Объектно-ориентированное программирование

Введение

Аттестация

- 14 лекций; 14 практических занятий
- Аттестация: 100-баллов максимум.

Баллы	Итоговая оценка
0 – 50	«Неудовлетворительно»
51 – 65	«Удовлетворительно»
66 – 75	«Хорошо»
76 – 100	«Отлично»

- Экзамен: максимум 51 балл;
- Практические работы: максимум 35 баллов;
- Промежуточное тестирование: максимум 14 баллов

Занятия

- Пары 25 сентября не будет.
- Она переносится на
  - **11 сентября, 15:20, ауд. 330/3б.**

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 C++. Специальное издание./Пер. с англ. - СПб.; –М.: "Невский диалект" - "Издательство БИНОМ", 2008 г.- 1104 с.
- *Павловская Т.А.* C/C++. 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.
- *Мейерс С.* Эффективное использование C++. 55 верных способов улучшить структуру и код ваших программ. Москва: ДМК-Пресс, 2006. — 300 с.
- *Фримен Э.* Паттерны проектирования. СпБ.:Питер, 2014. - 656 с.
- *Эрих Гамма, Ричард Хелм, Ральф Джонсон, Джон Влиссидес.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. СпБ.:Питер, 2014. – 368 с.

Сложность ПО

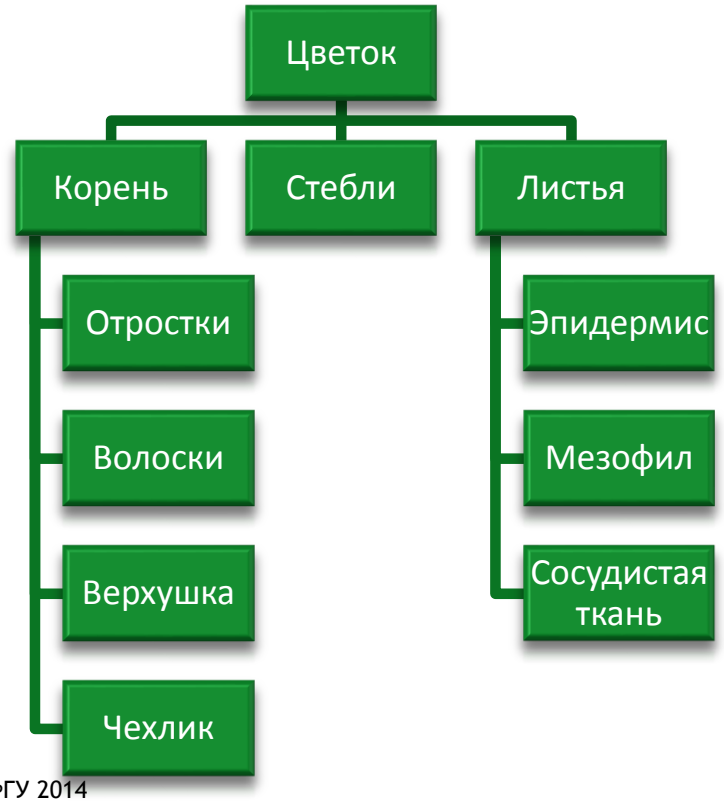
Сложность

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:

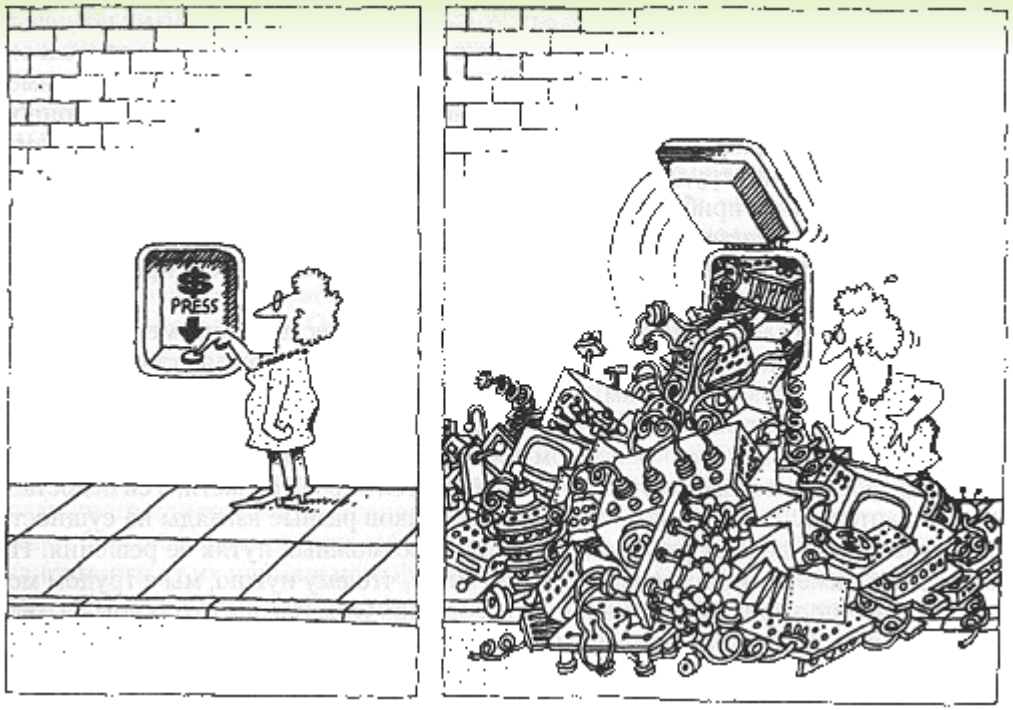
- сложностью реальной предметной области, из которой исходит заказ на разработку;
- трудностью управления проектированием;
- необходимостью обеспечить достаточную гибкость программы;
- сложность описания поведения больших дискретных систем.

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.

Буч Г. и др. Объектно-ориентированный анализ…– М.:«И.Д. Вильямс», 2010.

Трудности управления проектированием

Мой Банк
СИСТЕМА ДИСТАНЦИОННОГО БАНКОВСКОГО ОБСЛУЖИВАНИЯ

- Главная страница
- Выписки
- Платежи
  - Оплатить услуги
  - Периодическая оплата
  - Одна кнопка
  - Архив операций
- Переводы между картами
- Конверсия по мультивкладам
- Мои карты
- Моя почта
- Сервис
- Полезные ссылки
- Выйти из системы

Предупреждение! При проведении всех типов платежей через систему "Мой Банк" следует избегать отправки одинаковых сумм в течение промежутка времени меньше одной минуты.

Внимание! Поля, помеченные звездочкой (*) обязательны для заполнения.

- Перечисление денег на получателя осуществляется на **СЛЕДУЮЩИЙ РАБОЧИЙ ДЕНЬ** после оплаты.

- Комиссия за перевод составляет минимум 500 белорусских рублей, 0.2% от суммы перевода. Максимальная сумма перевода составляет 9 999 999 белорусских рублей.

- Создать перевод на основе уже существующего можно из списка совершенных платежей.

- Назначение платежа обязательно должно содержать ФИО и реквизиты того, за что Вы платите! Назначение платежа не должно содержать:символ '//' и ',' '№', 'ё', 'Ё' текст с 204 по 236 и с 300 по 332 символ не должен содержать '/'. Символы '-', ':' не должны находиться в позициях: 1,36,71,106,141,171,204,237,267,300 (эти параметры контролируются программой).

Пример:Иванов Иван Иванович оплата за стоянку за 01.2007 по договору 14 от 01.01.2000 за место 5.

- Перечисление денег возможно на маски счетов:2421, 2422, 2427, 2475, 2476, 2477, 3011, 3012, 3013, 3014, 3015, 3104, 3134, 3135, 3603, 3604, 3632, 3642, 3812, 3819, 4731, 6399, 8115, 8510.

- В случае, если платеж совершается за счет овердрафтного кредита или с кредитной карты, дополнительно взимается комиссия и/или начисляются проценты

Буч Г. и др. Объектно-ориентированный анализ…– М.:«И.Д. Вильямс», 2010.

