

Независимая  
научно-практическая конференция  
«Разработка ПО 2011»

31 октября - 3 ноября, Москва



# Teaching Programming: What We Miss in Academia, *Or What Would Mr. Platt Say?*



**Evgeny Pyshkin / Евгений Пышкин**  
Санкт-Петербургский государственный  
политехнический университет

**КСИПТ** | КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И  
ПРОГРАММНЫХ ТЕХНОЛОГИЙ ФТК СПбПУ

# classified

<http://kspt.ftk.spbstu.ru/info/staff/pyshkin/>



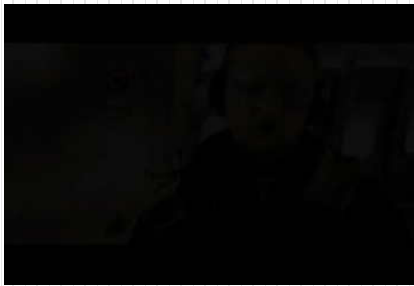
- ◆ Изучение программирования
  - ◆ ~ с 1982 года / Программируемые калькуляторы, Pascal, Fortran, PL/I, Lisp, Prolog, C, C++, JavaScript, C#, Visual Basic, Java...
- ◆ Преподавание программирования
  - ◆ с 1992 года / Pascal, C/C++, C#, Java
- ◆ Курсы
  - ◆ Основы алгоритмизации и программирования
  - ◆ Объектно-ориентированное программирование
  - ◆ Структуры и алгоритмы компьютерной обработки данных
  - ◆ Object-Oriented Software Engineering

“If programmers will be building a plane...”



“If programmers will be building a plane...”

---



# ОСОБЕННОСТИ ПРОГРАММИРОВАНИЯ КАК АКАДЕМИЧЕСКОЙ ДИСЦИПЛИНЫ

Programming as an Academic Course

# Особенности программирования как академической дисциплины

- Разнородность задач, стоящих перед преподавателем
  - Научить подходам и концепциям
  - Научить конкретному языку
  - Выбрать задачи для иллюстрации изучаемых идей
  - Сформировать проектные навыки
  - Обеспечить поддержку связанных курсов
- Изменение языковых стандартов
- Многообразие и развитие технологий
  - Средства и интегрированные среды разработки
  - Библиотеки построения пользовательского интерфейса
  - Различие архитектур
- Необходимость постоянной корректировки содержания лекционного курса и практических занятий

# Связь с другими курсами

**Информатика**

**Математика**

**Русский язык**

**Вычислительная  
математика**

**Программирование**

**Иностранный язык**

**Теория алгоритмов и  
автоматов**

**Системный анализ и принятие решений**

**Базы данных**

**Технологии  
компьютерных сетей**

**Языки  
программирования**

**Системное ПО**

**Распределенные  
системы**

**Математическая  
логика**

**Трансляторы**

**Телекоммуникационные  
системы**

**Автоматизация  
проектирования  
программно-  
аппаратных систем**

**Промышленные  
технологии  
разработки ПО**

**Инженерия  
разработки ПО**

# ПРОБЛЕМЫ, С КОТОРЫМИ МЫ СТАЛКИВАЕМСЯ

---

What Kind of Discipline Do We Teach  
Teach To Translate the Tasks



# Проблемы, с которыми мы сталкиваемся

- Изучение программирования подменяется изучением смежной дисциплины



- Теория алгоритмов
- Вычислительная математика
- Системный анализ
- Компьютерная графика
- Информатика
- ...

# Проблемы, с которыми мы сталкиваемся

- Ограниченные представления студентов
  - о переходе от математической модели к программной
  - о формулировании требований к разрабатываемой программе
  - о других активностях (отладка, тестирование, сопровождение)

*Well, you like writing code,  
but you don't like to test it,  
so let's take JUnit.*

*Don't test, simply write the test code.*

## Переход от языка предметной области на язык программирования: пример

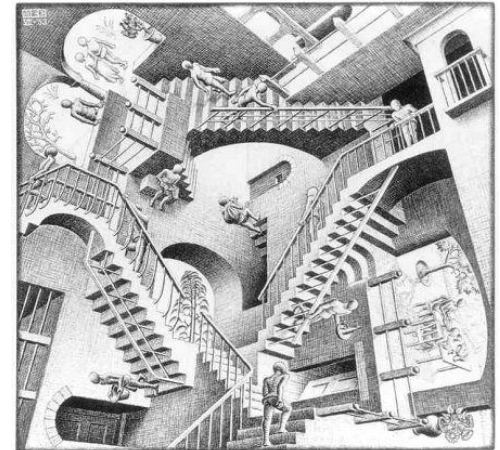
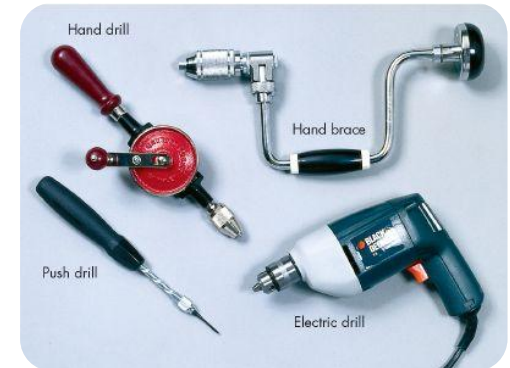
$$\sin(t) = \sum_{k=0}^{\infty} (-1)^k \frac{t^{2k+1}}{(2k+1)!}$$

