

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Введение

СОДЕРЖАНИЕ

- ◎ Содержание курса
- ◎ Сложность , присущая ПО
- ◎ История языков программирования и борьба со сложностью
- ◎ Объектно-ориентированный подход, язык C++

АТТЕСТАЦИЯ

- ◎ 18 лекций; 18 практических занятий
- ◎ Аттестация: 100-баллов максимум.

0-50 <<2>>
51-65 <<3>>
66-75 <<4>>
76-100 <<5>>

- ◎ Экзамен: максимум 50 баллов;
- ◎ Лабораторные: максимум 50 баллов;
- ◎ Если по лабораторным менее 26 баллов - то зачетное задание (максимум – 26 баллов).

СОДЕРЖАНИЕ КУРСА

1. Основные концепции объектно-ориентированного программирования
2. Языки C/C++.
3. Объектно-ориентированные средства языка C++
4. Наследование классов
5. Механизмы обработки исключительных ситуаций
6. Шаблоны C++
7. Паттерны проектирования

ОСНОВНАЯ ЛИТЕРАТУРА

- ◎ *Страуструп Б.* Язык программирования С++. Специальное издание./Пер. с англ. - СПб.; –М.: "Невский диалект" - "Издательство БИНОМ", 2008 г.- 1104 с.
- ◎ *Павловская Т.А.* С/С++. Программирование на языке высокого уровня. –СПб.: Питер, 2009. -461 с.
- ◎ *Подбельский В.В.* Язык Си++ : Учеб. пособие для вузов по направлениям "Приклад. математика" и "Вычисл. машины, комплексы, системы и сети" -М. : Финансы и статистика , 2001 г. - 559 с.
- ◎ *Буч Г. и др.* Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е изд.: Пер. с англ.. – М.:«И.Д. Вильямс», 2010. – 720 с.
- ◎ *Макконнелл С.* Совершенный код. СПб.: Питер, 2007. – 896 с.

Сложность ПО

Сложность

Мы окружены сложными системами:

- ◎ персональный компьютер
- ◎ любое дерево, цветок, животное
- ◎ любая материя – от атома до звезд и галактик
- ◎ общественные институты – корпорации и сообщества

СТРУКТУРА СЛОЖНЫХ СИСТЕМ

Большинство сложных систем обладает иерархической структурой.



Сложность ПО

Не все ПО – сложное. Существует класс приложений которые проектируются разрабатываются и используются одним и тем же человеком. Но они имеют ограниченную область применения.

Вопросы сложности появляются при разработке корпоративного ПО, промышленного программирования.

Сложность ПО

Сложность ПО вызывается четырьмя основными причинами:

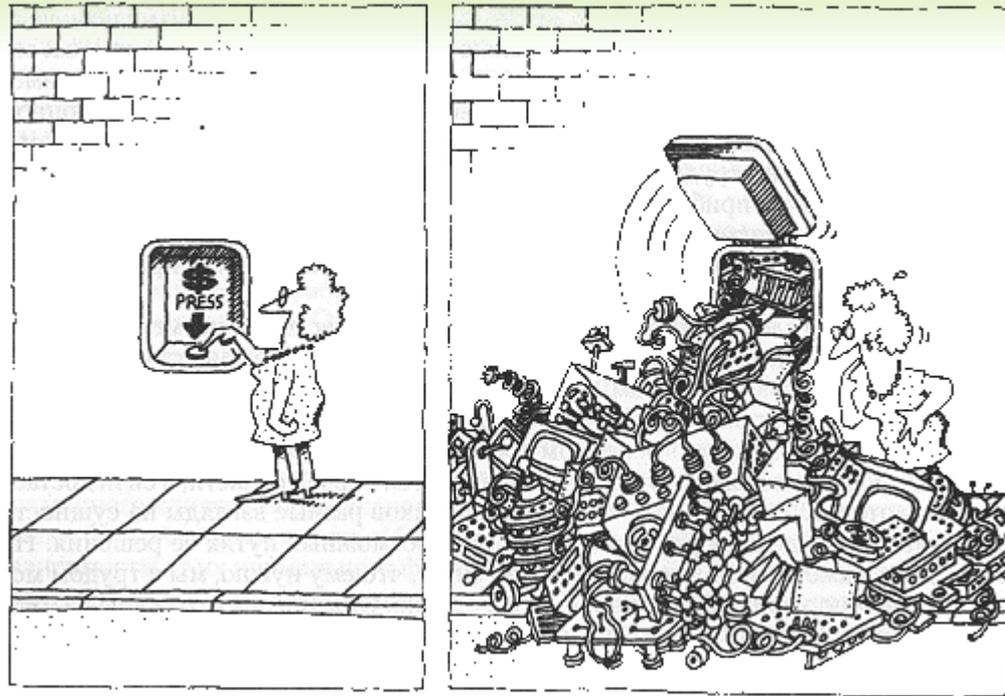
- ⊙ сложностью реальной предметной области, из которой исходит заказ на разработку;
- ⊙ трудностью управления проектированием;
- ⊙ необходимостью обеспечить достаточную гибкость программы;
- ⊙ сложностью описания поведения больших дискретных систем.

