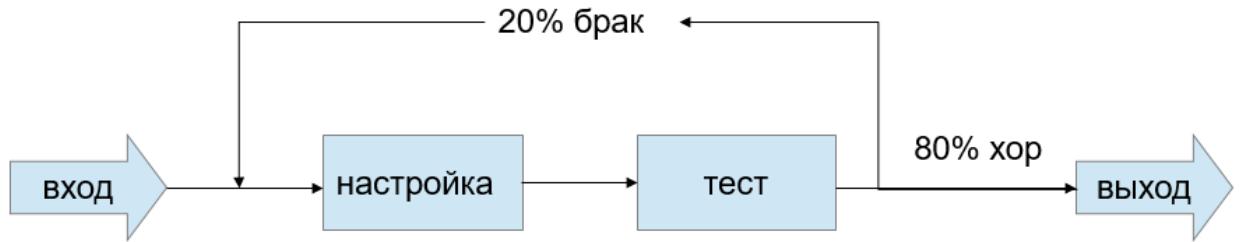


Практическая работа

Выполнение экспериментов с моделью в AnyLogic

Возьмем для примера такую задачу: промоделировать работу технологического процесса по настройке и тестированию компьютерных мониторов.



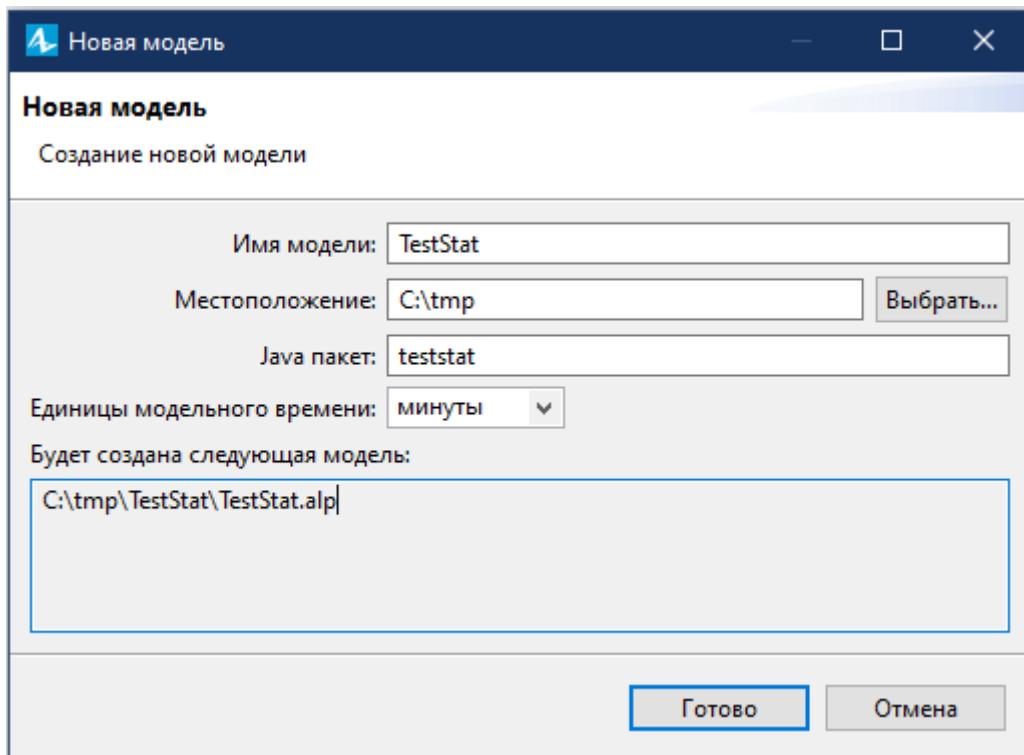
Мониторы на участок настройки поступают в среднем по 20 шт в час с экспоненциальным распределением времени между прибытиями. Настройка происходит с экспоненциальным распределением для среднего значения времени 2 мин. Тестирование — примерно 1.75 минут с экспоненциальным распределением времени, причем 20% мониторов возвращаются на настройку повторно.

