# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

**Факультет физико-математических и естественных наук Кафедра прикладной информатики и теории вероятностей**

# ОТЧЕТ

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 11**

# дисциплина: Моделирование информационных процессов

Студент: Тейшейра Боа Морте Селмилтон

Группа: НкНбд-01-20

**МОСКВА**

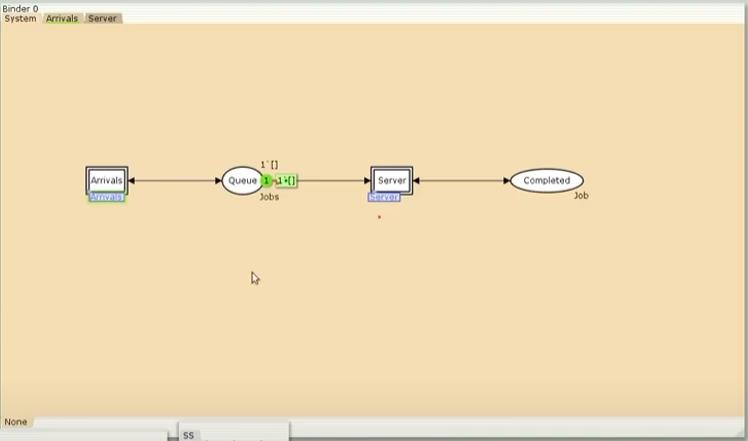
2023 г.

## Постановка задачи

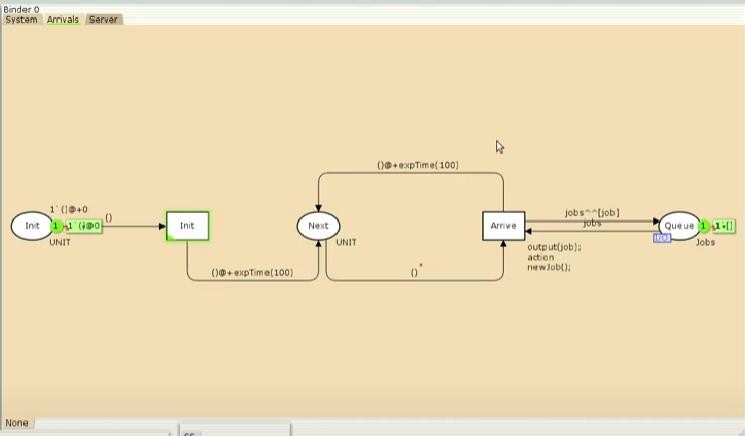
Цель работы решить Задача об Модель системы массового обслуживания.

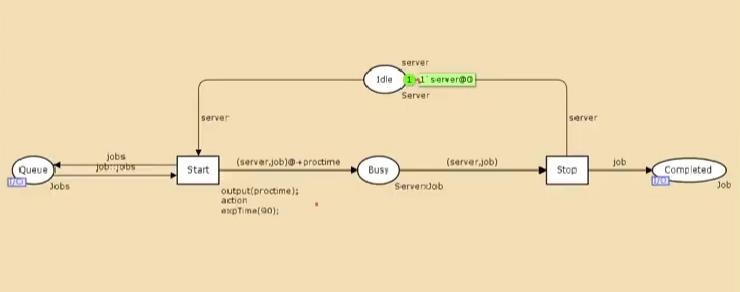
## Выполнение работы

Построение модели с помощью CPNTools 1. Будем использовать три отдельных листа: на первом листе опишем граф системы на втором — генератор заявок ,на третьем — сервер обработки заявок .

Сеть имеет 2 позиции (очередь — Queue, обслуженные заявки — Complited) и два перехода (генерировать заявку — Arrivals, передать заявку на обработку серверу — Server). Переходы имеют сложную иерархическую структуру, задаваемую на отдельных листах модели (с помощью соответствующего инструмента меню — Hierarchy).

Между переходом Arrivals и позицией Queue, а также между позицией Queue и переходом Server установлена дуплексная связь. Между переходом Server и по зицией Complited — односторонняя связь. Граф генератора заявок имеет 3 позиции (текущая заявка — Init, следующая заявка — Next, очередь — Queue из листа System) и 2 перехода (Init — определяет распределение поступления заявок по экспоненциальному закону с интенсивностью 100 заявок в единицу времени, Arrive — определяет поступление заявок в очередь).



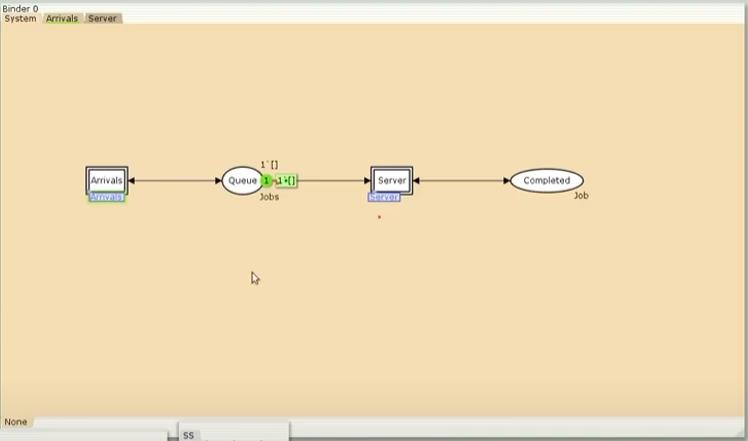
Граф процесса обработки заявок на сервере имеет 4 позиции (Busy — сервер занят, Idle — сервер в режиме ожидания, Queue и Complited из листа System) и 2 перехода (Start — начать обработку заявки, Stop — закончить обработку заявки).

