


ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 3

Задание транспортной базы




На этом этапе необходимо задать логику и процессы модели на диаграммах каждого агента в отдельности. Начнем с конфигурации типа агента  Main, которая задает начальную позицию грузовиков и вертолетов в пространстве.

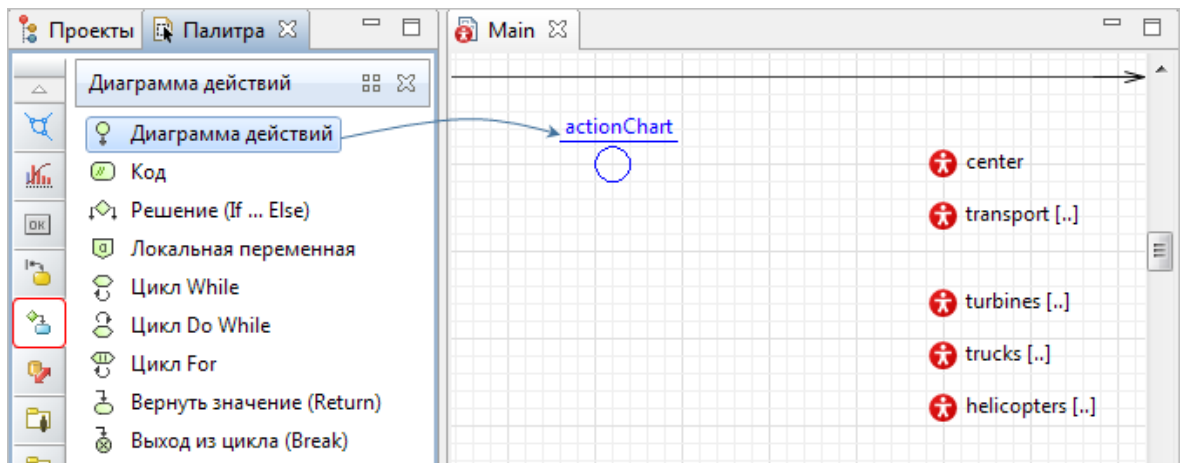
Сложное имитационное моделирование не может существовать без возможности задания алгоритмов, обычно выполняющих определенную обработку данных или вычисления. AnyLogic поддерживает **Диаграммы действий** – структурированные блок-схемы, позволяющие задавать алгоритмы графически в стиле структурированного программирования. При задании алгоритмов используется расширение подхода, предложенного в свое время Дейкстра. Суть подхода состоит в том, что алгоритмы разбиваются в подразделы с одной точкой входа. Утверждается, что трех способов объединения программ – упорядочения, повторения и выбора – достаточно для задания алгоритма любой сложности. Такой стиль сводит понимание целого алгоритма к пониманию составляющих его частей.

Диаграммы действий облегчают задание алгоритмов, делая необязательным знание синтаксиса Java операторов.

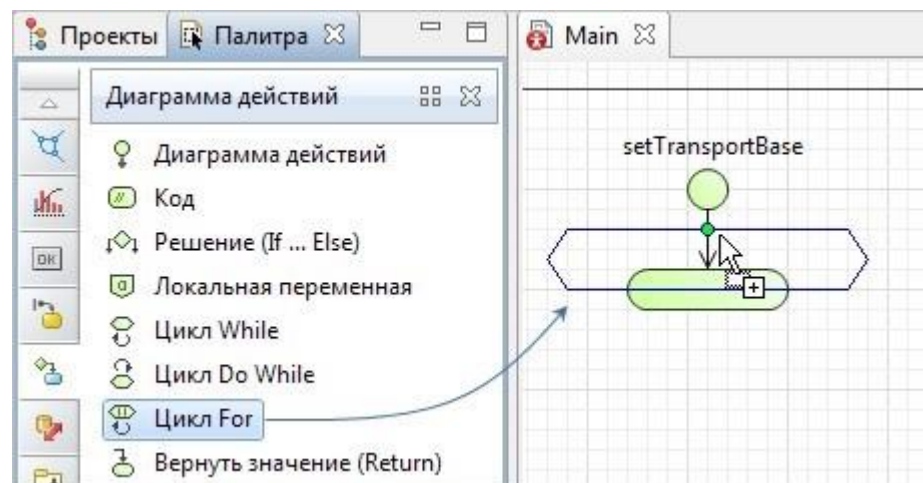
По сути диаграммы действий используются для визуального задания функций, и Вы можете использовать для этого обычные функции. Но использование диаграмм действий дает еще одно преимущество: с их помощью Вы можете визуализировать алгоритмы, делая их более понятными для других пользователей модели.