Показателями качества процесса являются:

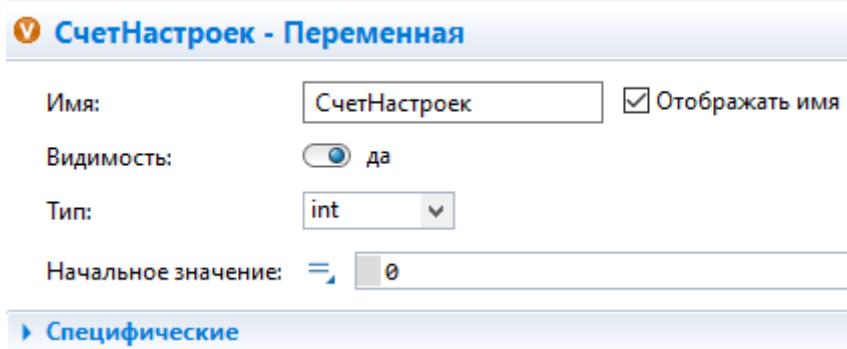
- суммарное время выполнения работ над монитором (**Time In System - TIS**)
- количество завершенных работ – сделанных мониторов (**Number In System - NIS**)
- коэффициент использования обработчиков – настройщик и тестировщик (**Utilization**)

Фаза 1. Подготовка модели

Создайте новую модель AnyLogic.



Создайте новый тип агента – Монитор. Никаких атрибутов пока не задавайте. После добавления агента в модель на странице агента Монитор добавьте компонент агента Переменная из палитры Агент. Эта переменная будет использоваться в экспериментах. Задайте переменной свойства как на рисунке:



Добавьте компонент из палитры Презентация скругленный прямоугольник и задайте ему свойства как на рисунке:

roundRectangle - Скругленный прямоугольник

Имя: Исключить

Отображается на верхнем агенте Значок Блокировать

Видимость: да

Внешний вид

Цвет заливки: (СчетНастроек==0) ? black: (СчетНастроек==1 ? green:(СчетНастроек==2 ? blue:red))

Цвет линии: Нет цвета

Толщина линии: 1 pt

Стиль линии:

Местоположение и размер

Уровень:

X: Ширина:

Y: Высота:

Поворот: градусы

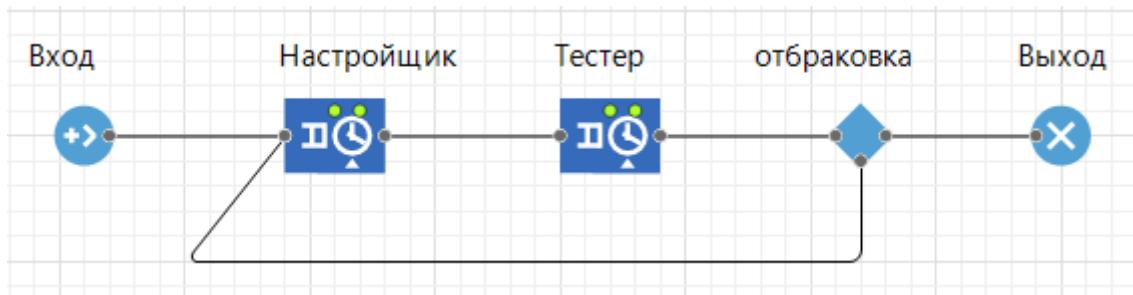
Радиус скругления углов:

На закладке агента Main в логической структуре модели определим основные технологические операции.

Из библиотеки процессов поставьте блоки **Source**, **Service (2wtm)**, **SelectOutput**, **Sink**.

Переименуйте их — **Вход**, **Настройщик**, **Тестер**, **отбраковка**, **Выход**

Соедините блоки в соответствии с рисунком.



Для учета использования ресурсов добавьте два компонента ResourcePool.

В одном ресурсном компоненте задайте такие свойства:

Настройщик - ResourcePool

Имя:	настройщик	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Тип:	=	Статический
Количество задано:	=	Напрямую
Количество ресурсов:	=	1
При уменьшении кол-ва:	=	ресурсы сохраняются (конец смены)
<hr/>		
Новый ресурс:	=	Агент
создать другой тип		
Базовое местоположение (узлы):	=	<input type="text"/>
Отображать анимацию по умолчанию:	=	<input checked="" type="checkbox"/>

В другом ресурсном компоненте задайте такие свойства:

Инспектор - ResourcePool

Имя:	инспектор	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Тип:	=	Статический
Количество задано:	=	Напрямую
Количество ресурсов:	=	1
При уменьшении кол-ва:	=	ресурсы сохраняются (конец смены)
<hr/>		
Новый ресурс:	=	Агент
создать другой тип		
Базовое местоположение (узлы):	=	<input type="text"/>
Отображать анимацию по умолчанию:	=	<input checked="" type="checkbox"/>

Задайте свойства блоку source так, как на рисунке:

⊕ Вход - Source

Имя:	<input type="text" value="Вход"/>	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Прибывают согласно:	= Времени между прибытиями	
Время между прибытиями:	<input type="text" value="exponential(1/3.0)"/>	минуты
Первое прибытие происходит:	= при запуске модели	
Считать параметры агентов из БД:	= <input type="checkbox"/>	
За 1 раз создается несколько агентов:	= <input type="checkbox"/>	
Ограниченнное кол-во прибытий:	= <input type="checkbox"/>	

Задайте свойства первому блоку service так, как на рисунке:

⊕ Настройщик - Service

Имя:	<input type="text" value="Настройщик"/>	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Захватить:	= <input type="radio"/> (альтернативный) набор ресурсов <input checked="" type="radio"/> ресурсы одного типа	
Тип ресурсов:	= <input type="text" value="настройщик"/>	
Количество ресурсов:	= <input type="text" value="1"/>	
Вместимость очереди:	= <input type="text" value="200"/>	
Максимальная вместимость:	= <input type="checkbox"/>	
Время задержки:	<input type="text" value="exponential(1/2.0)"/>	минуты

Задайте аналогично свойства второму блоку service, изменив имя на Тестер, ресурс – на Инспектор, время – на exponential(1/1.7) мин.

В блоке SelectOutput задайте такие свойства:

◇ отбраковка - SelectOutput

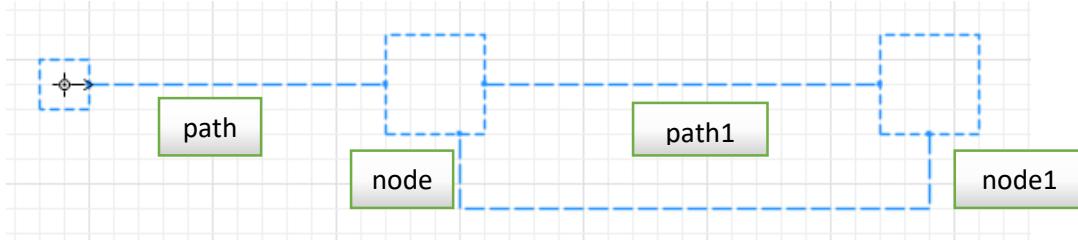
Имя:	<input type="text" value="отбраковка"/>	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Выход true выбирается:	<input checked="" type="radio"/> Заданной вероятностью <input type="radio"/> При выполнении условия	
Вероятность:	<input type="text" value="0.8"/>	

Так как в данном случае нас не интересует конкретное расположение объектов в пространстве, то можно просто добавить **схематическую анимацию** интересующих нас объектов — обработчики и очереди мониторов к ним. Анимация модели рисуется в той же Диаграмме в графическом редакторе, в которой и Диаграмма моделируемого процесса.

Нарисуйте прямоугольные узлы, которые будет обозначать на анимации обработку. Откройте палитру Разметка пространства, содержащей в качестве элементов различные примитивные фигуры, используемые для рисования презентаций моделей: путь, прямоугольный узел, многоугольный узел, точечный узел и пр.

Выделите элемент Прямоугольный узел и перетащите его на диаграмму класса Main. Поместите элементы Прямоугольный узел так, как показано на рисунке ниже. Нарисуйте пути, которые будут обозначать на анимации очередь к обработчику. Чтобы нарисовать путь, сделайте двойной щелчок мышью по элементу Путь, чтобы перейти в режим рисования. Теперь вы можете рисовать путь точка за точкой, последовательно щелкнув мышью в тех точках диаграммы, куда вы хотите поместить вершины пути. Чтобы завершить рисование, добавьте последнюю точку пути двойным щелчком мыши. Важно, какую точку пути вы создаёте первой. Агенты будут располагаться вдоль нарисованного вами пути в направлении от конечной точки к начальной точке. Поэтому обязательно начните рисование пути слева и поместите рядом с узлом (node) конечную точку пути, которая будет соответствовать в этом случае началу очереди.

