Параллельные процессы. Альтернативы

Параллельное программирование

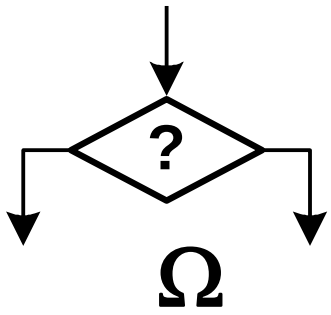
РГАТУ
2019
Грызлова Т. П.

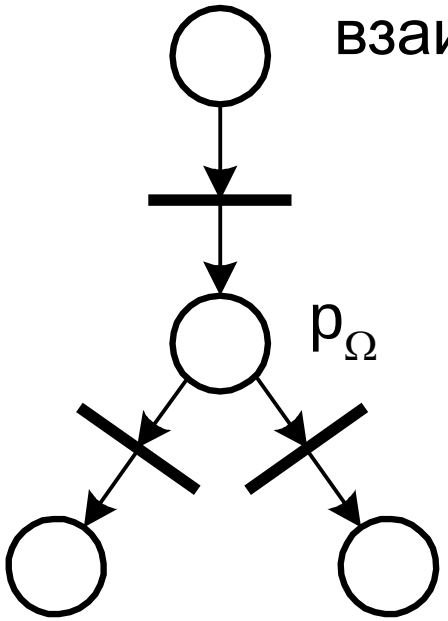
Управляющие вершины ПГС

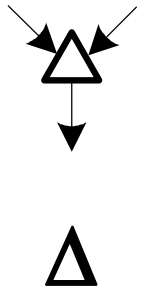
Условные

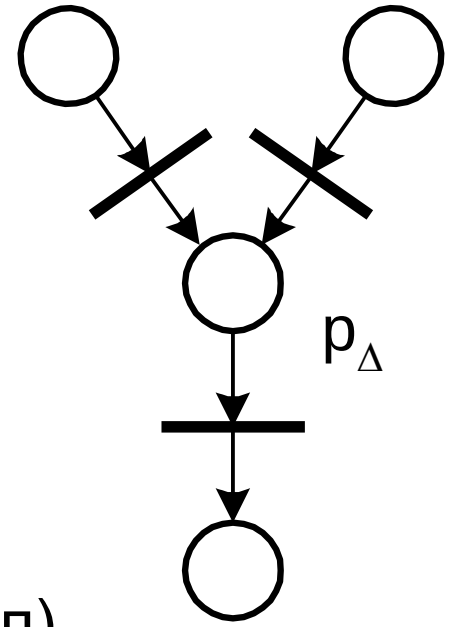
Вход в цикл.

