

УДК 372.851

ПРОЕКТНАЯ ДЕЯТЕЛЬНОСТЬ СТАРШЕКЛАССНИКОВ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Аннотация. В данной статье рассмотрены возможности использования сред имитационного моделирования в проектной деятельности старшеклассников при обучении информатике. Предлагается разработанная практическая работа в среде AnyLogic и продолжение этой работы в виде проектной задачи. Таким образом, показано, как интегрировать работу по изучению новой среды моделирования и выполнение заданий практической работы, а затем полученную заготовку использовать для проектной работы. Приводится обзор сред имитационного моделирования, доступных для освоения школьниками.

Ключевые слова: имитационное моделирование, проектная деятельность, обучение информатике, AnyLogic, старшая школа.

Введение. Актуальность расширения тематики имитационного моделирования в старшей школе

Имитационное моделирование как частный случай компьютерного моделирования в школьной информатике изучается и используется более 20 лет в виде виртуальных лабораторий и различных образовательных платформ, внедряемых в школах, а также специально созданных



Наталья Ивановна Волынчук,
кандидат педагогических наук,
заведующий Центром математического и
естественно-научного общего
образования
ФГБНУ «Институт содержания
и методов обучения
им. В. С. Леднева»,
г. Москва, Россия
E-mail: Volynchuk@instrao.ru

Как цитировать статью: Волынчук Н. И., Магомедова А. М. Проектная деятельность старшеклассников с использованием среды имитационного моделирования // Образ действия. 2026. Вып. 3 «Естественно-научное образование (лучшие практики)». С. 104–121.

сред моделирования для выполнения практических работ [1]. Методы имитационного моделирования реализуются в виде программного продукта самыми разными компаниями и образовательными учреждениями высшего образования. Существуют отечественные среды имитационного моделирования, например Actor Pilgrim (МЭИ), Visual Imitak, Imitak Project (ГУУ), GPSS STUDIO (ООО «Элина-Компьютер»), AnyDynamics и AnyLogic (доступные без ограничений образовательные версии), которые можно использовать для крупных проектов и даже школьных стартапов [3]. Есть свободно распространяемая среда NetLogo, которая отличается простотой освоения, но ограничена одним подходом моделирования. Среда имитационного моделирования Powersim ориентирована на системную динамику и имеет ограниченные возможности. Программы Arena и GPSS требуют специальной подготовки и сложны для использования в школе. Потребность в изучении сред имитационного моделирования в старших профильных классах и классах специализации (ИТ, инженерных) возрастает. В таких средах можно без программирования решать задачи моделирования сложных динамических систем, разработки стратегий и выбора оптимальных решений, например, массового обслуживания, распределения финансовых потоков, ресурсов, транспортных потоков и пр. Для профиля «Робототехника» Всероссийской олимпиады школьников по информатике умение работать с имитационными моделями необходимо для виртуального прототипирования и тестирования роботов в симуляторах (Gazebo, CoppeliaSim) до создания физического образца.

Востребованность сред имитационного моделирования для учебных и исследовательских задач и проектной деятельности требует более детального представления умений при-



**Асма Магомедовна
Магомедова,**
*Институт математики
и информатики,
Московский педагогический
государственный университет,
г. Москва, Россия
E-mail: asma.magomedova.04@
mail.ru*

менять среды имитационного моделирования в нормативных документах для их применения не только в практической, но и проектной деятельности старшеклассников. Сейчас они представлены в предметных результатах ФГОС СОО в следующем виде:

«– умение разрабатывать и исследовать несложные имитационные модели, в частности агентные, дискретно-событийные, системно-динамические в выбранной среде моделирования;

– умение планировать и осуществлять учебные исследования и эксперименты с моделями, для решения задач прогнозирования, поиска и оптимизации решений» [4].

Детализация методов имитационного моделирования в предметных результатах по информатике в ФГОС СОО (агентное, системно-динамическое, дискретно-событийное) проведена с учетом достижений современной науки.

От практической работы до проектного исследования. Способы освоения и использования среды имитационного моделирования

Имитационное моделирование является методом исследования, который основан на создании моделей, предназначенных для воспроизведения поведения реальных систем во времени.

Основными методами моделирования (видами моделей) являются:

- системная динамика — ориентирована на анализ взаимосвязей и изменений в системе;
- дискретно-событийный метод — применяется для систем с событиями во времени;
- агентный метод — рассматривает систему как совокупность взаимодействующих элементов.

Для использования моделирования в обучении старшеклассников необходимо выбрать подходящую программную среду. Ключевыми критериями являются доступность, наглядность, функциональность и возможность масштабирования.

