

The background of the image is a spiral-bound notebook with a light beige, textured cover. The metal spiral binding is visible along the left edge. The title is written in a bold, red, serif font, centered on the page.

ОСНОВНЫЕ ПОНЯТИЯ СТРУКТУРНОГО ПРОГРАММИРОВАНИЯ

**Структурное
(систематическое)
программирование -
грамотное, рациональное
программирование.**

Принципы структурного программирования

- Преимущественное использование базовых алгоритмических структур: следование, развилка, цикл.
- Метод нисходящего проектирования алгоритмов.
- Использование обозначений, соответствующих содержанию задачи и облегчающих понимание программы.

Принципы структурного программирования

Преимущественное использование базовых алгоритмических структур $\Rightarrow$ ограниченное использование операторов безусловного перехода – `goto`, `break`, `continue`, `exit` и др.

Преимущественное использование базовых алгоритмических структур $\Rightarrow$ уменьшение разнообразия алгоритмов, их стандартизация $\Rightarrow$ улучшение наглядности алгоритмов, понятности алгоритмов $\Rightarrow$ уменьшение количества ошибок при разработке алгоритмов.

Структурное программирование основано на теореме Дейкстры.

Принципы структурного программирования

Метод нисходящего проектирования алгоритмов:

- ◆ разделение задачи на связанные между собой подзадачи;
- ◆ алгоритм каждой подзадачи имеет один вход и один выход;
- ◆ разбиение на подзадачи отражает логику программы;
- ◆ алгоритм, устанавливающий связи между подзадачами, строится с использованием базовых алгоритмических структур;
- ◆ подзадачи автономны - кодируются как подпрограммы, преимущественно использующие локальные переменные.

Принципы структурного программирования

Использование обозначений, соответствующих содержанию задачи и облегчающих понимание программы

Все имена сначала описываются (объявляются), а потом используются

Методы структурирования алгоритмов

Структурирование: приведение неструктурных алгоритмов к структурным.

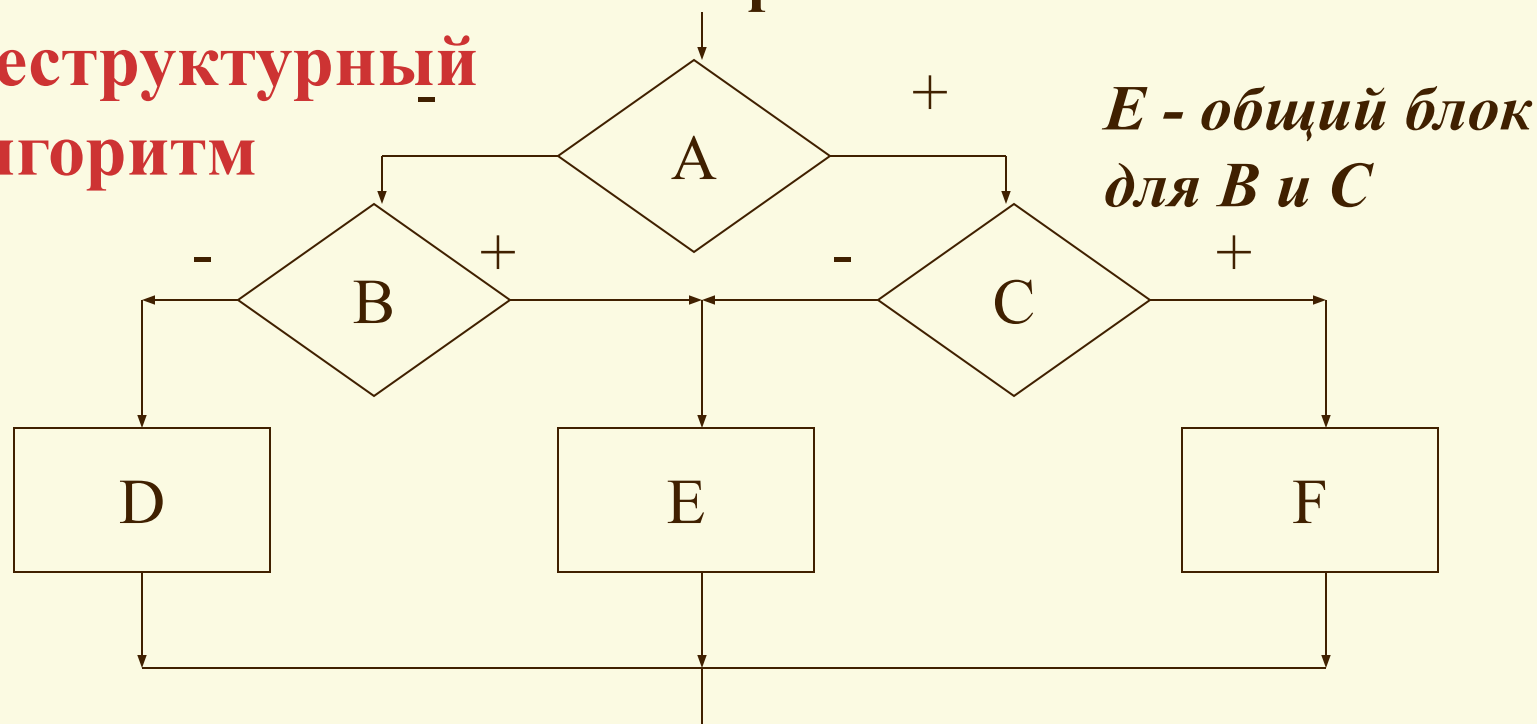
Цель изучения методов структурирования: различение неструктурных алгоритмов, использование при программировании только структурных алгоритмов.

