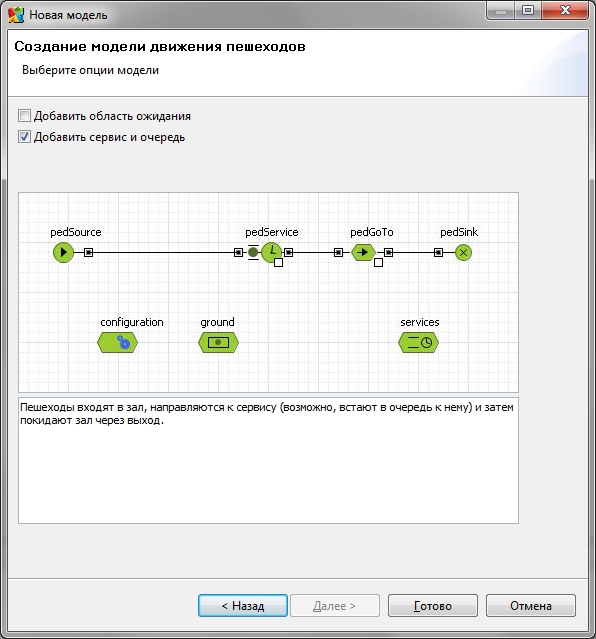
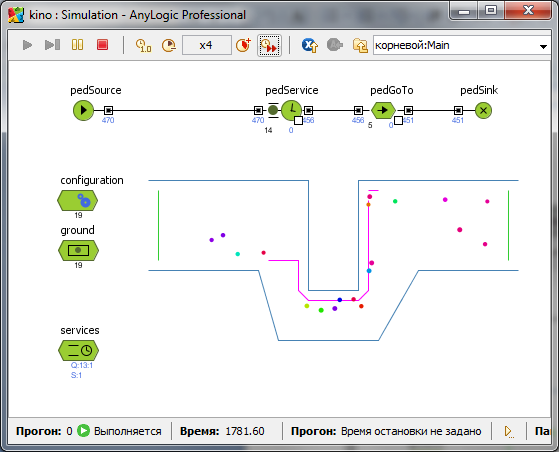
**Модель кинотеатра.**

Построим имитационную модель посещения зрителей в кинотеатре с точки зрения их распределения согласно купленным билетам и их выходу по окончании кино.

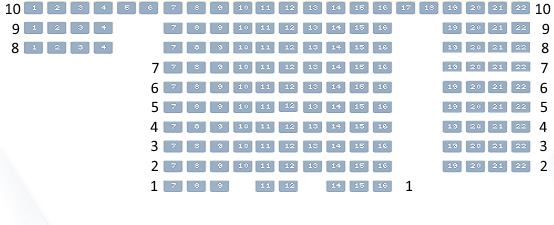
1. Создайте новый проект, используя шаблон «Моделирование движения пешеходов». При создании поставьте галочку напротив «Добавить сервис и очередь», так как нам потребуется ОУ для генерации задержки на зрительских местах (просмотр кино). Тогда по умолчанию будут добавлено большинство необходимых блоков.



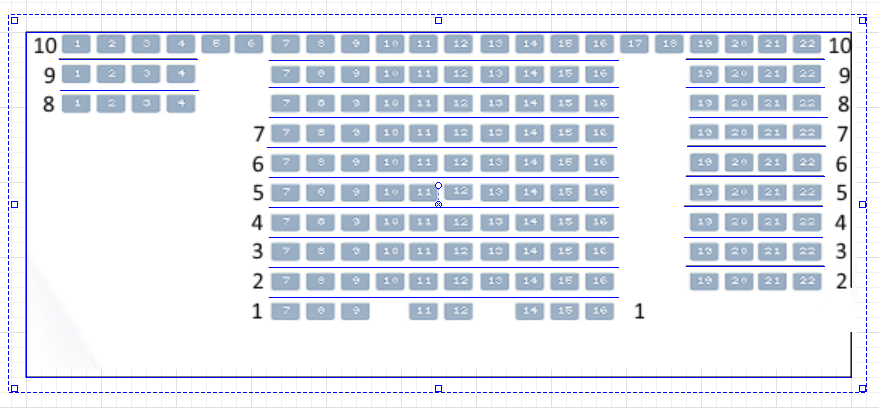
1. В принципе, готовый шаблон презентации уже можно запустить и посмотреть, как зрители выстраиваются в очередь перед кассой.