Сравнение программных средств показало, что именно среда AnyLogic сочетает реализацию всех видов моделирования, развитые средства визуализации и возможность создания и исследования сложных моделей.

Подбор заданий для моделирования динамических процессов для старшеклассников должен соответствовать их познавательным возможностям. Самое главное, что вначале учащиеся должны освоить приемы работы в среде имитационного моделирования. Поэтому учебное задание формулируется в практической работе на основе всем известных процессов и выполняется пошагово с визуальной поддержкой. Например, работа небольшого магазина, кафе, фитнес-центра и пр. После выполнения пошаговой практической работы, когда первоначальные навыки работы в среде сформированы, каждому выдается индивидуальное проектное задание как продолже-

ние выполненного всеми задания. Но теперь каждый учащийся реализует собственную проектную задачу. Например, смоделировать максимальную проходимость фитнес-центра в пиковые часы загрузки, оптимизировать посадочные места в кафе, сделать безопасными проходы в магазин для маломобильных покупателей, расширить количество предоставляемых банковских услуг и т. д.

Рассмотрим в качестве примера работу небольшого отделения банка. В банке процесс обслуживания чаще всего выглядит следующим образом: посетитель входит в отделение, если это необходимо, попадает в зону ожидания, далее переходит к освободившемуся сотруднику банка, который готов ему помочь, после получения услуги покидает отделение банка.

Время, которое клиенты проводят в банке, зависит от некоторых факторов: например, от текущей загруженности отделения, количества работающих сотрудников и времени выполнения различных операций. Каждый из посетителей по очереди проходит все этапы обслуживания, но на некоторых из них возможно образование очереди, которая влияет на общее время получения услуги.

В процессе реализации модели необходимо определить число посетителей, которое может быть обслужено за заданный промежуток времени, узнать среднее и максимальное время пребывания клиента в отделении банка [2].

Практическая работа. Создание модели работы отделения банка

Этап 1. Создание новой модели

1.1. В основном меню необходимо выбрать Файл, далее отметить пункт Создать (рис. 1).

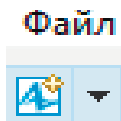


Рисунок 1. Создание нового файла

1.2. Из выпадающего списка типов документов выбираем Модель (рис. 2).

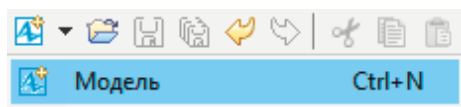


Рисунок 2. Выбор типа документа

1.3. На экране появляется диалоговое окно Создание модели. В это окно надо ввести следующие данные: имя проекта BankServiceModel, название должно быть уникальным в рамках рабочей области, не содержать пробелов.

лов и специальных символов. В качестве единицы модельного времени оптимально будет выбрать Минуты (рис. 3).

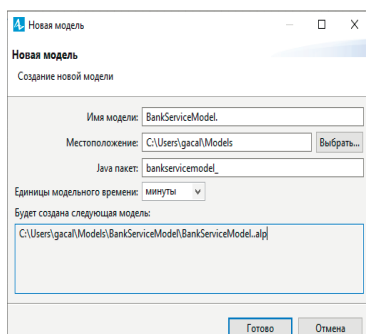


Рисунок 3. Диалоговое окно создания модели

1.4. После того как все параметры заполнены, нажимаем кнопку Готово, расположенную в нижней части диалогового окна. Среда разработки AnyLogic автоматически сгенерирует пустую модель, которая содержит основной компонент Main. Теперь можно перейти к редактированию модели.

Этап 2. Построение области имитации — банковского отделения

Для того чтобы имитация была наглядной и позволяла отслеживать перемещение клиентов в реальном времени, необходимо создать графическое представление банковского отделения. Это делается в два подэтапа: сначала загружается фоновое изображение (план помещения), а затем поверх него с помощью специальных инструментов рисуются стены и другие элементы, ограничивающие перемещение пешеходов. Без этих ограничений виртуальные клиенты могли бы проходить сквозь стойки и стены, что нарушило бы реалистичность модели.

2.1. В правой части окна AnyLogic расположена палитра компонентов, сгруппированная по тематическим вкладкам. Находим вкладку с названием Презентация и кликаем по ней. Внутри этой вкладки представлены элементы, предназначенные для визуального оформления модели, нам необходим компонент Изображение. Его надо перетащить в центральную область (рис. 4).