Соединение взаимоисключающих ветвей





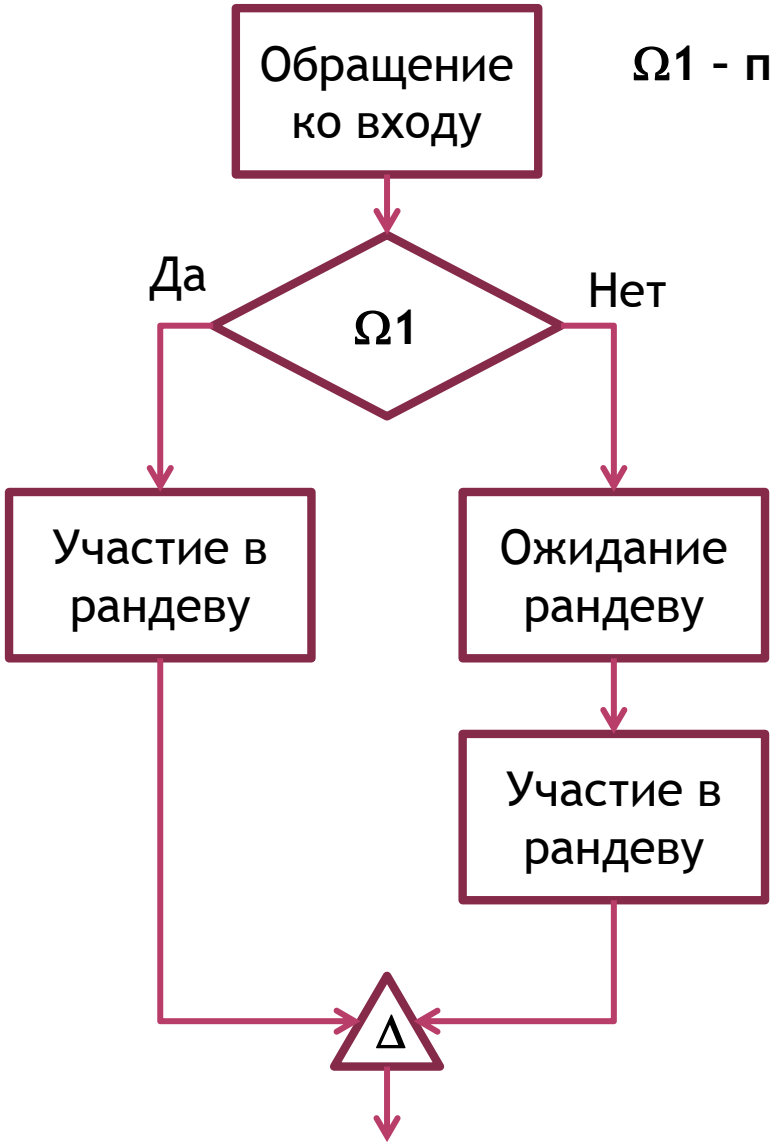




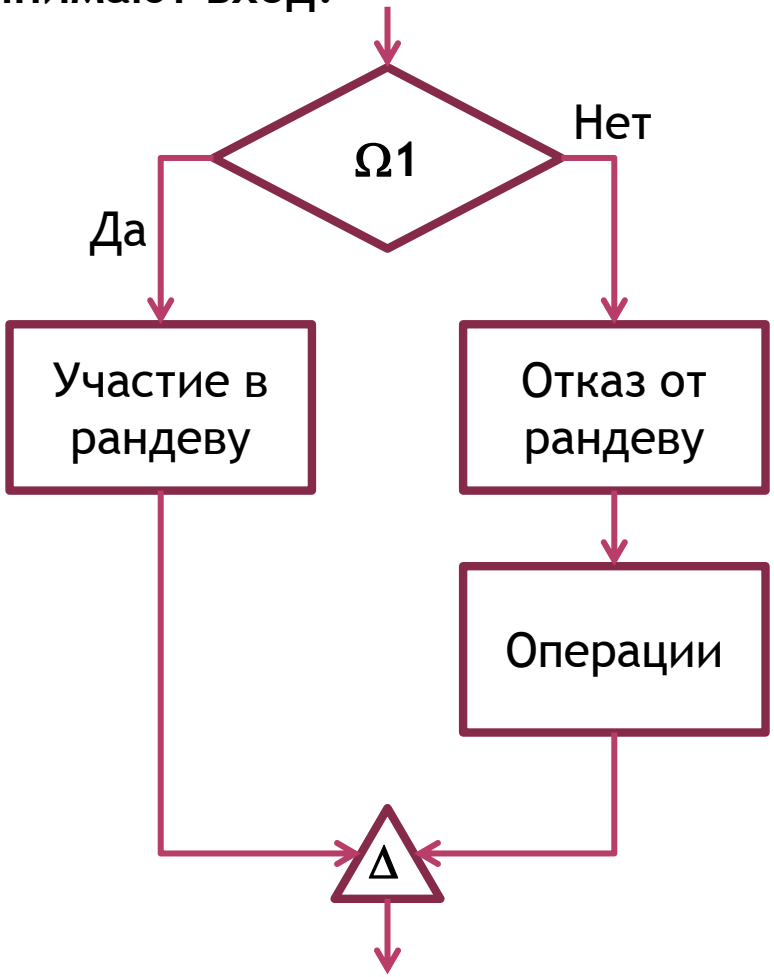
Выход из цикла.

