

### Запуск скрипта start.py

Перед запуском убедитесь, что в системе установлен и зарегистрирован в системных переменных (для Windows в системной переменной Path) Python 2.7.\*.

Для запуска скрипта из командной строки необходимо написать и выполнить:

**python start.py** – в таком случае скрипт будет пытаться открыть конфигурационный файл (который задан по умолчанию) config.conf.

Либо:

**python start.py myconfig.conf** – в данном случае, скрипт будет считывать информацию о параметрах из указанного файла myconfig.conf

### Этапы работы скрипта start.py

Скрипт start.py выполняет следующие действия:

1. Создает файлы конфигурации в папке gen\_conf\_files, исходя из параметров исходного файла config.conf (о возможных параметрах данного файла ниже);
2. Итерационно (по количеству сгенерированных файлов конфигурации) запускает скрипт генерации вспомогательных файлов в соответствующих директориях (по названию файла конфигурации) в папке gen\_net\_files для создания сетки \*.edg.xml, \*.nod.xml, \*.con.xml, затем, если все прошло без ошибок формирует на основе данных файлов в той же папке файл \*.net.xml;
3. итерационно запускает симуляцию с параметром '-nogui' и записывает файлы статистики в соответствующих директориях (по названию файла сети) в папке gen\_stat\_files;
4. завершает работу и выводит информацию о проделанной работе.

### Параметры файла config.conf

На картинке ниже представлена структура файла

```
[Parametrization]
name.=.test1
toll_scheme.=.00001100
truck_scheme.=.00000000
toll_spec.=.1
road_lanes.=.2
exp_length.=.700
arr_length.=.80
toll_lanes.=.8
stop_length=.20
drive_length=.20
conv_length=.5
dep_length.=.80
nar_length.=.700

[Scheme.of.Section]
1.=.Expansion__start0.:.road_lanes;.toll_lanes;.exp_length
2.=.One-to-Many__E: .toll_lanes;.arr_length
3.=.One-to-One__qpay: .drive_length
4.=.One-to-One__stoppay: .stop_length;.toll_spec
5.=.Many-to-One__O: .conv_length;.dep_length
6.=.Narrows__end0.:.toll_lanes;.road_lanes;.nar_length
```

Раздел [Parametrization] содержит общие параметры, такие как:

- name – имя проекта (будет использовано в качестве имени файлов);
- toll\_scheme – схема шлюзов по типам поддерживаемого способа оплаты (0 – наличные, 1 – транспондер);
- truck\_scheme – схема запрета проезда для грузовых автомобилей на участке, где применяется paying (0 – проезд разрешен, 1 – проезд запрещен). Если согласно scheme\_pay предусмотрена оплата посредством транспондера, то проезд грузовым автомобилям в любом случае будет разрешен;
- toll\_spec – указатель на применение toll\_scheme и truck\_scheme на участке

- переменные, используемые в проекте (можно создавать любое количество переменных с любыми именами из латинских букв и цифр)

Раздел [Scheme of Section] представляет собой описание построения топологии дороги и поддерживает следующие схемы:

- Expansion (var: x1, y1, z1) – расширение дороги, x1 – начальное число полос, y1 – конечное число полос (до которого необходимо расширить дорогу), z1 – расстояние участка от начала до конца
- Narrows (var: x2, y2, z2) – сужение дороги, x2 – начальное число полос, y2 – конечное число полос (до которого необходимо сузить дорогу), z2 – расстояние участка от начала до конца
- Straight (var: x3, y3) – прямой участок дороги, x3 – число полос, y3 – расстояние участка от начала до конца
- One-to-Many (var: x4, y4) – создание edge в количестве x4 на участке y4. *Данная схема не может применяться первой! Предполагается применение данной схемы после схем: Expansion, Straight*
- One-to-One (var: x5, toll\_spec=0) – “подхватывает” все конечные точки (junction), созданные на предыдущем шаге, и создает из каждой прямой участок дороги (смещение происходит только по оси абсцисс) расстоянием x5, параметр paying характеризует если ли система оплаты проезда на созданных участках, если да, то применяется схема, описанная в scheme\_pay (число знаков в данной переменной должно совпадать с числом создаваемых участков). *Данная схема не может применяться первой! Предполагается применение данной схемы после схемы One-to-Many или One-to-One.*
- Many-to-One (var: x6, y6) – “соединяет” все участки по конечным точкам (junction), созданных на предыдущем шаге, x6 – отступ до первого junction, y6 – расстояние, на протяжении которого происходят соединения. *Данная схема не может применяться первой! Предполагается применение данной схемы после схемы One-to-One.*

**Стоит отметить, что указанные выше для каждой схемы параметры являются обязательными!**

Теперь немного подробнее про правильность описания схемы, на примере первого пункта:

1 = Expansion\_\_start0.: road\_lanes; toll\_lanes; exp\_length

- Число 1 – порядковый номер шага;
- Expansion – имя применяемой на данном шаге схемы
- start0. – наименование участка дороги (между именем применяемой схемы и наименованием участка разделитель два нижних подчеркивания ‘\_\_’)
- road\_lanes; toll\_lanes; exp\_length – переменные, используемые в данной схеме, которые объявлены в разделе [Parametrization] (между наименованием участка и переменными разделитель двоеточие ‘:’)

В разделе [Parametrization] любой переменной (кроме name, toll\_spec, truck\_scheme – эти переменные исключительно с постоянным значением, некоторые отличия есть у переменной toll\_scheme\*) могут быть присвоены:

- постоянные значения (например, road\_lanes = 2),
- набор значений (например, road\_lanes = values(2,3,4,5))
- генератор значений (например, road\_lanes = generate(2,6,1), где 2 – начальное значение, 6 – конечное значение, 1 – шаг).

\*Переменная toll\_scheme тоже поддерживает все эти схемы присвоения значений, но с некоторыми отличиями:

- toll\_scheme = 00001100 – постоянная последовательность
- toll\_scheme = values(00001100,00011100) – набор последовательностей

- `toll_scheme = generate(01,8)` – генератор значений, где '01' – возможные значения в последовательности, '8' – длина генерируемой последовательности (ее необходимо ставить равной числу шлюзов, если она будет меньше, то ошибок не будет, по умолчанию будет ставиться 0 на соответствующий шлюз)

### **Результаты работы скрипта `start.py`**

В результате работы скрипта `start.py`, в той же директории создается папка `Output` с содержимым:

- Директория `gen_conf_files` – содержит сгенерированные файлы конфигурации
- Директория `gen_net_files` – содержит вложенные поддиректории (с названиями соответствующих файлов конфигураций), в которых находятся сгенерированные файлы для построения сети: `*.edg.xml`, `*.nod.xml`, `*.con.xml`, и сам файл сети `*.net.xml`
- Директория `gen_stat_files` – содержит вложенные поддиректории (с названиями соответствующих файлов сети), в которых находятся все выходные статистические данные по отработанным сценариям.