

УДК 004.94

DOI 10.21661/r-561539

*Артемьев Богдан Олегович*

## **ДИСКРЕТНО-СОБЫТИЙНОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССА ЗАПИСИ ПАЦИЕНТА НА ПРИЁМ В ПОЛИКЛИНИКЕ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ANYLOGIC**

***Аннотация:** в статье рассматривается процесс регистрации пациента в поликлинике, с использованием ERP-системы, разработанной авторами. Для имитации данного процесса используется программа AnyLogic. Создается модель Clinic, в которой заявки (пациенты) генерируются блоком Source с интенсивностью 10 в час. Результаты моделирования могут быть использованы для оптимизации процесса регистрации и улучшения качества обслуживания пациентов в поликлиниках.*

***Ключевые слова:** ERP-система, AnyLogic, интеллектуальные системы, дискретно-событийное моделирование.*

В наше время современные информационные системы (ИС) играют такую же значимую роль, как когда-то появление машин. Они лежат в основе развития экономики государств и всего мира и его важнейших отраслей. Базисом ИС предприятий и организаций выступают так называемые системы планирования ресурсов предприятия (ERP-системы). Их грамотное внедрение значительно улучшает управляемость организацией и увеличивает эффективность ее работы.

Цель настоящей работы – промоделировать процесс регистрации пациента в поликлинике с использованием разработанной ERP-системы. Допускаем, что поликлиника работает только в будние дни с 8:00 до 20:00, поток пациентов меняется в зависимости от времени суток; время обслуживания одного пациента составляет пять минут.

Сначала необходимо создать новую модель. Для этого в программе AnyLogic нажимаем на кнопку «Файл» в левом верхнем углу. Далее из выпадающего меню

выбираем пункт «Создать» и выбираем единственный доступный вариант «Модель» (рис. 1).

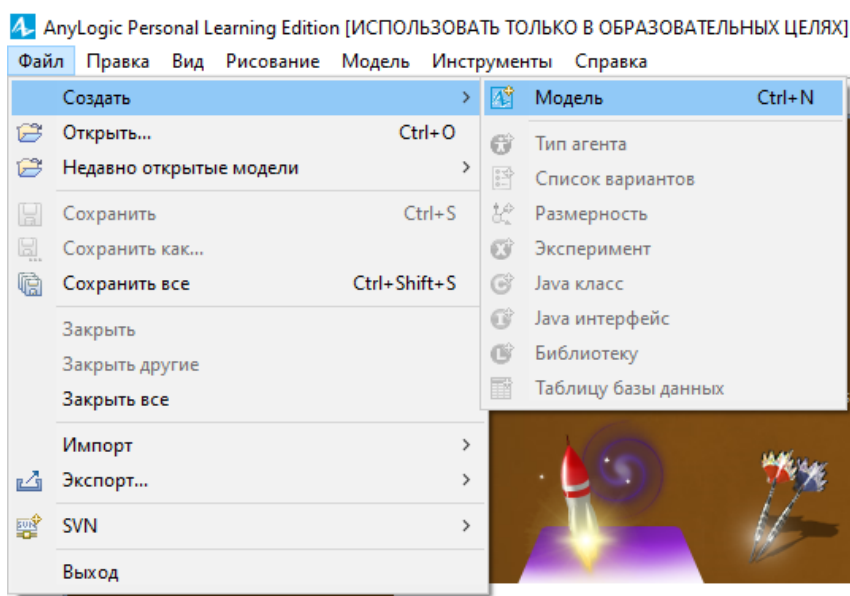


Рис. 1. Создание модели в среде AnyLogic

В появившемся диалоговом окне необходимо задать имя модели «Clinic» и единицы модельного времени – минуты, после чего нажать на кнопку «Готово» (рис. 2).

