

СХЕМА

для чайников

*Как записать тренировку одной строкой
и постепенно научиться понимать
собственную нагрузку*

30/30#

Практическое руководство по языку Схема и Zaruba Timer

Серёра Photoskiff

создатель языка «Схема» и Zaruba Timer

О книге

Это не спецификация языка и не экзамен по синтаксису. Книга построена так, чтобы каждые несколько минут вы получали новую вещь, которую уже можете применить в своей тренировке.

Сначала будет одна простая строка. Потом — ещё один знак. Потом — ещё одна возможность. Сложные слова появляются только тогда, когда без них уже действительно удобнее.

Главная идея. Не учите Схему целиком. Каждый раз спрашивайте: «Что ещё я хочу сообщить таймеру?» — и добавляйте только это.

Содержание

I. Первые пять минут — работа, отдых, повтор, последний отдых.

II. Нагрузка, повторения и темп — @, интенсивность, r, t и две стороны атома.

III. Собираем тренировку — атомы, группы, метки, dur: и reps:.

IV. Оставляем себе свободу — ~, бесконечные циклы, Регистратор, материализация.

V. Почему неполная Схема всё равно работает — правило трафарета и настройка интенсивности.

VI. Учимся узнавать свою нагрузку — личная база проверенных Схем.

VII. Вариативность — прогрессии, темпо-рампы, случайность и адаптивность.

VIII. Пишем и читаем уверенно — константы, арифметика, разбор длинных Схем, ошибки, шпаргалка.

ЧАСТЬ I

Первые пять минут

Сначала научимся записывать самый обычный интервал. Никакой теории заранее — только то, что уже можно запустить в таймере.

1. Начнём с тренировки

Допустим, ваша тренировка выглядит так:

30 секунд работаю, 30 секунд отдыхаю.

В языке Схема это записывается:

30/30#

Вот и всё. Вы уже написали свою первую Схему.

Первое число — работа. Второе — отдых.

РАБОТА / ОТДЫХ #

Поэтому:

60/20#

60 секунд работать, 20 секунд отдыхать и закончить тренировку.

Знак # означает: здесь Схема заканчивается.

2. А если отдыха нет?

60/0#

60 секунд работать, отдых — 0 секунд, конец.

Можно встретить более короткие записи, но пока привыкайте писать явно 60/0#, а не просто 60.

Это не формальность. Если часть Схемы не указать, таймер или другое устройство может дополнить её своими настройками. /0 явно говорит «отдыха нет», а # — «закончить именно здесь». Почему так устроено, разберём позже в главе «Правило трафарета».

РАБОТА / ОТДЫХ

3. Повторим это несколько раз

Хотим: 30 секунд работать, 30 отдыхать — пять раз.

30/30*5#

Знак * означает «повторить». Поэтому *5 читается: повторить пять раз.

- 30 работа / 30 отдых
- 30 работа / 30 отдых
- 30 работа / 30 отдых
- 30 работа / 30 отдых
- 30 работа / 30 отдых

У нас уже получилась настоящая интервальная тренировка.

4. Последний отдых не нужен?

40/30*5#

После пятой работы таймер выполнит ещё и последние 30 секунд отдыха. Если тренировка сразу заканчивается, этот отдых часто не нужен.

40/30*5`#

Знак ` — обратная кавычка. Она убирает только последний отдых последнего круга.

работа/отдых → работа/отдых → ... → работа/конец

ЧАСТЬ II

Нагрузка, повторения и темп

Теперь время уже умеем задавать. Добавим смысл: насколько тяжело, сколько сделать и с какой скоростью.

5. Добавим нагрузку

30 секунд прогулки и 30 секунд предельного спринта — совершенно разные 30 секунд. Для описания нагрузки используется знак @.

30/30@8#

30 секунд работать с интенсивностью 8, затем 30 секунд отдыхать.

Интенсивность в Схеме задаётся по шкале от 0 до 10, где 10 — примерно 100% доступного усилия, абсолютный максимум.

Не знаете точное число? Это нормально. Одной из задач работы со Схемой как раз будет постепенно научиться находить свои интенсивности, а не угадывать их с потолка.