СЛОЖНОСТЬ ПРЕДМЕТНОЙ ОБЛАСТИ

- ◎ Система управления авиалайнером
- ◎ ПО коммутатора сотовой связи
- ◎ Система управления движением поездов
- ◎ Автономный робот на поверхности Марса

«Мало того, что системы сложны сами по себе, так еще и появляются нюансы «недопонимания» между разработчиками и пользователями, вызванный недостатком знаний в предметной области другой группы»

ТРУДНОСТИ УПРАВЛЕНИЯ ПРОЕКТИРОВАНИЕМ



Задача группы проектировщиков – создать
иллюзию простоты.

НЕОБХОДИМОСТЬ ОБЕСПЕЧЕНИЯ ГИБКОСТИ

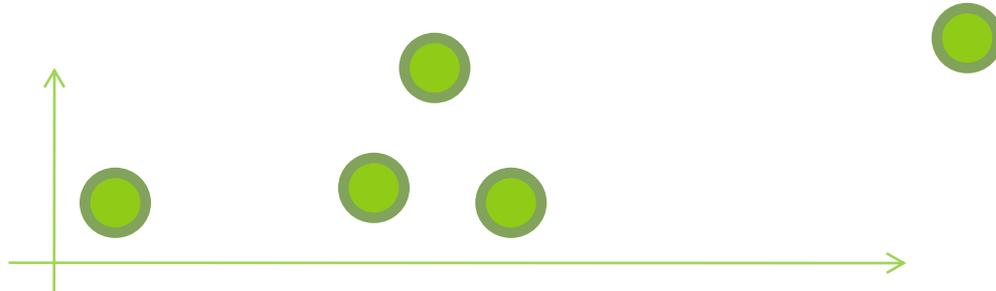
Строительная компания обычно не владеет лесхозом для создания пиломатериалов; не владеет металлургическим заводом для создания уникальных стальных балок.

В процессе разработки ПО такой подход используется повсеместно для обеспечения чрезвычайной гибкости.

СЛОЖНОСТЬ ОПИСАНИЯ ДИСКРЕТНЫХ СИСТЕМ

Непрерывные функции меняются предсказуемо:
малое изменение параметра приводит к
малому изменению значения.

Дискретные системы не могут быть описаны
непрерывными функциями, и небольшое
изменение входных значений может
непредсказуемо отразиться на результате.



ПРЕДЕЛЫ ЧЕЛОВЕЧЕСКИХ ВОЗМОЖНОСТЕЙ

Эксперименты показывают, что максимальное количество единиц информации, за которыми человеческий мозг может одновременно следить, приблизительно равно семи.

Но при организации системы в процессе ее проектирования, необходимо думать сразу о многом, необходимо рассматривать большие, сложные и не всегда детерминированные пространства состояний.