1. Чему равно  $(-1)^k$  ?
2. Как вычислить  $(2k+1)!$  ?
3. Четный или нечетный элемент ряда?
4. Как вычислить значение очередного члена ряда?
5. Как вычислить сумму, если число членов ряда  $\infty$  ?

**Что такое «типичные ошибки»?**

# Проблемы, с которыми мы сталкиваемся

- Студенты умеют пользоваться только ограниченным набором средств проектирования
- Задания неинтересны студентам
  - Потому что слишком сложные
  - Потому что слишком научные
  - Потому что оторваны от окружающего ИХ мира
  - ...



# ИЗМЕНЧИВОСТЬ И СТРУКТУРА

---

Teach to Change the Code

Teach to See the Code Structure

# Проблемы

- Студенты не понимают, что изменчивость ПО является его неотъемлемым качеством
  - Использование средств управления проектами, задачами, изменениями
- Студенты (и преподаватели) абсолютизируют факт работающей программы
- Мы не учим студентов обсуждать проектные решения

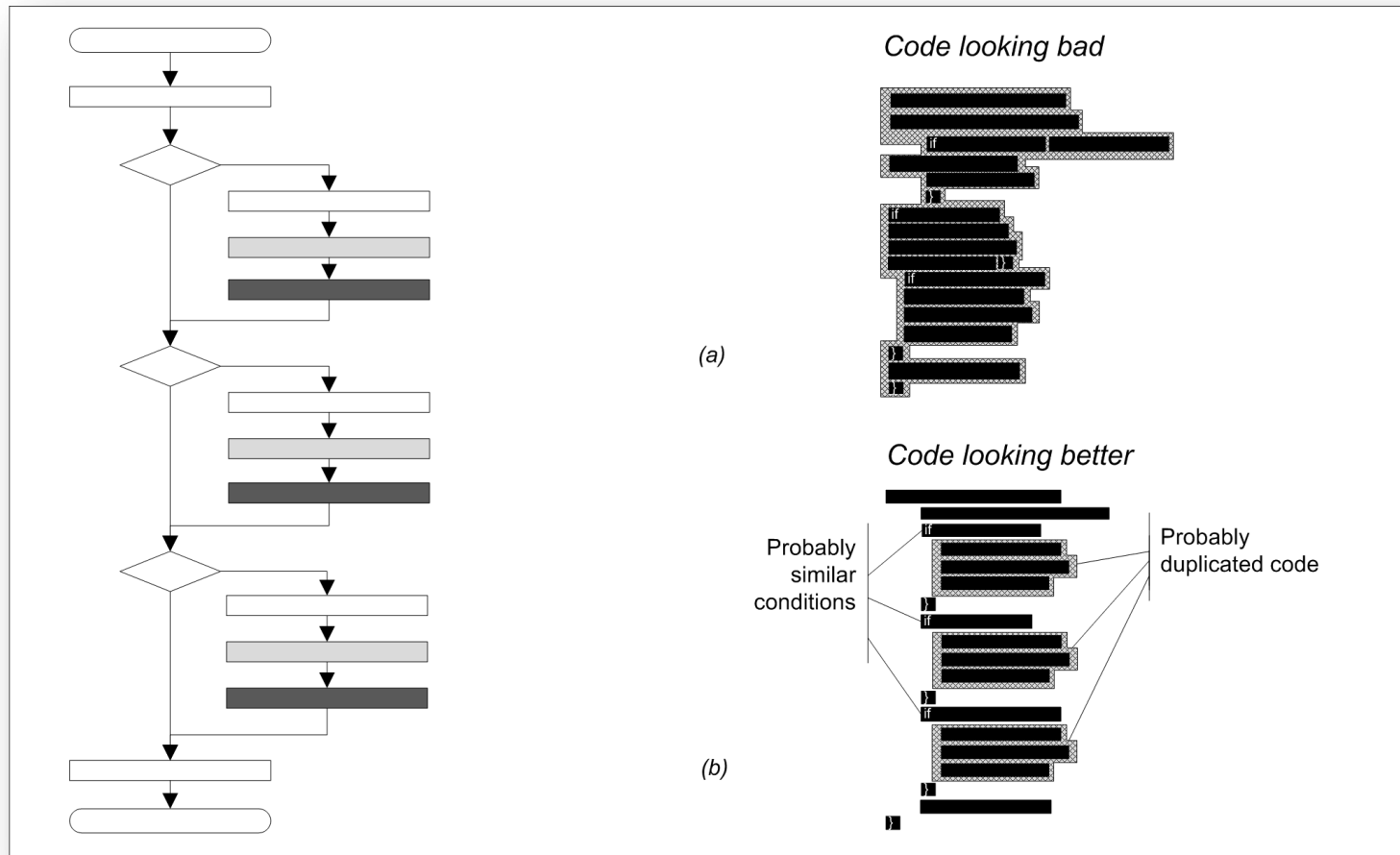
– *Каким языком в первую очередь должен владеть разработчик?*  
– *Русским.*