Ветвление по условию (условный цикл)

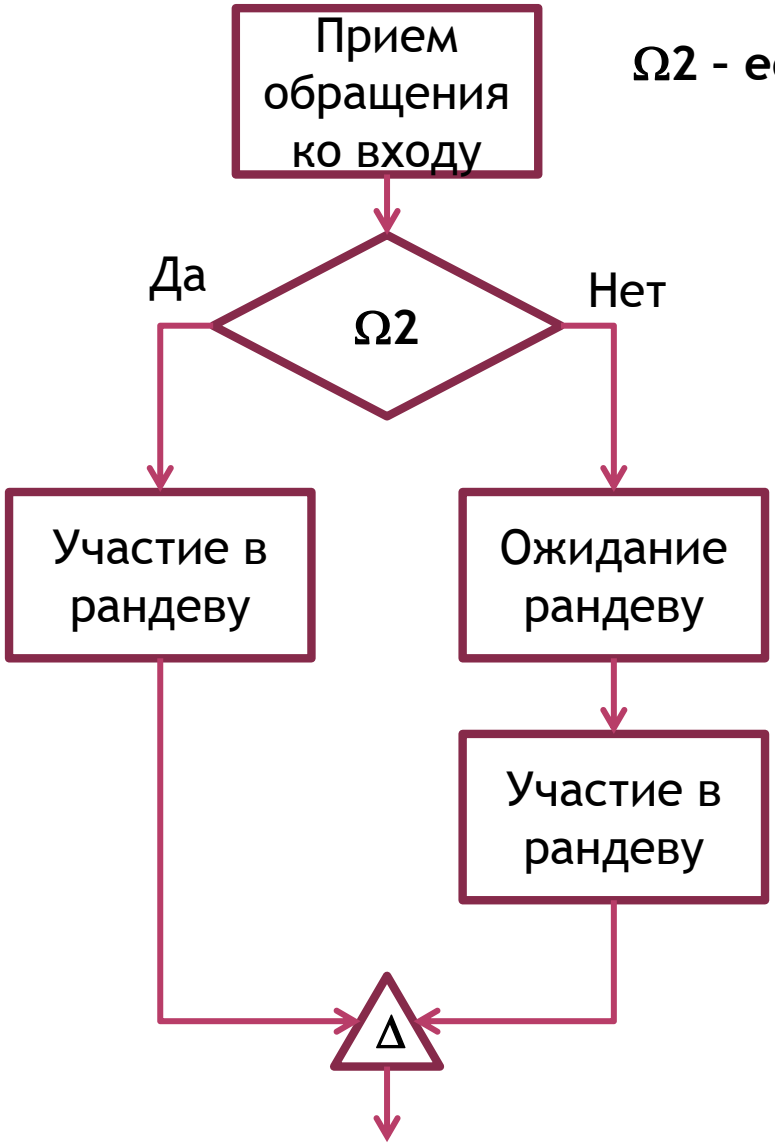
Альтернативные процессы



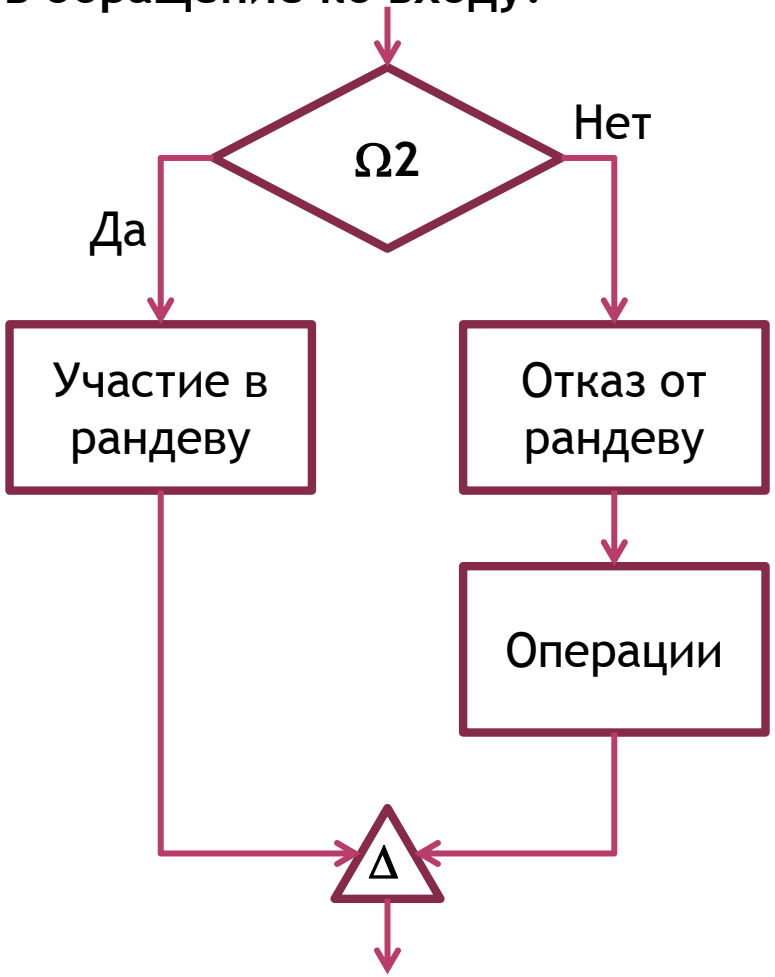
Ω1 - принимают вход?