1. Зададим декларации системы. Определим множества цветов системы (colorset): – фишки типа UNIT определяют моменты времени; – фишки типа INT определяют моменты поступления заявок в систему. – фишки типа JobType определяют 2 типа заявок — A и B; – кортеж Job имеет 2 поля: jobType определяет тип работы (соответственно име ет тип JobType, поле AT имеет тип INT и используется для хранения времени нахождения заявки в системе; – фишки Jobs — список заявок; – фишки типа ServerxJob — определяют состояние сервера, занятого обработкой заявок

Переменные модели: – proctime — определяет время обработки заявки; – job —

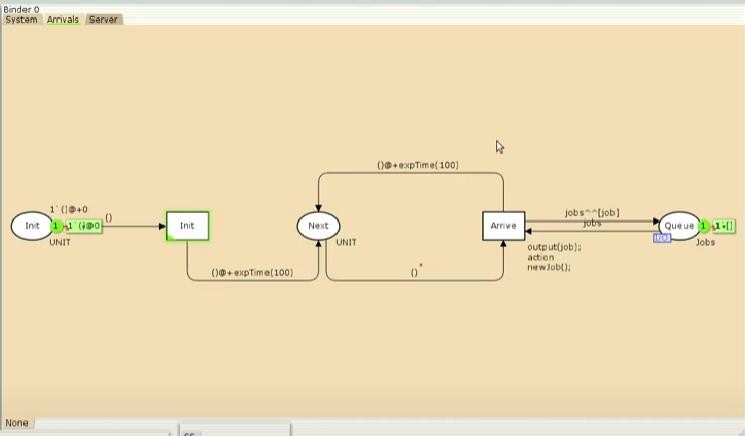
определяет тип заявки; – jobs — определяет поступление заявок в очередь. Определим функции системы:

* + функция expTime описывает генерацию целочисленных значений через интервалы времени, распределённые по экспоненциальному закону;
  + функция intTime преобразует текущее модельное время в целое число;
  + функция newJob возвращает значение из набора Job — случайный выбор типа заявки (A или B).

1. Зададим параметры модели на графах сети. 3.1. На листе System (рис. 11.4): – у позиции Queue множество цветов фишек — Jobs; начальная маркировка 1`[] определяет, что изначально очередь пуста. – у позиции Completed множество цветов фишек — Job.

3.2. На листе Arrivals (рис. 11.5): – у позиции Init: множество цветов фишек — UNIT; начальная маркировка 1`()@0 определяет, что поступление заявок в систему начинается с

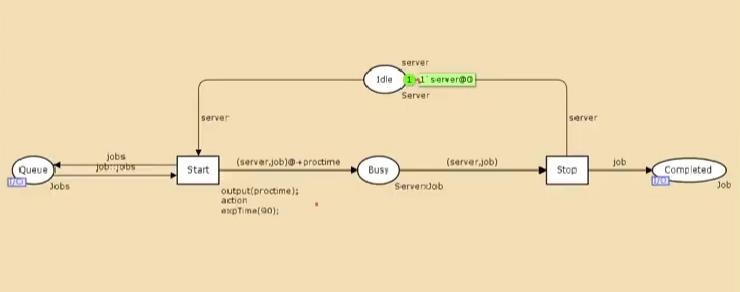
нулевого момента времени; – у позиции Next: множество цветов фишек — UNIT



* на дуге от позиции Init к переходу Init выражение () задаёт генерацию заявок; – на дуге от переходов Init и Arrive к позиции Next выражение ()@+expTime(100) задаёт экспоненциальное распределение времени между поступлениями заявок;
* на дуге от позиции Next к переходу Arrive выражение () задаёт перемещение фишки; – на дуге от перехода Arrive к позиции Queue выражение jobs^^[job] задает поступление заявки в очередь;
* на дуге от позиции Queue к переходу Arrive выражение jobs задаёт обратную связь. 3.3. На листе Server (рис. 11.6):
* у позиции Busy: множество цветов фишек — Server, начальное значение маркировки — 1`server@0 определяет, что изначально на сервере нет заявок на обслуживание;
* у позиции Idle: множество цветов фишек — ServerxJob;
* переход Start имеет сегмент кода output (proctime); action expTime(90); о

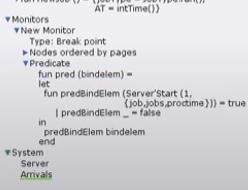
пределяющий, что время обслуживания заявки распределено по экспоненциальному закону со средним временем обработки в 90 единиц времени;