Надписи на этой схеме соответствуют именам этих узлов и путей.



Теперь нужно задать созданные анимационные объекты в качестве анимационных фигур объектов диаграммы нашего процесса. Задайте пути в качестве фигуры анимации очереди. Выделите щелчком объект Настройщик. На панели свойств объекта установите такие значения:

Место агентов (queue):  

Место агентов (delay):  

› **Приоритеты / вытеснение**

Выделите щелчком объект Тестер. На панели свойств объекта установите такие значения:

Место агентов (queue):  

Место агентов (delay):  

› **Приоритеты / вытеснение**

Теперь для визуализации состояния обрабатываемых изделий добавьте действие «При выходе» в свойствах сервиса Настройщик:

Настройщик - Service

Имя:	<input type="text" value="Настройщик"/>	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Захватить:	<input type="radio"/> (альтернативный) набор ресурсов <input checked="" type="radio"/> ресурсы одного типа	
Тип ресурсов:	<input type="button" value="="/>  настройщик	<input type="button" value="<"/> , <input type="button" value=">"/>
Количество ресурсов:	<input type="button" value="="/>  1	<input type="button" value="<"/> , <input type="button" value=">"/>
Вместимость очереди:	<input type="button" value="="/> 200	<input type="button" value="<"/> , <input type="button" value=">"/>
Максимальная вместимость:	<input type="button" value="="/> <input type="checkbox"/>	<input type="button" value="<"/> , <input type="button" value=">"/>
Время задержки:	<input type="button" value="="/>  exponential(1/2.0)	<input type="button" value="<"/> , <input type="button" value=">"/> минуты
Пересыпать захваченные ресурсы:	<input type="checkbox"/>	<input type="button" value="<"/> , <input type="button" value=">"/>
Место агентов (queue):	<input type="button" value="="/>  path	<input type="button" value="<"/> , <input type="button" value=">"/>
Место агентов (delay):	<input type="button" value="="/>  node	<input type="button" value="<"/> , <input type="button" value=">"/>

Приоритеты / вытеснение

Специфические

Действия

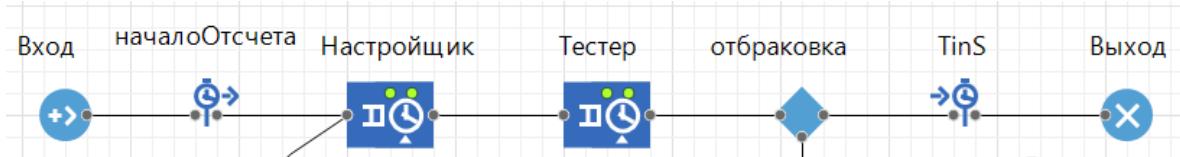
При входе:	<input type="button" value="="/> 	<input type="button" value="<"/> , <input type="button" value=">"/>
При захвате ресурса:	<input type="button" value="="/> 	<input type="button" value="<"/> , <input type="button" value=">"/>
При начале задержки:	<input type="button" value="="/> 	<input type="button" value="<"/> , <input type="button" value=">"/>
При подходе к выходу:	<input type="button" value="="/> 	<input type="button" value="<"/> , <input type="button" value=">"/>
При выходе:	<input type="button" value="="/> agent.СчетНастроек ++;	<input type="button" value="<"/> , <input type="button" value=">"/>
При извлечении:	<input type="button" value="="/> 	<input type="button" value="<"/> , <input type="button" value=">"/>

Предварительная фаза настроек закончена, для проверки результатов запустите модель.
Убедитесь по анимации в появлении мониторов всех использованных цветов.

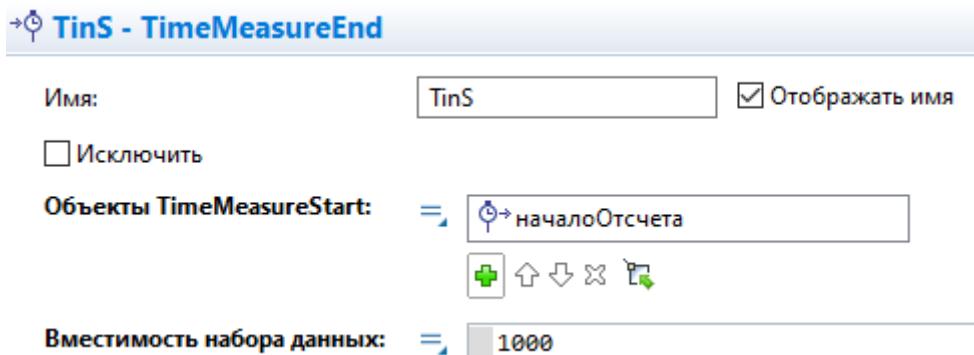
Фаза 2. Настройка сбора статистики в простом эксперименте

