МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информационные технологии»

Лабораторная работа №3

**«Программное обеспечение для имитационного моделирования: часть 3»**

по дисциплине

Математическое и имитационное моделирование

Выполнил: студент гр. БЭИ2202

Кулешов А. С.

Проверил: Степанова И.В.

Москва, 2024 г

1. Цель работы

Добавление дополнительных факторов, сравнение прогонов модели. Работа происходит на модели, построенной на предыдущих занятиях

1. Выполнение

Добавлю новый параметр – DiscardTime, после чего сделаю переход из User в новое состояние – WantsToBuy, переход из которого обратно с состояние User будет происходить через 2 секунды.

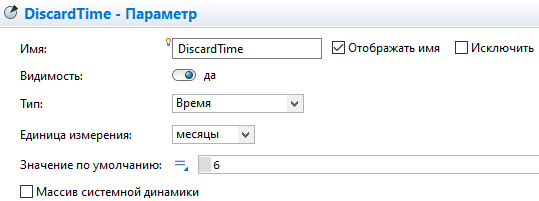


Рисунок 1 – параметр DiscardTime

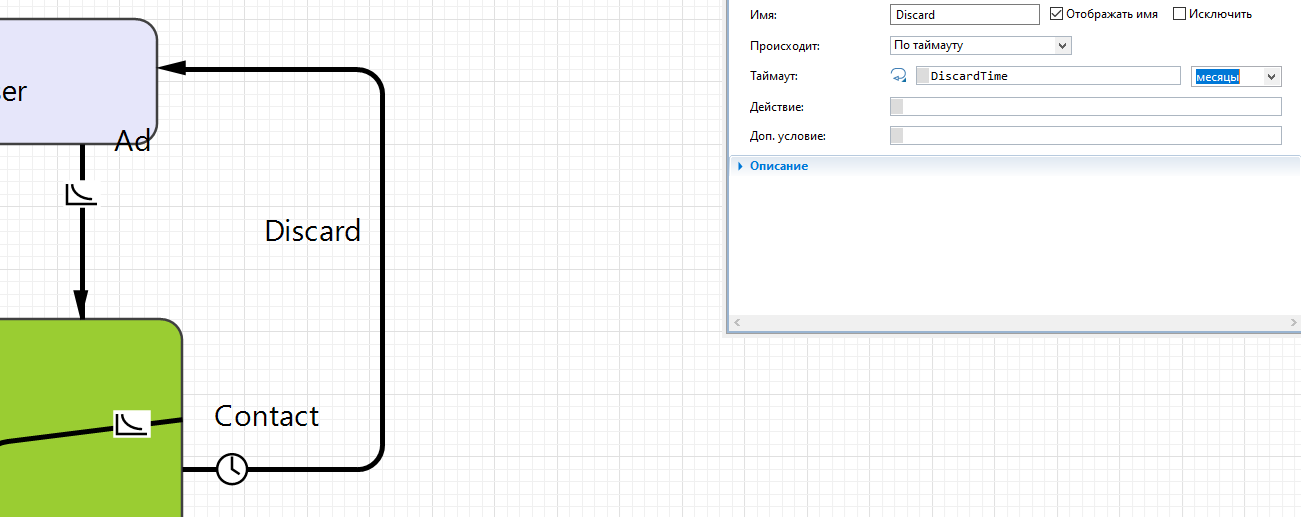


Рисунок 2 – переход Discard

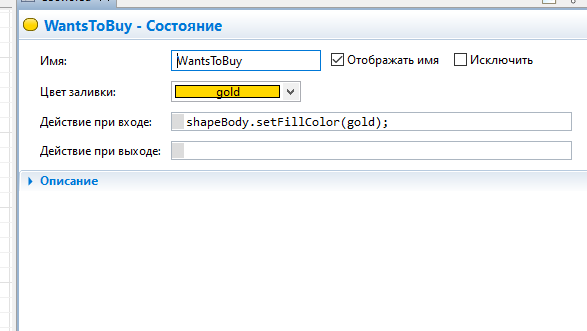


Рисунок 3 – состояние WantsToBuy

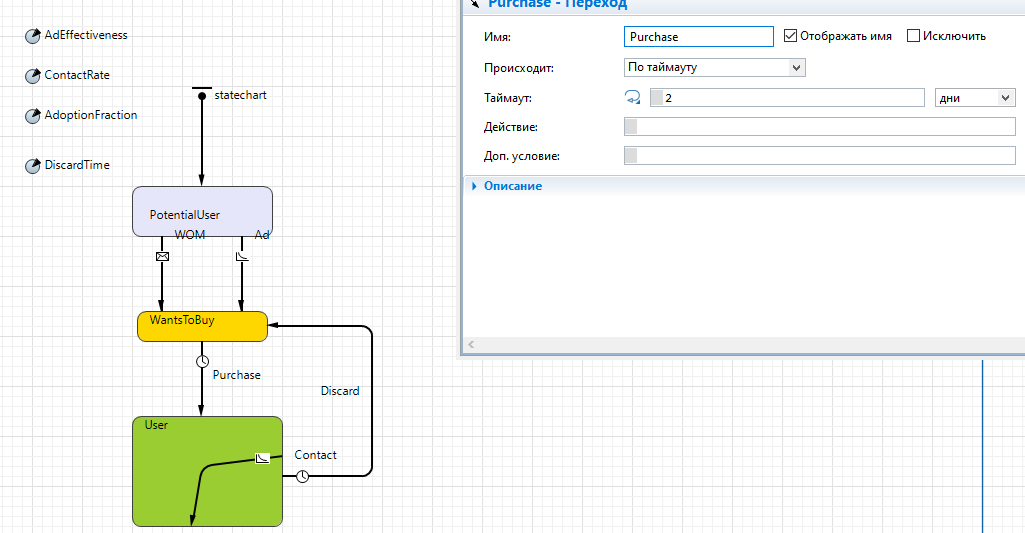


Рисунок 4 – переход Purchase

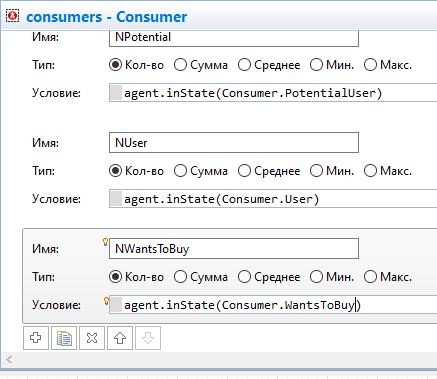


Рисунок 5 – функция для счёта числа желающих

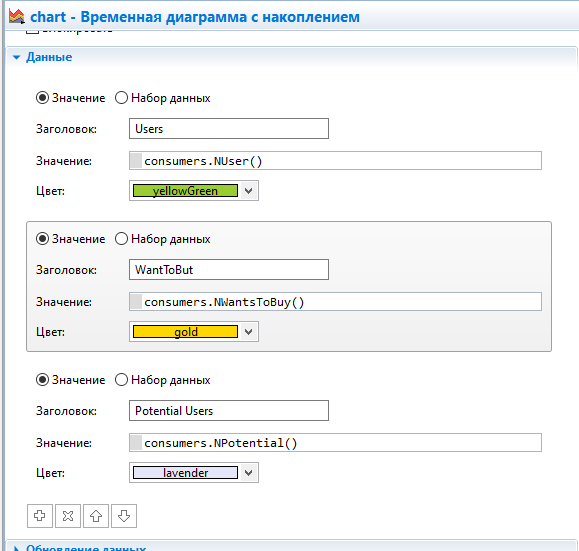


Рисунок 6 – отображение числа желающих на графике

Теперь реализую отказ от товара в случае слишком долгого ожидания

