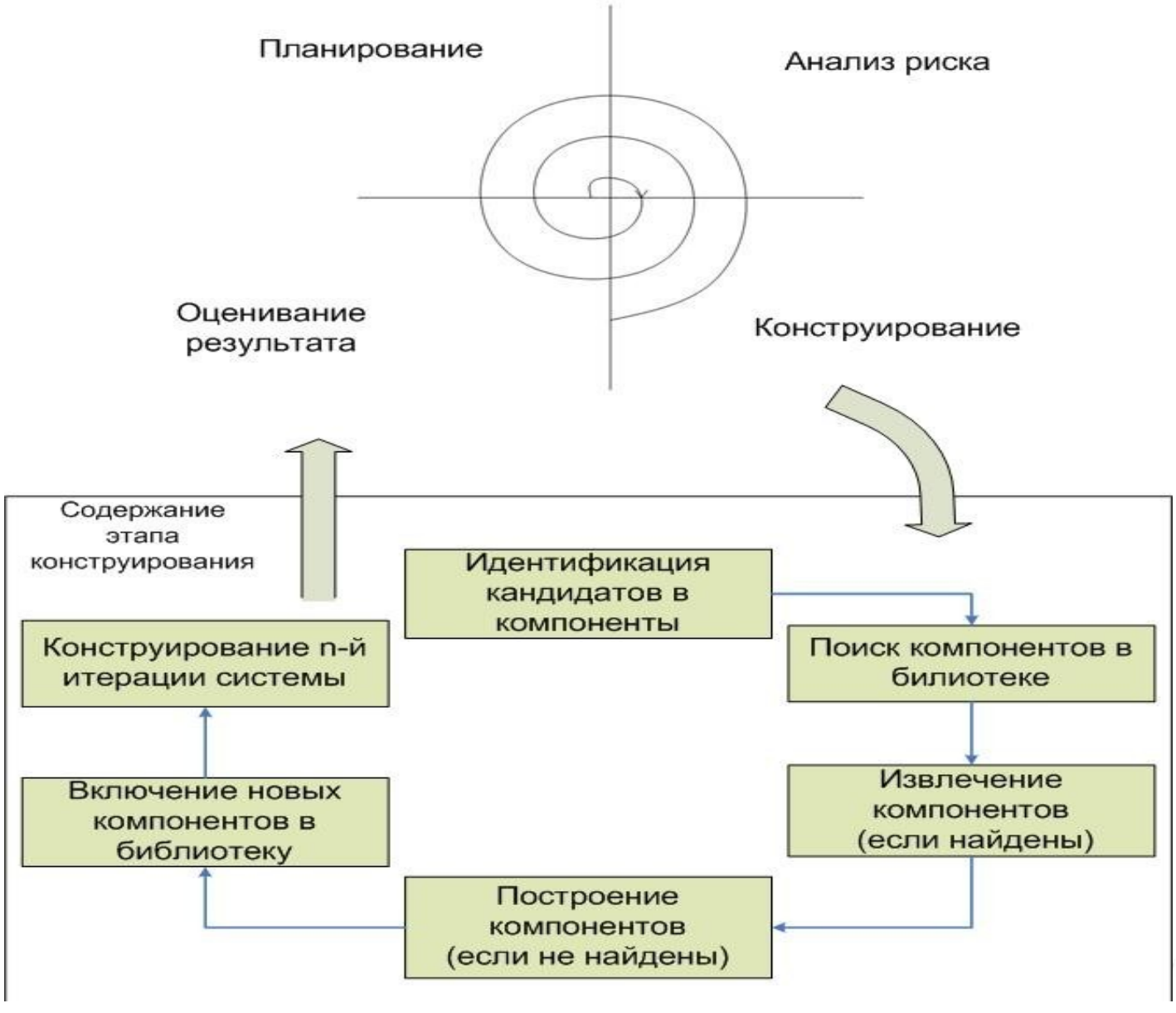
КПО

Компонентный подход
Модульность
Сложность системы

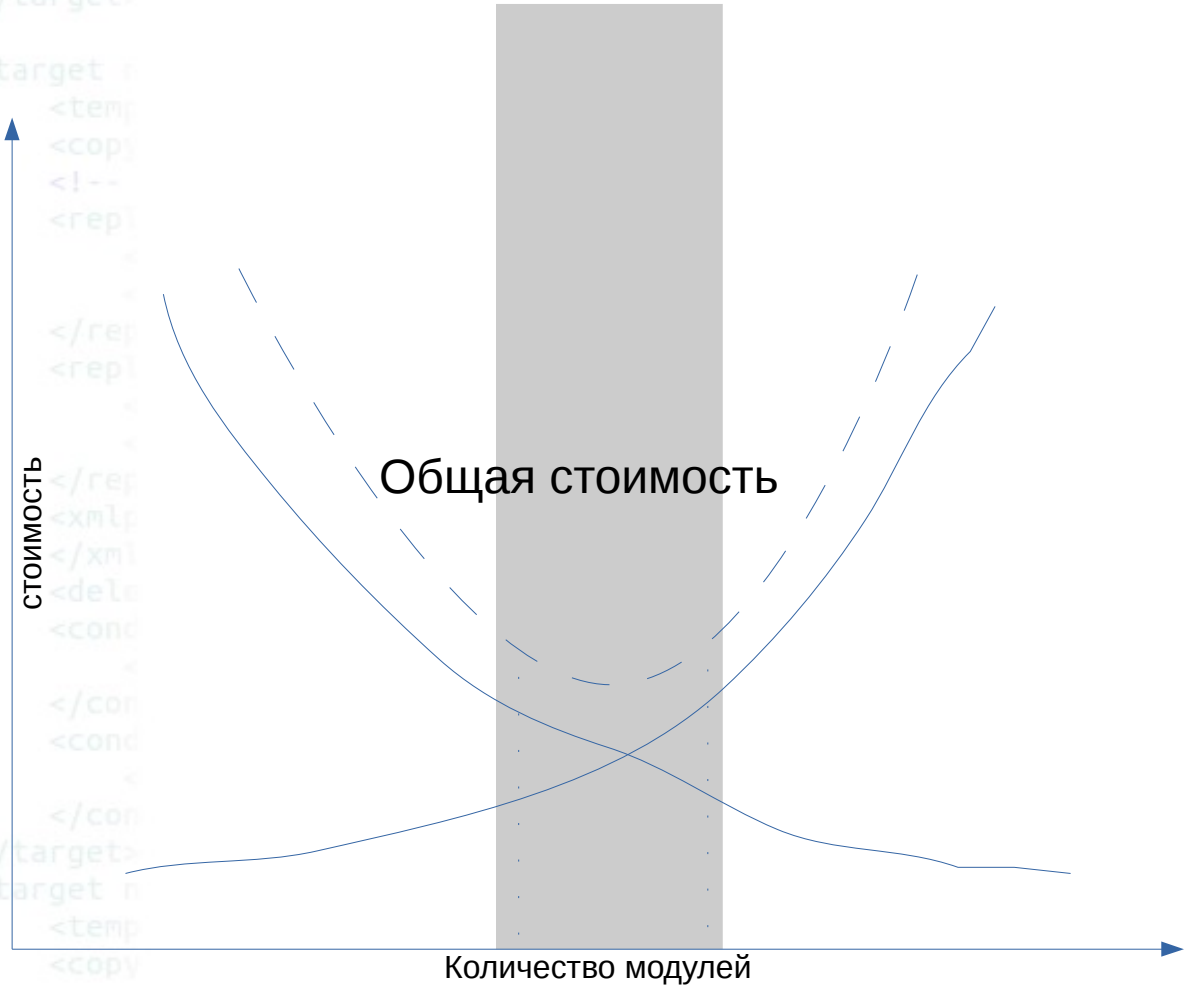
Лекция №7

(версия 1.0)

компонентно-ориентированная модель



Модульность



Оптимальный модуль

Снаружи проще, чем внутри

Его проще использовать, чем построить

Информационная закрытость

Принцип информационной закрытости:
Содержание(процедуры, данные) модулей должно быть скрыто друг от друга.

Это означает следующее:
все модули независимы, обмениваются только той информацией, которая необходима для работы доступ к операциям и структурам модуля ограничен.

Достоинства: возможность работы разных команд, легкая модификация системы. Идеальный модуль – «черный ящик».

Связность модуля

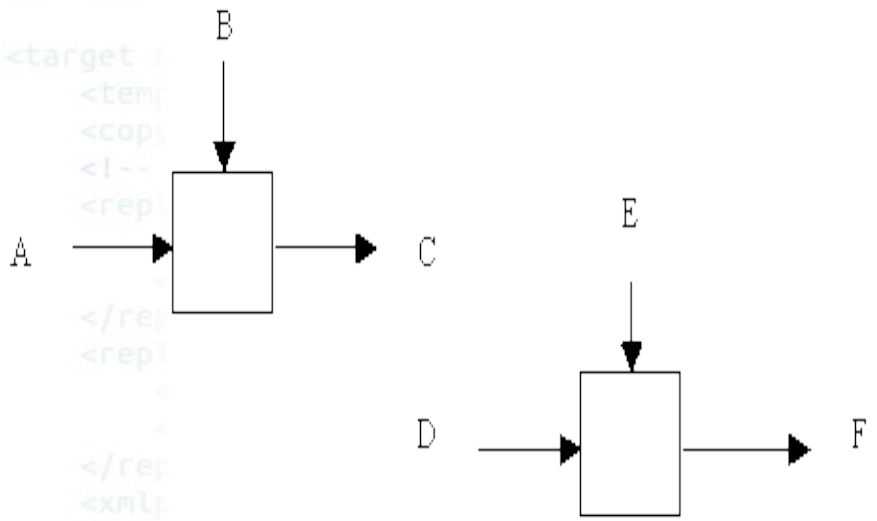
Связность модуля (cohesion) – внутренняя характеристика модуля, характеризующая меру прочности соединения функциональных и информационных объектов внутри одного модуля. При проектировании модулей нужно стремиться к высокой связности, ибо чем выше связность, тем лучше спроектирован модуль.

