать второй смысловой слой и взаимодействие смысловых слоёв.

При написании псевдокода программист не ограничен языком программирования и выражает понятия в тех обозначениях, которые ему кажутся наиболее уместными и понятными (себе и коллегам).

Филология программирования должна изучать псевдокоды и используемые в них выразительные средства.

Выразительные средства программирования

Выразительные средства программирования

Единица изучения филологии программирования — **выразительное средство**.

- ▶ Выразительное средство **языка программирования** — конструкция, непосредственно входящая в сам язык. Например, цикл со счётчиком в Паскале:

```
for i := 1 to N do ...
```

- ▶ **Идиома** языка программирования — устойчивый способ записи высокоуровневого понятия с использованием более низкоуровневых средств, предоставляемых самим ЯП. Идиома привязана к конкретному ЯП (иногда, семейству ЯП со схожим синтаксисом). Например, в Си-подобных языках цикл `for` — цикл с предусловием, а цикл со счётчиком выражается идиоматически:

```
for (int i = 0; i < N; ++i) ...
```

- ▶ **Паттерн** проектирования — выражение понятия при помощи высокоуровневых средств ЯП. Паттерн, в отличие от идиомы, не привязан к конкретному языку и может быть выражен на самых разных ЯП

Разные выразительные средства на примере

- ▶ Паттерн проектирования Одиночка (Синглтон, Singleton) может быть выражен на любом ОО языке с поддержкой инкапсуляции и полей класса (статических полей).
- ▶ Синглтон Мейерса и счётчик Шварца — идиомы языка C++, используемые для инициализации экземпляра одиночки. Они существенно используют особенности C++ (инициализация статических переменных при первом обращении и препроцессор Си со статической областью видимости в единице трансляции) и на другие языки не переводятся.
- ▶ Язык Java, начиная с версии 1.5, поддерживает перечисления (enum) — классы, для которых создаётся конечное количество экземпляров, т.е. паттерн поддерживается на уровне синтаксиса (до версии 1.5 он был идиомой и был описан в книге Джошуа Блоха «Java™. Эффективное программирование. Второе издание»).

Каталоги выразительных средств

Одной из задач филологии программирования является создание каталогов выразительных средств.

Описания выразительных средств должны включать способы их выражения в ЯП:

- ▶ синтаксис и семантику для встроенных конструкций ЯП, где они поддерживаются нативно,
- ▶ для языков с идиоматической поддержкой — на каких встроенных средствах ЯП эта идиома основана,
- ▶ для паттернов — какой минимум высокоуровневых средств должен поддерживать данный паттерн.

Предыдущий слайд — сильно сокращённый пример разбора для паттерна Синглтон.

Выразительные средства и филология программирования

Замечено, что основные выразительные средства для ЯП используют, как правило, наиболее компактный синтаксис. Предельно компактный синтаксис — пустое место.

- ▶ В Хаскеле пустым местом выражается операция вызова функции.
- ▶ В Рефале пустым местом выражается конкатенация.

Изучение этой тенденции тоже в рамках предлагаемой дисциплины.

Также филология программирования должна изучать **синтаксический сахар** и **синтаксическую соль**.

Эволюция языков программирования

Эволюция ЯП и филология программирования

Филология программирования должна изучать эволюцию ЯП и выявлять тенденции их развития.

- ▶ Докладчик заметил, что одной из тенденций развития языков программирования является включение синтаксического сахара для распространённых идиом (можно привести массу примеров).
- ▶ Другая тенденция — рост размеров ЯП. См. **Свердлов, С. З.** *Арифметика синтаксиса / С.З. Свердлов // PC Week: RE (Компьютерная неделя). — 1998. — № 42-43(166–167). — С. 84. — EDN WDWMGJ.*

Работы по филологии программирования

Работы по филологии программирования

Дисциплины «филология программирования» нет (докладчик надеется, что пока нет), однако, ряд ранее написанных работ вполне можно к ней отнести.

Прежде всего, это книги, описывающие хорошие практики программирования на конкретных языках. Например, серия книг Скотта Мейерса «Effective C++ ...» и книги для других ЯП, написанные в том же стиле.

Однако, это практика. А нужна теория, ибо «нет ничего практичнее хорошей теории».

С.З. Свердлов «Арифметика синтаксиса»

Свердлов, С. З. Арифметика синтаксиса / С.З. Свердлов // PC Week: RE (Компьютерная неделя). — 1998. — № 42-43(166–167). — С. 84. — EDN WDWMGJ.

В данной статье автор измеряет количественные характеристики грамматик нескольких языков программирования (длина грамматики, количество нетерминалов) и показывает тенденцию их роста с выпуском новых версий.

Иллюстрация из книги *Свердлов С.З. Языки программирования и методы трансляции: Учебное пособие.* — СПб.: Питер, 2007. — 638 с., ил., куда данная статья включена как параграф.