6. Самое важное правило после @

15/15@8/2#

До @ мы видим 15/15: работа / отдых. После @ действует точно такое же соответствие: 8/2 — параметры работы / параметры отдыха.

РАБОТА / ОТДЫХ @ ПАРАМЕТРЫ РАБОТЫ / ПАРАМЕТРЫ ОТДЫХА #

- Работа: 15 секунд, интенсивность 8.

- Отдых: 15 секунд, интенсивность 2.

Это правило будет действовать и дальше для других параметров.

7. А если после @ нет второй части?

30/30@8#

После @ нет /. Значит всё написанное после @ относится к работе.

- Работа: 30 секунд, интенсивность 8.
- Отдых: 30 секунд.

@ПАРАМЕТРЫ = параметры работы
@ПАРАМЕТРЫ/ПАРАМЕТРЫ = параметры работы / параметры отдыха

8. Что означает интенсивность 10

Легко подумать, что шкала 1–10 устроена как обычная линейка и расстояние между всеми числами одинаковое. Для организма это не так.

Интенсивности 1–3 могут соответствовать лёгкой или умеренной работе, которую человек способен выполнять очень долго — например ходьбе.

А 9 — уже очень тяжёлая работа, которую можно выдерживать лишь ограниченное время.

9 — порядка минут
9.9 — порядка десятков секунд
10 — абсолютный максимум, порядка 20 секунд

Это не медицинская таблица и не секундомер: у разных людей, упражнений и условий цифры различаются. Главное другое: разница между 1 и 3 совсем не такая, как между 8 и 10. Чем ближе к пределу организма, тем дороже становится каждый следующий шаг.

Математики называют такое поведение нелинейным.

Запоминать термин необязательно. Достаточно помнить: около 10 нагрузка растёт очень резко.

Zaruba Timer учитывает это при расчёте нагрузки и модели пульса.

9. Интенсивность и время могут быть дробными

@7.5 @8.45 @9.9

В русском тексте мы пишем «интенсивность 8,45», но внутри Схемы используется точка:

30/30@8.45#

То же самое со временем:

30.5/10#

30,5 секунды работы, 10 секунд отдыха.

10. Зададим количество повторений

Допустим: за 30 секунд надо сделать 10 повторений.

30/30@r10#

r происходит от английского repetitions — «повторения».

r10 = 10 повторений

40/20@r15#

40 секунд работы, цель 15 повторений, затем 20 секунд отдыха.

11. Интенсивность и повторения вместе

30/30@8r10#

Работа теперь имеет две характеристики: интенсивность 8 и цель 10 повторений. Мы просто добавили ещё одно свойство к уже знакомой работе.

12. Темп: t

Иногда важно не «сделать 20 повторений», а «делать со скоростью 20 повторений в минуту». Для этого используется t — от английского tempo, «темп».

60/0@t20#

Работать 60 секунд в темпе 20 повторений в минуту.

13. Не путайте r и t

r20 = всего 20 повторений

t20 = скорость 20 повторений в минуту

Поэтому следующие две Схемы — разные:

30/0@r20#

30/0@t20#

В первой за 30 секунд требуется сделать 20 повторений. Во второй темп равен 20 в минуту, поэтому за 30 секунд получится примерно 10.

r = сколько

t = как быстро

На одной стороне атома r и t одновременно не используются.

Полезная зацепка. Темп и повторения дают объективную характеристику работы. Позже их особенно удобно связывать с вашей реальной интенсивностью и постепенно строить собственную карту нагрузок.

14. Теперь прочитаем страшную запись

15/15@8.5t40/2t20#

Она уже не должна пугать.

- Работа: 15 секунд, интенсивность 8.5, темп 40 повторений в минуту.

- Отдых: 15 секунд, интенсивность 2, темп 20 в минуту.

РАБОТА / ОТДЫХ @ РАБОТА / ОТДЫХ

ЧАСТЬ III

Собираем тренировку

Один интервал — хорошо. Теперь научимся соединять интервалы в блоки и ограничивать их временем или объёмом.

15. Что такое атом

30/30@8r10

Это атом. Не пугайтесь слова: атом — просто один законченный кусочек тренировки.

