Объектно-ориентированное программирование

Object-oriented programming

I. Введение

Introduction

The core of what I now have to call "**real OOP**" – namely *encapsulated modules all the way down with pure messaging* – still hangs in there strongly because **it is nothing more than an abstract view of complex systems**.

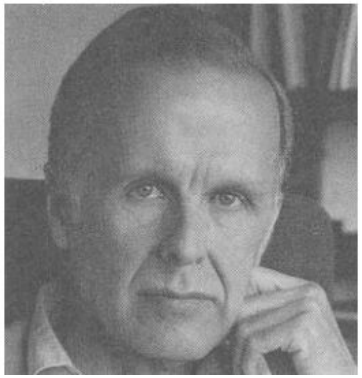
Alan C. Kay

https://www.youtube.com/watch?v=zwucsB2EfXc?t=1029

leon.brinzan@iis.utm.md

Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs

John Backus
IBM Research Laboratory, San Jose



Conventional programming languages are growing ever more enormous, but not stronger. Inherent defects at the most basic level cause them to be both fat and weak: their primitive word-at-a-time style of programming inherited from their common ancestor—the von Neumann computer, their close coupling of semantics to state transitions, their division of programming into a world of expressions and a world of statements, their inability to effectively use powerful combining forms for building new programs from existing ones, and their lack of useful mathematical properties for reasoning about programs.

http://worrydream.com/refs/Backus-CanProgrammingBeLiberated.pdf

Критерии к моделям исчисления

1. Наличие математического обоснования (описания).
2. Наличие системы хранения состояния программы (“истории”).
3. Семантический тип (state-transition, reduction).
4. Ясность и полезность (выражают ли программы процесс исчисления ясно; воплощают ли программы концепции, которые помогают формулировать процессы).

Классификация моделей исчисления

1. Простые (конечные автоматы, машины Тьюринга и т.п.).
2. Аппликативные (λ-исчисление, Lisp и т.п.).
3. Фон-Неймановские (компьютеры, Fortran и т.п.).
4. Другие (напр. data-flow-языки программирования).

Воронка фон Неймана (von Neumann bottleneck)

- Предназначение программы – изменить данные в хранилище
- Данные передаются из хранилища на ЦПУ только по шине
- Большая часть данных – мета-данные (адреса изменяемых значений)
- Как результат: программист находится в рамках мышления “**по одному слову за раз**” (переменные, прыжки между инструкциями, присваивание)

Скалярное произведение векторов

```
c := 0
for i := 1 step 1 until n do
    c := c + a[i] * b[i]
```

- Невидимое состояние
- Отсутствие иерархии (комбинации)
- Выражает повторение инструкции (трудно мысленно проследить)
- Состоит из повторных присваиваний результата выражения
- Работает только для векторов длины **n**

Скалярное произведение векторов

```
(Insert +) ◦ (ApplyToAll *) ◦ Transpose
```

Например, для вектора {{1, 2, 3}, {6, 5, 4}}:

1. **Transpose** {{1, 2, 3}, {6, 5, 4}} : {{1, 6}, {2, 5}, {3, 4}}
2. (**ApplyToAll ***) {{1, 6}, {2, 5}, {3, 4}} : {6, 10, 12}
3. (**Insert +**) {6, 10, 12} : (**+** {6, (**+** {10, 12})})
4. **+** {6, 22} : 28

