

На протяжении всех лет существования практика программирования требовала совершенствования технологических приемов и создания на их основе таких средств программирования, которые упростили бы процесс разработки программ, позволяя создавать все более сложные программные системы.

Первые программы были организованы очень просто. Они состояли из собственно программы на машинном языке и обрабатываемых данных. Создание сначала ассемблеров, а затем и языков высокого уровня сделало программу более обзримой за счет снижения уровня детализации. Появление в языках средств, позволяющих оперировать подпрограммами, существенно снизило трудоемкость разработки программ.



Архитектура: программа с глобальной областью данных

Подпрограммы можно было сохранять и использовать в других программах. Были накоплены огромные библиотеки расчетных и служебных подпрограмм, которые по мере надобности вызывались из разрабатываемой программы.



Архитектура: программа с локальными данными подпрограмм

Структурное программирование представляет собой совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения.

Основные принципы:

- принцип нисходящей разработки, рекомендующий на всех этапах вначале определять наиболее общие моменты, а затем поэтапно выполнять детализацию;
- структурное программирование, рекомендующее определенные структуры алгоритмов и стиль программирования;
- принцип сквозного структурного контроля для проведения контроля всех этапов разработки.

В основе структурного программирования лежит *декомпозиция* (разбиение на части) сложных систем с целью последующей реализации в виде отдельных небольших подпрограмм. С появлением других принципов декомпозиции получил название *процедурной декомпозиции*.

Метод пошаговой детализации:

определяется структура программы в виде одного из трех вариантов:

- последовательности подзадач
- альтернативы подзадач
- повторение подзадач

процесс продолжается, пока на очередном уровне не получается подзадача, которая достаточно просто реализуется средствами используемого языка.



Основные структуры процедурной декомпозиции

Процедурная декомпозиция («Записная книжка»)

Основная программа:

Начать работу

Вывести меню на экран.

Ввести команду.

Выполнить цикл обработки вводимых команд.

Завершить работу

Операция: « Выполнить цикл обработки »

Выполнить цикл обработки вводимых команд:

цикл-пока команда «завершить работу»

Выполнить команду

Ввести команду

все-цикл

Детализация операции: «Выполнить команду»

Выполнить команду:

если команда= «открыть книжку»

то Открыть книжку

иначе если команда= «добавить»

то Добавить запись

иначе если команда= «найти»

то Найти запись

все-если

все-если

все-если

Основная программа

Начать работу.

Вывести меню на экран.

Ввести команду,

цикл-пока команда «завершить работу»

если команда= «открыть книжку»

то Открыть книжку

иначе если команда= «добавить»

то Добавить запись

иначе если команда= «найти»

то Найти запись

все-если

все-если

все-если

Ввести команду

все-цикл

Завершить работу.

На первом уровне выделены подзадачи:

«Вывести меню», «Ввести команду», «Открыть книжку»,
«Добавить запись» и «Найти запись».

Дальше определяются подзадачи задач второго уровня:

Открыть_книжку:

Ввести имя файла

если существует файл Имя_книжки

то Открыть файл

иначе Вывести сообщение об ошибке

все-если

Получаем подзадачи: «Ввести имя файла» и «Открыть файл».

Схема двухуровневой процедурной декомпозиции



Сформулированная методика декомпозиции закрепила сложившийся в то время процедурный (алгоритмический подход) к программированию, при котором основное внимание концентрируется на определении последовательности действий.

Дальнейший рост сложности и размеров разрабатываемого программного обеспечения потребовал развития структурирования данных (в языках появляется возможность определения пользовательских типов данных). Результат - появление и развитие технологии модульного программирования.