Для сбора статистических данных о времени обработки деталей необходимо добавить элемент статистики. Этот элемент будет запоминать значения времён для каждого монитора. На основе собранных данных он предоставит пользователю стандартную статистическую информацию (среднее, минимальное, максимальное из измеренных значений, среднеквадратичное отклонение и т.д.).

Из палитры моделирования процессов добавьте блок «Time Measure Start» после source «Вход» и задайте ему имя началоОтсчета:



Блок «Time Measure End» поставьте перед блоком sink «Выход» и задайте такие свойства:



Имя TinS означает сокращенно Time In System – часто используемый показатель процесса. В этом блоке создается и будет заполняться временем жизни проходящих агентов набор данных TinS-dataset.

Чтобы получить доступ к данным о состоянии всех обрабатываемых в каждый момент времени мониторах, добавьте компонент Популяция агентов. Для этого из палитры Агент перетащите иконку Агент на страницу Main и в панели мастера создания агента выберите Популяция агентов. Затем укажите «Я хочу использовать существующий тип агента», из списка выберите Монитор и задайте имя популяции Мониторы. Далее укажите «Я хочу добавить агентов позже» и оставьте популяцию пустой.

После появления популяции в агенте Main зайдите в ее свойства и добавьте следующие настройки в разделе Статистика:

Мониторы - Монитор

Имя: Отображать имя

Исключить

Одиночный агент Популяция агентов

Популяция: Изначально пуста
 Содержит заданное кол-во агентов
 Загружается из базы данных

► Размеры и движение

► Начальное местоположение

▼ Статистика

Имя:

Тип: Кол-во Сумма Среднее Мин. Макс.

Выражение:

Условие:



► Специфические

Для заполнения популяции агентами нужно указать ее в свойствах source Вход:

▼ Агент

Новый агент:

Изменить размеры:

Добавить агентов в: Популяцию по умолчанию
 Другую популяцию агентов

Популяция агентов:

Выталкивать агентов:

► Действия

Теперь популяция будет источником сведений о состоянии процесса обработки.

Добавьте из палитры Статистика компонент Статистика и задайте ему такие свойства:

МониторСтат - Статистика

Имя: Исключить

Видимость: да

 Дискретная
  Непрерывная

Значение:

Обновлять данные автоматически
 Не обновлять данные автоматически

Использовать модельное время Использовать календарные даты

Время первого обновления: минуты

Дата обновления:

Период: минуты

Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▶ Описание

Этот набор данных будет накапливать данные о количестве мониторов в системе каждую минуту.

Добавьте из палитры Статистика компонент Набор данных и задайте ему такие свойства:

Настройки - Набор данных

Имя: Отображать имя Исключить

Видимость: да

Использовать время в качестве значения по оси X

Значение по оси X:

Значение по оси Y:

Хранить до последних измерений

Обновлять данные автоматически
 Не обновлять данные автоматически

Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▶ Описание

Изначально этот набор данных пустой, заполним его позже.

Добавьте из палитры Статистика компонент «Данные гистограммы» и задайте ему такие свойства:

СделанoDS - Данные гистограммы

Имя: Отображать имя

Исключить

Видимость: да

Значение:

Кол-во интервалов:

Считать CDF

Вычислять процентили: Нижний: Верхний:

Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▼ Диапазон значений

Выбирается автоматически

Фиксированный

Нач. размер интервала:

▼ Обновление данных

Обновлять данные автоматически

Не обновлять данные автоматически

Для заполнения этих наборов данных дополним действия блока Выход:

Выход - Sink

Имя: Отображать имя

Исключить

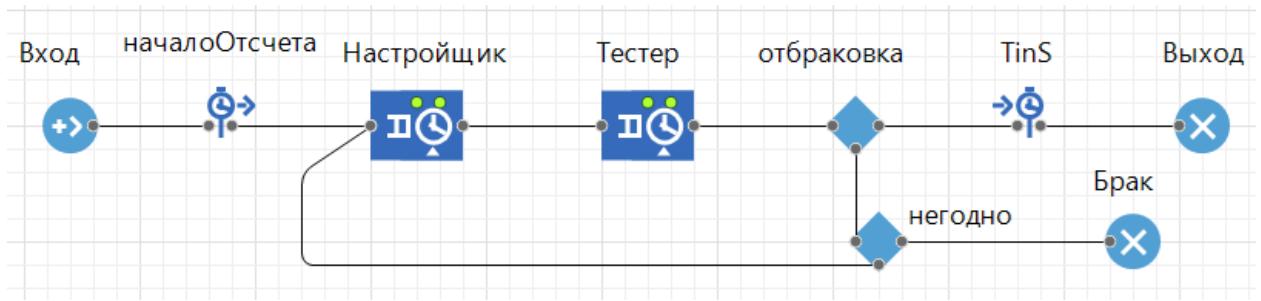
▼ Действия

При входе:

▶ Специфические

В наборе данных Настройки будет собираться информация о выполненном количестве настроек для каждого монитора. В наборе данных СделаноДС будет запоминаться количество успешно сделанных мониторов.

Для моделирования отбраковки в технологическом процессе неудачных экземпляров добавим еще один блок SelectOutput в ветвь обратной связи после первого SelectOutput и еще один блок sink с именем Брак.



Зададим для нового ветвления свойства в виде условия отбраковки в таком виде:

◇ **негодно - SelectOutput**

Имя:	<input type="text" value="негодно"/>	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Выход true выбирается:	=	<input type="radio"/> Заданной вероятностью <input checked="" type="radio"/> При выполнении условия
Условие:	<input type="button" value="agent.СчетНастроек > 3"/>	
▼ Действия		
При входе:	<input type="button"/>	
При выходе (true):	<input type="button"/>	
При выходе (false):	<input type="button"/>	
► Специфические		

Теперь запустите модель и дождитесь появления бракованных мониторов, т.е. срабатывания блока Брак. Убедитесь в заполнении всех наборов данных.

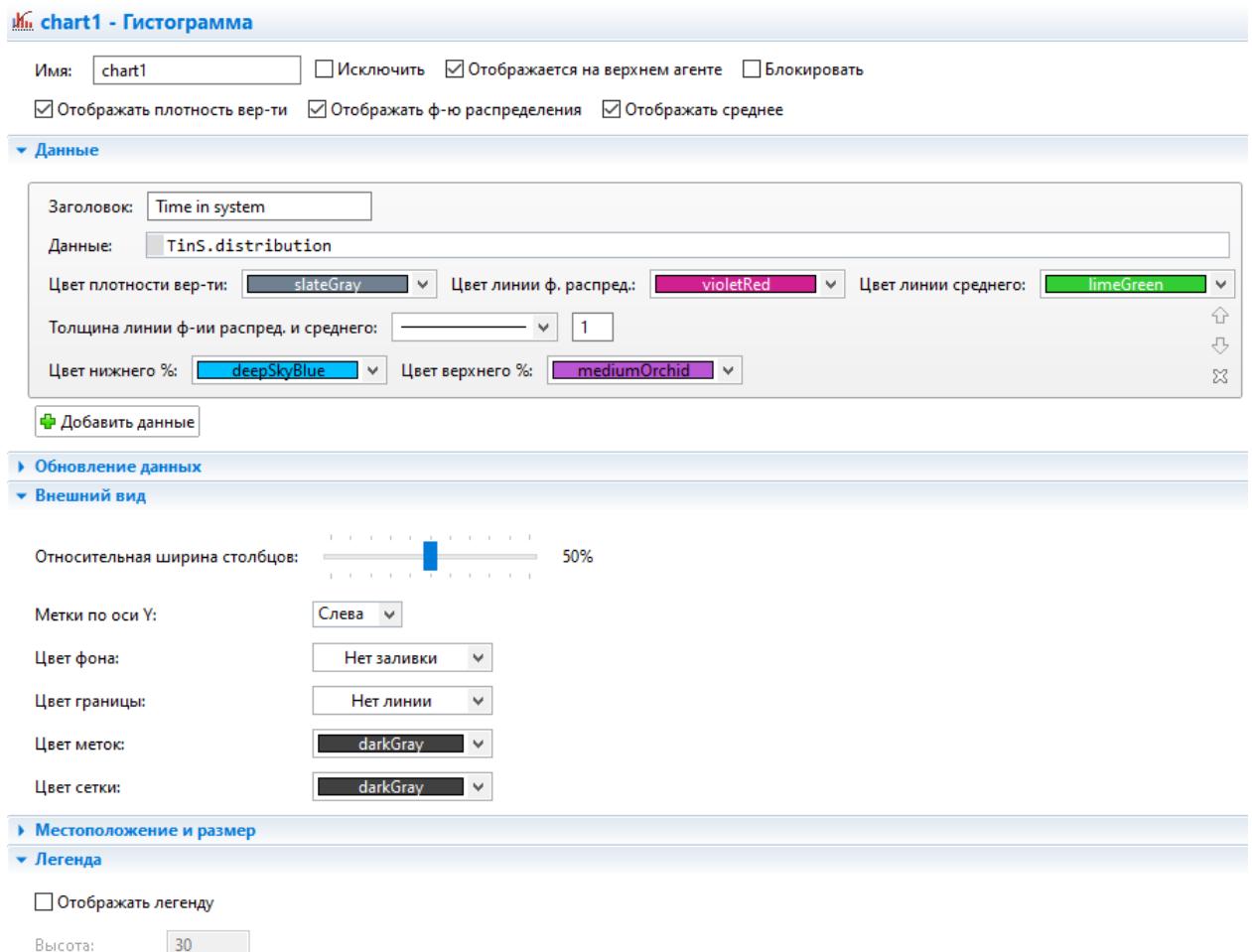
Фаза 3. Настройка визуализации статистики в простом эксперименте