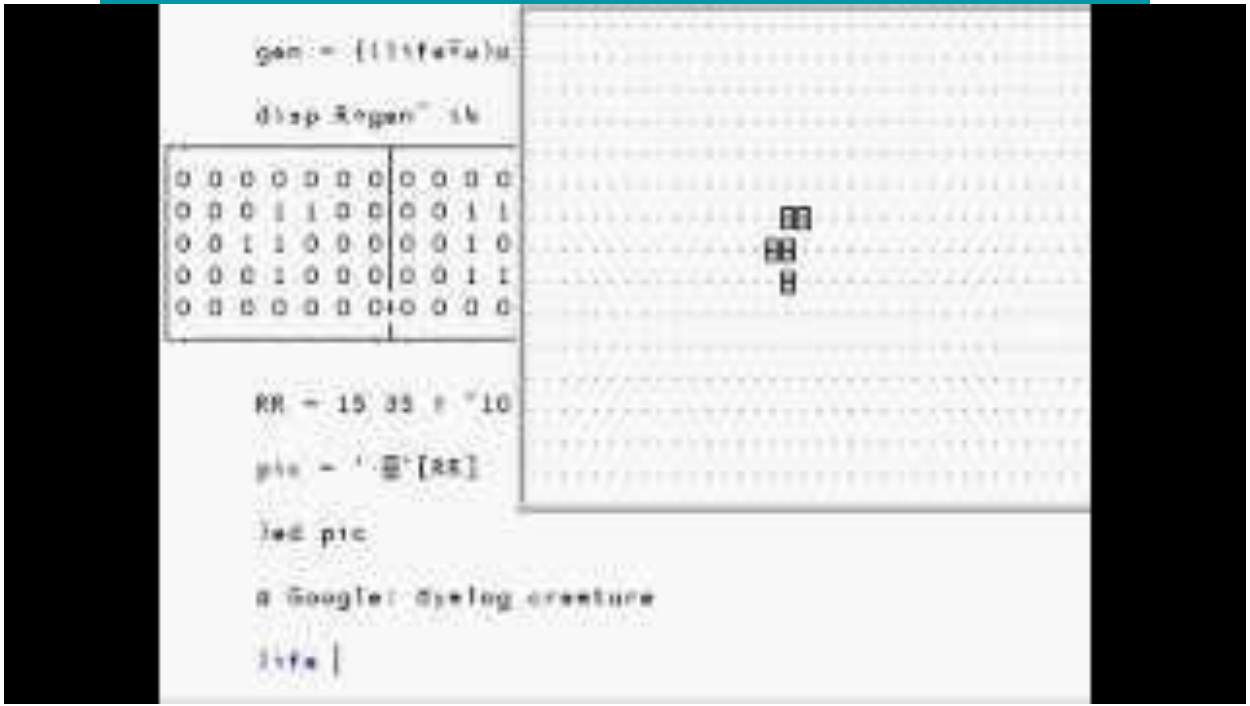
- Отсутствие состояния
- Есть строгая иерархия (композиция)
- Оперирует цельными концепциями, а не машинными словами
- Полностью обобщена (нет привязки к конкретным именам, длине векторов и т.д.)

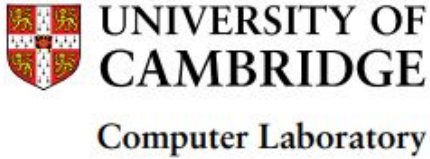
Альтернативный стиль

1. Функциональный стиль **без использования переменных.**
2. **Алгебра** функциональных программ.
3. **Формализованная** система (позволяет комбинировать стиль и алгебру системы для построения более сложных систем).
4. **Аппликативная** система состояний и переходов между ними.

Альтернативный стиль – APL

https://www.youtube.com/watch?v=a9xAKttWqP4





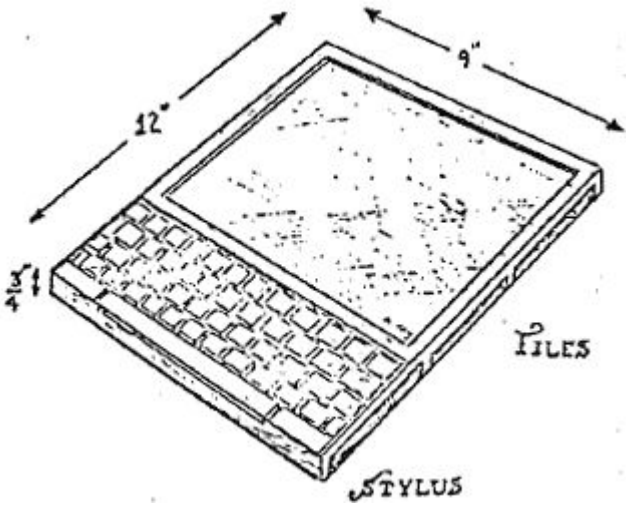
Sketchpad: A man-machine graphical communication system