30/30

60/0@t20

15/15@8.5t40/2t20

Все три записи — атомы.

16. Несколько атомов подряд

30/30 20/40#

Сначала выполняется 30/30, затем 20/40. Схема читается слева направо.

17. Объединим их в группу

Хотим: 30/30, затем 20/40, и весь этот блок повторить три раза.

(30/30 20/40)*3#

Круглые скобки создают группу. (...)*3 означает: всё внутри скобок повторить три раза.

18. Один простой закон групп

Интенсивность, g и t относятся к конкретным атомам, а не ко всей группе.

(30/30@8 20/40@8)*3#

После закрывающей скобки группы обычно ставится её повтор: $)^*3$.

19. Метки !1, !2 — даём участкам имена

Иногда мало сказать таймеру, сколько длится участок. Хочется ещё сказать, что именно здесь нужно делать. Для этого есть метки !1, !2, !3 и так далее.

15!1/0 15!2/0# 1=подтягивания 2=отжимания

Читаем: 15 секунд подтягиваний, затем 15 секунд отжиманий. !1 означает «возьми подпись №1», а !2 — «подпись №2». Сами подписи записаны после #.

!N — это метка с номером N. Она даёт участку имя, но сама по себе не меняет его время, темп или интенсивность.

Если в Схеме использована !1, для неё обязательно должна быть подпись 1=... после #. То же правило действует для !2, !3 и остальных номеров.

Позже можно встретить и запись ~!1. Номер после ! работает так же: он связывает отложенное, пока неизвестное значение с подписью №1. Пока достаточно запомнить простую вещь: !N = метка N.

20. А если трудно посчитать, сколько кругов поместится?

Допустим, мы хотим выполнять какой-то цикл 10 минут. Можно вручную считать круги и остаток. А можно сказать таймеру: этому блоку дай 600 секунд.

Для этого есть `dur` — сокращение от английского `duration`, «длительность».

```
dur:600(40/20*100)#
```

Выполнять цикл 40/20, но весь блок ограничить 600 секундами.

Число 100 просто гарантирует, что циклов заведомо хватит. Главное здесь — `dur:600(...)`.

21. Зачем `dur`: нужен в обычной Схеме

С простой тренировкой 40/20 легко увидеть, что круг занимает минуту. Но внутри могут быть разные атомы, группы и неудобные числа. И вдруг 20 минут не делятся на круг красиво.

Вместо вычислений и специального последнего хвоста можно написать:

```
dur:1200( . . . )#
```

Этому блоку дано 20 минут.

22. То же самое с повторениями

Есть похожая конструкция `reps` — сокращение от `repetitions`, «повторения».

```
reps:100(30/0@r10*20)#
```

Выполнять этот блок, пока общий заданный объём не достигнет 100 повторений.

dur:N(...) = ограничение временем
reps:N(...) = ограничение повторениями

Очень удобно, когда считать вручную лень. А считать вручную обычно лень.

ЧАСТЬ IV

Оставляем себе свободу

Схема не обязана знать всё заранее. Можно оставить неизвестным ровно то, что хочется решить уже во время тренировки.

23. А что, если я не знаю, сколько кругов сделаю?

Каждый круг известен:

180/0

Но сколько их будет сегодня — неизвестно:

180/0*~#title=бесконечный ЕЗМОМ

Знак ~ означает: это значение определится позже. Здесь неизвестно только количество кругов.

24. Как закончить такую тренировку

Делаете круг за кругом столько, сколько хотите. Когда решаете заканчивать, нажимаете в таймере кнопку «Последний круг». Текущий круг становится последним, и тренировка заканчивается нормально.

***~ = повторяй, пока я сам не решу остановиться**

25. Можно заранее задать темп

180/0@t20*~#

Каждый круг длится 180 секунд и идёт в темпе 20 повторений в минуту. Количество кругов решает спортсмен.

26. Или количество повторений

180/0@r50*~#

Каждый круг: 180 секунд, цель 50 повторений. Кругов — сколько захотите.

27. ~ не означает хаос

180/0@t20*~#

Здесь известно почти всё. Неизвестно только одно число: сколько будет кругов. ~ означает не «ничего не знаю», а «вот это конкретное значение узнаем позже».