Альтернативные процессы

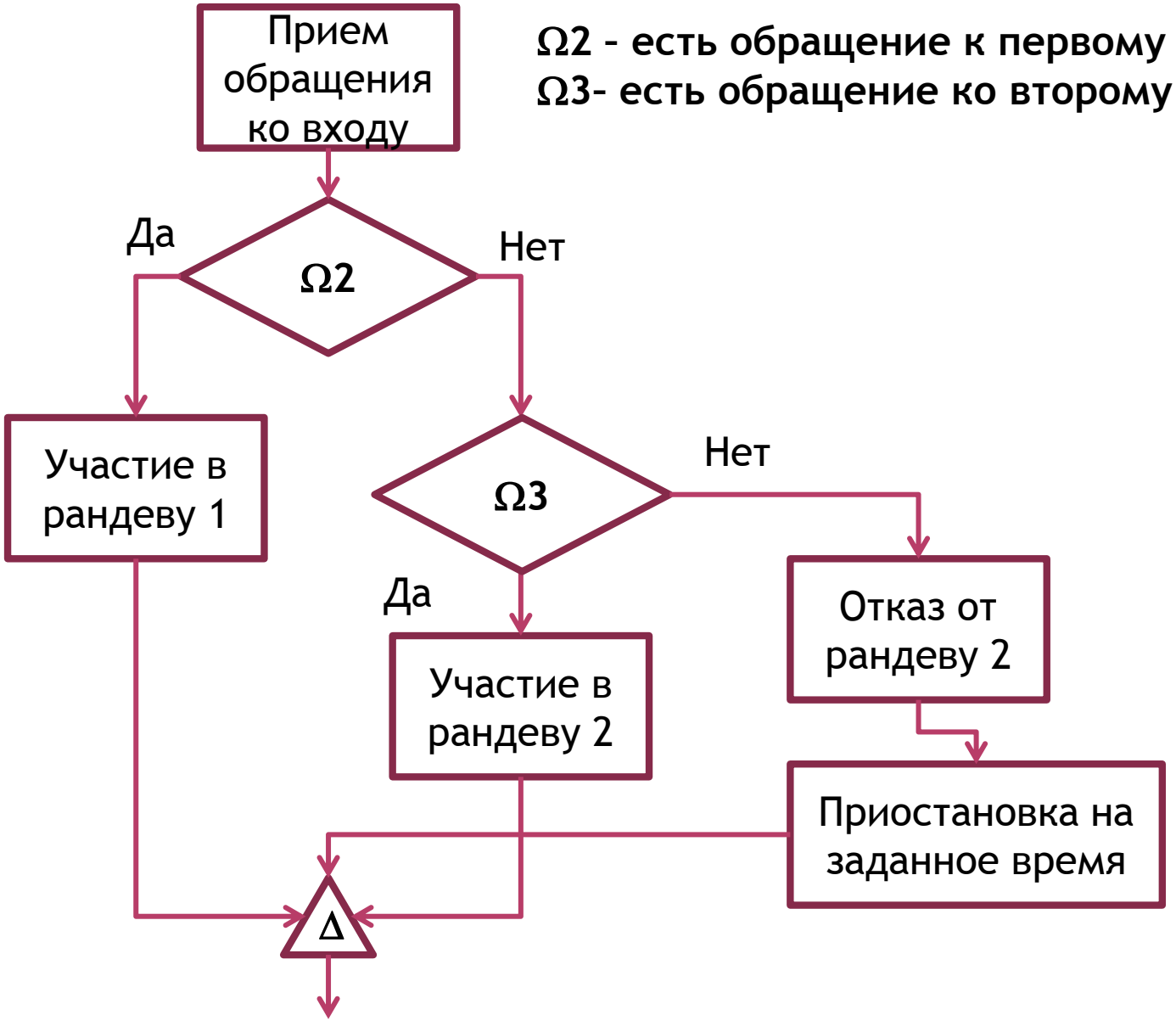


Ω2 - есть обращение ко входу?