Неверный подход: написание неструктурных алгоритмов и последующее приведение их у структурным.

Методы структурирования развилки

Типичная неструктурность: наличие общих
блоков в ветвях «+» и «-» развилки:

**Неструктурный
алгоритм**

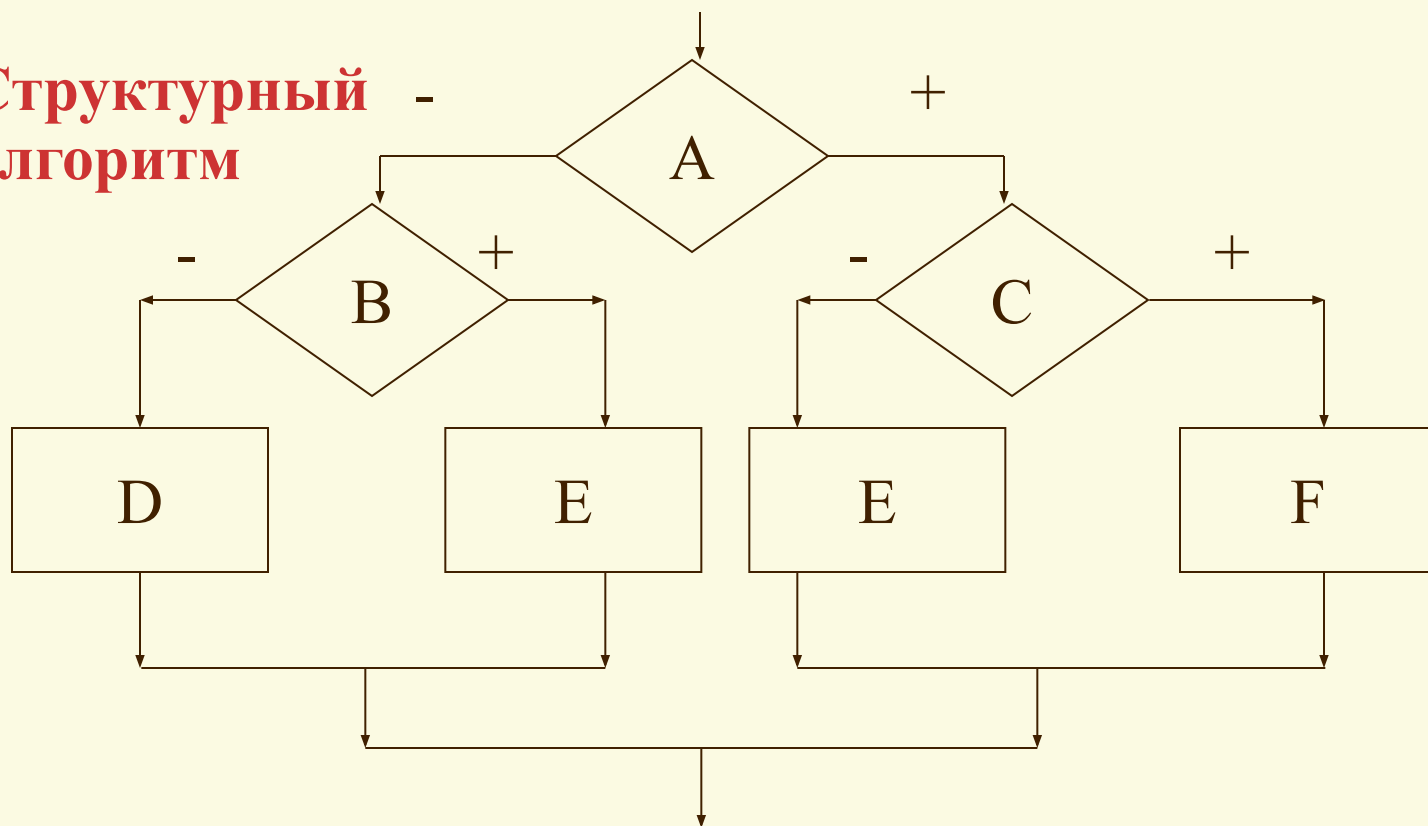


*У структурной развилки должна четко определяться
автономная точка соединения ветвей*