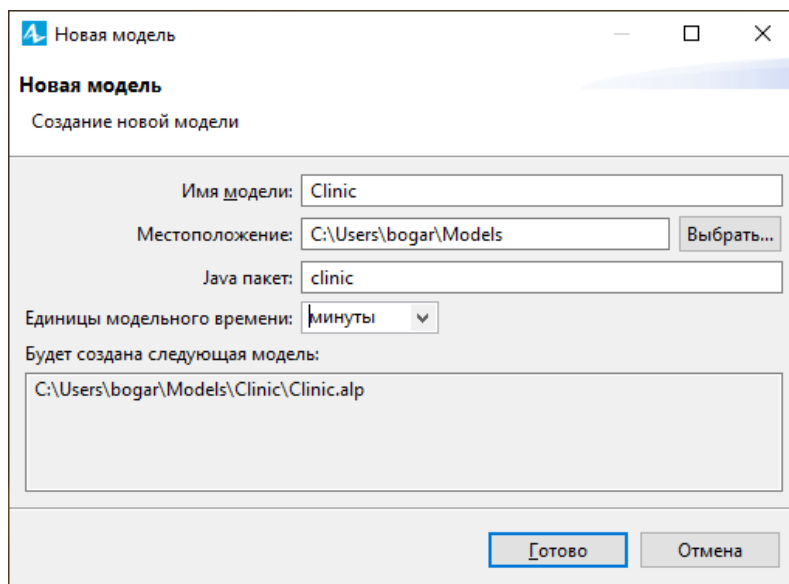


Рис. 2. Свойства новой модели

Рабочая область программы разделена на три части. В левой части располагается «Палитра» – набор библиотек и содержащихся в них блоков. Центральная часть программы (обозначается как Main) представляет собой рабочее поле для

моделирования систем. Моделирование осуществляется переносом необходимых блоков из палитры, их настройкой и заданием связей между ними. Наконец, правая часть программы – «Свойства». Здесь осуществляется настройка блоков в модели. В целом, базовый интерфейс программы схож с интерфейсом программного продукта Microsoft Visio.

В процессе моделирования будут использоваться блоки из Библиотеки моделирования процессов. Данная библиотека как раз и предназначена для дискретно-событийного или «процессного» моделирования. Процессы задаются в форме потоковых диаграмм (блок-схем) – графическом представлении, принятом во многих областях: производстве, бизнес-процессах, центрах обработки звонков, логистике, здравоохранении и т. д.

В первую очередь перетаскиваем на рабочее поле из библиотеки моделирования процессов блок Source, который обеспечивает появление заявок (в данном случае – пациентов) в модели. Задаём имя блоку – «patient», в параметре «Прибывают согласно» выбираем пункт «Интенсивности», а параметр «Интенсивность прибытия» задаём равным «10 в час», что означает, что в поликлинику будут приходить в среднем десять пациентов в час (рис. 3).

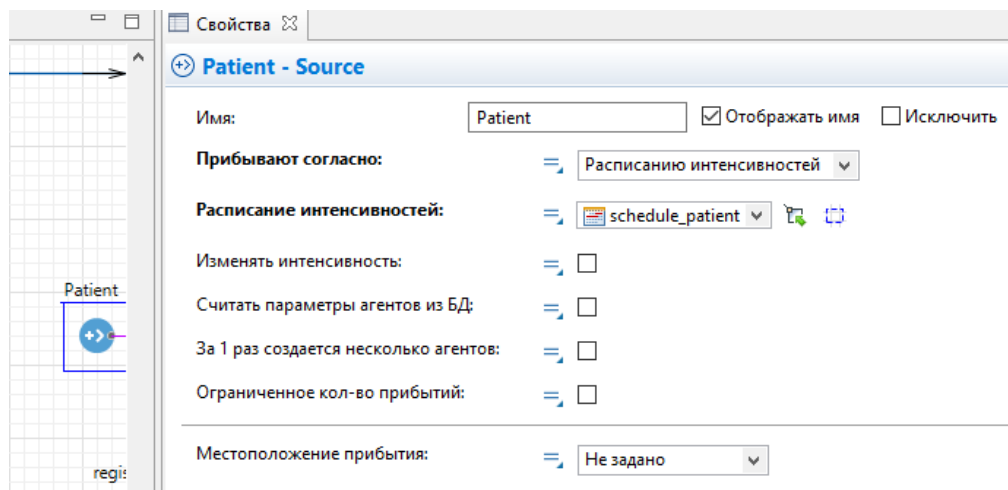


Рис. 3. Блок Source (patient) и его свойства

Следующий шаг – моделирование очереди пациентов. Для этого из библиотеки моделирования процессов переносим на рабочую область блок Queue, причём желательно перенести его так, чтобы между блоком Queue и блоком Source (patient) сразу образовалась связь. Запись на приём к врачу проходит по линиям