Ivan Edward Sutherland

https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf

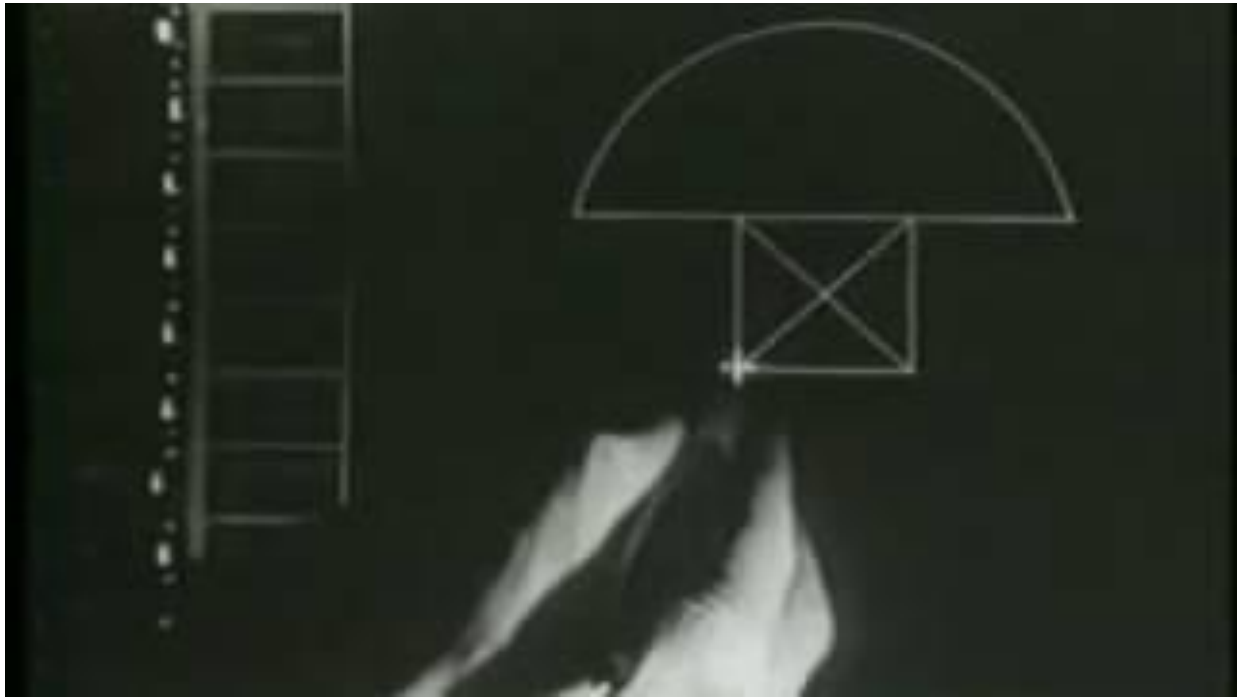
Связь Sketchpad с ООП

- Понятие “класса” и “экземпляра” (основаны на базе “плексов” в Algol 68; напрямую повлиял на появление классов и объектов в языках SIMULA и Smalltalk)
- Концепция Dynabook* была основана на Sketchpad (реализована в Smalltalk)
- Понятие “абстрактного типа данных” (проникло в систему через работы Ч. Хоара, почти одновременно с SIMULA, а оттуда – в C++)
- Ранняя реализация связного списка для автоматической очистки памяти