Методы структурирования развилки

1. Метод дублирования блоков (блока E)

Структурный
алгоритм



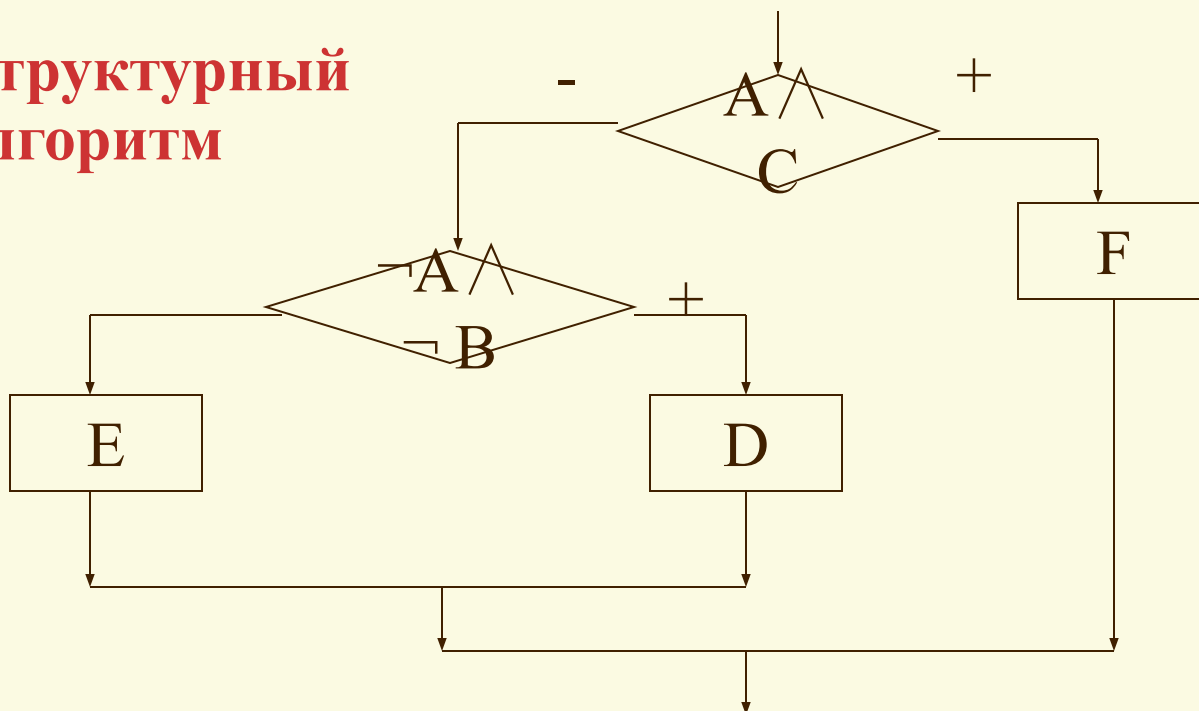
Методы структурирования алгоритмов

ВНИМАНИЕ! Переход от неструктурного алгоритма к структурному по формальным правилам приводит к «утяжелению» алгоритма $\Rightarrow$ необходимо изначально проектировать структурные алгоритмы.