Связность модуля

Существует 7 типов связности:

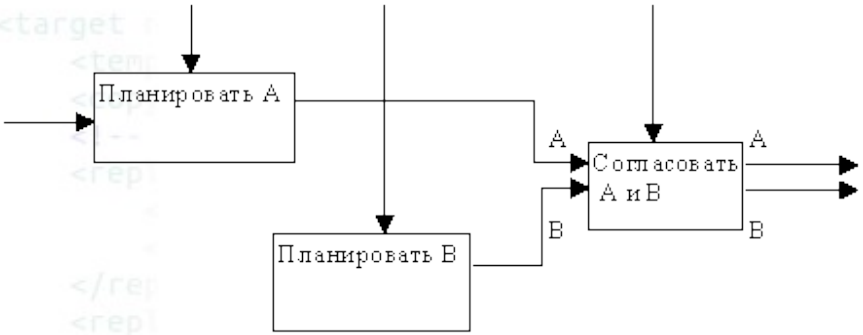
1. Связность по совпадению (CC=0)
2. Логическая связность (CC=1)
3. Временная связность (CC=3)
4. Процедурная связность (CC=5)
5. Коммуникативная (Информационная) связность (CC=7)
6. Последовательная связность (CC=9)
7. Функциональная связность (CC=10)

Связность по совпадению



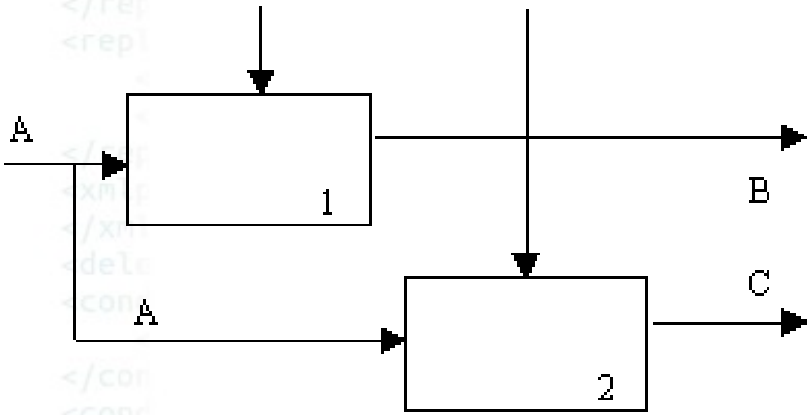
Случайная связность возникает, когда конкретная связь между функциями мала или полностью отсутствует.

Процедурная связность



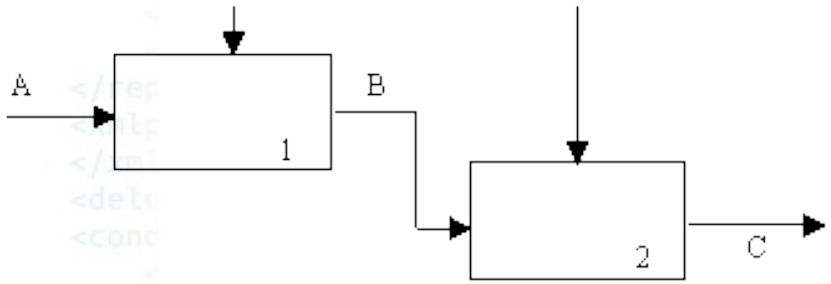
Процедурно-связанные элементы появляются сгруппированными вместе вследствие того, что они выполняются в течение одной и той же части цикла или процесса.

Коммуникативная (Информационная) связность



Диаграммы демонстрируют коммуникационные связи, когда блоки группируются вследствие того, что они используют одни и те же входные данные и/или производят одни и те же выходные данные

Последовательная связность



На диаграммах, имеющих последовательные связи, выход одной функции служит входными данными для следующей функции. Связь между элементами на диаграмме является более тесной, чем на рассмотренных выше уровнях связок, поскольку моделируются причинно-следственные зависимости

Функциональная связность

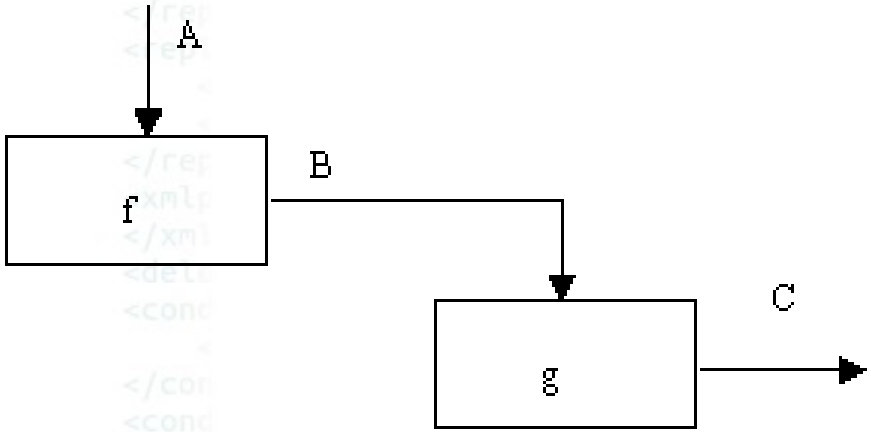


Диаграмма отражает полную функциональную связность, при наличии полной зависимости одной функции от другой. Диаграмма, которая является чисто функциональной, не содержит чужеродных элементов, относящихся к последовательному или более слабому типу связности.