Sketchpad – прародитель ООП

https://www.youtube.com/watch?v=5RyU50qbvzQ



Основные возможности системы

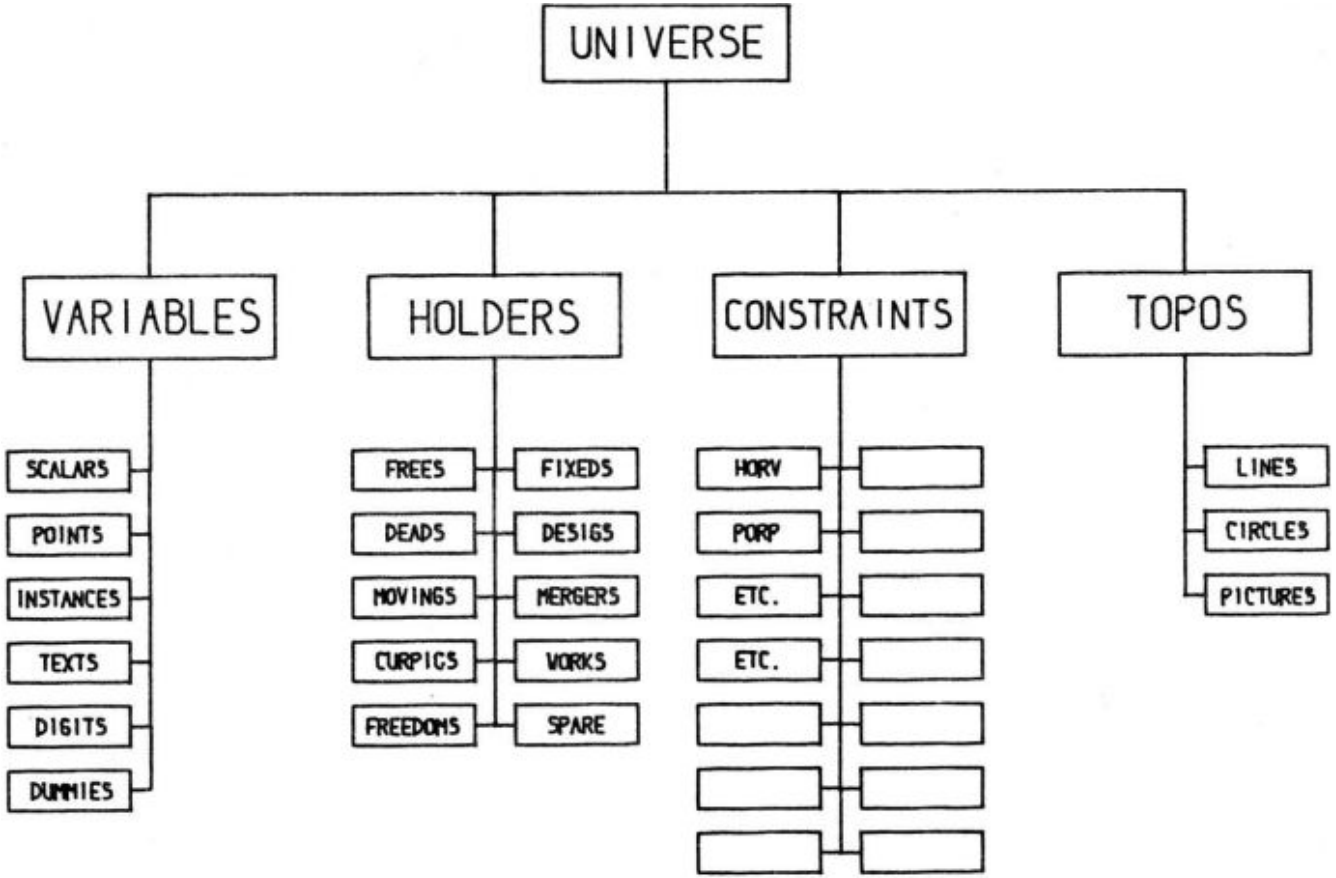
Master drawing – единый чертеж фигуры, который используется для создания копий, при этом изменение “мастера” отражается на копиях.

Экземпляр – рекурсивная структура данных, обладающая топологическими свойствами, которая может содержать другие экземпляры; используется для отображения фигур чертежа.

Атомарные ограничения – базовые отношения между фигурами (напр., требование чтобы шестиугольник вписывался в круг), которые насаждаются системой автоматически.

Наследование – возможность хранить информацию о фигурах в “иерархиях”, вершиной которых является “обобщенная структура данных”.

Обобщенные операции – части системы, отвечающие за поведение всех сущностей на экране, независимо от их “типа”.



24	4	VARIABLES	TYPE
-2		0000	SPECB
TYPEWRITER CODE NAME			NAME
SUBROUTINE ENTRY			DISPLAY
FIT SCOPE AROUND IT			HOWBIG
APPLY TRANSFORMATION			MOVIT
24.16..			SIZE
NORMAL PICTURE KIND			KIND
FOUR COMPONENTS			TUPLE
VALUE AT IVAL			VARLOC