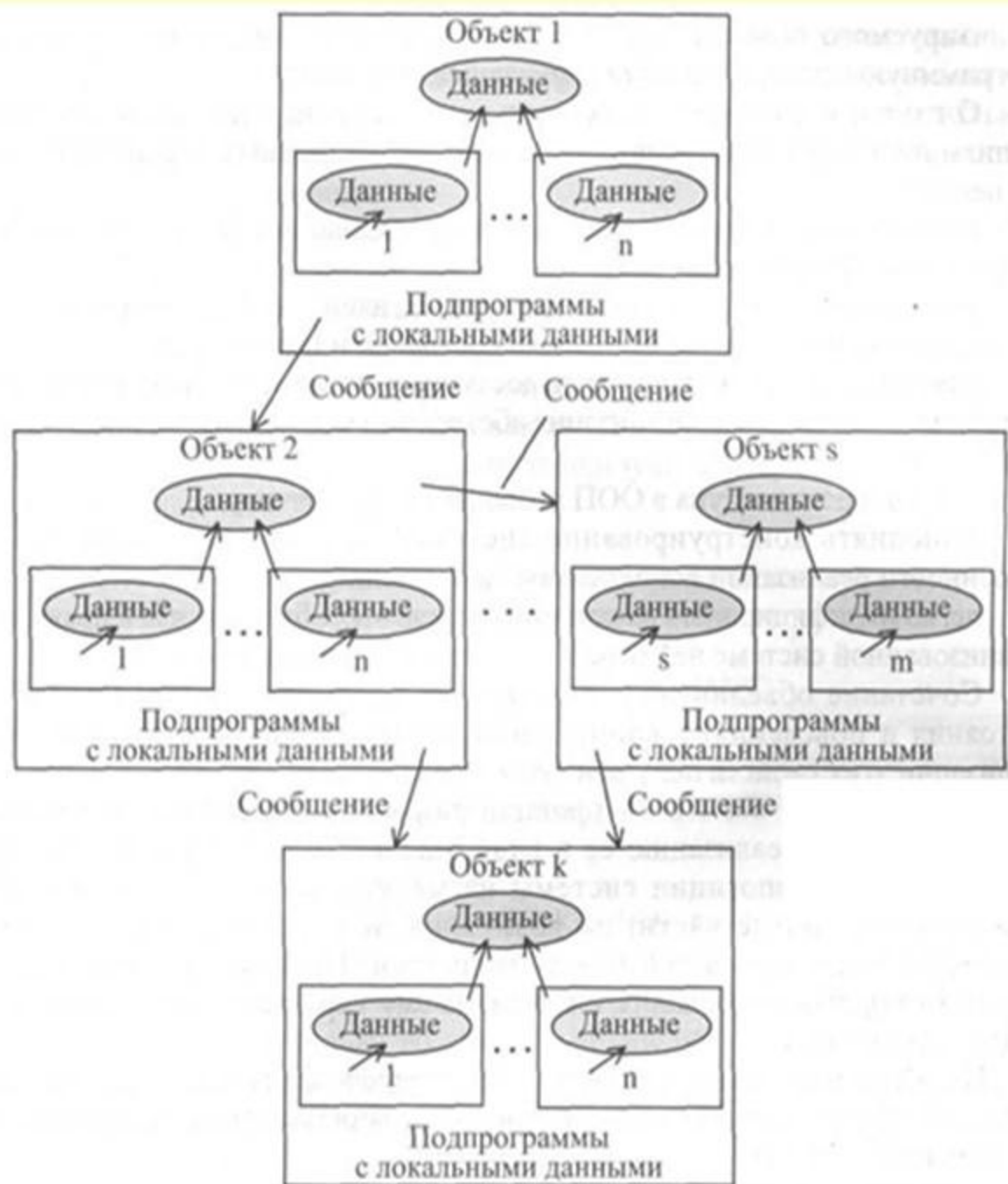
Модульное программирование предполагает выделение подпрограмм, использующих одни и те же глобальные данные, в отдельно компилируемые модули (библиотеки подпрограмм),. Связи между модулями осуществляются через специальный интерфейс, в то время как доступ к реализации модуля запрещен.



Архитектура программы, состоящей из модулей

ООП - технология создания сложного программного обеспечения, которая основана на представлении программы в виде совокупности *объектов*, каждый из которых является экземпляром определенного типа (*класса*), а классы образуют иерархию с *наследованием свойств*. Взаимодействие программных объектов в такой системе осуществляется путем передачи *сообщений*.

Принципы ООП: абстрагирование, ограничение доступа, модульность, иерархичность, типизация, параллелизм, устойчивость.



Архитектура программы при ООП

Объектная декомпозиция («Записная книжка»)

Диаграмма состояний интерфейса пользователя



Полная диаграмма состояний интерфейса пользователя



Диаграмма объектов системы «Записная книжка»



Объектная декомпозиция программы, которая по запросу пользователя рисует одну из двух фигур: квадрат или круг

Диаграмма объектов графического редактора