связей между блоками. В параметрах блока Queue необходимо отметить галочкой пункт «Максимальная вместимость» (длина очереди не ограничена); остальные параметры оставляем по умолчанию без внесения изменений (рис. 4).

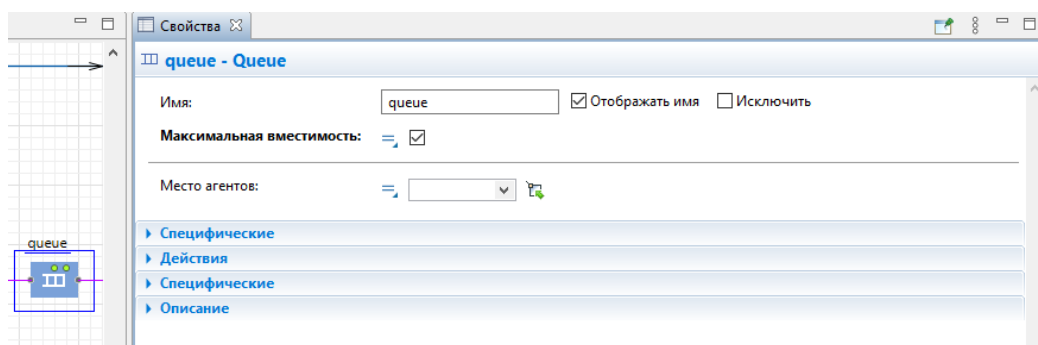


Рис. 4. Блок Queue и его параметры

В качестве ресурса для процедуры записи на приём в нашей модели будет использоваться Регистратура. Для создания ресурса в модели Библиотека моделирования процессов системы AnyLogic содержит блок ResourcePool. Особенностью данного блока является то, что он не имеет связей с другими блоками модели. Его необходимо перетащить на рабочее поле модели и задать параметры: имя – «registrationPoint», тип ресурса – «Статический», остальные параметры блока пока изменять не следует.

Для моделирования процесса записи пациента на приём будем использовать блок Service. В его свойствах задаём имя «registration», вместимость очереди – 1 и время задержки, которое задаётся треугольным распределением (triangular) из трёх параметров: минимального значения (в нашем случае – 1 минута), среднего значения (3 минуты) и максимального значения (5 минут). В пункте «Наборы ресурсов» выбираем регистратуру registrationPoint (рис. 5).