– *What language has the developer to know first of all?*  
– *English.*

開発者はまず  
はじめに何語  
を知るのか  
日本語

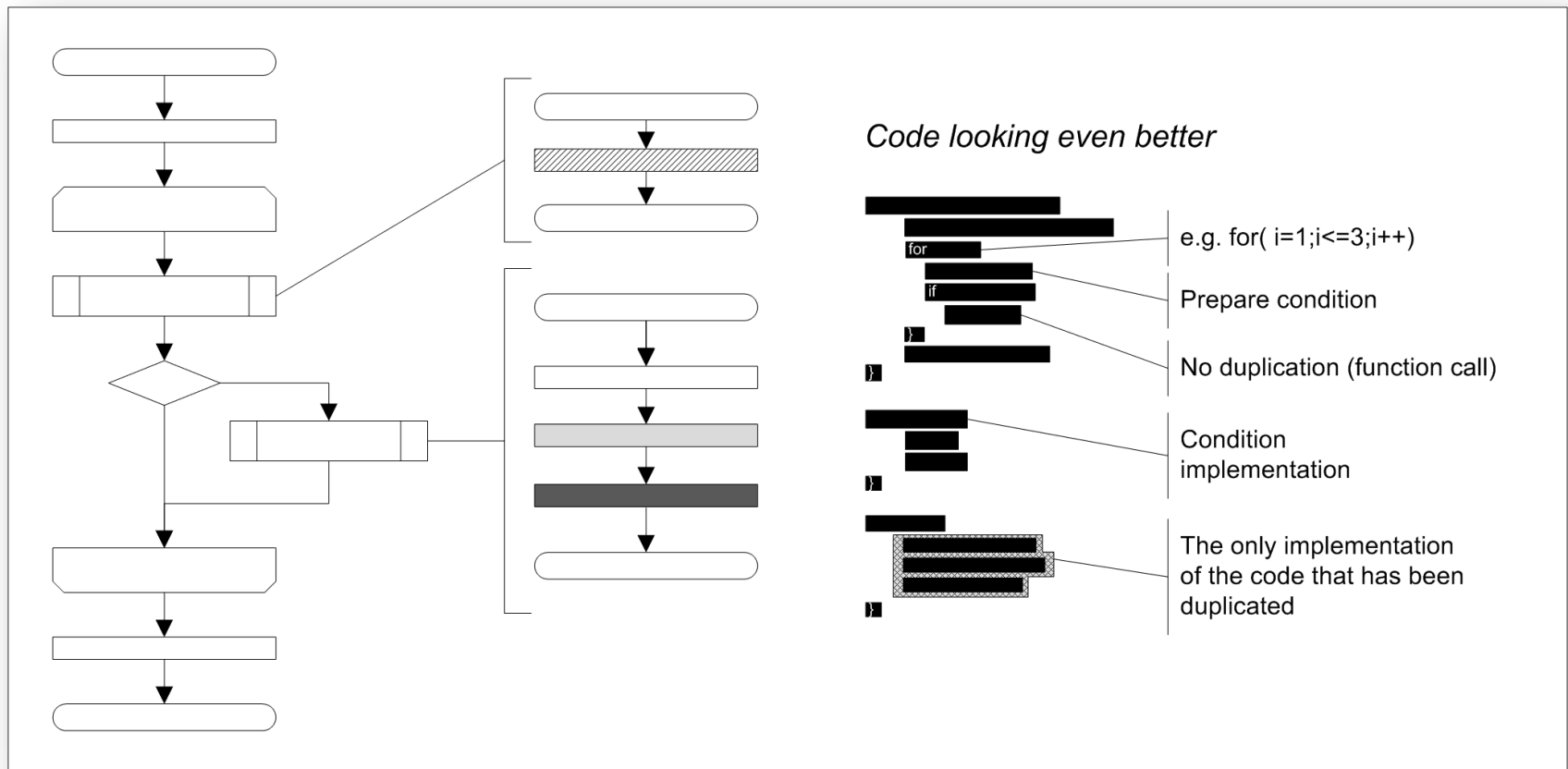
# О структуре кода и об «оценке на расстоянии»