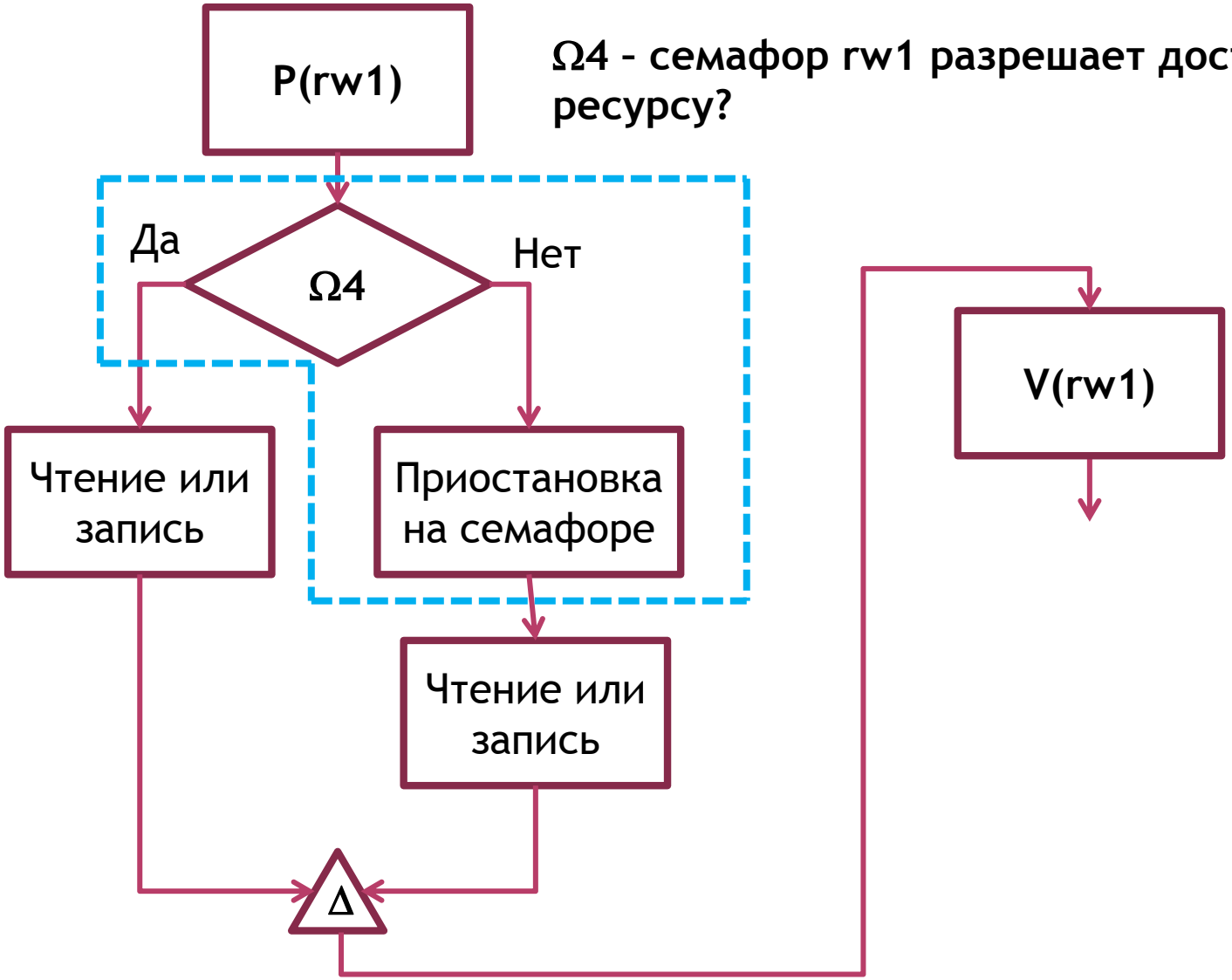
Альтернативные процессы

Ω2 - есть обращение к первому входу?
Ω3- есть обращение ко второму входу?