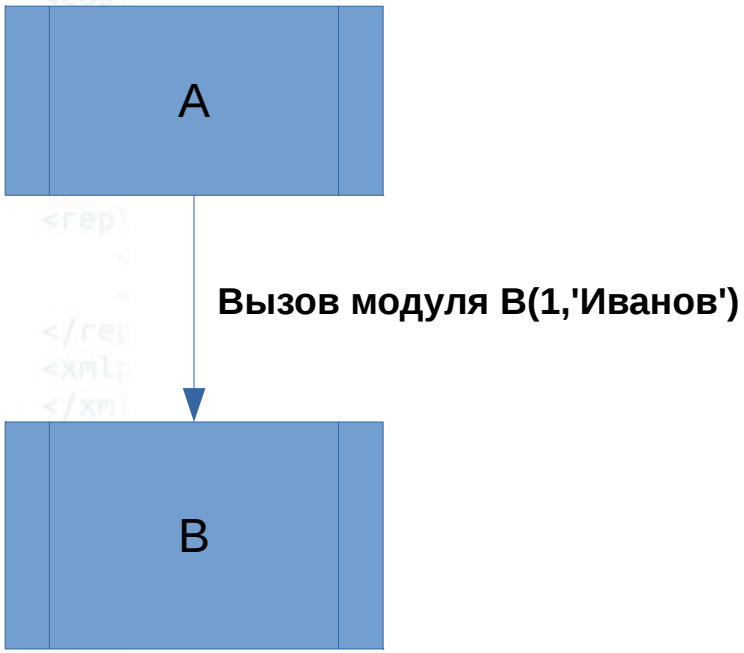
Связность модуля

Мера связности	Сцепление	Модифицируемость	Понятность	Сопровождаемость
Функциональная	хорошее	хорошая	хорошая	хорошая
Последовательная	хорошее	хорошая	близкая к хорошей	хорошая
Информационная	среднее	средняя	средняя	средняя
Процедурная	приемлемое	приемлемая	приемлемая	плохая
Временная	плохое	плохая	средняя	плохая
Логическая	плохое	плохая	плохая	плохая
Случайная	плохое	плохая	плохая	плохая

Сцепление модулей

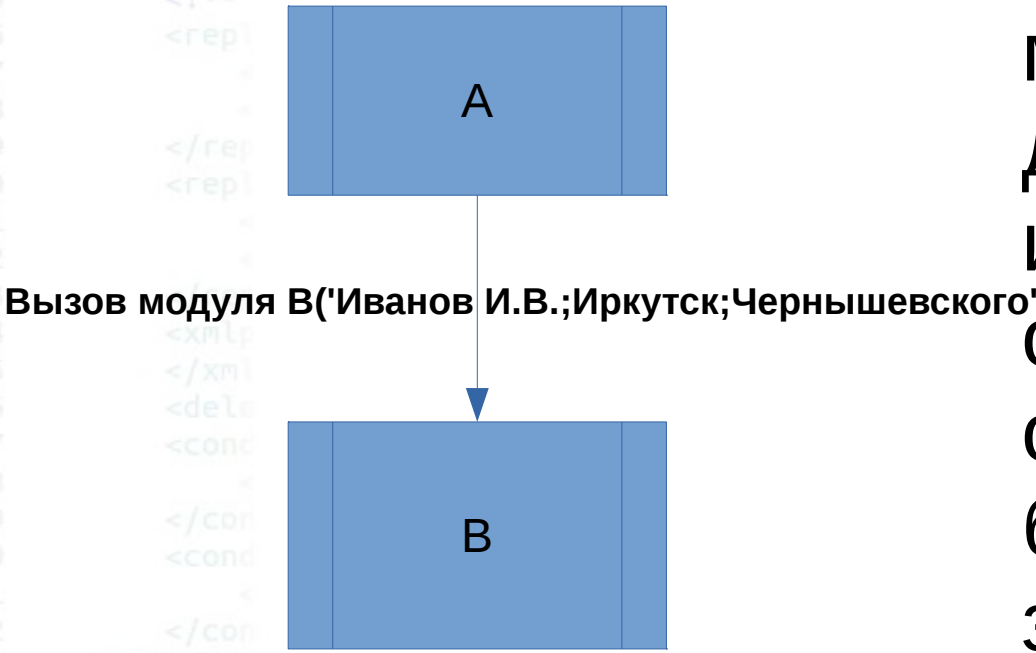
Связанность (coupling) модуля является мерой взаимозависимости модулей. При создании систем необходимо стремиться к максимальной независимости модулей, т.е. связанность модулей должна быть минимальной.