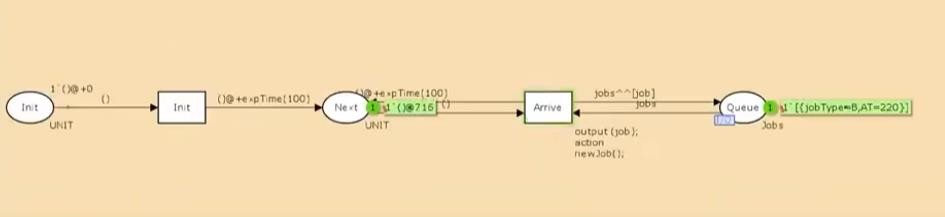
* на дуге от позиции Queue к переходу Start выражение job::jobs определяет, что сервер может начать обработку заявки, если в очереди есть хотя бы одна заявка;
* на дуге от перехода Start к позиции Busy выражение (server,job)@+proctime запускает функцию расчёта времени обработки заявки на сервере;
* на дуге от позиции Busy к переходу Stop выражение (server,job) говорит о завершении обработки заявки на сервере; – на дуге от перехода Stop к позиции Completed выражение job показывает, что заявка считается обслуженной; – выражение server на дугах от и к позиции Idle определяет изменение состояние сервера (обрабатывает заявки или ожидает);
* на дуге от перехода Start к позиции Queue выражение jobs задаёт обратную связь

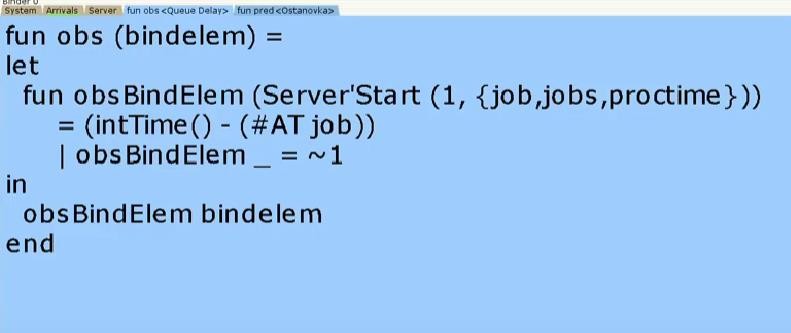
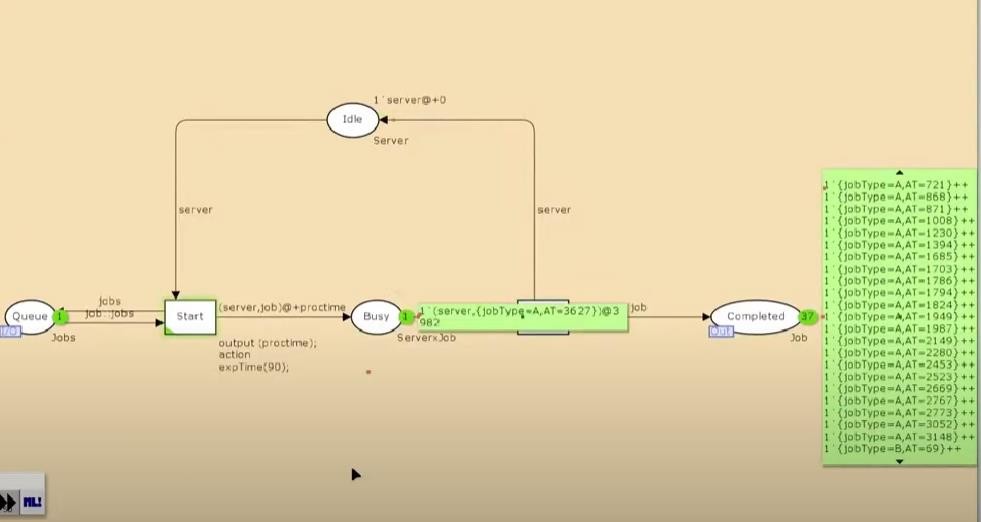


Мониторинг параметров моделируемой системы Цель: мониторинг параметров очереди системы M|M|1. Потребуется палитра Monitoring. Выбираем Break Point (точка останова) и устанавливаем её на переход Start. После этого в разделе меню Monitor появится новый подраздел, который назовём Ostanovka. В этом подразделе необходимо внести изме нения в функцию Predicate, которая будет выполняться при запуске монитора

Изначально, когда функция начинает работать, она возвращает значение true, в противном случае — false. В теле функции вызывается процедура predBindElem, которую определяем в предварительных декларациях. Зададим число шагов, через которое будем останавливать мониторинг. Для этого true заменим на Queue\_Delay.count()=200

необходимо определить конструкцию Queue\_Delay.count(). С помощью па литры Monitoring выбираем Data Call и устанавливаем на переходе Start. Появившийся в меню монитор называем Queue Delay (без подчеркивания).





## Заключение

В систему поступает поток заявок двух типов, распределённый по пуассоновскому закону. Заявки поступают в очередь сервера на обработку. Дисциплина очереди - FIFO. Если сервер находится в режиме ожидания (нет заявок на сервере), то заявка поступает на обработку сервером.