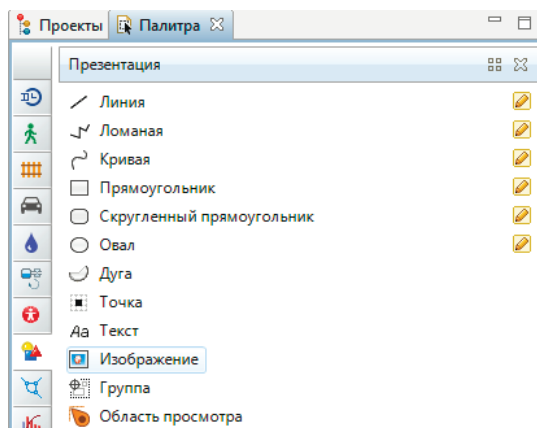


Рисунок 4. Добавление компонента Изображение

2.2. Открывается диалоговое окно, в котором необходимо выбрать файл с именем ПР_1-банк. Это графический файл, который содержит схему расположения стоек и касс банковского отделения. Необходимо сделать два клика левой кнопкой мыши или однократно кликнуть по файлу, а затем нажать кнопку Открыть (рис. 5).

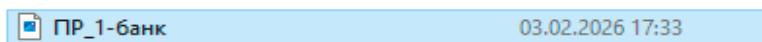


Рисунок 5. Выбор графического файла

2.3. После загрузки изображение появится на экране. Но оно может быть неподходящего размера, слишком большим или маленьким. Чтобы это изменить, необходимо его выделить и, зажимая левую кнопку мыши на одном из угловых маркеров, вытянуть по диагонали, увеличивая или уменьшая картинку (рис. 6).

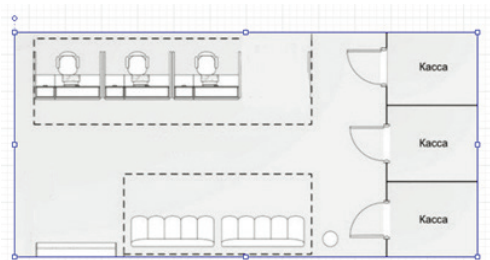


Рисунок 6. Масштабирование фонового изображения

2.4. При построении модели часто будут производиться перемещения и редактирования различных объектов. Если фоновое изображение останется незаблокированным, его можно случайно сдвинуть. Чтобы этого избежать, необходимо заблокировать, как на фотографии (рис. 7):

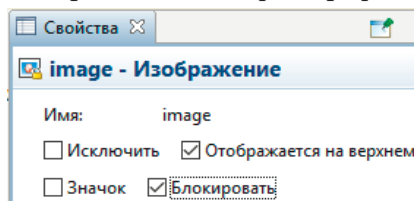


Рисунок 7. Блокировка фонового изображения

2.5. Для того чтобы задать границы, за пределы которых не могут выходить клиенты, применяется инструмент Стена из Пешеходной библиотеки. Чтобы активировать режим рисования, надо сделать двойной щелчок по элементу. После этого рядом со значком Стена появляется иконка Карандаш, являющаяся сигналом для рисования линии на холсте (рис. 8).

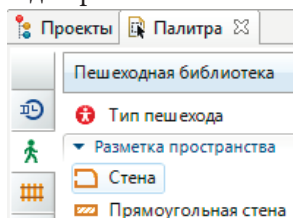


Рисунок 8. Добавление компонента Стена

2.6. Стена будет нарисована как последовательность отрезков (рис. 9).

Алгоритм следующий:

- Первый клик левой кнопкой мыши ставится в начальную точку стены.
- Курсор перемещается в то место, где стена должна изменить направление, и снова кликнуть — появится первый отрезок и фиксируется точка излома.
- Кликать надо в каждой точке, где стена поворачивает.
- В конечной точке последнего отрезка надо сделать двойной клик.

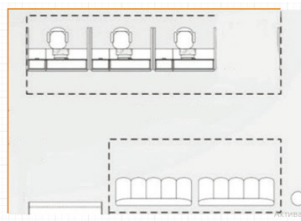


Рисунок 9. Построение стен

2.7. Данную процедуру необходимо повторить для каждого участка стен на плане банка. Особенно стоит уделить внимание следующим местам (рис. 10):

Стенки вокруг кассовых окошек, должны образовывать узкий проход, чтобы клиенты подходили по одному.

Стойки операторов нужно обрисовать так, чтобы клиенты не могли пройти сквозь рабочее место сотрудников.

