

ДРАКОН Схема

Это стандартизированная блок-схема для описания бизнес-процессов, которая подчиняется строгим правилам, чтобы сделать ее понятной и эргономичной

Программирование на гибридном языке происходит следующим образом:

- Рисуем ДРАКОН-схему.
 - Внутри икон помещаем небольшие кусочки кода на соответствующем языке программирования.
 - Программа-транслятор преобразует ДРАКОН-схему в текстовый файл с исходным кодом.
 - Этот текстовый файл включается в проект обычным образом.
- Генерацию кода из диаграмм на сегодняшний день поддерживают несколько редакторов.

Ключевые моменты

Главный принцип — удобство восприятия человеком. Схемы строятся так, чтобы минимизировать количество пересечений линий и изгибов.

Используются иконки (пиконы) различной формы для разных операций (действие, вопрос, цикл, начало/конец и т.д.).

Алгоритм представляется в виде дерева, что делает его логику стройной и избавляет от хаотичного "лабиринта" из стрелок.

Высокая наглядность и простота восприятия.

Подходит для описания процессов в различных областях: от программирования до медицины и управления.

Диаграмма деятельности UML

UML (от англ. Unified Modeling Language) переводится как «унифицированный язык моделирования». Это графический язык, в котором каждой фигуре, символу, стрелке или их сочетаниям присвоены конкретные значения. Он позволяет визуализировать явление или процесс так, чтобы схема была понятна всем, кто знаком с UML.

Язык UML нужен, чтобы описать и визуализировать какую-то абстрактную модель. На практике это может быть:

- Создание модели объекта. Например, описание структуры базы данных.
- Создание модели процессов. Например, последовательность выполнения запросов ПО, чтобы клиент получил ожидаемый результат.

Основные элементы диаграммы деятельности UML

Начальный узел: Символизирует старт процесса или действия.

Конечный узел: Обозначает завершение процесса.

Действия: Это прямоугольники с закругленными углами, описывающие конкретные задачи или шаги в рамках процесса.

Потоки управления: Стрелки, которые соединяют действия и узлы, показывая порядок их выполнения.

Узлы решений: Ромбовидные фигуры, используемые для отображения условий, которые определяют, какой из потоков управления будет выбран далее.

Узлы объединения: Также ромбовидные фигуры, которые применяются для слияния нескольких потоков управления в один, после того как различные ветви процесса завершены.

Диаграммы потоков данных (DFD)

DFD диаграммы в отличии от других нотаций позволяют визуально показать все процессы с точки зрения данных. Это может быть полезно:

- при разработке информационной системы;
- при интеграции системы;
- при миграции данных и функционала с одной системы на другую;
- в проектах, связанных с Data Management;
- в процессе построения аналитического хранилища, BI-решения.

Диаграмма позволяет визуализировать как движение данных между объектами системы, так и преобразования данных, которые могут применяться на разных шагах процесса.

Идеальна для анализа и проектирования информационных систем на ранних этапах.

Позволяет ясно увидеть, какие данные куда поступают и где хранятся.

Основные элементы диаграммы потоков данных (DFD)

- Процесс.

Процессы, при которых идет изменение потока данных (обработка, трансформация и др. изменения). Процесс как и в других диаграммах обычно прописывается с помощью глагола, например: “Отправка заполненной формы”.

- Внешняя сущность.

Сущность (объект), которая получает или отправляет данные при взаимодействии с описанным процессом.

- Хранилище данных.

Все хранилища данных или отдельные файлы, которые хранят исходные или выходные данные, а также все промежуточные хранилища.

- Поток данных.

Поток данных, который отображает направление и сами данные, которые перемещаются между внешними сущностями и хранилищами данных с помощью процессов.

Записи алгоритма с помощью Р-схемы

Р-схема – это разновидность блочной схемы, которая использует символы или пиктограммы (графические изображения) для представления отдельных команд или этапов алгоритма. Такие схемы наглядны, поскольку каждый символ обозначает определенное действие, а стрелки показывают последовательность выполнения команд, что делает процесс более понятным для анализа и отладки алгоритма.

Основные элементы Р-схемы

Процесс: Прямоугольник; вычислительное действие или последовательность операций.

Решение: Ромб; проверка условия. Имеет один вход и два (или более) выхода.

Модификация: Шестиугольник; начало цикла (инициализация счетчика, условие продолжения).

Предопределенный процесс: Прямоугольник с двойными вертикальными линиями; вызов подпрограммы или другого алгоритма.

Ввод/Вывод: Параллелограмм; операция ввода или вывода данных.

Терминатор: Скругленный прямоугольник; начало или конец алгоритма.

Преимущества:

- Четкий, международный стандарт.
- Интуитивно понятна для представления последовательности шагов и ветвлений.
- Универсальна и широко известна.

Диаграммы Нэсси-Шнейдермана

Диаграмма Нэсси-Шнейдермана — это графический метод представления структурированных алгоритмов и программ, разработанный в 1972 году американскими аспирантами Айзеком Нэсси и Беном Шнейдерманом. Она использует вложенные прямоугольники для отображения структурных элементов программирования, таких как последовательность, условие, цикл, и других блоков управления, что позволяет наглядно представить логику программы.

Преимущества:

- Наглядно демонстрирует структуру программы, ее вложенность и scope (область видимости).
- Является прямой "графической проекцией" структурного кода. Легко преобразуется в код.
- Широко используется в образовании для обучения основам структурного программирования. Является основой для визуальных языков типа Scratch, Blockly, App Inventor.

Основные элементы диаграммы Нэсси-Шнейдермана

Следование: Действия располагаются друг под другом в одном столбце.

Ветвление (IF-THEN-ELSE): Блок делится на несколько веток (например, левая — THEN, правая — ELSE).

Цикл (WHILE, DO-WHILE, FOR): Тело цикла представляется как вложенный прямоугольник внутри заголовка цикла, где указано условие.