Поместите транспорт в сервисный центр

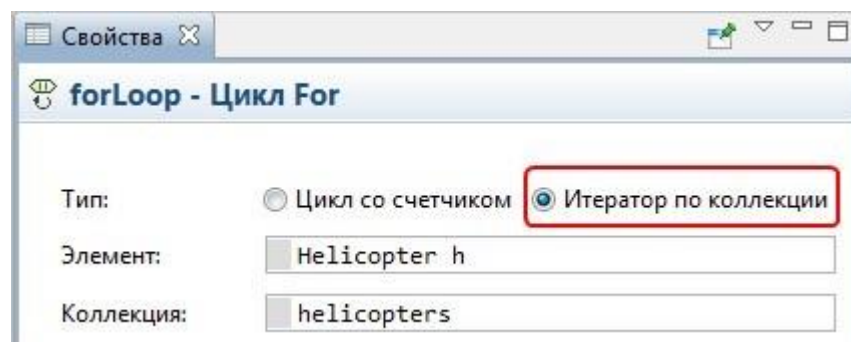
1. Перейдите на диаграмму  Main и откройте палитру **Диаграмма действий** в панели **Палитра**.
2. Перетащите элемент **Диаграмма действий**  в графический редактор  Main.



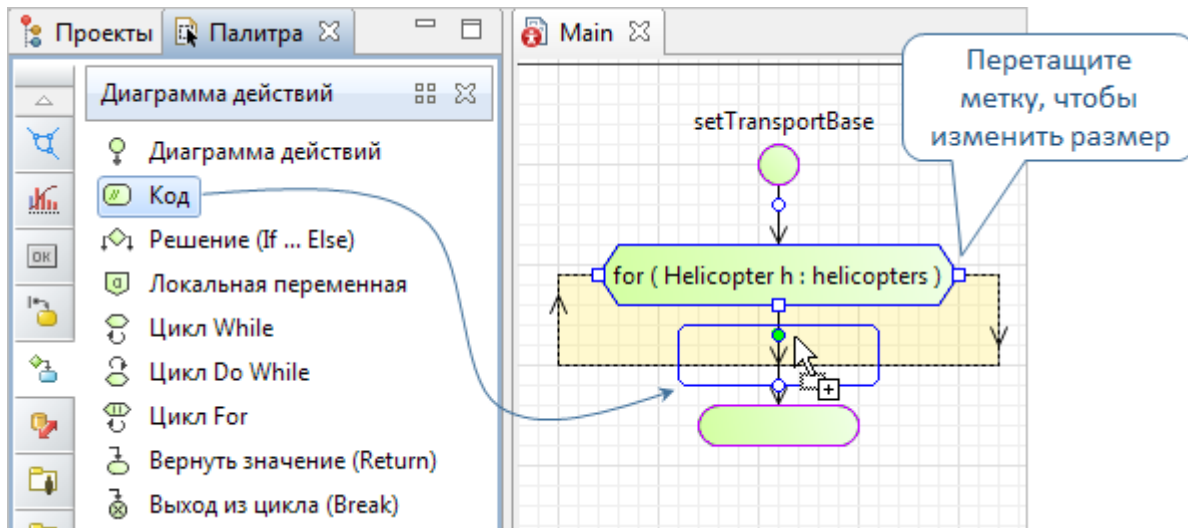
3. Назовите диаграмму **setTransportBase**. Эта диаграмма действий ничего не возвращает, то есть, ее тип - **Действие**. Далее, добавьте в диаграмму элемент **Цикл For**. Работая с элементами диаграммы действий, проверяйте, что вставляете элементы в нужное место:



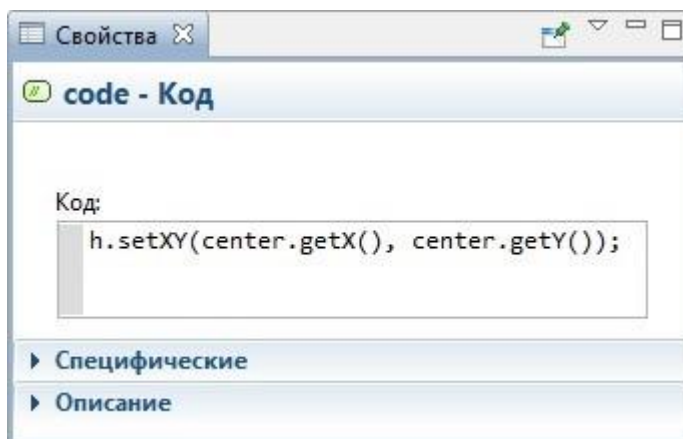
4. Этот цикл **for** является **Итератором по коллекции**, где **Коллекция** - **helicopters**. Определите Элемент как **Helicopter h**.








5. Далее, вставьте элемент **Код**  в цикл **for**.