1. Доработаем готовый шаблон до нужной нам модели кинотеатра.
2. Сначала добавьте изображение плана кинотеатра. Для этого вначале откройте закладку Презентация панели Палитра и перетащите элемент Изображение. Измените его свойства, чтобы выбрать нужное изображение и отметьте галочку «Исходный размер». Само изображение, очевидно, должно быть подготовлено студентом заранее, примерно подобного вида:



1. Далее мы нарисуем границу моделируемого пространства, играющую роль стен здания. Чтобы было легче рисовать, лучше отключить сетку и увеличить масштаб анимации с помощью соответствующих кнопок панели инструментов. С помощью инструмента Ломанная или Линия, находящиеся в палитре Презентация, нарисуйте границы, как показано на рисунке.



1. Теперь сгруппируем нарисованные линии. Для этого выделите нарисованные границы, нажмите ЛКМ и выберите Группировка – Создать группу. Переименуйте эту группу в свойствах группы на **walls**. Группа фигур используется для группировки фигур презентации. С помощью динамических свойств группы фигур (X, Y, Поворот, и т.п.) можно передвигать группу фигур и поворачивать ее вокруг опорной точки. Сама опорная точка группы фигур на анимации не отображается.
2. Для корректного функционирования схемы нам нужны блоки:

☑ **pedSource**

☑ **pedService**

☑ **pedGoTo**

☑ **pedSink**

☑ **pedConfiguration**

☑ **pedGround**

☑ **pedServices**

Проверьте, что они присутствуют на схеме.



1. **PedSource** – создает посетителей. Обычно используется в качестве начальной точки блок-схемы, формализующей поток пешеходов. Создает заявки-посетителей любых подклассов базового класса *Ped* через случайные промежутки времени.
2. **PedServices** – задает группу одинаковых физических объектов обслуживания (например, несколько турникетов или автоматов по продаже билетов). Объект позволяет задавать очереди и сервисы в любой комбинации, и задавать правила выбора сервисов: какую очередь выбрать, какой сервис выбрать, к какой очереди должен обращаться сервис, может ли сервис обслуживать несколько очередей и т.д. Используйте этот объект, чтобы создавать ряды сервисов и задавать их параметры по умолчанию.
3. **PedGoTo** – заставляет посетителей перейти в заданное место моделируемого пространства, которое может быть задано линией или точкой. Переход будет считаться выполненным, когда пешеход пересечет заданную линию или достигнет заданной точки. Люди-заявки будут искать путь к заданному транзиту в пределах текущего этажа.
4. **PedSink** – удаляет поступивших в объект посетителей из моделируемой среды. Обычно объект используется в качестве конечной точки блок-схемы, формализующей поток пешеходов. *PedSink* автоматически ведет подсчет заявок. Также объект можно использовать для удаления любого посетителя. Если выходной порт объекта библиотеки *Pedestrian Library* ни к чему не подсоединен, то он не сможет произвести вывод пешеходов, поэтому, если Вам нужно удалять заявки, то обязательно поместите объект *PedSink*.
5. **PedConfiguration** – это главный объект *Pedestrian Library*. Объект *PedConfiguration* должен присутствовать на диаграмме в каждой модели *AnyLogic*, использующей *Pedestrian Library*, поскольку он поддерживает перемещение посетителей и их анимацию. Он позволяет задавать общие параметры, относящиеся ко всем объектам *Pedestrian Library*, и настраивать модель для конкретной задачи с целью получения максимальной производительности.
6. **PedGround** – позволяет задавать двумерное пространство в моделируемой среде, представляющее собой «этаж», т.е. поверхность, по которой будут перемещаться люди. Этажи могут быть ограничены какой-то стеной или быть неограниченными. Стены – это объекты, которые пешеходы не могут пересекать. Стены являются частью этажа, то есть одна стена не может быть использована несколькими этажами.
7. В свойствах объекта **pedSource** следует сделать следующие изменения:

пешеходы прибывают согласно: **Интенсивности**;

расписание, пешеходов в ед. времени: **150 / minute()**;

этаж (PedGround): **ground**;

начальное направление: **Запад**;

место появления (линия, ломаная): **exit**.