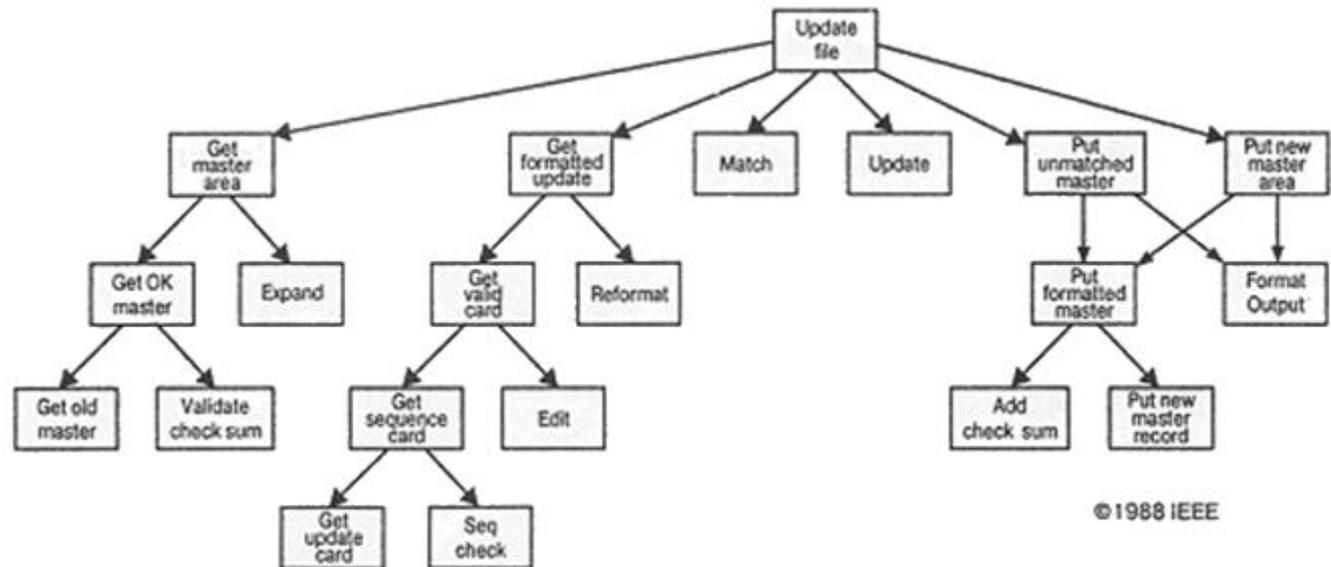
ДЕКОМПОЗИЦИЯ - БОРЬБА СО СЛОЖНОСТЬЮ

Divide et Impera (разделяй и властвуй)

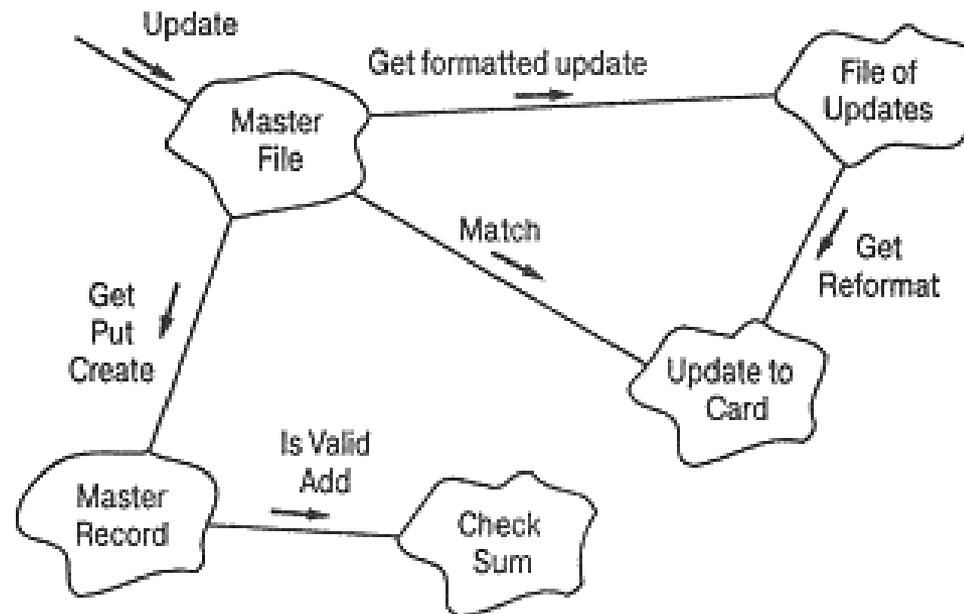
Необходимо разделять систему на независимые подсистемы, каждую из которых разрабатывать отдельно. Методы декомпозиции:

- ⊙ алгоритмическая декомпозиция
- ⊙ объектно-ориентированная декомпозиция

АЛГОРИТМИЧЕСКАЯ ДЕКОМПОЗИЦИЯ



ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ ДЕКОМПОЗИЦИЯ



КРАТКАЯ ИСТОРИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

ПОКОЛЕНИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ (1-2)

Выделяют следующие этапы развития ЯП высокого уровня:

- ◎ Языки первого поколения (1954-1958)
 - ◎ FORTRAN 1 Математические формулы
 - ◎ ALGOL-58 Математические формулы
- ◎ Языки второго поколения (1959-1961)
 - ◎ FORTRAN II Подпрограммы
 - ◎ ALGOL-60 Блочная структура, типы данных
 - ◎ COBOL Описание данных, работа с файлами
 - ◎ LISP Обработка списков, указатели, сборка мусора

ПОКОЛЕНИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ (4)

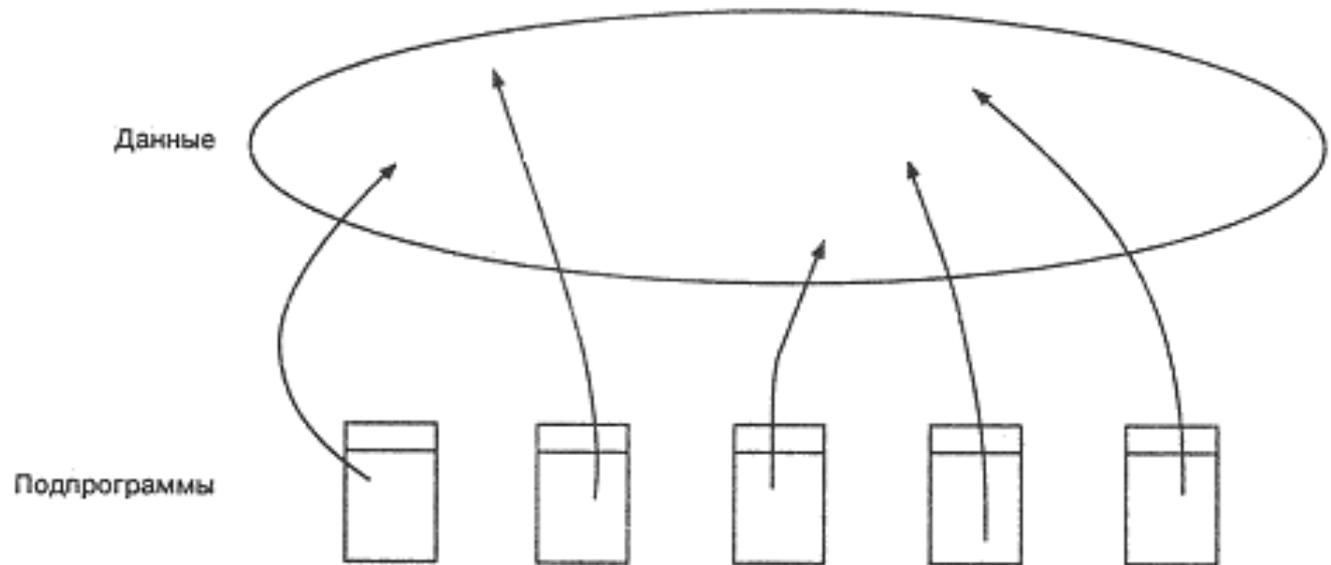
◎ Бум ООП (1980-1990)

- ◎ Smalltalk 80 Чисто объектно-ориентированный язык
- ◎ C++ C + Simula
- ◎ Ada83 Строгая типизация; сильное влияние Pascal

◎ Появление инфраструктур (1990-...)