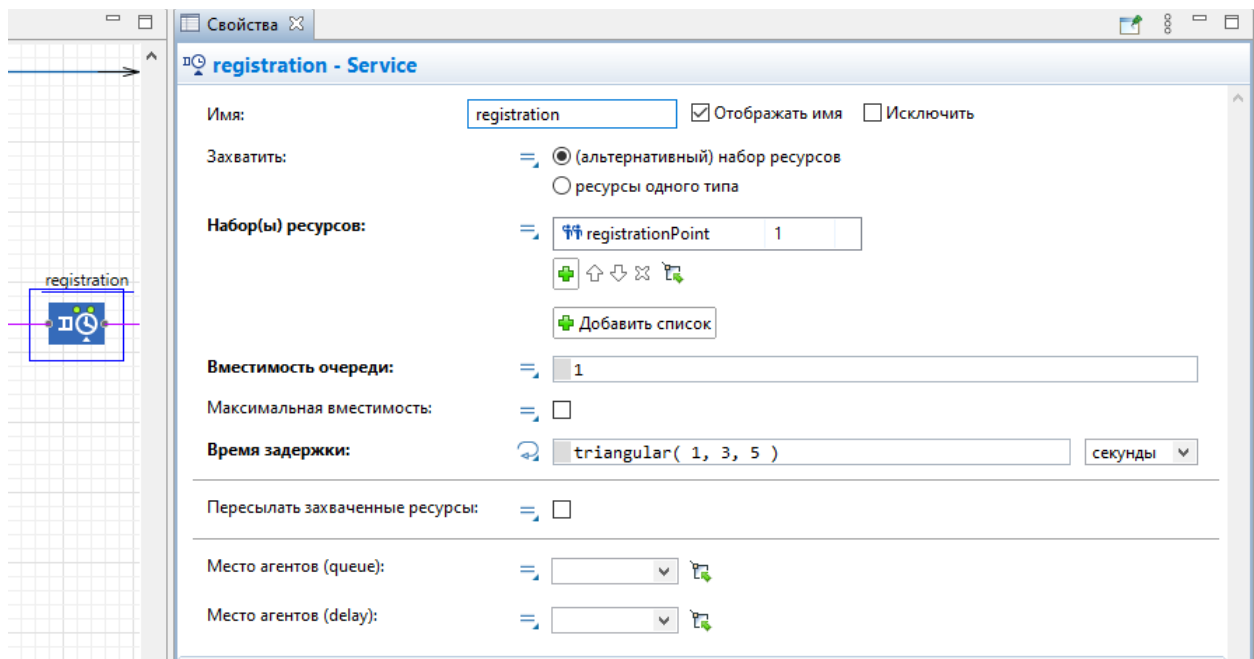


Рис. 5. Блок Service (registration) и его параметры

После окончания процесса записи на приём, пациент уходит из регистратуры. Для задания конечной точки процесса используется объект Sink, находящийся в библиотеке моделирования процессов. Блок размещается после блока Service, в качестве свойства блока задаётся только имя «exit» – выход из клиники или регистратуры (рис. 6).

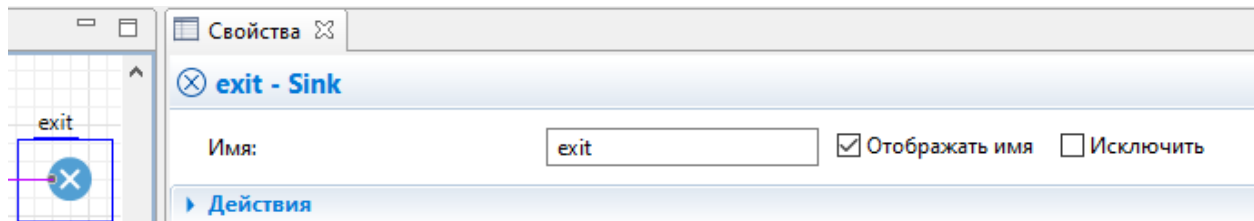


Рис. 6. Блок sink (exit)

Ещё одним важным этапом в построении модели является задание расписаний. Любые периодические изменения в модели задаются блоком «Расписание», находящемся в библиотеке моделирования процессов. Так же, как и блок ResourcePool, данный блок не объединяется и не имеет связей с другими блоками.

Необходимо перетащить блок «Расписание» на рабочее поле и в его свойствах задать имя «schedule\_patient». В данном расписании размещена информация об интенсивности прибытия пациентов в клинику, соответственно в парамет-

рах расписания в разделе «Данные» выбираем тип «Интенсивность», длительность «Неделя» (расписание задаётся на неделю), расписание задаёт «Интервалы (Начало, Конец)». Значение по умолчанию оставляем равным 0, это будет означать, что в неуказанные в расписании часы клиника не работает. В таблице «Повторять расписание еженедельно» необходимо задать временные интервалы со значениями, соответствующими интенсивности прихода пациентов в клинику. Допустим, что с 8 до 13 часов средняя интенсивность будет составлять 10 человек в час, с 13 до 18 часов – 20 человек в час, с 18 до 20 часов – 5 человек в час (рис. 7).