Необходимость обеспечения гибкости

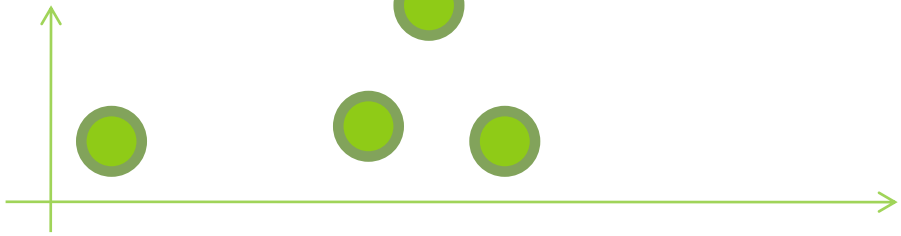
Строительная компания обычно не владеет лесхозом для создания пиломатериалов; не владеет металлургическим заводом для создания уникальных стальных балок.

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.

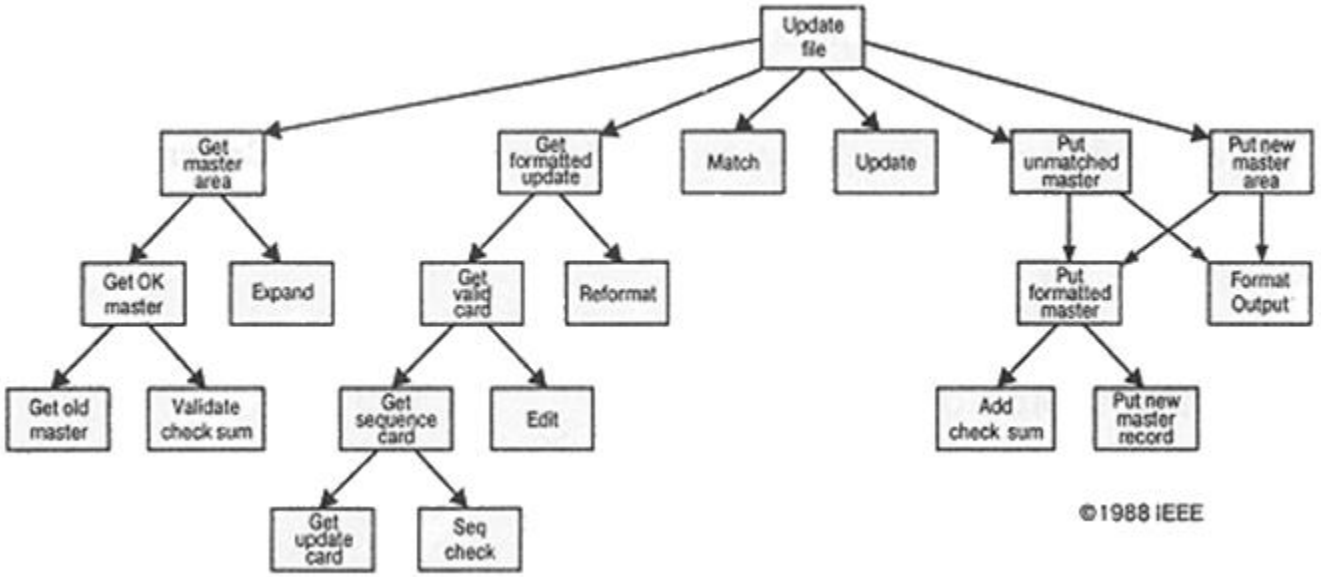
Декомпозиция - борьба со сложностью

Divide et Impera (разделяй и властвуй)