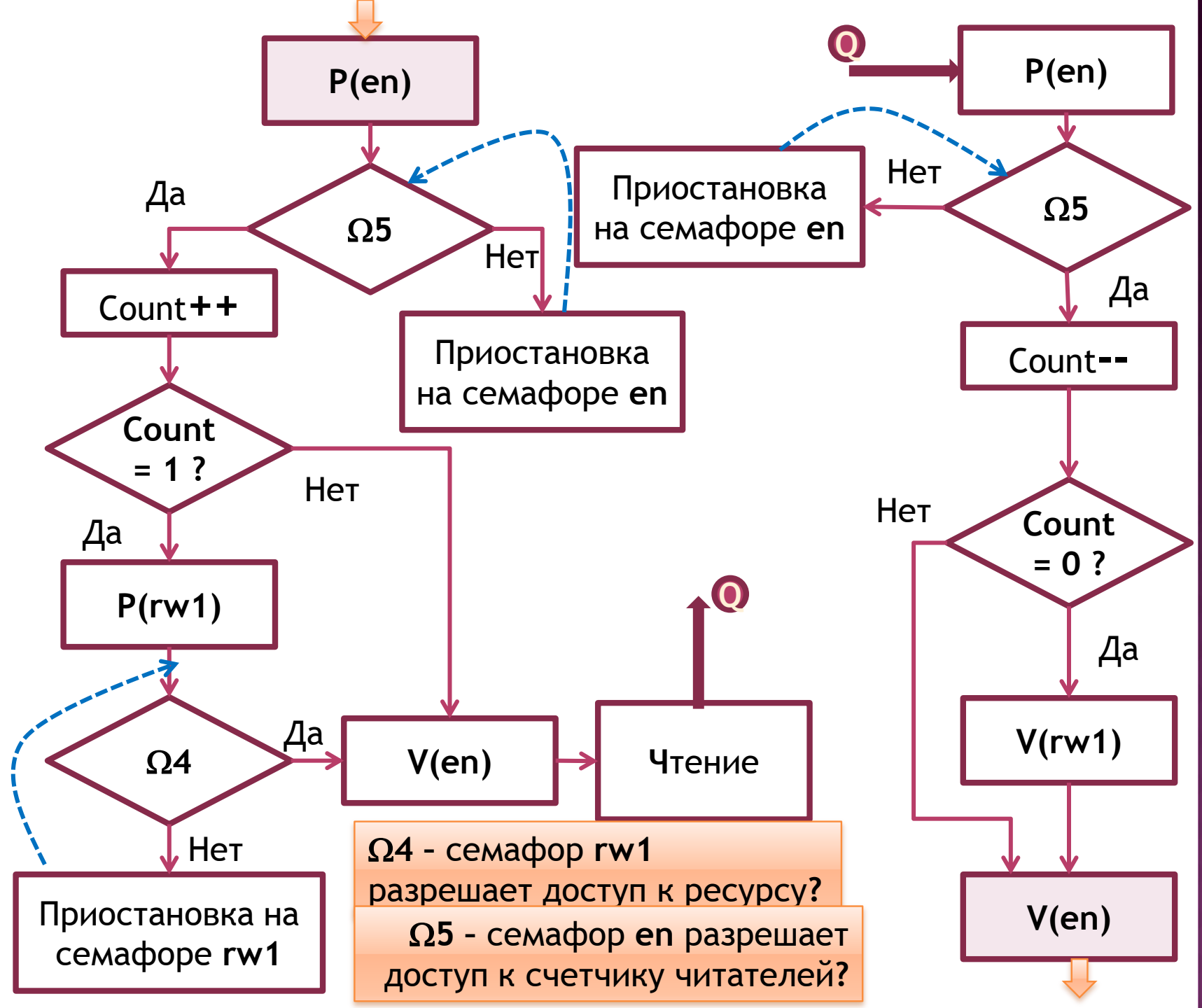


НИТИ

Альтернативные процессы



Ω4 - семафор rw1 разрешает доступ к ресурсу?


P(en)
Ω5
Да
Нет
Count++
Приостановка на семафоре en
Count = 1 ?
Нет
Да
P(rw1)
Ω4
Да
V(en)
Чтение
Q
Нет
Приостановка на семафоре rw1
Q
P(en)
Ω5
Нет
Приостановка на семафоре en
Да
Count--
Count = 0 ?
Нет
Да
V(rw1)
V(en)
Ω4 - семафор rw1 разрешает доступ к ресурсу?
Ω5 - семафор en разрешает доступ к счетчику читателей?