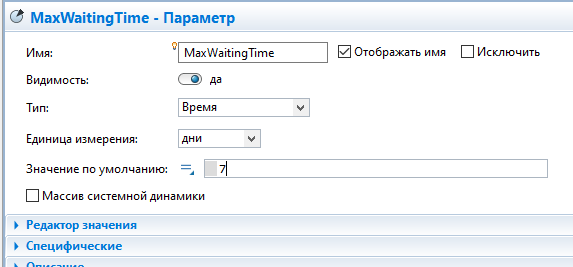


Рисунок 7 – параметр максимального времени ожидания товара

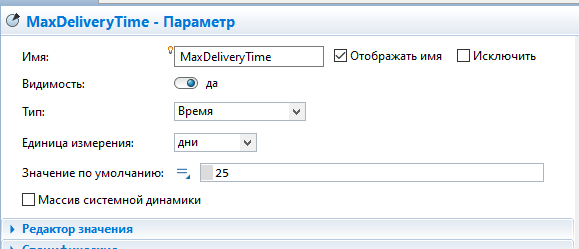


Рисунок 8 – параметр максимального времени доставки

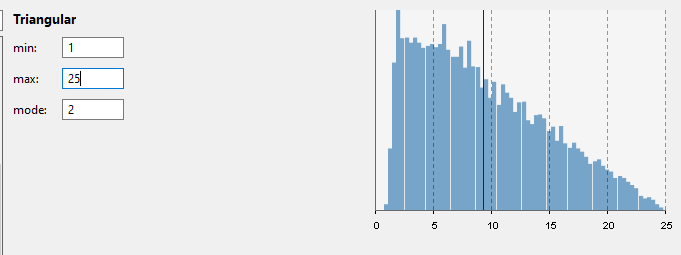


Рисунок 9 – распределение времени доставки

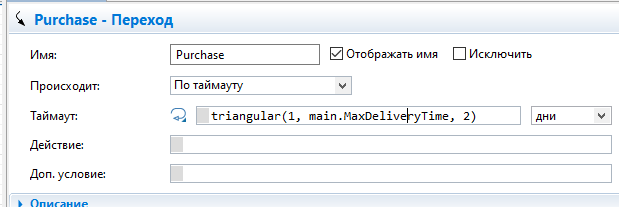


Рисунок 10 – привязывание распределения к параметру

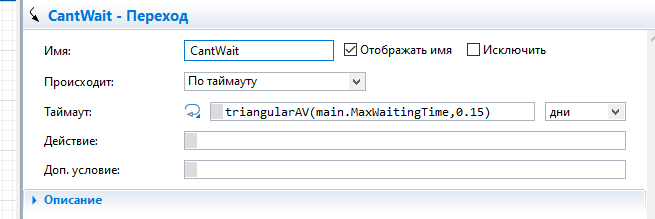


Рисунок 11 – переход в случае слишком долгого ожидания

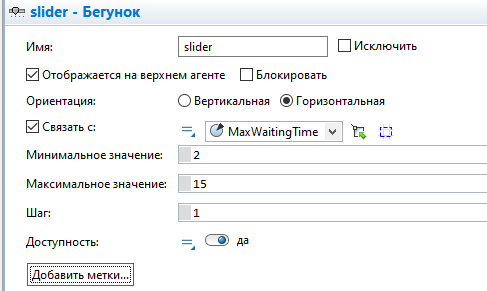


Рисунок 12 – бегунок для изменения время ожидания

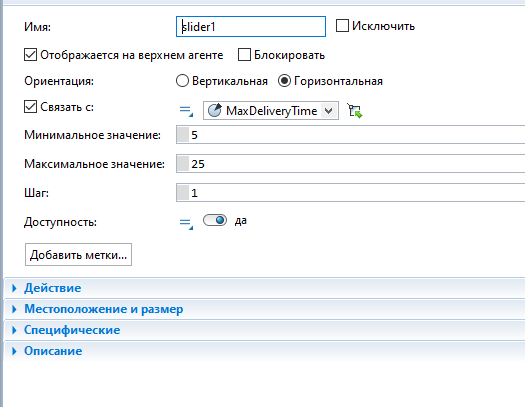