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

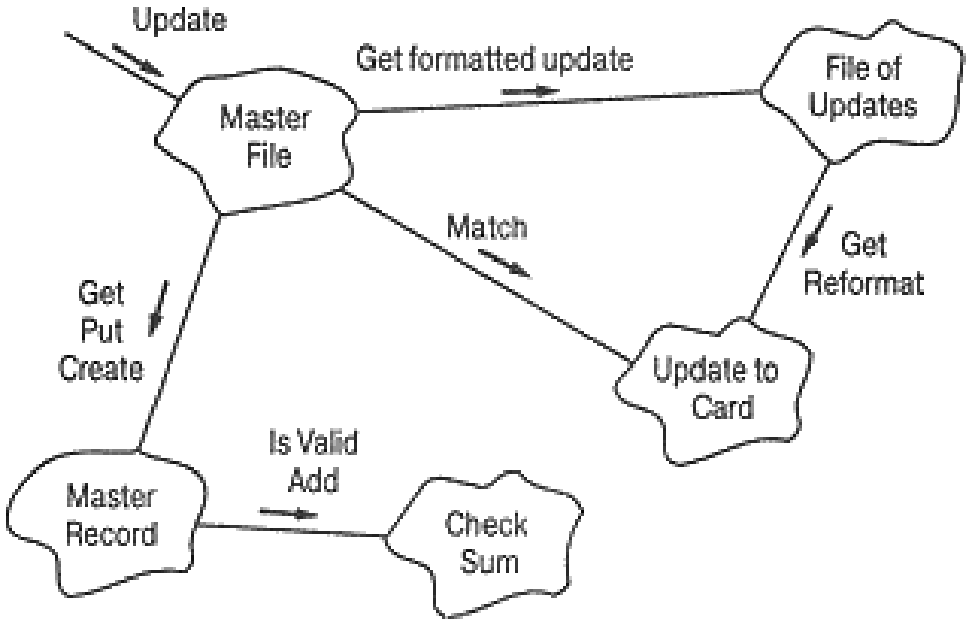
- алгоритмическая декомпозиция
- объектно-ориентированная декомпозиция

Алгоритмическая декомпозиция



Буч Г. и др. Объектно-ориентированный анализ…– М.:«И.Д. Вильямс», 2010.

Объектно-ориентированная декомпозиция



Буч Г. и др. Объектно-ориентированный анализ…– М.:«И.Д. Вильямс», 2010.

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

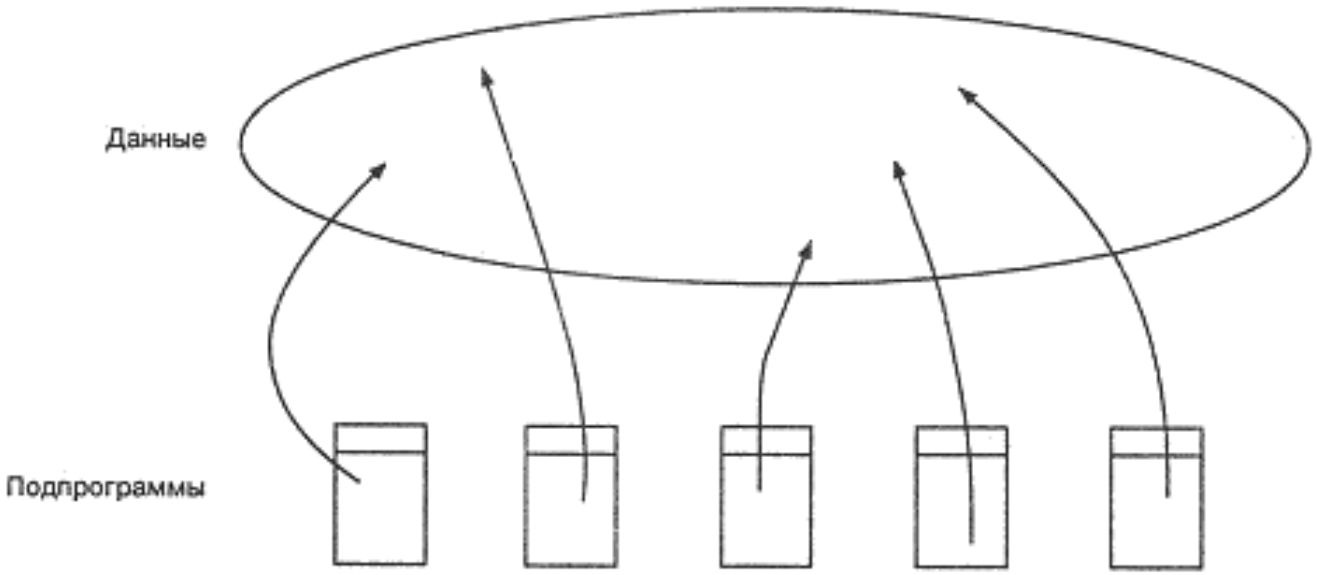
Поколения языков программирования (3)