28. Регистратор

~/0#

Это Регистратор. Даже время работы заранее не задано. Запускаете таймер — и делаете что хотите.

- ходить;
- бегать;
- работать с гирями;
- подтягиваться;
- отдыхать;
- менять упражнения и темп.

Таймер ничего не навязывает. Он просто записывает сессию.

29. «Поставил и забыл»

~/0#

Запустили. Убрали телефон. Занимаетесь.

Если подключён пульсометр, таймер продолжает записывать пульс и другие доступные данные. Это самая свободная Схема из возможных.

Но между полностью запрограммированной тренировкой и Регистратором существует огромное количество смешанных вариантов.

30. Отложенная Схема может материализоваться

180/0*~#

Допустим, во время тренировки вы сделали семь кругов.

Неизвестное ~ стало конкретным числом 7:

180/0*7#

Это называется материализацией. Слово страшнее смысла: неизвестные значения стали известными. Получившуюся Схему можно сохранить и потом использовать как обычную готовую тренировку.

31. Четыре состояния — совсем коротко

- Активная — всё известно заранее: 30/30*5#.
- Отложенная — что-то узнаем во время выполнения: 30/30*~#.
- Материализованная — неизвестные значения уже стали конкретными.
- Реализованная — то, что спортсмен фактически выполнил.

План и реальность — не всегда одно и то же. Пока этого достаточно.

32. Название тренировки

180/0*~#title=бесконечный ЕЗМОМ

title по-английски означает «название». Всё после # здесь — дополнительная информация о Схеме.

ЧАСТЬ V

Почему неполная Схема всё равно работает

Теперь мы уже достаточно умеем, чтобы открыть обещанный секрет: почему /0, # и отсутствующая интенсивность — части одного правила.

33. Правило трафарета

В Zaruba Timer есть обычные настройки: работа, отдых, количество кругов, общая длительность, интенсивность работы и отдыха. Их можно выставить ползунками вообще без редактора.

Из этих значений получается простая полная Схема. Будем называть её трафаретом.

Главное. Трафарет всегда полный. Пользовательская Схема может заменить только те части, которые вы записали явно.

34. Что делает трафарет

Представим, что в нём сейчас стоят:

работа	30
отдых	20
круги	5
интенсивность работы	7
интенсивность отдыха	2

Теперь мы даём таймеру пользовательскую Схему. Всё, что записано явно, заменяет соответствующую часть трафарета. А то,

чего нет, устройство при необходимости может взять из трафарета.

Вот и весь принцип.

35. Поэтому /0 — не украшение

Если написать только работу и не задать отдых, устройство может дополнить отдых само. Но:

20/0

Отдых действительно равен нулю.

Мы не оставили это место трафарету.

36. Поэтому # тоже не украшение

#

Тренировка заканчивается здесь.

Без явного конца устройство может иметь возможность продолжить или дополнить тренировку своими параметрами.

20/0#

20 секунд работы, никакого отдыха, после этого конец.

37. И поэтому отсутствие интенсивности не означает ноль

30/30*10#

Интенсивность нигде не указана. Но это не значит, что она равна нулю. Она просто не записана и может быть взята из трафарета.

Таймеру она нужна, например, чтобы построить теоретическую модель нагрузки и график пульса.

38. Полезная стратегия новой Схемы

60/0@t20*10#

Темп известен, а какая интенсивность действительно соответствует этой работе именно для вас — пока нет.

Необязательно придумывать число. Оставьте интенсивность трафарету, выполните тренировку, посмотрите реальный пульс и подстройте модель.

При следующих проходах снова сравните модель с реальностью. Когда совпадение станет хорошим, найденную интенсивность можно зафиксировать прямо в Схеме.

39. Фиксируем найденную нагрузку

Допустим, после нескольких тренировок хорошее совпадение получилось при интенсивности 7.4:

60/0@7.4t20*10#

Теперь это значение стало частью самой Схемы и больше не зависит от текущего трафарета.

Таймер уже может использовать эту проверенную модель как ориентир и отслеживать, насколько реальный пульс сегодня отклоняется от ожидаемого.