- Method of distant look examination 😊



# О структуре кода и об «оценке на расстоянии»

- Method of distant look examination 😊





# ВЫБОР ЯЗЫКА ПРОГРАММИРОВАНИЯ

---

Choose the Right Language(s)

# Проблема выбора языка программирования

- Выбор языка программирования, используемого в базовом курсе
  - Должен позволять иллюстрировать большинство современных концепций проектирования
  - Должен быть языком общего назначения
  - Должен быть актуальным
    - Использоваться при промышленной разработке ПО
    - Использоваться в научно-исследовательской работе
- **Должен быть “cool”**

*Преподаватель должен объяснить студентам, почему он использует этот язык*

# Проблема выбора языка программирования

TIOBE Index: August 2011

<http://www.tiobe.com>

Position Aug 2011	Position Aug 2010	Delta in Position	Programming Language	Ratings Aug 2011	Delta Aug 2010	Status
1	1	=	Java	19.409%	+1.42%	A
2	2	=	C	17.390%	-0.48%	A
3	3	=	C++	8.433%	-1.23%	A
4	4	=	PHP	6.134%	-3.05%	A
5	6	↑	C#	6.042%	+1.06%	A
6	9	↑↑↑	Objective-C	5.494%	+2.34%	A
7	5	↓↓	(Visual) Basic	5.013%	-0.40%	A
8	7	↓	Python	3.415%	-0.81%	A
9	8	↓	Perl	2.315%	-1.11%	A
10	11	↑	JavaScript	1.557%	-0.84%	A
11	23	↑↑↑↑↑↑↑↑↑↑	Lua	1.362%	+0.83%	A
12	12	=	Ruby	1.329%	-0.65%	A

# О форме и содержании компьютерной программы как объекта искусства (послесловие)

---

Afterword

About the Form and Content of the Software as an Object of  
Art

# О форме и содержании компьютерной программы

- Форма
  - Модульная структура
  - Аннотации
  - Комментарии
  - Конструирование имен
  - Форматирование

```
// State NXTLIT: Next Litera Expected
void FSM_Model::Process_NXTLIT( const Litera& lit, Lexema& ident )
{
    switch( lit.GetSynterm() )
    {
        case Litera::SPACE : ident.LexStop();
                            output << ident << " ";
                            state = S0;
                            break;

        case Litera::ENDFILE : ident.LexStop();
                              output << ident << " ";
                              state = STOP;
                              break;

        case Litera::LETTER :
        case Litera::DIGIT : ident.LexNext( lit );
                            state = NXTLIT;
                            break;

        default : state = ERROR;
                break;
    }
}
```

***Форма произведения не просто то, что «окружает» содержание, отделяя его от внешнего мира, но и то, что связывает его с окружающим миром (по Д. Лихачеву)***

# Из мыслей Дмитрия Лихачева

- Статья «Несколько мыслей о «неточности» искусства и стилистических направлениях»
  - Создавая произведение искусства, автор вкладывает в это произведение («*программирует*») акт "воспроизводства" в сознании рецептора
  - Воспроизводство это только условно повторяет акт творения художника и имеет широкие потенциальные возможности своей реализации в акте сотворчества читателя, зрителя, слушателя



# Из мыслей Павла Флоренского

- Статья «Анализ пространственности в художественно-изобразительных произведениях»
  - «Ведь произведения изобразительных искусств, пока они не прочитаны и не осуществлены во времени, вообще для нас не стали искусством. В этом смысле они не более понятны музыкальной партитуре, если таковая, будучи сама в себе лишь родом плоского орнамента, не зазвучала ритмически инструментальным или внутренним чтением»

The image displays a page from a musical score for orchestra. The staves are labeled as follows: Flauti, Oboi, Clarinetti in C, Fagotti, Corni in C, Trombe in C, Tromboni, Timpani in C.G., Violino I, Violino II, Viola, Violoncello, and Basso. The score is marked 'Andate' at the top. The music is written in a common time signature (C) and includes various dynamic markings such as 'pp' (pianissimo) and 'p' (piano). The score is presented in a layered, overlapping fashion, with the top sheet being the most prominent and subsequent sheets visible behind it, creating a sense of depth and complexity.

Независимая  
научно-практическая конференция  
«Разработка ПО 2011»

31 октября - 3 ноября, Москва



# Teaching Programming: What We Miss in Academia, *Or What Would Mr. Platt Say?*



**Evgeny Pyshkin / Евгений Пышкин**  
Санкт-Петербургский государственный  
политехнический университет

**КСИПТ** | КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И  
ПРОГРАММНЫХ ТЕХНОЛОГИЙ ФТК СПбГУ