1. В свойствах объектов **pedService** ничего менять не надо.
2. В свойствах объектов **PedGoTo** следует сделать следующие изменения:

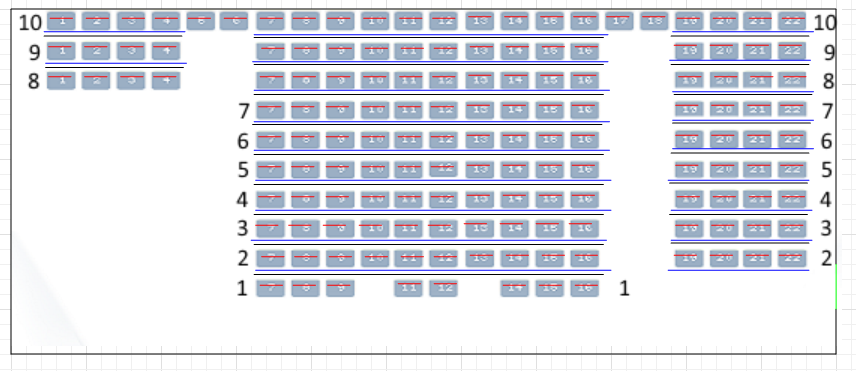
цель (точка, линия): **exit.**

1. В свойствах объектов **PedGround** следует сделать следующие изменения:

стены (группа, необязательный): **walls**

1. Не забудьте добавить саму линию **exit** – вход в кинотеатр.
2. Теперь создадим сервисы обслуживания и очереди. Их можно сделать с помощью инструмента Ломаная или Линия. Сервисами будут места в кинотеатре, а очередями – промежутки между рядами.

Ниже на рисунке: зеленая линия = вход, красные линии = места, синие линии = очереди.



1. Нужно сгруппировать сервисы и переименовать в **mesta**, затем сгруппировать очереди и переименовать в **groupqueue**. В их свойствах (*Динамические*) следует установить видимость ***false***.
2. В свойствах объектов **PedServices** следует сделать следующие изменения:

сервисы (группа линий): **mesta**,

время задержки: **uniform(5.0\*minute(), 5.0\*minute())**

правило выбора очереди: **самая короткая очередь**,

обслуживается: **самая длинная очередь**.

1. В текущей модели интенсивность потока зрителей задана фиксированной величиной. Мы же хотим иметь возможность интерактивно изменять интенсивность во время выполнения модели, чтобы на тот или иной сеанс было разное количество зрителей. Для этого мы добавим специальный элемент управления.
2. Откроем палитру *Элементы управления* и перетащим элемент *Бегунок* из палитры на диаграмму класса *Main* (справа от плана кинотеатра).
3. В свойствах объекта **Бегунок** следует сделать следующие изменения:

связать с: **pedSource.rate**,

минимальное значение: **0**,

максимальное: **150 / minute()**.

1. Добавим текстовую метку для бегунка. Для этого добавим к бегунку текстовую метку с заголовком. Перетащим элемент *Текст* из палитры *Презентация* на диаграмму (выше бегунка). Введем текст, который мы хотим показать с помощью этой метки, в поле **Текст**. Например: **Изменение интенсивности зрителей**.
2. Теперь можно смело запускать модель и наблюдать, как люди рассаживаются в кинотеатре на сеанс.
3. Настройте модель до режима, наиболее приближенного к реальной жизни, меняя различные параметры. Продемонстрируйте полученный результат преподавателю.