Хорошая стратегия. Если в новой Схеме уже есть t или r , интенсивность часто разумно сначала не фиксировать. Сначала измерьте себя, потом запишите найденное число.

ЧАСТЬ VI

Учимся узнавать свою нагрузку

Вот здесь Схема перестаёт быть только языком записи. Проверенные Схемы постепенно становятся личной базой знаний о том, как именно ваш организм реагирует на разные условия.

40. Не просто каталог тренировок

Работая со Схемами, полезно постепенно собирать не только сами тренировки, но и соответствия нагрузки конкретным условиям.

- темпу;
- количеству повторений;
- весу снаряда;
- упражнению;
- скорости движения;
- длительности;
- другим важным условиям.

Не всё из этого обязано быть записано одним и тем же символом внутри Схемы. Важно другое: проверенная Схема хранится вместе с понятным контекстом и становится ещё одной точкой вашей личной карты нагрузки.

41. Полная проверенная Схема — маленький кусочек знания

60/0@7.4t20*10#

После нескольких проверок эта строка означает уже больше, чем просто «десять минут работай в темпе 20».

В этих условиях такая работа для меня обычно соответствует примерно такой нагрузке и такой реакции организма.

Если модель несколько раз хорошо совпала с реальным пульсом, Схема становится хорошо подогнанной опорной точкой.

42. Чем больше знаешь, тем точнее можешь варьировать

Допустим, вы уже хорошо проверили несколько темпов для одного упражнения и веса:

t18	t20	t22
-----	-----	-----

Теперь хотите попробовать t21 или построить рампу t[18->22]. Вы уже не гадаете с нуля: рядом есть проверенные точки.

После выполнения новая тренировка добавит ещё одну точку в вашу базу.

**придумал → выполнил → сравнил → уточнил → сохранил
→ использовал дальше**

43. Самая важная привычка автора Схем

Полезно спрашивать не только: «Какую тренировку я хочу сделать?»

Что эта тренировка научит меня о моей нагрузке?

Хорошо проверенная Схема — не просто сохранённая тренировка. Это ещё одна измеренная точка на вашей личной карте нагрузки.

ЧАСТЬ VII

Вариативность

Теперь базовый язык уже знаком. Можно переходить к вещам, ради которых Схема особенно интересна: управляемому изменению нагрузки во времени.

44. Прогрессия

Представим тренировку:

```
30/30
25/35
20/40
```

Работа каждый раз уменьшается на 5 секунд, а отдых увеличивается на 5. Можно записать всё вручную, а можно:

```
30/30->[for:3,delta:-5/+5]#
```

delta означает «изменение»: работа -5, отдых +5. for:3 — получить три значения последовательности.

45. Можно двигаться до цели

Вместо количества шагов можно использовать to — «до».

```
->[to:...,delta:...]
```

Изменяй такими шагами, пока не дойдёшь до заданного значения.

Это пригодится в более сложных прогрессиях.

46. Темпо-рампа

Название звучит страшнее, чем сама вещь. Темпо ramp — «темпо-рампа» — выражение, пришедшее из музыки. Там это

постепенное ускорение или замедление темпа композиции, трека или диджей-сета.

медленно → быстрее → ещё быстрее

В тренировке идея та же.

47. Разгоняем или замедляем темп

60/10@t[15->25]#

За рабочий интервал увеличить темп от 15 до 25 повторений в минуту.

Если написать наоборот:

60/10@t[25->15]#

Постепенно замедляться от 25 до 15.

Вот и вся темпо-рампа.

48. Можно указать шаг

60/10@t[15->25 d:2]#

d — короткая запись delta, то есть изменения. Темпы будут:

15 17 19 21 23 25

Можно написать и полностью: delta:2.

49. Свяжем изменение темпа с нагрузкой

60/10@t[15->25 i:6->9]#

Здесь i:6->9 не приказывает нагрузке изменяться от 6 до 9. Темп уже меняется по t[15->25].

i сообщает таймеру другое: насколько тяжёлы для вас два конца этого диапазона.

Допустим, речь идёт о рывке гири 24 кг. Темп 15 для вас комфортный — примерно интенсивность 6. А темп 25 уже близок к пределу — примерно 9.

t[15->25 i:6->9]

Темп меняется от 15 до 25; для меня нагрузке при темпе 15 соответствует примерно 6, а при темпе 25 — примерно 9.