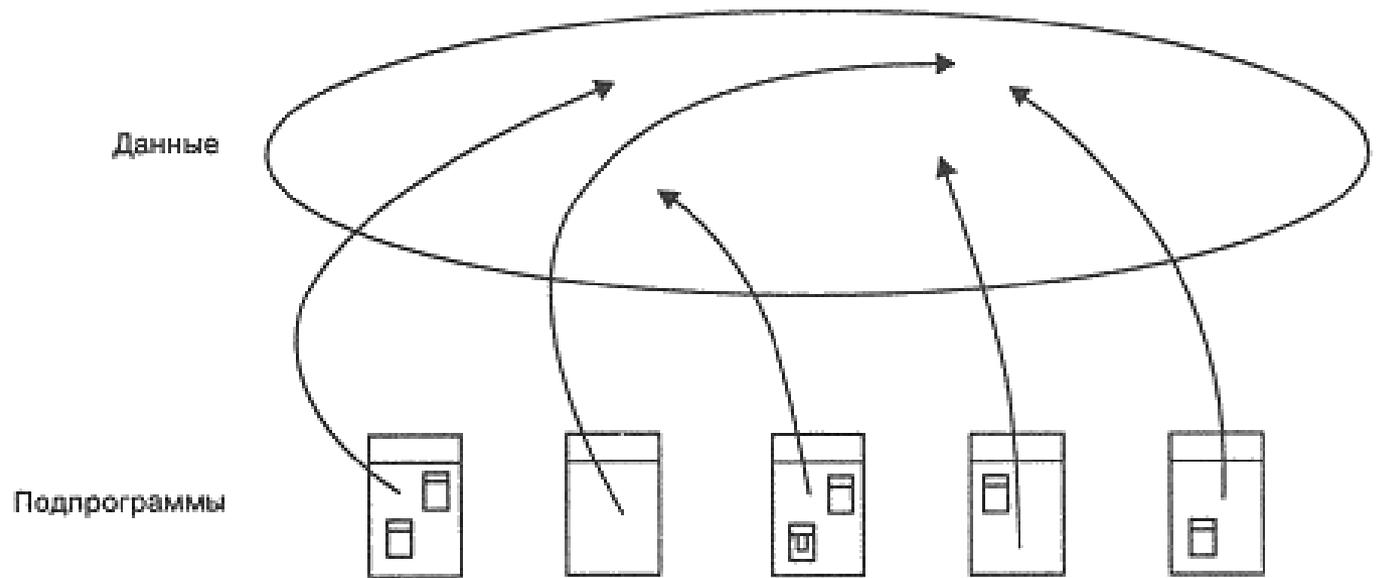
- ◎ Java Блочная структура, типы данных
- ◎ Python Объектно-ориентированный язык сценариев
- ◎ Visual C# Конкурент языка Java для среды Microsoft .NET

Топология языков первого поколения



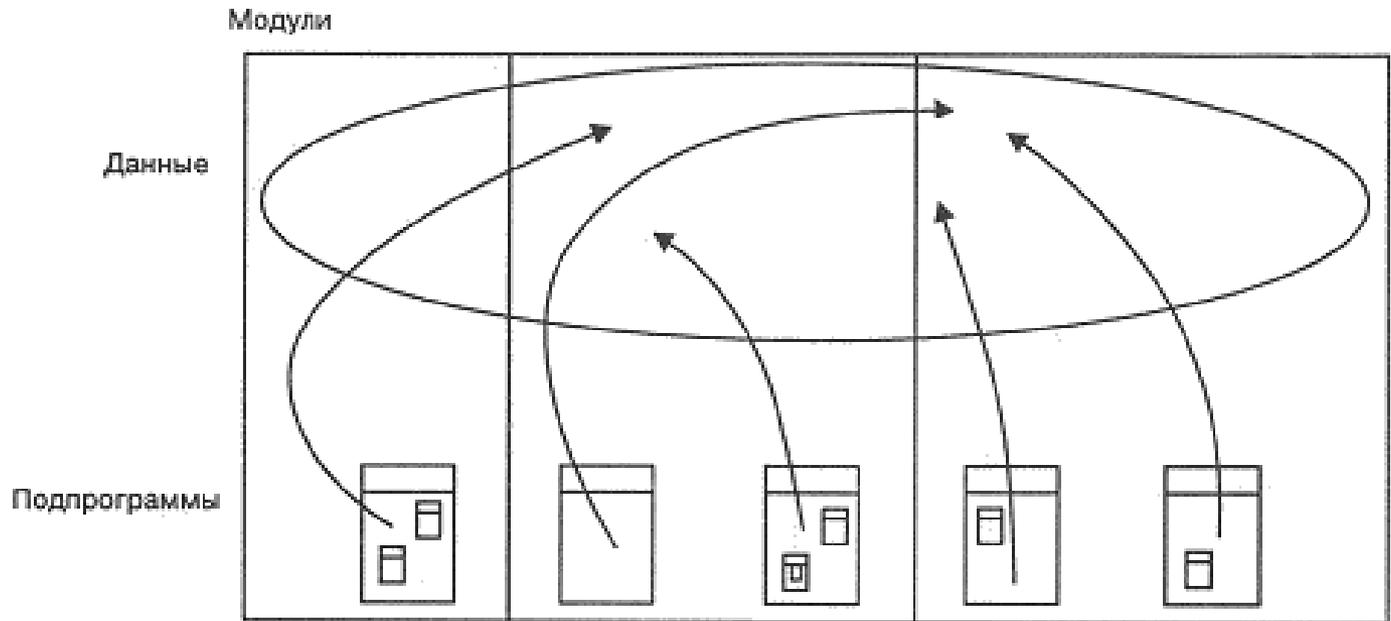
Программы имеют относительно простую структуру, состоящую только из глобальных данных и подпрограмм.