Рисунок 13 – бегунок для изменения времени доставки

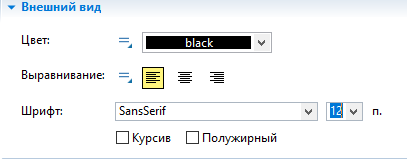


Рисунок 14 – стиль текста

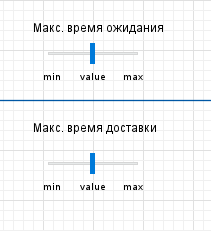


Рисунок 15 – бегунки с текстом

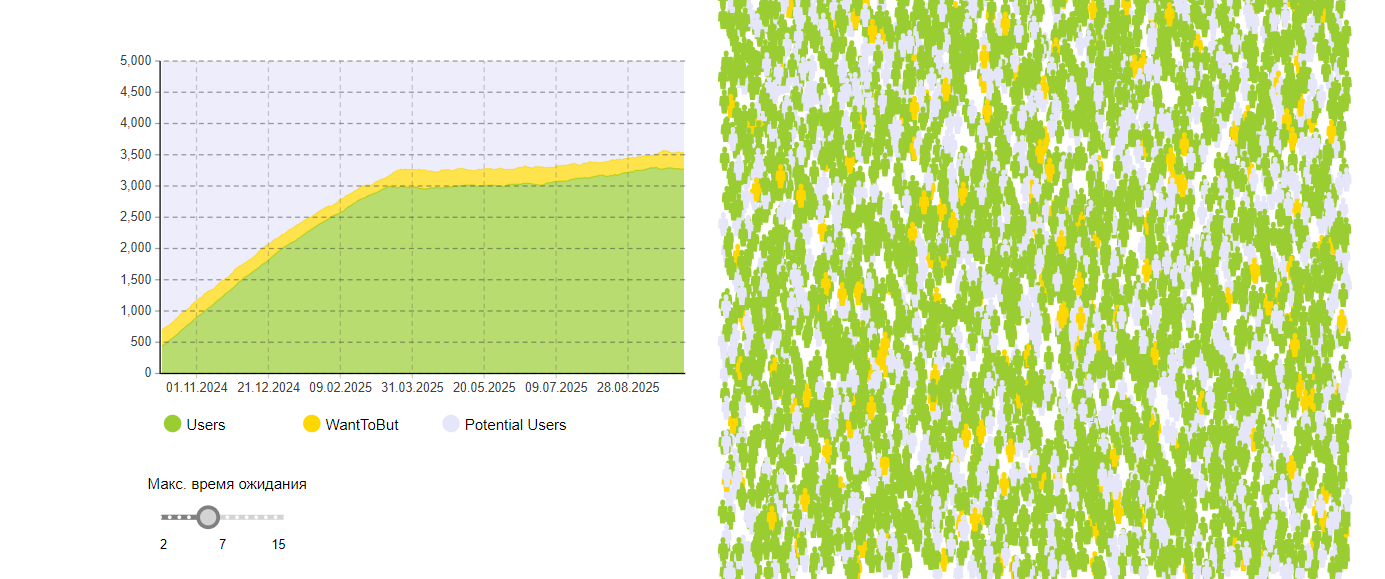


Рисунок 16 – симуляция запущена

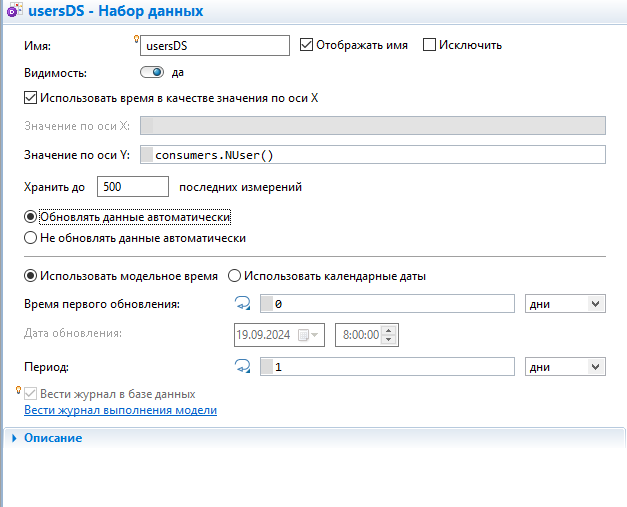


Рисунок 17 – создание набора данных

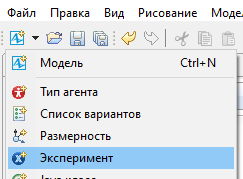


Рисунок 18 – создание эксперимента

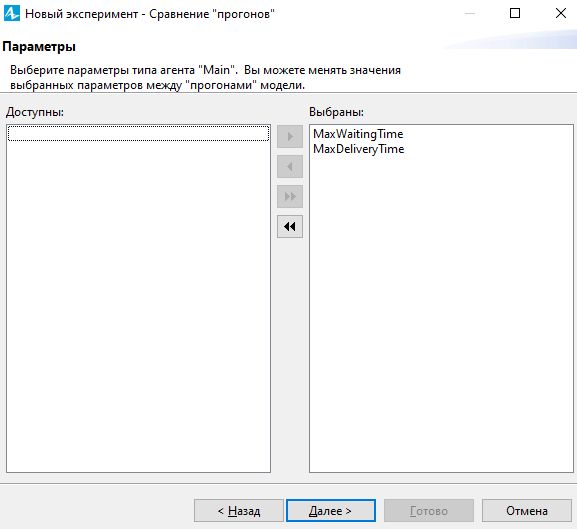


Рисунок 19 – добавление параметров в эксперимент

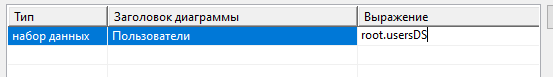


Рисунок 20 – добавление названия

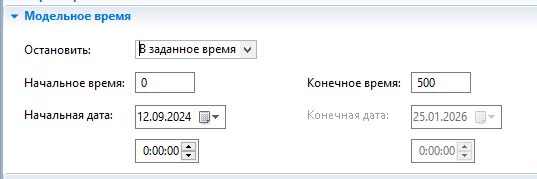