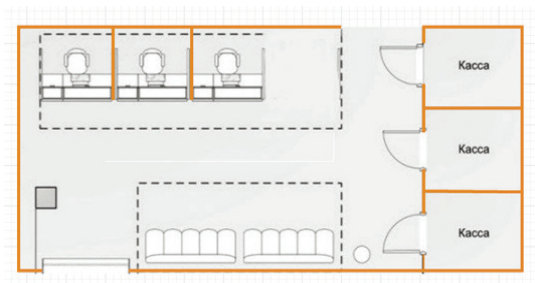


Рисунок 10. Схема расположения стен

Далее из палитры Пешеходная библиотека надо перетащить объект Целевая линия к входу в банк, который необходим для создания пешеходов (клиентов). Также добавим Целевую линию около касс — объект, определяющий место исчезновения пешеходов из модели, то есть имитацию того, что клиенты, получив услугу, ушли из отделения (рис. 11 и рис. 12).



Рисунок 11. Размещение целевой линии

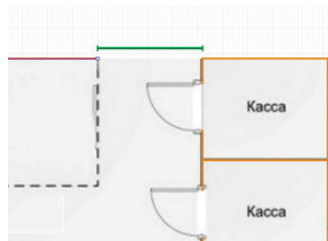


Рисунок 12. Размещение целевой линии

2.8. Зададим для линий имена entrance — от английского слова «вход» — и exit — от слова «выход» (рис. 13 и 14).

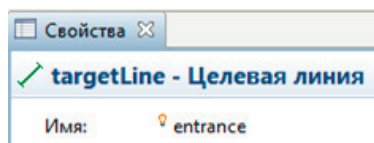


Рисунок 13. Задание имени

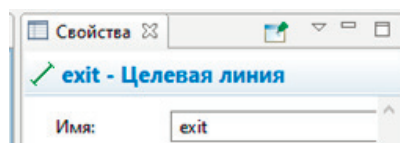


Рисунок 14. Задание имени

2.9. Добавим еще несколько Целевых линий — эти объекты будут определять кассы, необходимые для выполнения финансовых операций, и рабочие места операторов, которые проводят консультацию (рис. 15).

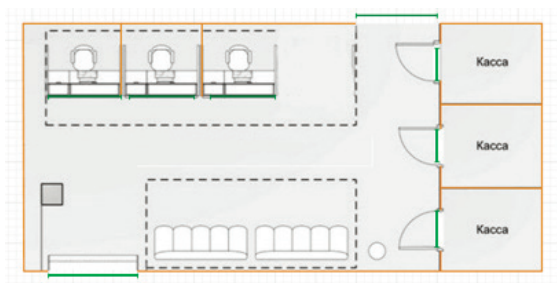


Рисунок 15. Целевые линии для касс и операторов

2.11. Далее зададим имена для линий. У операторов: operator1, operator2, operator3 (справа налево) и у касс: cash1, cash2, cash3 (сверху вниз).

2.12. Возле кресел добавим зону для клиентов, где они будут ожидать вызова к кассе или оператору, для этого используется объект Прямоугольная область из палитры Пешеходная библиотека. Назовем ее waitingArea (рис. 16).

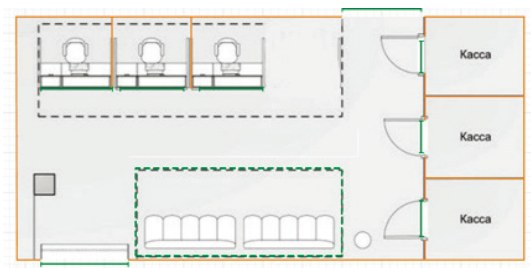


Рисунок 16. Добавление зоны Ожидание

Этап 3. Настройка логики движения и обслуживания клиентов

Теперь, когда визуальная среда готова, необходимо создать диаграмму процесса, то есть последовательность блоков, которая описывает жизненный цикл клиента от момента появления до момента ухода.

3.1. Открыв Пешеходную библиотеку, необходимо по очереди перетаскивать на холст следующие компоненты, располагая их в произвольном месте ниже нарисованного плана банка, также их необходимо соединить так, как показано на фото (рис. 17):

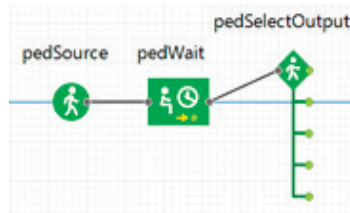


Рисунок 17. Фрагмент диаграммы

3.2. Выделив блок pedSource, необходимо в его свойствах задать следующие параметры:

- Местом появления будет выбрана линия. Это означает, что клиенты будут появляться не в случайной точке, а строго на линии entrance.
- Целевой линией будет выбрана entrance.
- Вводится интенсивность 30 человек в час.