Топология языков ВТОРОГО ПОКОЛЕНИЯ



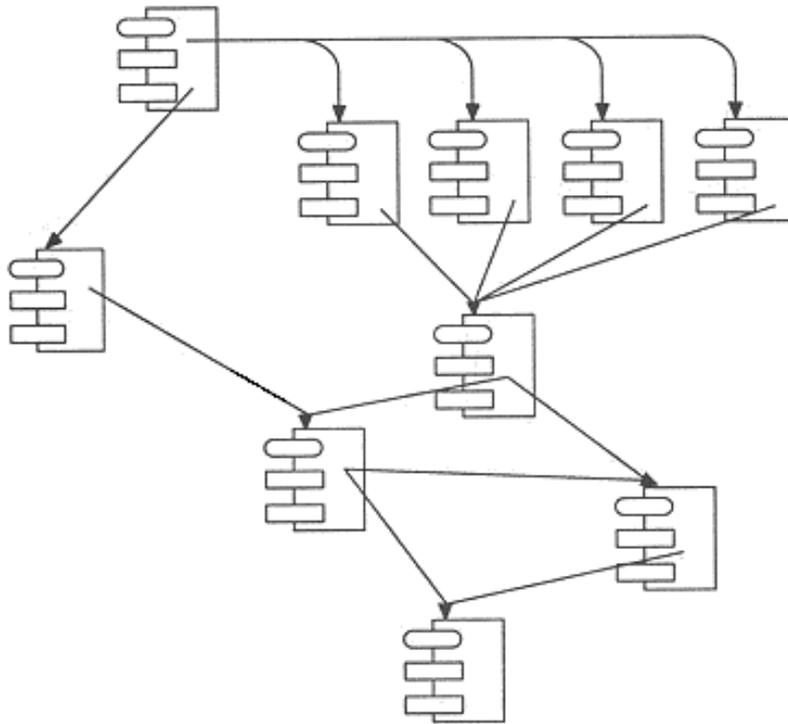
Появление процедурной абстракции, которая позволила описать абстрактные программные функции в виде подпрограмм.

Топология языков ТРЕТЬЕГО ПОКОЛЕНИЯ



В модули собирали подпрограммы, которые будут изменяться совместно, но их не рассматривали как новую технику абстракции.

Топология ООП



Основным элементом конструкции служит модуль, составленный из логически связанных классов и объектов, а не подпрограмма.

- ◎ Описывать реальные системы с использованием компьютера – сложно:
 - ◎ Предметная область
 - ◎ Управление проектированием
 - ◎ Обеспечение гибкости
 - ◎ Описание дискретных систем
- ◎ Для борьбы со сложностью используется декомпозиция (алгоритмическая или объектно-ориентированная).
- ◎ В языках разных поколений применялись различные топологии декомпозиции (от процедурной абстракции к абстракции ООП).