Сцепление модулей



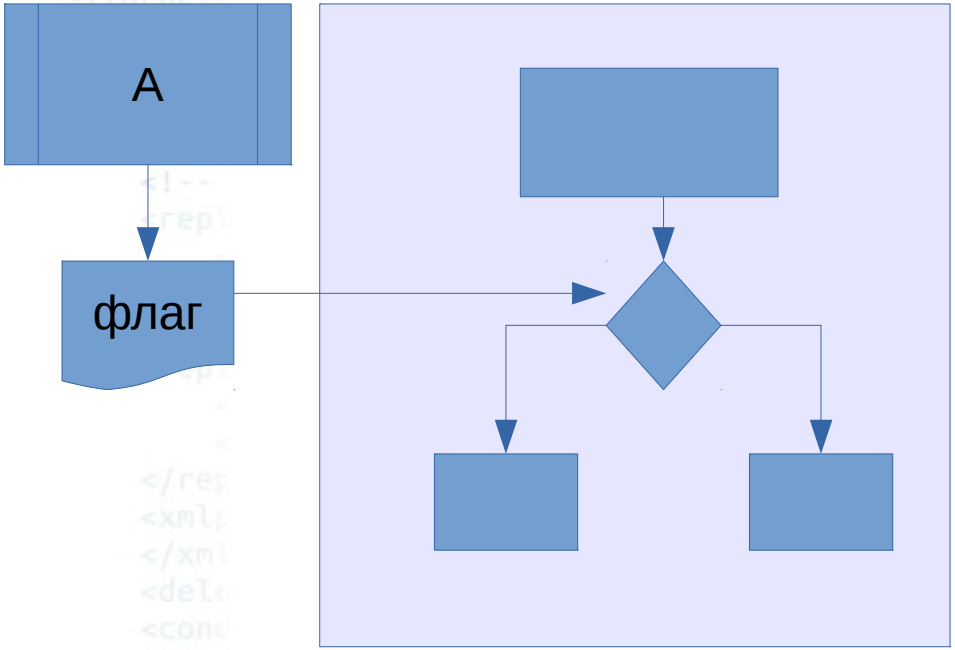
Модули связаны по данным, если они взаимодействуют через передачу параметров и при этом каждый параметр является элементарным информационным объектом. Это наиболее предпочтительный тип связанности (сцепления).

Сцепление модулей



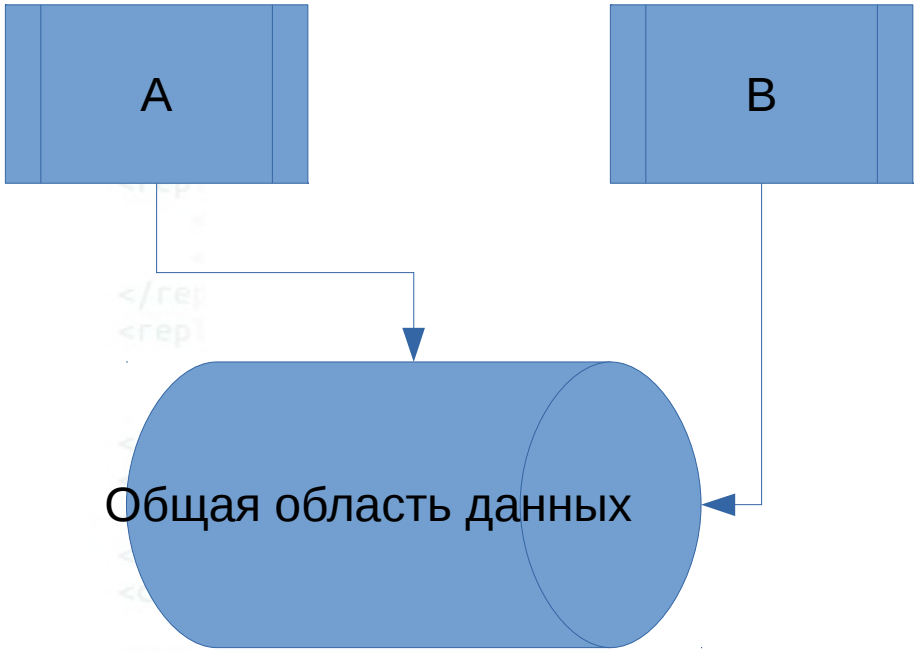
Модули связаны по образцу если один модуль посылает другому составной информационный объект (например, объект – библиографическая запись, которая содержит имя автора, название книги и т.д.).

Сцепление модулей



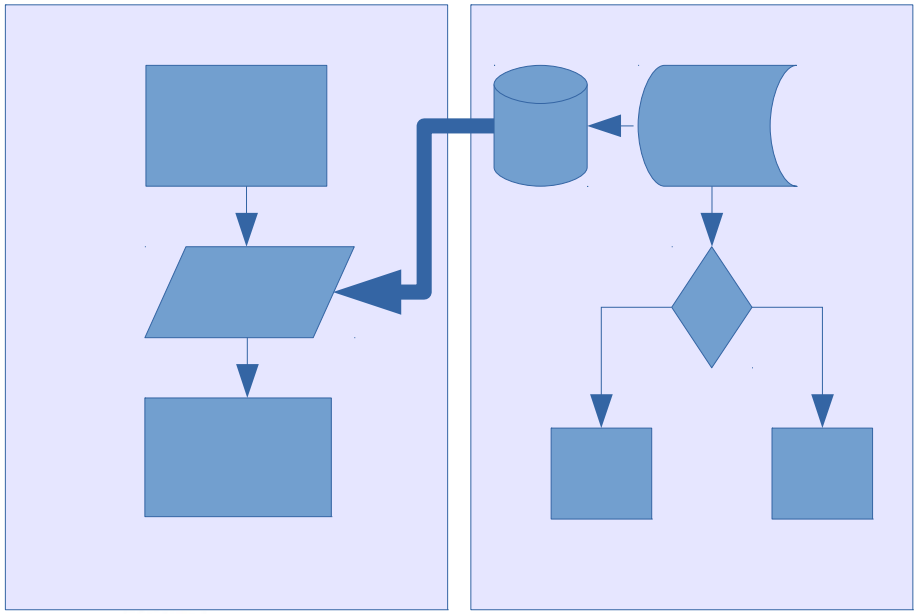
Модули связаны по управлению, если один посылает другому информационный объект – флаг, предназначенный для управления его внутренней логикой.