Количество прибытий будет неограниченным, то есть флажок будет снят. Это необходимо, чтобы клиенты генерировались бесконечно (рис. 18).

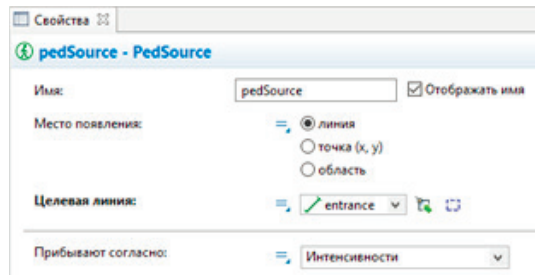


Рисунок 18. Настройка параметров

3.3. Выделяем блок pedWait, этот блок заставляет остановиться и ждать, пока не выполнится некоторое условие. В свойствах указывается:

Область — waitingArea. Тогда клиенты будут перемещаться в эту зону и там ожидать своей очереди.

Остальные параметры остаются без изменений (рис. 19).

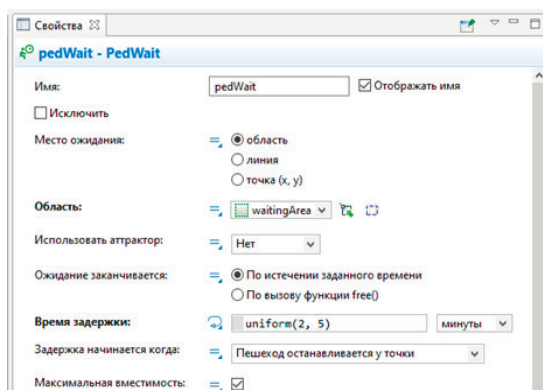


Рисунок 19. Настройка блока

3.4. Далее нужно определить, куда надо клиенту — к оператору или в кассу. Выбор происходит равномерно, поэтому свойства в объекте `pedSelectOutput` будут настроены следующим образом (рис. 20):

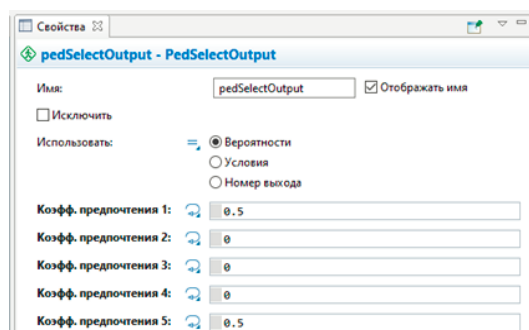


Рисунок 20. Настройка блока

3.5. В результате созданная схема должна будет выглядеть подобным образом (рис. 21):

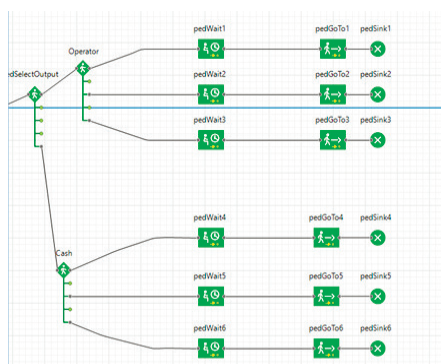


Рисунок 21. Диаграмма процесса

3.6. После того как клиент попал в ветку Оператор, он должен выбрать, к какому из трех операторов подойти. Аналогично для касс необходимо заполнить свойства подобным образом (рис. 22):

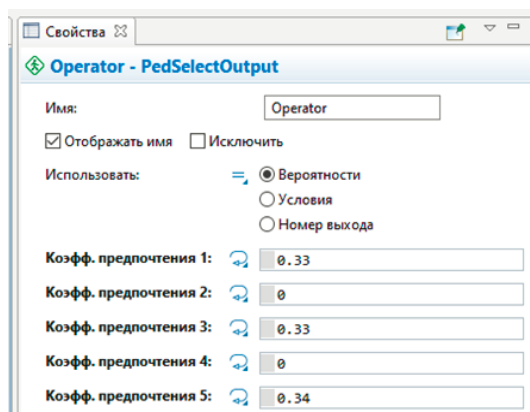


Рисунок 22. Настройка выбора оператора

3.7. Чтобы упростить схему, в объектах pedWait1, pedWait2, pedWait3, pedWait4, pedWait5, pedWait6 в качестве места ожидания выбрать линию, а дальше для каждого объекта указать соответственно Целевую линию, так как показаны свойства для pedWait1 (рис. 23):

