

# Машина Кузьмина. (Управляемая недетерминированная машина Тьюринга.)

*Расчитывая что читатель знаком с МТ оформим описание останавливаясь на отличиях предлагаемой машины от МТ.*

Вместо ленты МТ машина Кузьмина представляет собой перечислимое множество концептов (ячеек) и порождаемое множество управляющих устройств (можно считать их головками, кому это удобно) назначение которых – адресация (активация) концептов. Номер концепта назовем адресом. Одна головка может адресоваться только к одному концепту.

Состоянием обладает не головка, а концепт. Наряду с понятием «состояние» иногда будем применять понятие «событие». Ибо событие это переход из одного состояния в другое и обычно именуется по названию нового состояния. Исходя из этого можем отметить сразу 5 событий концептов- адресация, адресация для чтения, адресация для записи, адресация для выполнения, адресация для инициализации. Устройства могут перемещаться (адресоваться) к любому концепту и порождаться подписками, как и концепты могут порождаться адресацией с инициализацией. По выполнению адресации устройство (шина) освобождается.

Работа машина заключается в адресации (активации) концепта и, возможно, изменения состояния адресуемого концепта (например, при адресации для записи). На этом функция головки завершается.

Каждый концепт может иметь подписки на события. Подписка это адрес (адреса, потому как может быть несколько подписок на одно событие) к которому осуществляется переход при активации концепта и при истинности соответствующего события. Таким образом, адресация концепта может породить лавину подписок.

Функционирование машины заключается в адресации концептов, которые остаются иммутабельными.

Управление конкретной машиной осуществляется добавлением/удалением подписок.

Программирование заключается в создании множества концептов и оформлении подписок для необходимого функционала.

У читателя правомерно возникает вопрос, а нахрена это все? Есть МТ. Найти невыполнимые задачи для нее согласно теореме о полноте затруднительно. И вообще нет проблем. А вот, однако, есть. Дело в том, что **Машина Тьюринга (МТ)** — абстрактный исполнитель (абстрактная вычислительная машина). Была предложена [Аланом Тьюрингом](#) в [1936 году](#) для формализации понятия [алгоритма](#). А не все задачи удобны ( я не говорю разрешимы) для решения алгоритмами. И вот я хочу показать нечто новое. Название которому я дал А-ритмы. И вместо блок-схем А-схемы. Смотрим программу счетчика электроэнергии.

# Счетчик электрической энергии.

```
@ Byte (Interval, Count) Class ElectricMeter #Класс с параметрами времени в миллисекундах и счетчик суммирования»
New / {Timer T (Interval) ! / { Tick Timer.Start} # Создание таймера, создание подписки, сохранение счетчика.»
{ ' Создание свойств
  Byte Counter :: Count #Количество тиков таймера до секунды при инициализации»
    := @( Byte X ) Counter += X #Ведем счетчик.»
    Null {Counter =Count) Value =(Voltage/Counter)* (Electricity/Counter)} #При событии 0, формируем счет и
восстанавливаем значение счетчика. Как произведение средних значений тока и напряжения.»
  ' Определение атрибутов Voltage, Electricity и Value
  Float { Voltage := @( Float X ) Voltage += X # Суммирование значения при событии Write»
    := 0 # При чтении обнуляем счет.»
    Electricity := @( Float X ) Electricity +=X # Суммирование значения при событии Write»
    := 0 # При чтении обнуляем счет.»
    Value 0 'Начальное значение
    := @( Float X ) Value+ X #Значение счетчика наращиваем при записи.»
  }
  ' По каждому тикку ведем счетчик, читаем значение тока и напряжения по физическому адресу порта и суммируем
  Void Tick :[Move Counter ,-1 Move Voltage, "vvvv" Move Electricity ,"eeee"]
}
```

Объект создается из класса с параметрами значения интервала времени и значения счетчика (Interval, Count).

При создании объекта событием выполнение «|» запускается таймер с значением интервала.

Класс состоит из атрибутов Counter, Voltage, Electricity и Value назначение которых, соответственно вести счетчик тиков, значений напряжения, тока и значения энергии.