Методы структурирования развилки

2. Метод объединения условий

Структурный
алгоритм



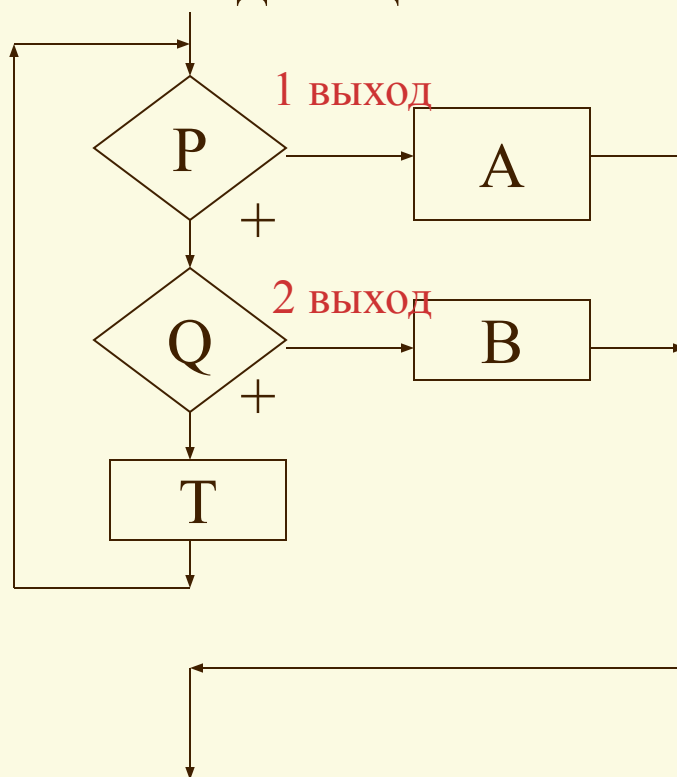
Методы структурирования цикла

Если цикл является структурным, то:

- ◆ Цикл имеет один блок анализа на выход из (продолжение) цикла.
- ◆ Блок анализа на выход из (продолжение) цикла стоит либо в начале (цикл ПОКА), либо в конце (цикл ДО) цикла.
- ◆ Ветвь «обратной связи» не содержит операторов.

Методы структурирования цикла

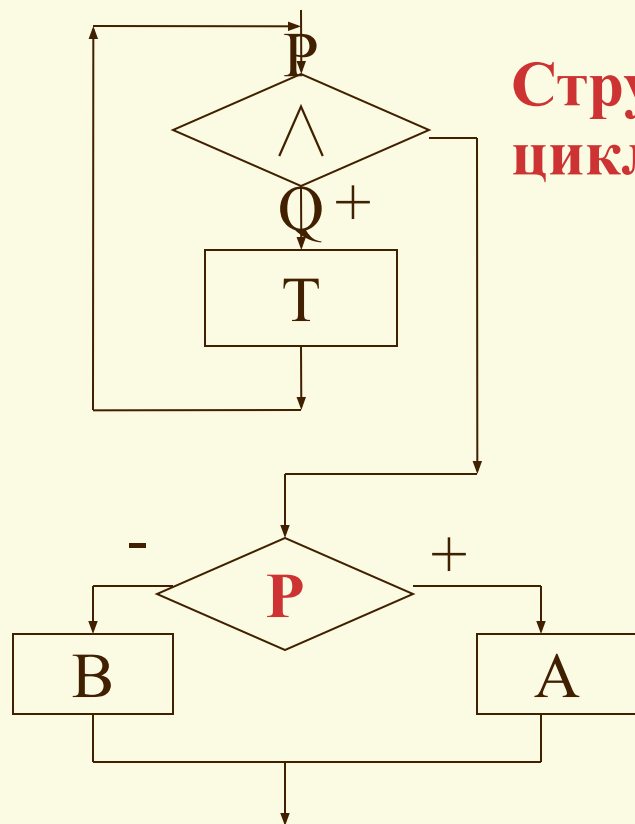
Типичная неструктурность: более одного блока анализа на выход из цикла.