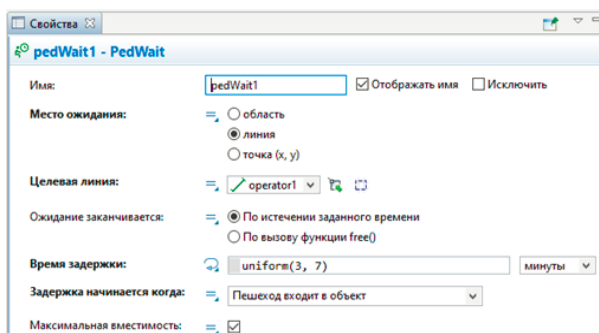


Рисунок 23. Настройка блока

Обратите внимание, что для объектов pedWait4, pedWait5, pedWait6 в окне свойства время задержки будет uniform (от 2-х до 5-и минут) вместо (от 3-х до 7-и минут) (рис. 23).

3.8. Используем объект pedGoTo, для того чтобы направить всех клиентов к выходу, после того как они получили услугу. Свойства выглядят следующим образом (рис. 24):

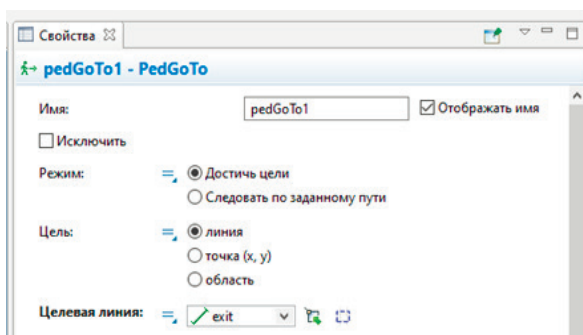


Рисунок 24. Настройка блока

3.9. Для запуска необходимо кликнуть кнопку Построить на верхней панели инструментов. Если ошибок нет, то кнопка Запуск станет активной. После нажатия откроется окно симуляции (рис. 25).

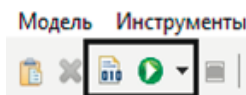


Рисунок 25. Кнопки Построить и Запуск

3.10. Запустив имитацию, можно пронаблюдать движение клиентов внутри банковского отделения. В течение 3600 секунд модельного време-

ни (1 час) клиенты постепенно поступают в систему и распределяются по зонам обслуживания в соответствии с заданной логикой. Таким образом, при текущих параметрах модели все клиенты, пришедшие в рассматриваемый временной интервал, имеют возможность попасть в зал ожидания и быть направленными к кассам или операторам без критических задержек (рис. 26).

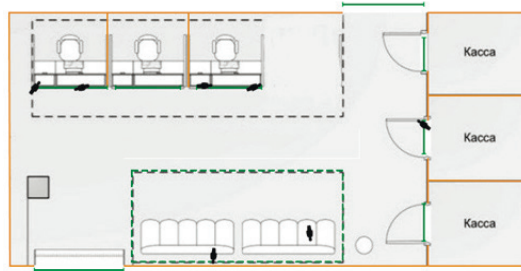


Рисунок 26. Результат имитации

При запуске модели можно заметить небольшой недочет: около одного оператора или одной кассы стоит несколько клиентов, что невозможно, так как один оператор может обслуживать только одного клиента. Этот недочет нуждается в исправлении на следующем этапе работы.

Этап 4. Устанавливаем ограничение на количество обслуживаемых клиентов

4.1. Для того чтобы к одному окошку мог подойти только один клиент, следует использовать объект `restrictedAreaStart` и `restrictedAreaEnd` из палитры Библиотека моделирования процессов. Визуально схема станет такой (рис. 27):

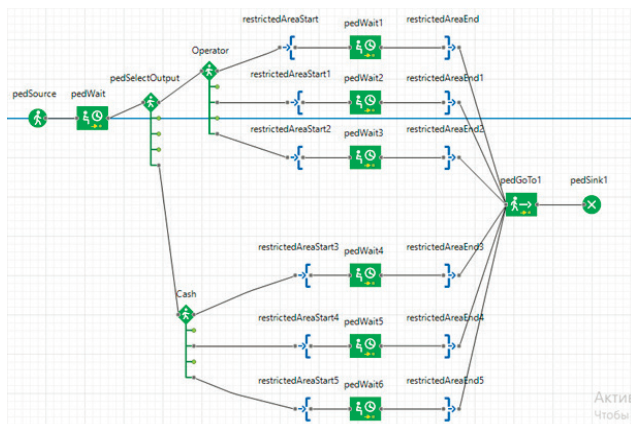


Рисунок 27. Использование блоков для ограничения

В свойствах для каждого `restrictedAreaStart` необходимо указать вместимость 1, а для `restrictedAreaEnd` выбрать соответствующий блок `restrictedAreaStart`.