НЕДЕЛИМЫЕ ОПЕРАЦИИ

❖ Для реализации **взаимного исключения** мелкомодульные неделимые операции объединяются в крупномодульные (составные) действия.

❖ Вторая форма синхронизации процессов – **условная синхронизация** реализуется задержкой выполнения процесса до достижения программой состояния, удовлетворяющего некоторому предикату

Неделимое действие выполняет неделимое преобразование состояния.

❖ Это означает, что любое ***промежуточное*** состояние, которое может возникнуть при выполнении этого действия, не должно быть видимо для других процессов.

❖ *Мелкомодульное* неделимое действие реализуется непосредственно аппаратным обеспечением, на котором выполняется программа.

В общем случае в **о**дном неделимом действии необходимо выполнять **п**оследовательность операторов.

Необходим механизм синхронизации, позволяющий задать ***к**рупномодульное* неделимое действие – последовательность мелкомодульных неделимых операций, которая выглядит как неделимая.

Неделимые действия задаются с помощью угловых скобок. Например, выражение <*ev*> указывает, что его должно вычислять неделимым образом. Синхронизация определяется с помощью оператора **await:**

$$\langle await(B)\ L; \rangle$$

Булево выражение *B* задает *условие задержки,*

L *– список последовательных операторов, завершение которых гарантировано.*

B гарантировано имеет значение « истина», когда начинается последовательность операторов *L,* и ни одно промежуточное состояние в последовательности не видно другим процессам.

Например, выполнение кода

$$\langle await(rw > 0)\ rw = rw - 1; \rangle$$

откладывается до момента, когда значение *rw* станет положительным, а затем оно уменьшается на 1. Гарантируется, что перед вычитанием *rw* положительно.

Оператор *await* является очень мощным, поскольку он может быть использован для определения любых **к**рупномодульных **н**еделимых действий. Это делает его удобным для выражения синхронизации при разработке первоначальных решений задач синхронизации.

Вместе с тем, **реализация** этого оператора в общей форме очень дорога. Поэтому существует множество **ч**астных случаев реализации оператора await, допускающих его эффективную реализацию. Например, приведенный выше пример *await* является частным случаем операции *P* над семафором.

Общая форма оператора *await* определяет как взаимное исключение, так и синхронизацию по условию. Для определения только взаимного исключения можно использовать сокращенную форму операторов *await:*

$$\langle L; \rangle \qquad \langle x = x + 1; y = y + 1; \rangle$$

Для задания только условной синхронизации оператор сокращается так:

(1) $\langle await(B); \rangle \quad \langle await(rCount > 0); \rangle$

Если выражение B удовлетворяет условию «не больше одного», то выражение (1) может быть реализовано как цикл ожидания (spin loop):

$$\langle while(not\ B); \rangle$$

Поскольку тело *while* пусто, он просто зацикливается до тех пор, когда значением *B* станет «ложь».

Типы неделимых действий

❖ **Б**езусловное неделимое действие – действие, которое не содержит в теле условия задержки B. Такое действие может быть выполнено немедленно.
❖ **А**ппаратно реализуемые (мелкомодульные) действия, выражения в угловых скобках и операторы await, в которых условие опущено или является константой «истина», являются безусловно неделимыми действиями.

❖ **У**словное неделимое действие – оператор *await* с условием *B*. Если *B* ложно, то оно может стать истинным только в результате действия других процессов. Таким образом, процесс, ожидающий выполнения условного неделимого действия, может оказаться задержанным непредсказуемо долго.

Семафоры. Операции с семафорами

Объекты типа **Semaphore** ограничивают число потоков, которые могут одновременно получать доступ к ресурсу или пулу ресурсов.

Метод **WaitOne** – вход в семафор, **Release** – освобождение семафора. Объект **Semaphore** создается с помощью конструктора, в который передаются параметры – **н**ачальное значение семафора и **к**оличество процессов, которые могут одновременно иметь доступ к ресурсу.

SemaphoreSlim – упрощенная альтернатива семафору **Semaphore**, ограничивающая количество потоков, которые могут параллельно обращаться к ресурсу или пулу ресурсов.