Сцепление модулей



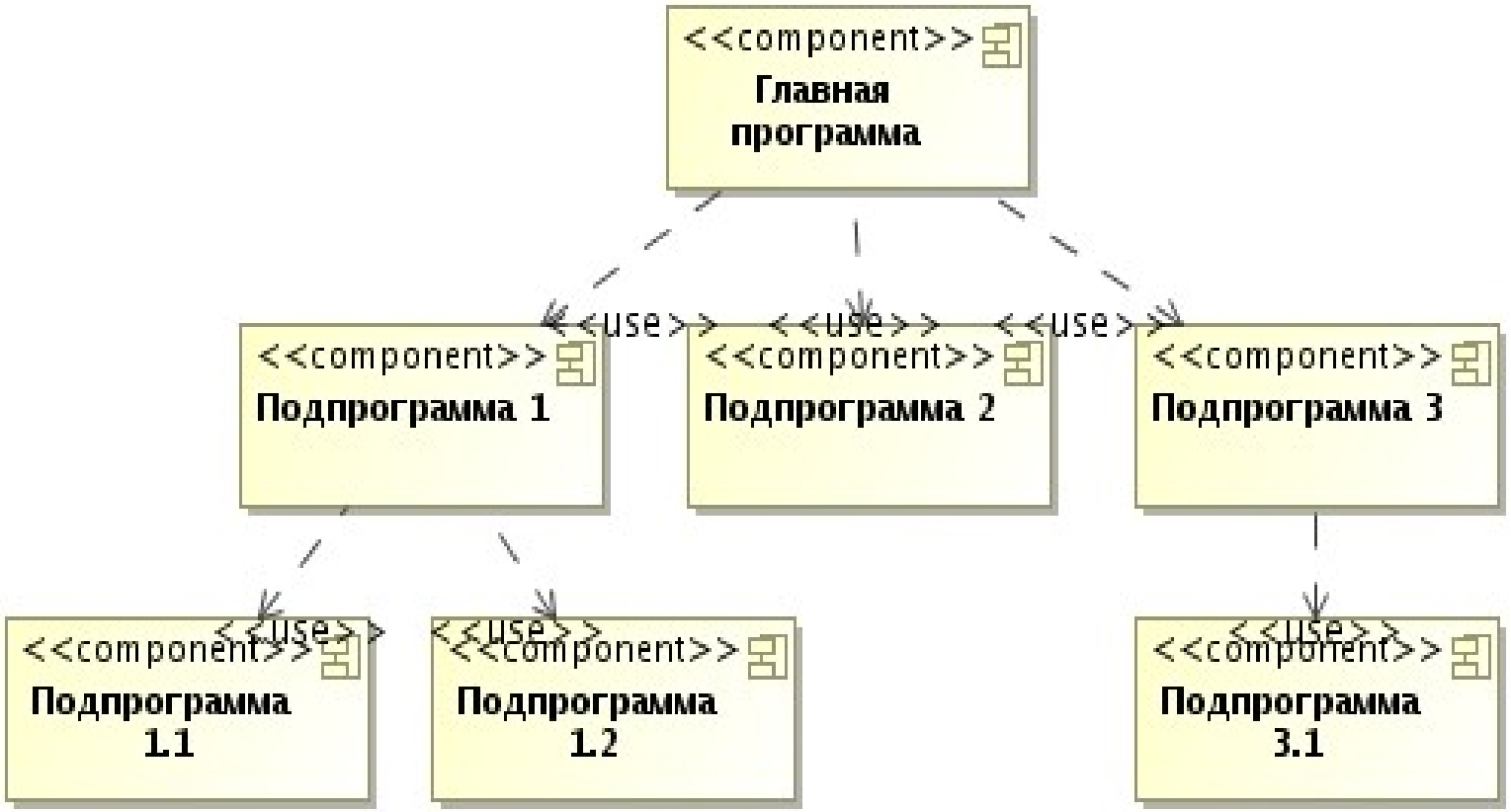
Модули связаны по общей области в том случае, если они ссылаются на одну область глобальных данных.