Запустив имитацию, еще раз пронаблюдаем движение клиентов в отделении банка.

Этап 5. Сбор статистики о времени обслуживания

Для анализа эффективности работы банка недостаточно просто визуально наблюдать за движением фигурок. Для этого необходимо время пребывания клиента в системе. Данный показатель демонстрирует, нет ли перегрузки, не слишком ли долго клиенты ждут, не нужно ли добавить дополнительных сотрудников. Сама программа AnyLogic не запоминает момент входа и выхода каждого конкретного пешехода. Для того чтобы это решить, необходимо создать собственного типа пешехода с параметром времени.

5.1. Открыв Пешеходную библиотеку, необходимо добавить на холст объект Тип пешехода, при этом автоматически открывается мастер создания нового типа (рис. 28).

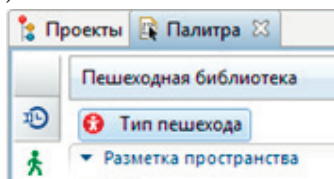


Рисунок 28. Добавление объекта

5.2. В первом окне вводится имя `bankClient`, затем необходимо кликнуть по кнопке Далее (рис. 29).

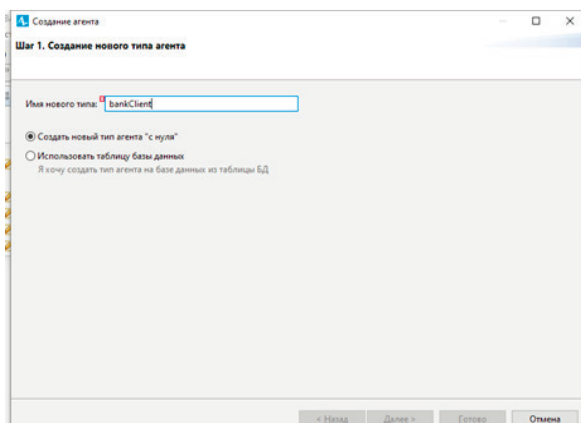


Рисунок 29. Создание нового типа пешехода

5.3. Во втором окне мастера предлагается выбрать фигурку анимации, которая будет представлять клиентов. Можно выбрать любую из предложенных, но для банка лучше подойдет фигурка в деловой одежде. После выбора необходимо кликнуть на кнопку Далее.

5.4. Третье окно предназначено для добавления параметров к типу пешеходов. Необходимо поле, которое будет хранить время входа в систему. Надо, кликнув на кнопку, в появившейся строке ввести Параметр arrivalTime (рис. 30):

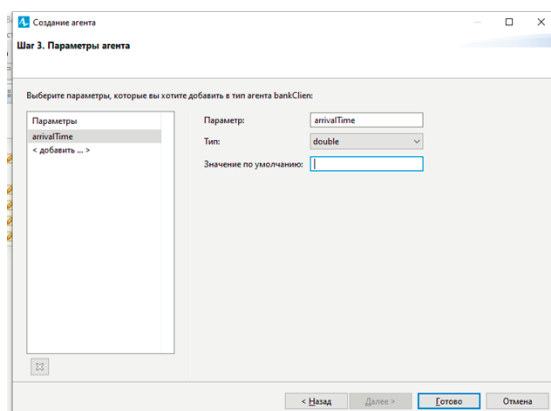


Рисунок 30. Добавление параметра

Замечание по умолчанию будет пустым, так как время входа заранее не известно. Далее нажимается Готово. Создается новый класс с рабочим полем, но менять в нем ничего не надо. Вкладку нужно закрыть и вернуться к компоненту Main.

5.5. Теперь в блоке pedSource необходимо изменить параметры так, чтобы он создавал пешеходов типа bankClient и чтобы при создании он сохранил текущее время в параметр arrivalTime.

- Выделим pedSource.

- В панели Свойства находим Новый пешеход. В выпадающем списке выбирается bankClient.

- В разделе Специфические в поле Тип пешехода выбирается bankClient.

- В разделе Действие, в поле Действие при выходе, вводится следующий код: `((bankClient)ped).arrivalTime = time()` (рис. 31).

