

Принципы проектирования

Проект — комплекс сценариев, файлов, источников данных и прочих элементов, предназначенных для решения отдельной аналитической задачи.

Проект может объединять в себе несколько пакетов благодаря тому, что каждый пакет имеет возможность предоставлять свои объекты другим пакетам через механизм ссылок.

Структурный подход

В основе построения проекта лежит методология *структурного проектирования* — представление алгоритма в виде иерархической структуры блоков.

Каждый блок на своем уровне иерархии может быть представлен в виде «черного ящика», выполняющего независимую подзадачу. Механизм решения подзадачи внутри «черного ящика» можно изменить, но в целом проект при этом останется работоспособным и будет выполнять поставленные задачи.

Спроектированный таким образом проект имеет четкую, легко читаемую структуру. Все это позволяет создавать и сопровождать сложные проекты, а также делегировать решение выделенных подзадач.

Особенностью подобного подхода является проектирование «сверху вниз» — от общей постановки задачи к отдельным подзадачам. На первом этапе проектирования описывают решение поставленной задачи, выделяя независимые подзадачи. На следующем аналогично описывают подзадачи, формулируя при этом элементы следующего уровня.

Таким образом, на каждом шаге происходит уточнение функций проекта. Процесс продолжают, пока не доходят до подзадач, алгоритмы, решения которых очевидны.

Механизм повторного использования алгоритмов

Выделение блоков, решающих независимые подзадачи, дает возможность применять их повторно в смежных задачах. Развитием идеи повторного использования ранее созданных алгоритмов является создание библиотек универсальных функций.

В Studio инструментом для создания таких функций являются *производные компоненты*. Создав один раз компонент «АВС-анализа», его можно применять как в задаче сегментации товаров, так и в сегментации клиентской базы. Подзадача, решаемая компонентом проверки адресов, может применяться как в очистке данных, так и в задачах скоринга.

В производных компонентах реализуется *механизм наследования*, преимуществом которого является то, что он позволяет модифицировать созданный пользователем универсальный компонент лишь в одном месте — в библиотеке функций, и эти изменения автоматически произойдут во всех сценариях, где он применяется.

Однако не всегда удастся встроить универсальный компонент в конкретный сценарий без внесения в него изменений. При этом изменить его в библиотеке функций нет возможности, поскольку сработает механизм наследования, и это может привести к ошибкам во всех сценариях, где он применяется.

Данную задачу решает *механизм переопределения*, позволяющий модифицировать производный компонент только в использующем его сценарии, при этом в библиотеке функций он останется неизменным.

Декомпозиция

Структура Проекта может быть представлена в иерархическом виде:

- Проект может состоять из связанных между собой Пакетов — это возможно благодаря тому, что *каждый пакет может предоставлять свои объекты другим пакетам* через механизм ссылок.
- Пакет включает в себя Модули — декомпозиция пакета на уровне модулей.
- **Модуль** — сам по себе не содержит узлов обработки данных, но предоставляет отдельное пространство для Сценариев и Подключений к различным источникам данных.
- **Сценарий** — содержит последовательность узлов обработки данных. Сценарий может:
 - Включать в себя подпрограммы — Подмодель.
 - Получать данные от узлов из других сценариев и пакетов через механизм Узел-ссылка.
 - Использовать настройки и обученные модели узлов из других сценариев и пакетов через механизм Выполнение узла.
 - Использовать готовые алгоритмы обработки данных, созданные в других сценариях и пакетах через механизм Производные компоненты.
- **Подмодель** — включает в себя другие узлы, предоставляя, таким образом, отдельное пространство для реализации произвольного алгоритма обработки данных. Подмодель в сценарии представлена в качестве узла, имеющего заданные пользователем входные и выходные порты. Может содержать в себе иерархию вложенных подмоделей. На базе подмодели может быть создан Производный компонент.