**schedule\_patient - Расписание**

Имя:  ☒ Отображать имя ☐ Исключить

Видимость: ☒ да

**Данные**

Тип:

Единица измерения:

Расписание задает: ☒ Интервалы (Начало, Конец) ☐ Моменты времени

Длительность: ☒ Неделя ☐ Дни/Недели ☐ Другая (нет привязки к календарю)

Значение по умолчанию:

Повторять расписание еженедельно

вс	пн	вт	ср	чт	пт	сб	Нача...	Конец	Значение
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	8:00	13:00	10.0
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	13:00	18:00	20.0
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	18:00	20:00	5.0

Рис. 7. Расписание интенсивности прибытия пациентов

Расписание работы регистратуры (schedule\_reg) задаёт время её работы (с 8 до 20 часов) и количество работающих окон, в нашем случае – одно. Тип расписания – Целое, значение по умолчанию – 0 (рис. 8).

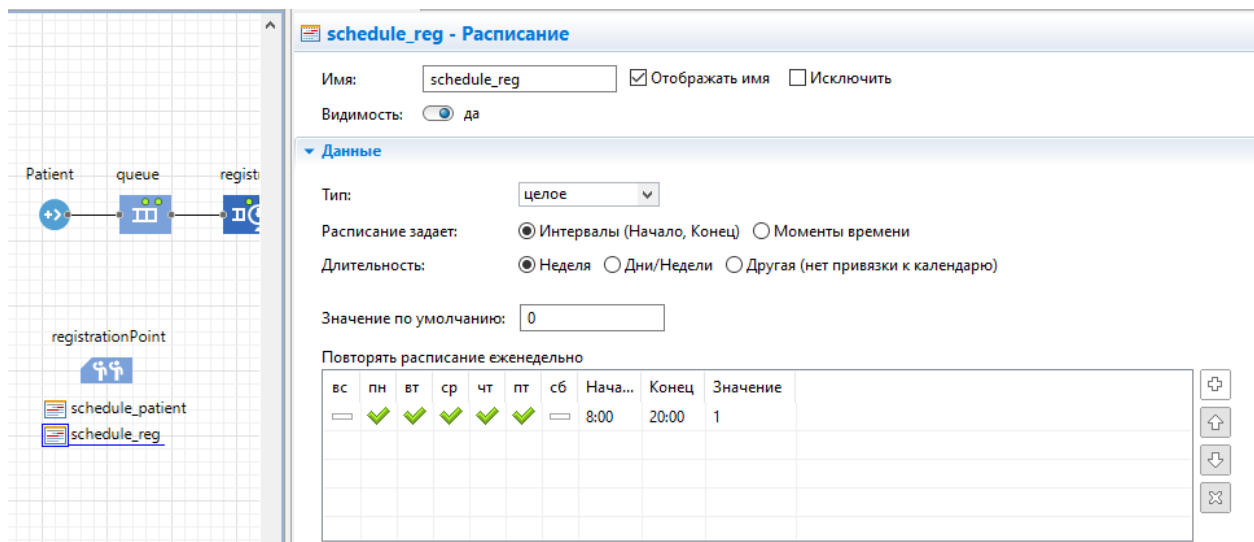


Рис. 8. Расписание работы регистратуры

Последним шагом в проектировании модели является привязка расписания прихода пациентов в поликлинику к объекту source и расписания работы регистратуры к объекту ResourcePool. В объекте source (Patient) в параметре Расписание интенсивностей указываем расписание «schedule\_patient» (см. рис. 3), а в объекте ResourcePool в параметре Расписание указываем таковое под названием schedule\_reg (рис. 9).