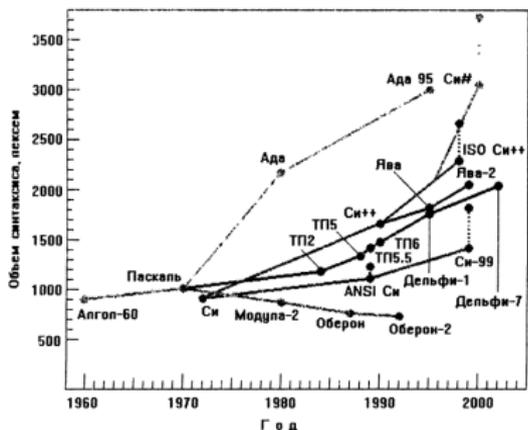
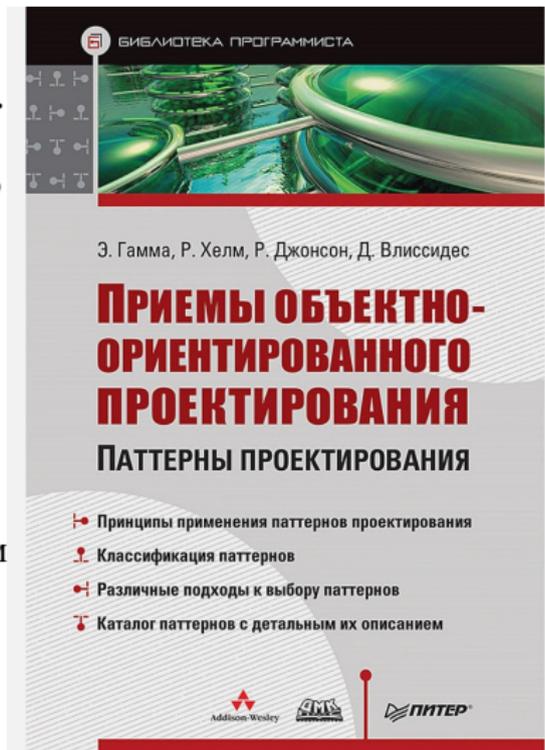


Рис. 1.21. Объем синтаксиса языков программирования

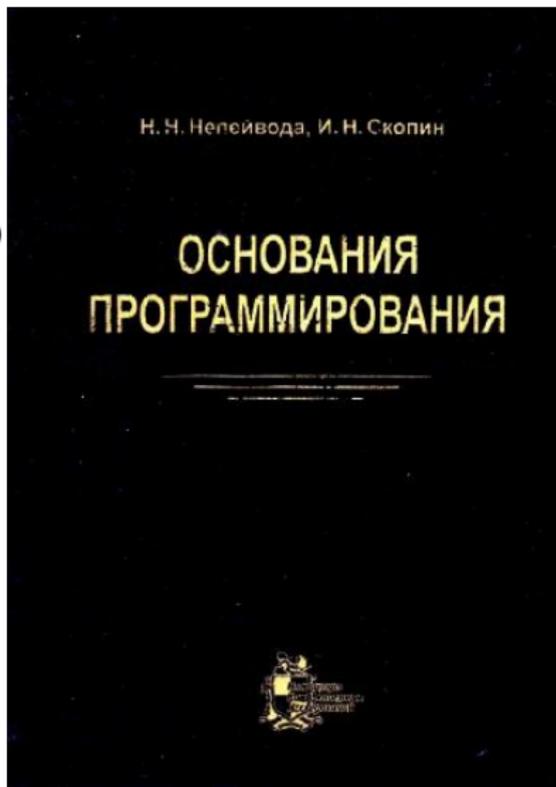
Банда четырёх

- ▶ *Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес. Паттерны объектно-ориентированного проектирования / Пер. с англ.: А. Слинкин. — СПб.: Питер, 2021. — 448 с. — ISBN 978-5-4461-1595-2*
- ▶ Классическая книга с описанием ОО паттернов.
- ▶ Книга должна быть образцом по дальнейшим исследованиям в области филологии программирования.



Основания программирования

- ▶ **Ненеявода Н.Н., Скопин И.Н.** *Основания программирования. 2002* (более точных выходных данных докладчик не нашёл)
- ▶ Учебное пособие систематизирует знания студентов о программировании.
- ▶ В контексте филологии программирования книга содержит обзор выразительных средств самых разных ЯП, их сильные стороны и недостатки.



Выводы

Конспект предыдущих слайдов

Конспект с предыдущих слайдов — что должна изучать новая дисциплина:

- ▶ Филология программирования должна изучать второй смысловой слой и взаимодействие смысловых слоёв.
- ▶ Филология программирования должна изучать псевдокоды и используемые в них выразительные средства.
- ▶ Одной из задач филологии программирования является создание каталогов выразительных средств.
- ▶ Филология программирования должна изучать лаконичность синтаксиса, синтаксический сахар и синтаксическую соль.
- ▶ Филология программирования должна изучать эволюцию ЯП и выявлять тенденции их развития.

Выводы

Филология программирования позволит:

- ▶ развивать хорошие практики программирования, например, разрабатывать новые идиомы для существующих ЯП, переносить на них известные паттерны,
- ▶ разрабатывать новые языки программирования сразу с наборами рекомендаций по их применению,
- ▶ позволит лучше понять программирование как языковую и коммуникативную деятельность.

Приложение

Идиома структурного цикла с предусловием в Фортране 77

Фортран 77 предоставляет только два структурных оператора — цикл со счётчиком DO и блочный оператор ветвления IF ... END IF. Предлагается цикл с предусловием моделировать при помощи помеченного блочного IF с безусловным переходом из строки перед END IF.

```
FUNCTION IGCD(M, N)
  IF (M .LT. 0) M = -M
  IF (N .LT. 0) N = -N
1  IF (N .NE. 0)                                WHILE
      K = MOD(M, N)
      M = N
      N = K
      GOTO 1
  END IF                                        WHILE
  IGCD = M
  RETURN
END
```

В поле комментариев слово WHILE подчёркивает, что блочный IF используется для изображения цикла.