Функционал определяется событиями. Итак по порядку:

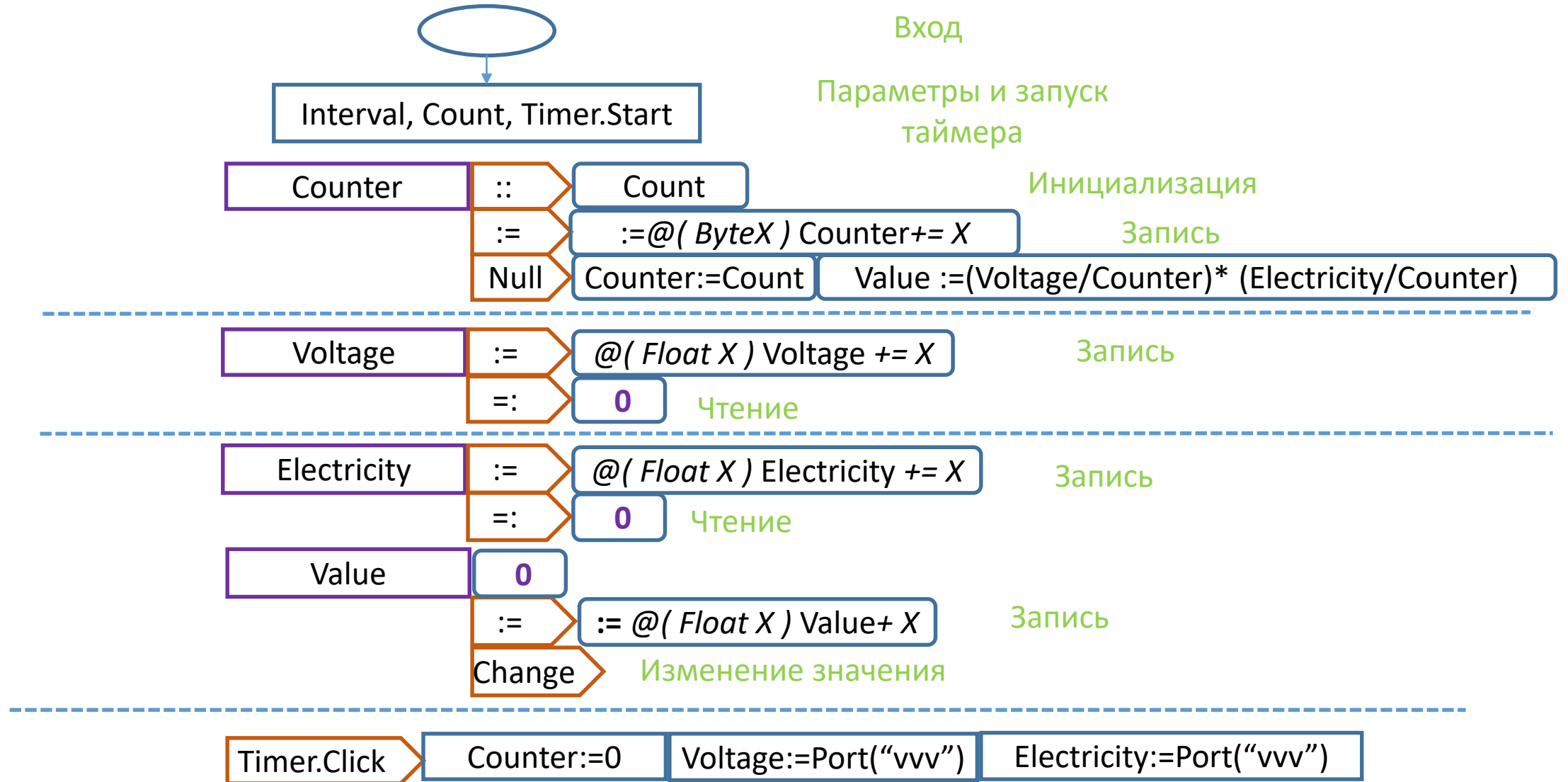
Счетчик Counter по событию инициализации «::»получает начальное значение (Count), при событии Запись «:=» ведет счет и по событию Null вычисляет значение для суммирования Value.

Атрибуты Voltage и Electricity при записи суммируют записываемые значения напряжения и тока и при чтении обнуляются. Событие чтения «=:» иницируется выражением по событию Null атрибута Counter

Атрибут Value только суммирует значения по событию запись «:=».

По каждому событию Tick одновременно (о чем говорят квадратные скобки) выполняется запись в атрибуты Counter, Voltage и Electricity.

# А-ритм. А схема



Фигуры обозначены фиолетовым цветом- атрибуты, оранжевым- события, синим – подписки, зеленым комментариями.  
Значение счетчика энергии можно получить организовав подписку к событию Value.Change