Рисунок 21 – ограничение времени до 500 дней

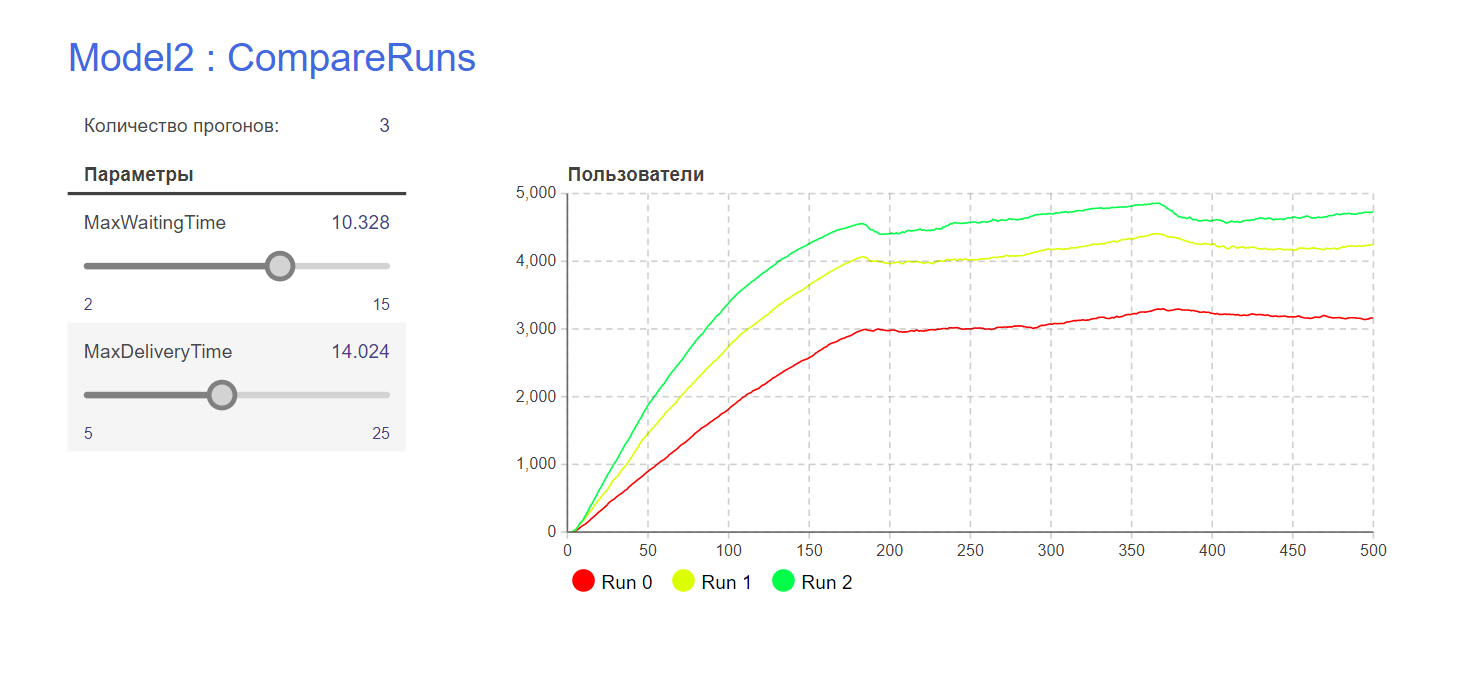


Рисунок 22 – запуск сравнений

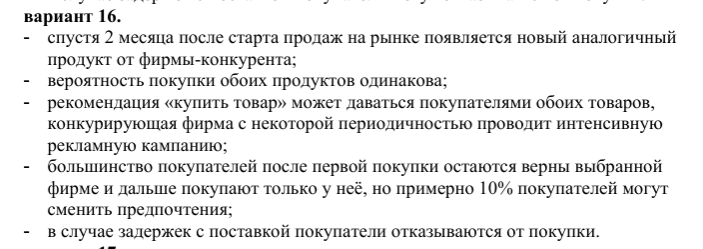


Рисунок 23 – условие индивидуального задания

Для реализации индивидуального задания я добавил новую переменную для агента – loyal, её значение означает компанию к которой клиент лоялен. Также были созданы новые состояния user2, wantstobuy2, работающие аналогично изначальным, в переходы ко второй компании было добавлено условие времени. Логика лояльности находится в блоках user, и написана в короткое количество Java строк кода