Подумайте о себе. Перед тем как ставить i:..., попробуйте почувствовать и проверить оба конца диапазона. Это не абстрактные числа: вы сообщаете таймеру своё знание о собственной нагрузке.

50. Одному темпу не соответствует одна вечная интенсивность

Теперь представим, что вместо гири 24 кг вы взяли 8 кг. Темпы остались 15 → 25, но ощущения будут совсем другими.

Темп 25 с 24 кг мог быть почти предельным, а с 8 кг — вполне умеренным. Значит и соответствующие интенсивности будут другими, а изменение нагрузки с ростом темпа может быть гораздо спокойнее.

Нагрузку создаёт не один темп. Важны упражнение, вес, темп, объём, длительность, состояние спортсмена и другие условия.

Схема не говорит: «темп 25 всегда означает интенсивность 9». Она позволяет сказать: «вот в этих условиях для этого спортсмена такой темп примерно соответствует такой нагрузке».

51. Таймер соединяет опорные точки

60/10@t[15->25 d:2 i:6->9]#

Темпы будут 15, 17, 19, 21, 23, 25. Между двумя указанными вами нагрузками таймер строит промежуточные значения с учётом нелинейности шкалы интенсивности.

ТЕМП	ИНТЕНСИВНОСТЬ
15	6.00
17	6.42
19	6.92
21	7.50
23	8.20
25	9.00

Обратите внимание: интенсивность прибавляется не одинаковыми кусочками. Возле верхней части шкалы цена следующего шага растёт.

52. Реальная Схема с разгоном

58/2@t[18->26 i:4->8]*10`#title=армейский рывок с ускорением

- 58/2 — 58 секунд работы, 2 отдыха.
- t[18->26] — темп разгоняется от 18 до 26.
- i:4->8 — этим крайним темпам соответствуют интенсивности 4 и 8.
- *10 — десять кругов.
- ` — последний отдых удалить.

И всё это помещается в одну строку.

53. Управляемая случайность

Можно даже сказать таймеру: вот набор допустимых элементов, а конкретную последовательность собери сам.

```
random(
  30/0@9t220
  30/0@6t140
  30/0@7t180
  turb:7 dur:580 min:8
)#
```

random означает «случайный выбор». Но это не хаос: вы заранее определили, из чего можно строить тренировку и какие ограничения должны соблюдаться.

Таймер выбирает только внутри заданных правил. Это и есть управляемая вариативность.

54. Повторяем случайность

Если нужен тот же случайный вариант ещё раз, используется seed — начальное число случайной последовательности.

Проще думать о нём как о номере конкретного случайного варианта. Одинаковый seed позволяет воспроизвести ту же последовательность.

Тренировка создана случайно, но её всё равно можно повторить.

55. Адаптивная работа

```
300/0@t180^#
```

Знак ^ обозначает адаптивный атом. Исходный план остаётся t180, но реальный темп во время выполнения может меняться.

Таймер сохраняет отдельно то, что было задумано, и то, что спортсмен сделал на самом деле. План не переписывается задним числом.

ЧАСТЬ VIII

Пишем и читаем уверенно

Остались инструменты, которые делают длинные Схемы короче, и несколько приёмов, которые спасают от паники при встрече с незнакомой строкой.

56. Константы

В длинной Схеме одно число может встречаться много раз. Можно дать ему имя:

```
[N=5]30/30*N#
```

Здесь $N=5$, поэтому $*N$ означает то же самое, что $*5$. Константы особенно полезны, когда один параметр управляет сразу несколькими местами.

57. Арифметика

В числовых полях можно использовать обычные действия $+$, $-$, $*$, $/$ и скобки.

```
[N=5]30/(N*6)#
```

$N*6 = 30$, поэтому фактически получится $30/30\#$.

Некоторые значения по смыслу должны оставаться целыми. Например число повторений $r10$. Таймер не обязан молча округлять $r10.37$ за пользователя.