**Два выхода из
цикла ⇒
неструктурный
алгоритм**

Методы структурирования цикла

1. Метод объединения условий

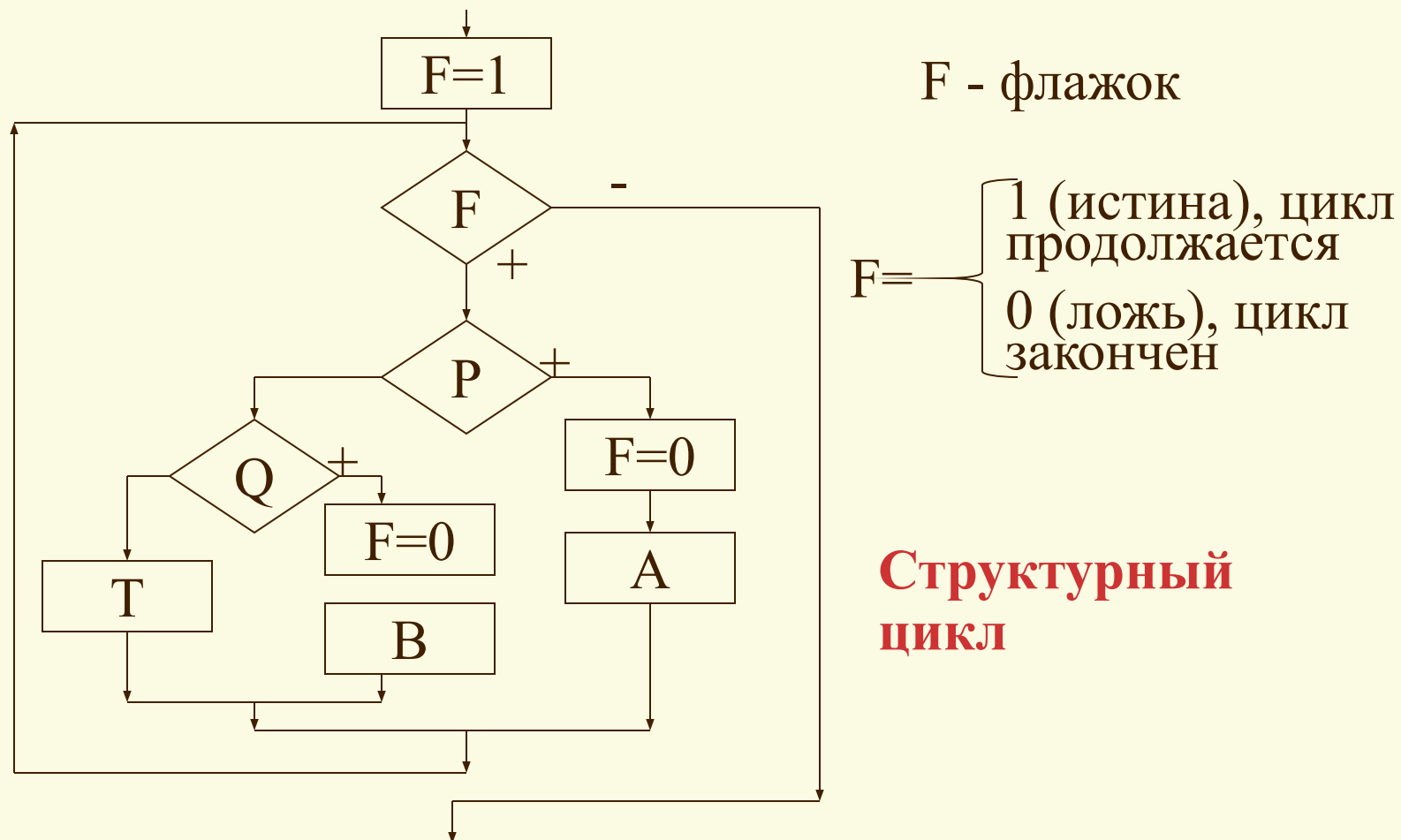


Структурный
цикл

P - приоритетное условие: при одновременном выполнении P и Q должен проработать оператор A, а не B

Методы структурирования цикла

2. Метод флажка (булева признака)



Методы структурирования цикла

**Преимущество метода объединения условий:
простота.**

**Преимущество метода флажка:
универсальность для любого числа условий
выхода из цикла.**