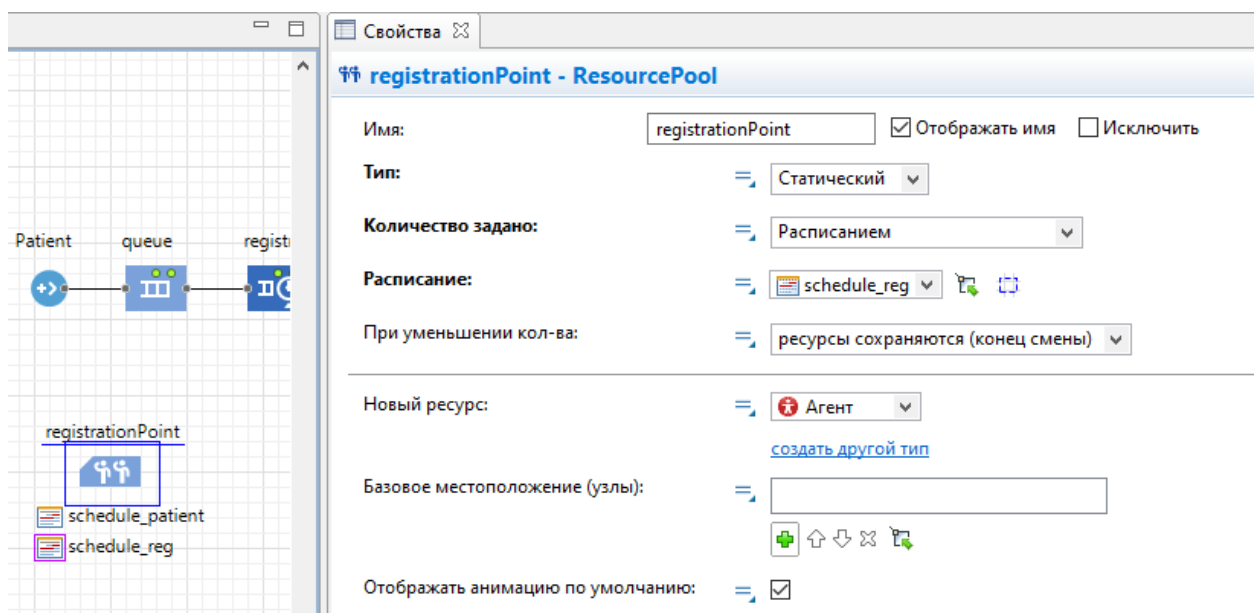


Рис. 9. Параметры блока ResourcePool (registrationPoint)

Окончательный вид модели представлен на рисунке 10. Стоит заметить, что данную модель является базовой, и её можно улучшать различными способами.

К примеру, можно добавить несколько окон регистратуры, в которые будет распределяться очередь. Для этого необходимо применить блок для разделения selectOutput. Среди других идей для улучшения: организация работы резервного окна регистратуры, организация ухода пациентов, не оформивших талон на приём, динамические графики и т. д.

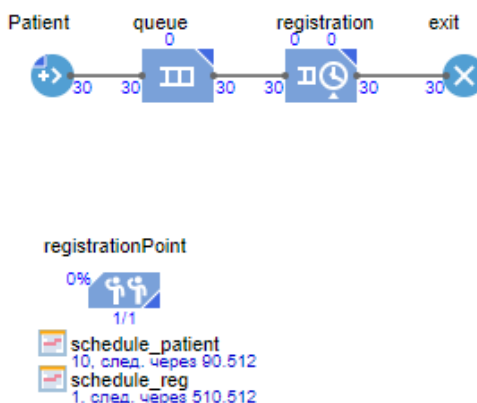


Рис. 10. Окончательный вид модели

### Список литературы

1. AnyLogic: имитационное моделирование для бизнеса [Электронный ресурс]. – Режим доступа: <https://www.anylogic.ru/> (дата обращения: 23.12.2023).
2. Документация AnyLogic [Электронный ресурс]. – Режим доступа: <https://anylogic.help/ru/> (дата обращения: 23.12.2023).
3. Григорьев И. AnyLogic за три дня. Практическое пособие по имитационному моделированию / И. Григорьев. – 2020. – 273 с.
4. Лимановская О.В. Имитационное моделирование в AnyLogic 7. В 2 ч. Ч. 1 и 2: учебное пособие / О.В. Лимановская. – Екатеринбург: Изд-во Урал. ун-та, 2017. – 152 с. EDN ZOQSHJ

---

**Артемов Богдан Олегович** – бакалавр, магистрант ФГБОУ ВО «Калужский государственный университет им. К.Э. Циолковского», Россия, Калуга.

---