58. Как читать любую страшную Схему

```
58/2@t[18->26 i:4->8]*10`#
```

Не пытайтесь понять её целиком. Разберите слоями:

- $58/2$ — работа и отдых;

- @ — начинаются параметры;
- t[18->26] — темп;
- i:4->8 — соответствие крайних темпов интенсивности;
- *10 — количество кругов;
- ` — убрать последний отдых;
- # — конец.

Вместо одной страшной строки получилось семь уже знакомых вещей.

59. Как самому строить Схему

```
30/30#
30/30*5#
30/30@8*5#
30/30@8r10*5#
30/30@8r10*5`#
```

Каждая следующая строка добавляет только одну новую мысль.

Главный вопрос. Не «как выучить Схему?», а «что ещё я хочу сообщить таймеру?»

60. Шесть ошибок, которые легко избежать

- Путать стороны: 15/15@8/2# — работа 8 / отдых 2, а не переход от 8 к 2.
- Путать r и t: r10 — десять повторений, t10 — десять повторений в минуту.
- Считать отсутствие параметра нулём: 30/30# не означает интенсивность 0.
- Использовать запятую внутри Схемы: правильно @8.5, а не @8,5.

- Считать ~ символом полной неизвестности: в 180/0@t20*~# неизвестно только количество кругов.
- Пытаться понять длинную Схему сразу: разбирайте её слоями.

61. Шпаргалка на одну страницу

30/30# — 30 секунд работы / 30 отдыха.

300/0# — непрерывная работа без отдыха.

30/30*5# — пять кругов.

30/30*5`# — пять кругов без последнего отдыха.

30/30@8# — интенсивность работы 8.

30/30@8/2# — интенсивность работы 8 / отдыха 2.

30/30@r10# — цель 10 повторений.

30/30@t20# — темп 20 повторений в минуту.

30/30@8r10# — интенсивность 8, цель 10 повторений.

(30/30 20/40)*3# — три повтора группы.

15!1/0 15!2/0# 1=подтягивания 2=отжимания — метки связывают участки с подписями.

dur:600(40/20*100)# — ограничить блок 600 секундами.

reps:100(30/0@r10*20)# — ограничить блок 100 повторениями.

180/0*~# — сколько угодно кругов.

~/0# — Регистратор.

30/30->[for:3,delta:-5/+5]# — 30/30 → 25/35 → 20/40.

60/10@t[15->25]# — разгон темпа от 15 до 25.

60/10@t[15->25 d:2 i:6->9]# — темп 15 → 25 шагом 2; крайним темпам соответствуют интенсивности 6 → 9.

300/0@t180^# — адаптивный атом.

Эпилог. Сначала вы собираете Схемы

В начале книги была одна строка:

30/30#

Потом появились нагрузка, темп, повторения, группы, свободные значения, рампы и вариативность.

Но, возможно, самое интересное начинается позже — когда Схемы перестают быть просто записями тренировок.

Одна проверенная Схема говорит: «вот это я уже знаю о себе». Другая добавляет ещё одну точку. Потом ещё одну.

- вот такой темп с этим весом для меня комфортный;
- здесь нагрузка уже становится тяжёлой;
- вот так обычно реагирует мой пульс;
- это раньше было почти пределом, а теперь стало обычной работой;
- а вот здесь я пока ещё только начинаю исследование.

Постепенно каталог тренировок превращается в каталог проверенных моделей собственного организма.

Сначала вы собираете Схемы. Потом замечаете, что на самом деле собираете знания о себе.

И у этой работы нет неприятной финальной точки. Каждая хорошо подогнанная Схема не заканчивает исследование — она делает следующую попытку точнее и интереснее.

Схема не говорит спортсмену, какую тренировку он обязан делать. Она позволяет точно записать, какую тренировку он

решил сделать, воспроизвести её, проверить на себе и узнать из неё что-то новое.

Итого. Схема — формальный язык управляемой вариативности тренировочной нагрузки, а Zaruba Timer — среда её воспроизводимого исполнения и физиологической проверки.

Но всё это по-прежнему начинается с одной строки:

30/30#

Если она понятна — Эверест уже не такой высокий.

СХЕМА

*Язык, в котором тренировка становится
точной, воспроизводимой и исследуемой.*

30/30#