- Языки третьего поколения (1962-1970)
  - PL/I — FORTRAN+ALGOL+COBOL
  - Pascal — Простой наследник ALGOL-60
  - Simula — Классы, абстракция данных
- Разрыв преемственности (1970-1980)
  - C — Эффективный высокоуровневый язык
  - FORTRAN 77 — Блочная структура, типы данных

Поколения языков программирования (4)

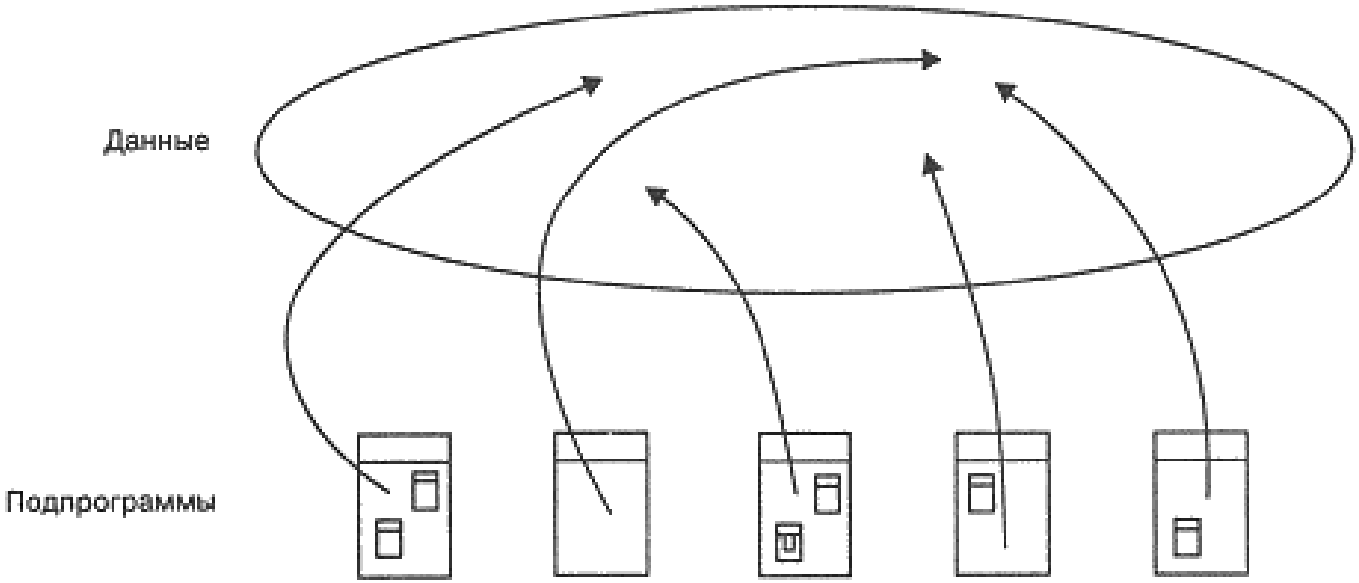
- Бум ООП (1980-1990)
  - Smalltalk 80 — Чисто объектно-ориентированный язык
  - C++ — C + Simula
  - Ada83 — Строгая типизация; сильное влияние Pascal
- Появление инфраструктур (1990-…)
  - Java — Блочная структура, типы данных
  - Python — Объектно-ориентированный язык сценариев
  - Visual C# — Конкурент языка Java для среды Microsoft .NET