Сцепление модулей



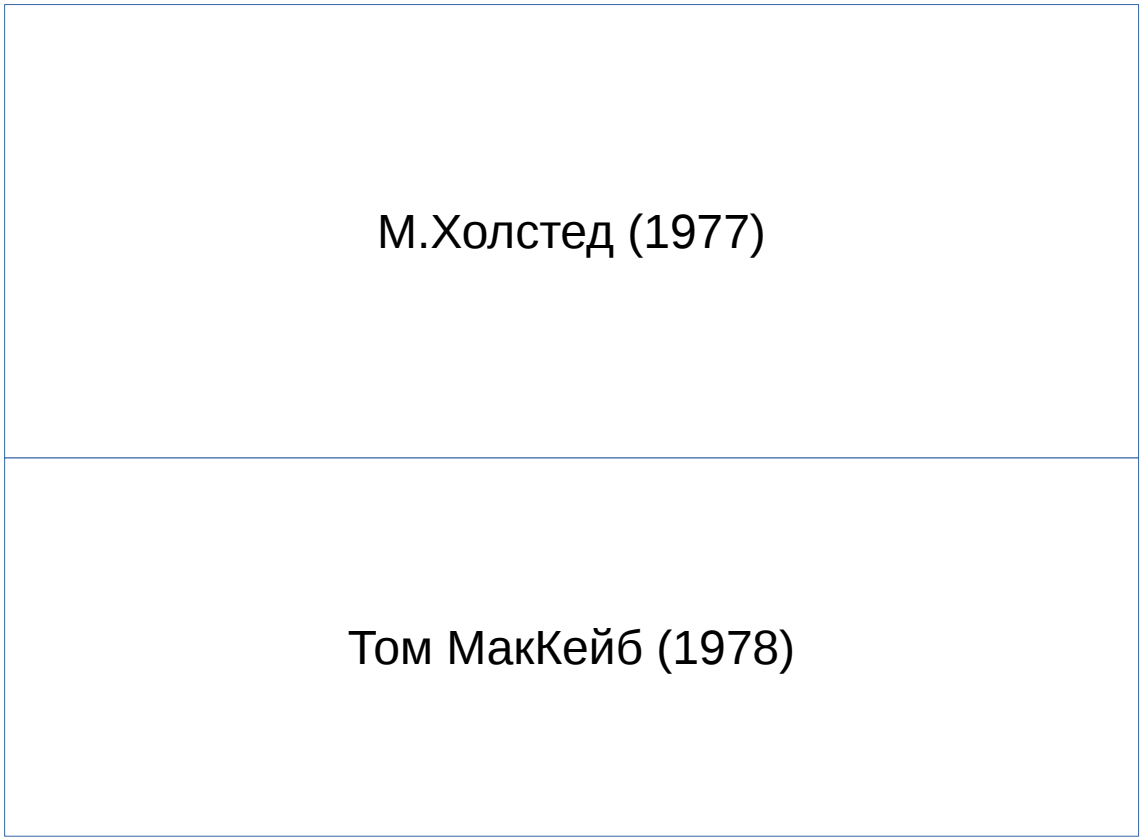
Модули связаны по содержимому в том случае, если один из них ссылается внутрь другого. Это недопустимый тип сцепления, ибо полностью противоречит принципу модульности, т.е. представления модуля в виде черного ящика.

Характеристика иерарх.структуры



Ширина $S = \sum_{i=1}^{n} length(i)\left(Fan_{in}(i) + Fun_{out}(i)\right)^2$

Сложность программной системы



Тестирование базового пути

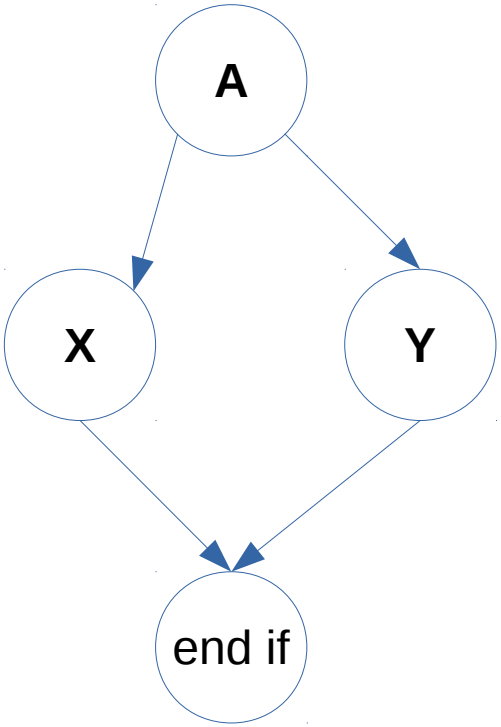
Том МакКейб, 1976

Потоковый граф

1) Отображение управляющей структуры;
2) Узлы (вершины) = линейным участка;
3) Дуги = поток управления;
4) Операторные и предикатные узлы;
5) Предикатный узел = простое условие;
6) Регионы;
7) Окружающая среда.