6. Вертолеты изначально находятся в сервисном центре, то есть, имеют те же координаты, что и он:



Свойства    



 **code - Код**

Код:

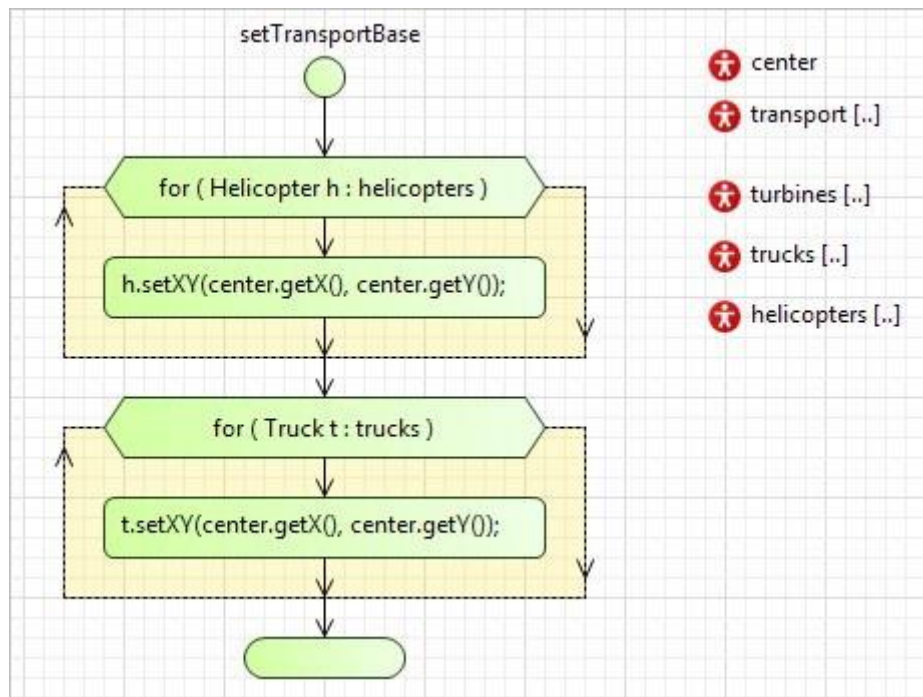
```
h.setXY(center.getX(), center.getY());
```

► Специфические

► Описание

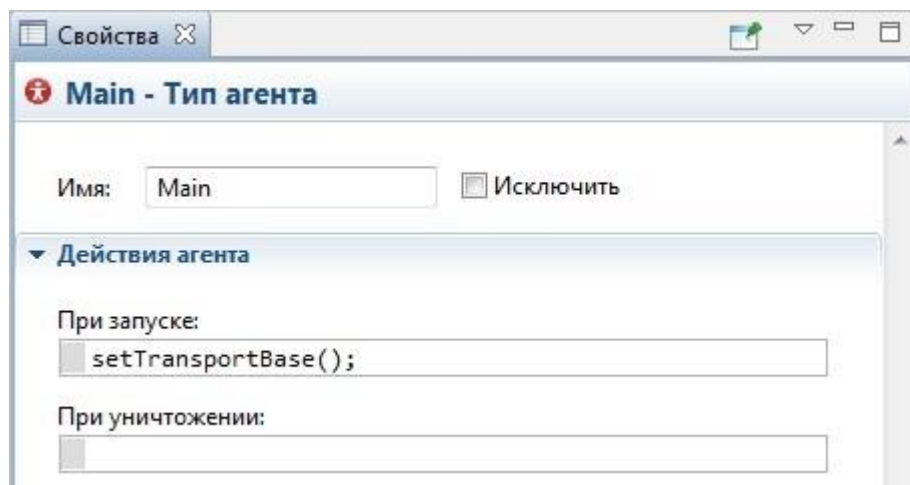
7. Добавьте еще один блок, состоящий из **Цикла For**  с элементом **Код**  внутри в эту диаграмму действий и сконфигурируйте его, размещая грузовики в сервисном центре.

8. При необходимости можно изменить размер блоков диаграммы действий, чтобы содержащийся в них код был полностью виден:



В итоге была создана диаграмма действий под названием **setTransportBase**, которая представляет функцию, размещающую транспортные средства в сервисном центре. Там же будет размещен транспорт при старте модели.

Выделите **Main** в дереве модели (или сделайте щелчок мышью в пустое место в графическом редакторе этого типа агента) и перейдите в панель **Свойства**. Разверните секцию свойств **Действия агента** и задайте запуск функции **При запуске** модели:



Запустите модель. Вы увидите, что теперь турбины распределены случайно, а транспортные средства находятся «в ангаре». Далее будет задано движение транспорта и поведение других агентов.