Топология языков первого поколения



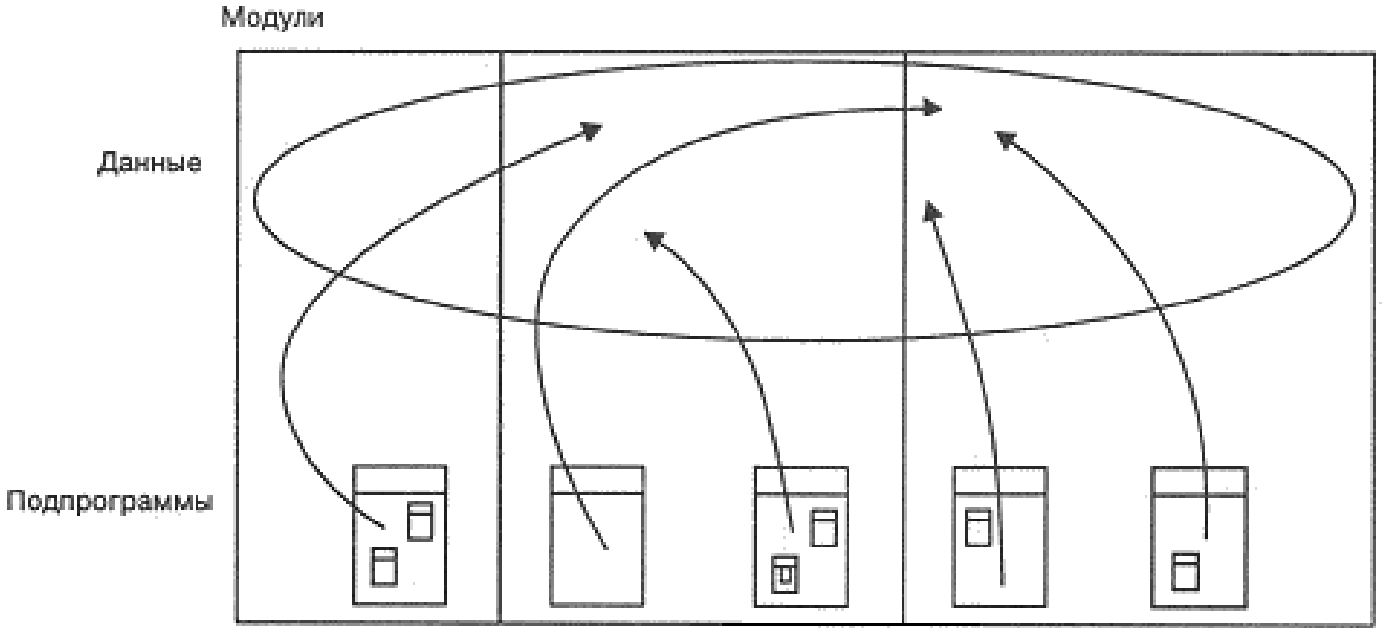
Программы имеют относительно простую структуру, состоящую только из глобальных данных и подпрограмм.

Топология языков второго поколения



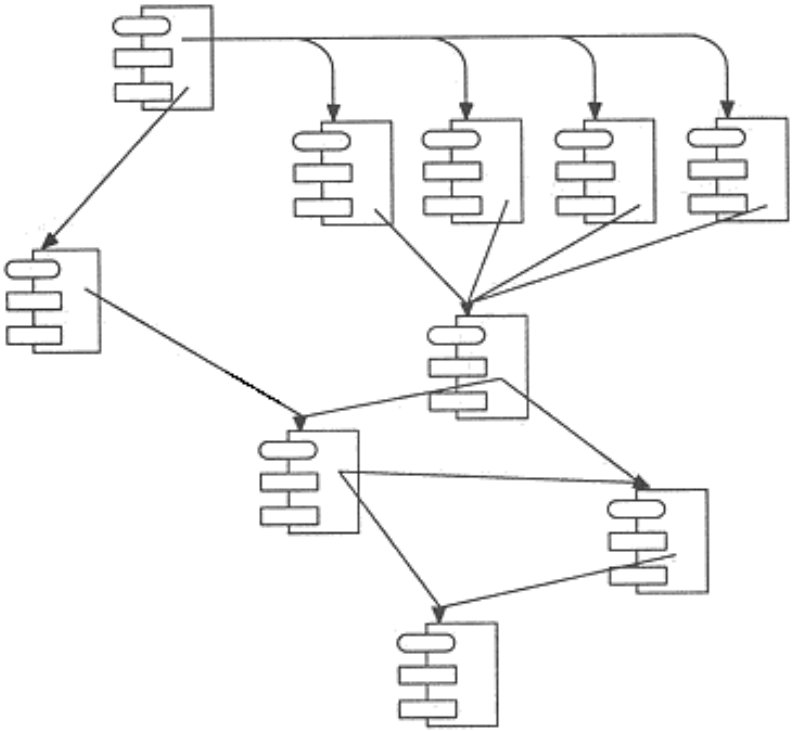
Появление процедурной абстракции, которая позволила описать абстрактные программные функции в виде подпрограмм.

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.

Итого

- 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).