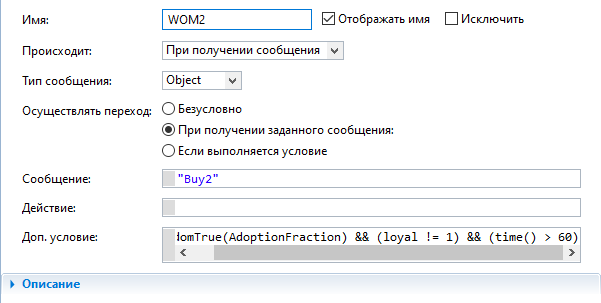


Рисунок 24 – условие времени для входа во вторую компанию

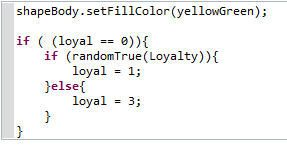


Рисунок 25 – код отвечающий за лояльность

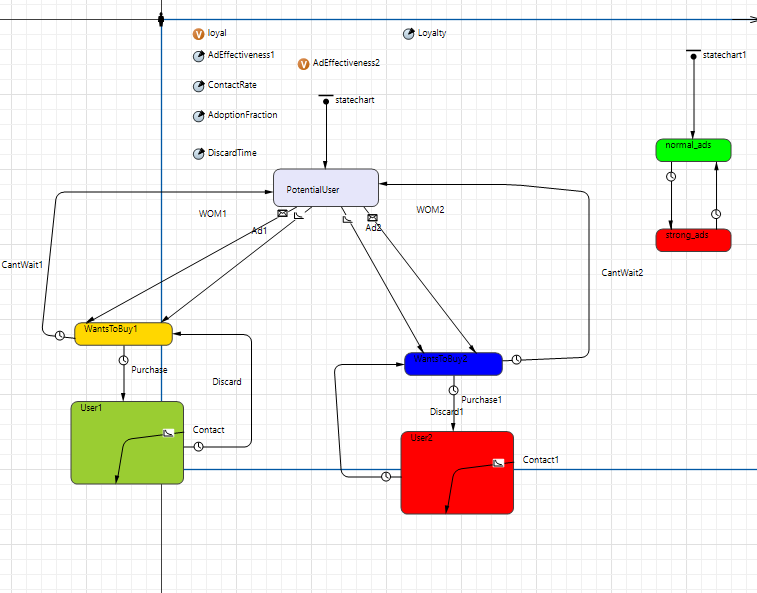


Рисунок 26 – итоговая диаграмма

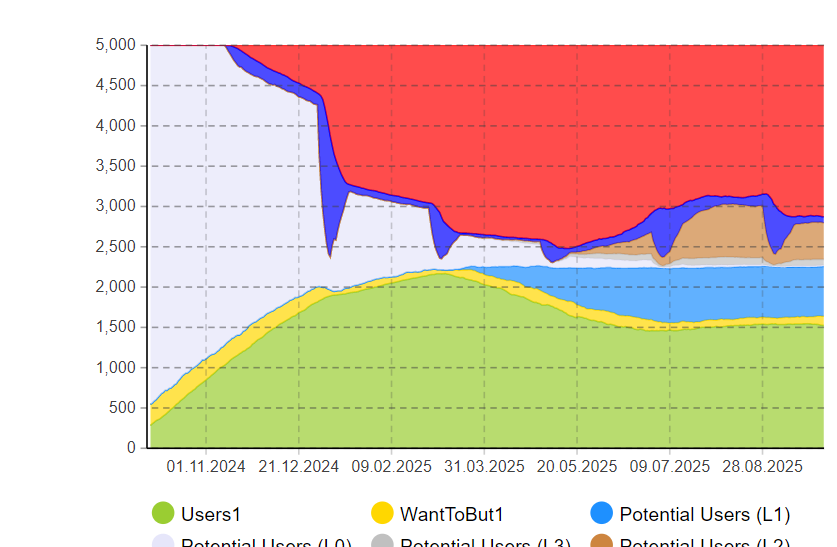


Рисунок 27 – начало симуляции

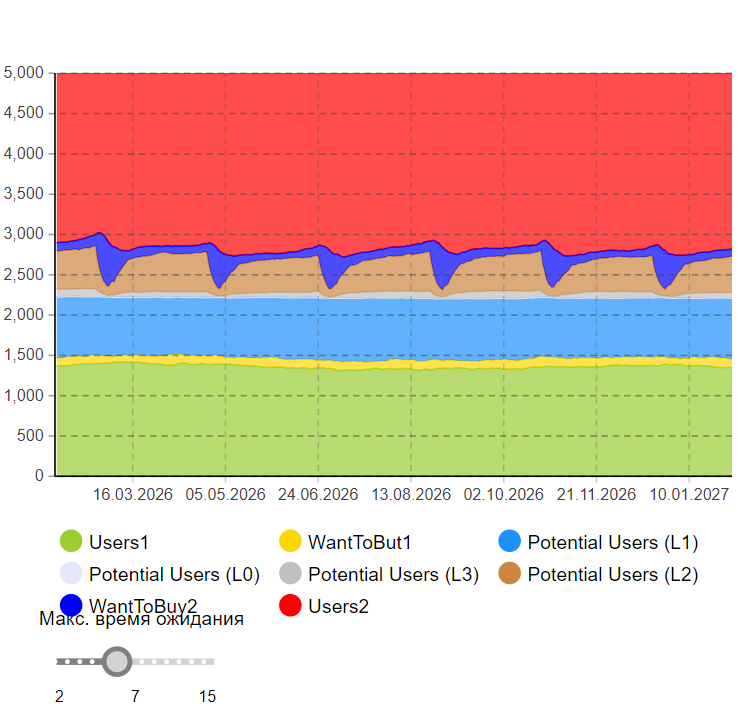


Рисунок 28 – стабилизация симуляции

Важно отметить «волновое» движение показателя users2, в действительности, если условие перехода Discard оставить в соответствии с методическими материалами, то в таком случае «волн» не будет, т.к. процесс прихода новых клиентов синхронизируется с таким же «волновым» поведением ухода клиентов. Для того, чтобы получить те самые «волны» следует либо сделать переход Discard случайным, либо чтобы время DiscardTime и время на цикл реклам не давали в результате деления число близкое к целому.

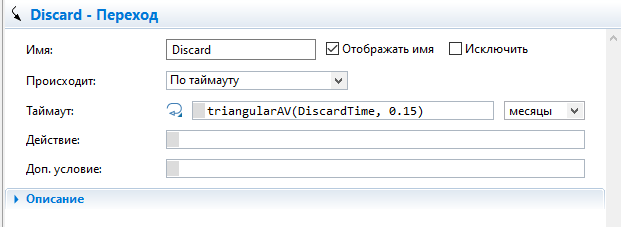


Рисунок 29 – изменение функционала перехода Discard

1. Вывод

После выполнения данной лабораторной работы мы научились создавать и анализировать более сложные математические модели