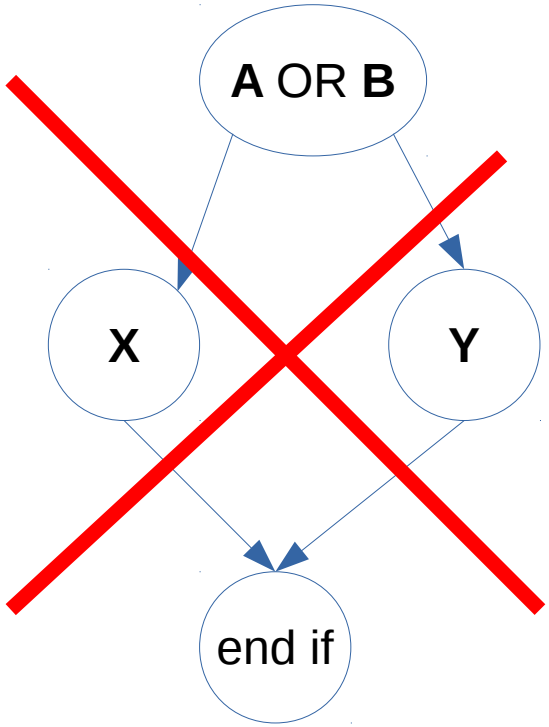
Тестирование базового пути

If a
then x
else y
end if



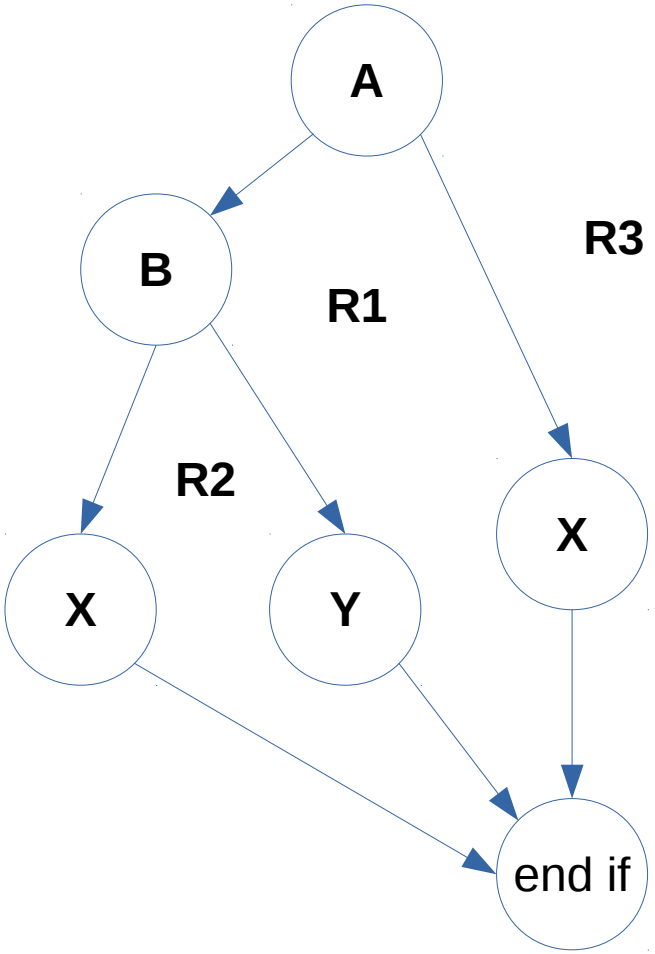
Тестирование базового пути

If a OR b
then x
else y
end if



Тестирование базового пути

If a OR b
then x
else y
end if



Цикломатическая сложность

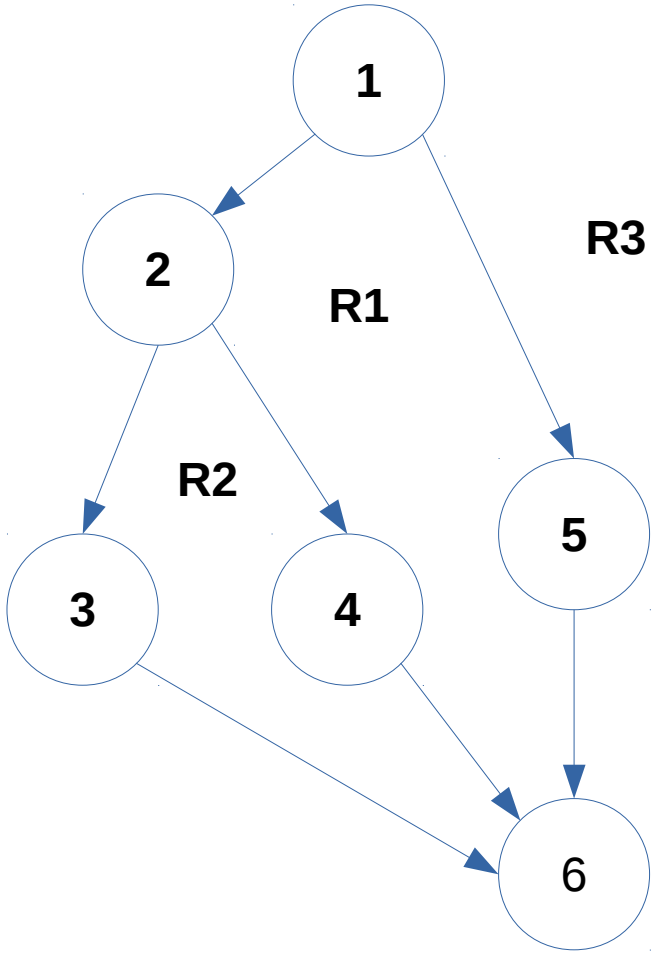
Цикломатическая сложность определяет:

- Количество независимых путей в базовом множестве алгоритма;
- Верхнюю оценку количества тестов, которая гарантирует однократное выполнение всех операторов.

Все независимые пути образуют базовое множество.

Цикломатическая сложность

1) 1-2-3-6
2) 1-2-4-6
3) 1-5-6



Цикломатическая сложность

Способы расчета:
1. Количество регионов потокового графа;
2. V(G)=E-N+2, где E — кол-во дуг, N — кол-во узлов;
3. V(G)=p+1, где p — кол-во предикатных узлов.