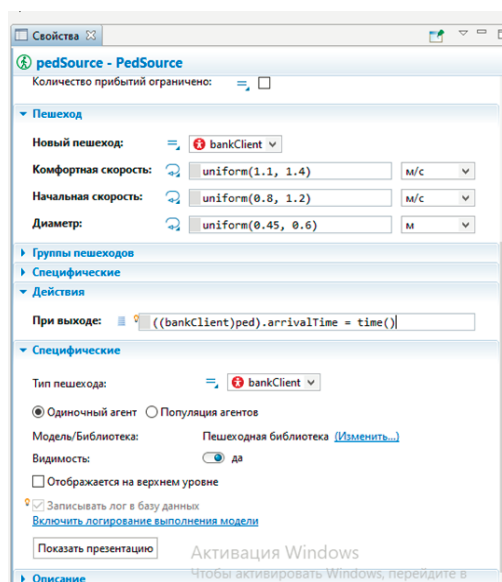


Рисунок 31. Настройка блока

5.6. Открывается палитра Статистика, а в ней выбирается объект Статистика. В панели Свойства дается имя statistics. Этот объект автоматически накапливает все значения, которые были добавлены, а также вычисляет минимальное, максимальное, среднее и другие величины (рис. 32).

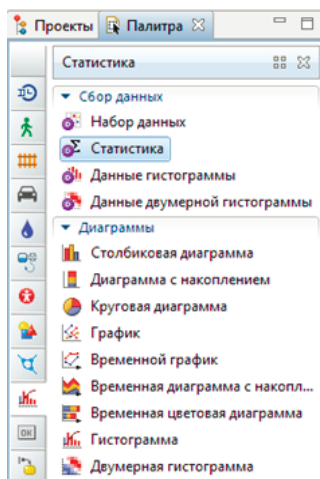


Рисунок 32. Добавление объекта

5.7. Теперь нужно в тот момент, когда клиент достигает блока `pedSink`, вычислить разницу между текущим временем и сохраненным временем входа и добавить это значение в статистику.

- Выделить каждый блок `pedSink` в модели.

- В панели Свойства, в разделе Действия и в поле Действие при входе, ввести следующий код: `statistics.add(time()-((bankClient)ped).arrivalTime)`.

Теперь при выполнении прогонов имитации можно не наблюдать за счетчиком модельного времени, а просто отслеживать накопленные данные: минимальное, максимальное и среднее время прохода клиентов, кликая на объект статистика.

В этом прогоне максимальное время, которое потратил клиент, чтобы получить услугу, — 18 минут, а среднее время — 9 минут [5].

Проектное задание. Спроектируйте установку двух дополнительных банкоматов в отделении. Предусмотрите разграничение потоков клиентов для получения разных услуг.

Заключение. В перспективе умение создавать и работать с имитационными моделями станет ключевым навыком для различных отраслей в связи с тенденцией цифровизации, работы с большими данными, необходимостью принимать решения в сложных и неопределенных ситуациях. Имитационное моделирование будет востребовано в управлении системами и инфраструктурой, инженерии и проектировании, аналитике, здравоохранении и биотехнологиях в связи с необходимостью комплексно анализировать динамические процессы, прогнозировать развитие событий и искать оптимальные решения [6].

Список литературы

1. Калинин И. А. Информатика. Углубленный уровень: задачи, практикум для 10–11 классов / И. А. Калинин, Н. Н. Самылкина, П. В. Бочаров. М.: БИНОМ. Лаборатория знаний, 2014. 248 с.
2. Курпияшкин А. Г. Основы моделирования систем: учебное пособие. Норильск: Норильский индустриальный институт. 2015. 135 с.
3. Малыгина С. Н. Обзор современных средств имитационного моделирования / С. Н. Малыгина, Е. О. Неупокоева // Труды Кольского научного центра РАН. Серия: Технические науки. 2022. Т. 13, № 2. С. 134–143. doi: 10.37614/2949-1215.2022.13.2.013.
4. Проект приказа Минпросвещения России «Об утверждении федерального государственного образовательного стандарта среднего общего образования» [Электронный ресурс]. URL: <https://docs.edu.gov.ru/document/95cddc2389fb11bda9cac0221b4fc45> (дата обращения 08.04.2026).
5. Самылкина Н. Н. Информатика (углубленный уровень). Реализация требований ФГОС среднего общего образования: метод. пособ. для учителя. М.: ФГБНУ «Институт стратегии развития образования», 2023. 226 с.
6. Юрченко Т. В. Информационные технологии в экономике. Решение экономических задач средствами MS EXCEL 2007: учеб. пособ. Н. Новгород: Нижегород. гос. архит.-строит. ун-т, 2010. 132 с.