Добавим компоненты визуализации данных.

Для оценки времени обработки мониторов в системе воспользуемся гистограммой.

Для наблюдения событий обработки мониторов в системе воспользуемся графиком.

Из палитры Статистика добавьте в модель Гистограмму и настройте ее свойства так:



Из палитры Статистика добавьте в модель Временной график и настройте его свойства так:

▼ Внешний вид

Метки по оси X:	Снизу
Метки по оси Y:	Слева
Формат временной оси:	Единицы модельного времени
Цвет фона:	Нет заливки
Цвет границы:	Нет линии
Цвет меток:	darkGray
Цвет сетки:	darkGray
<input checked="" type="checkbox"/> Рисовать линию	<input type="checkbox"/> Заливка области под линией
Интерполяция:	<input type="radio"/> Линейный <input checked="" type="radio"/> Ступенчатая

▶ Местоположение и размер

▼ Легенда

<input checked="" type="checkbox"/> Отображать легенду	
Высота:	30
Цвет текста:	black
Расположение:	<input type="radio"/>  <input type="radio"/>  <input type="radio"/>  <input checked="" type="radio"/> 

▶ Область диаграммы

plot1 - Временной график

Имя: Исключить Отображается на верхнем агенте Блокировать

Данные

Значение Набор данных

Заголовок:

Набор данных:

Стиль маркера:

Толщина линии: pt

Цвет:

Обновление данных

Обновлять данные автоматически
 Не обновлять данные автоматически

Использовать модельное время Использовать календарные даты

Время первого обновления:

Дата обновления:

Период:

Отображать до последних значений (для данных типа "Значение")

Масштаб

Временной диапазон: единицы мод. времени

Вертикальная шкала: Авто Фиксированный

Для наблюдения текущего значения количества мониторов в системе добавьте диаграмму.

Из палитры Статистика добавьте в модель Столбиковую диаграмму и настройте ее свойства так:

chart2 - Столбиковая диаграмма

Имя: Исключить Отображается на верхнем агенте Блокировать

Масштаб: Авто Фиксированный 100%

От: До:

Обновлять данные автоматически

Не обновлять данные автоматически

Использовать модельное время Использовать календарные даты

Время первого обновления: минуты

Дата обновления:

Период: минуты

▼ Данные

Заголовок:

Цвет:

Значение:



► Внешний вид

► Местоположение и размер

▼ Легенда

Отображать легенду

Высота:

Цвет текста:

Расположение:

Теперь настройте свойства созданного по умолчанию простого эксперимента Simulation:

✖ Simulation1 - Простой эксперимент

Имя: Исключить

Агент верхнего уровня:

Максимальный размер памяти: Мб

Пропустить экран эксперимента и запустить модель

▶ Параметры

▼ Модельное время

Режим выполнения: Виртуальное время (максимальная скорость)

Реальное время со скоростью

Остановить:

Начальное время:

Конечное время:

Начальная дата:

Конечная дата:

▼ Случайность

Генератор случайных чисел:

Случайное начальное число (уникальные "прогоны")

Фиксированное начальное число (воспроизводимые "прогоны") Начальное число:

Нестандартный генератор (подкласс класса Random):

▼ Окно

Заголовок:

Разрешить изменение масштаба и перетаскивание

Панель разработчика

Показать панель разработчика при старте модели

Запустите этот эксперимент, должен получиться вид модели примерно, как на рисунке ниже.

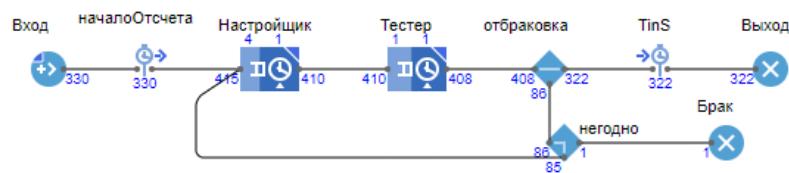
Проанализируйте полученные результаты эксперимента и сделайте выводы, отразив их в отчете.

Мониторы
Монитор [7]

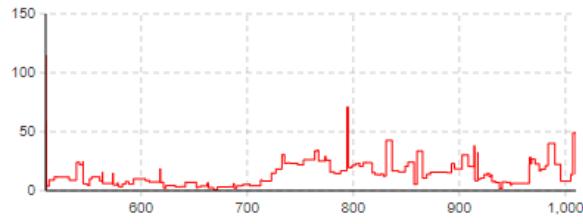
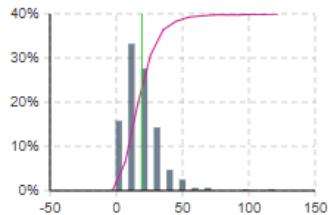
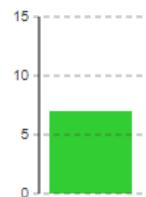
настройщик
82%
0/1

инспектор
63%
0/1

Σ МониторСтат
1,010 измерений [0...19], Среднее=0.361322 измерений ...[1,009,982, 1]
Настройки
СделаноDS
322 измерений [1...322], Среднее=161.5



мониторы 7



Time in System

Фаза 4. Эксперимент варьирования параметров

Подготовим эксперимент варьирования параметров. Использование этого эксперимента позволит сравнить различные конфигурации и отклики модели при изменении и сочетании нескольких параметров.

Сначала надо добавить в модель параметры, используемые в эксперименте.

Из палитры Агент добавьте два параметра. Один параметр МаксТюнинг с такими свойствами:

МаксТюнинг - Параметр

Имя:	МаксТюнинг	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Видимость:	<input checked="" type="radio"/> да	
Тип:	int	<input type="button" value="▼"/>
Значение по умолчанию:	<input type="button" value="≡"/> 3	
<input type="checkbox"/> Массив системной динамики		

Другой параметр ProcessTimeAdjust с такими свойствами:

ProcessTimeAdjust - Параметр

Имя:	ProcessTimeAdjust	<input checked="" type="checkbox"/> Отображать имя	<input type="checkbox"/> Исключить
Видимость:	<input checked="" type="radio"/> да		
Тип:	double	<input type="button" value="▼"/>	
Значение по умолчанию:	<input type="button" value="≡"/> 2.0		
<input type="checkbox"/> Массив системной динамики			

Редактор значения

Метка:	ВремяНастройки
Тип управления:	Бегунок
мин.:	1.0
макс.:	3.0

Параметр МаксТюнинг используем в выражении блока «негодно»:

◇ негодно - SelectOutput

Имя:	<input type="text" value="негодно"/>	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Выход true выбирается:	<input type="radio"/> Заданной вероятностью <input checked="" type="radio"/> При выполнении условия	
Условие:	<input type="text" value="agent.СчетНастроек > МаксТюнинг"/>	

Параметр ProcessTimeAdjust используем в выражении блока «Настройщик»:

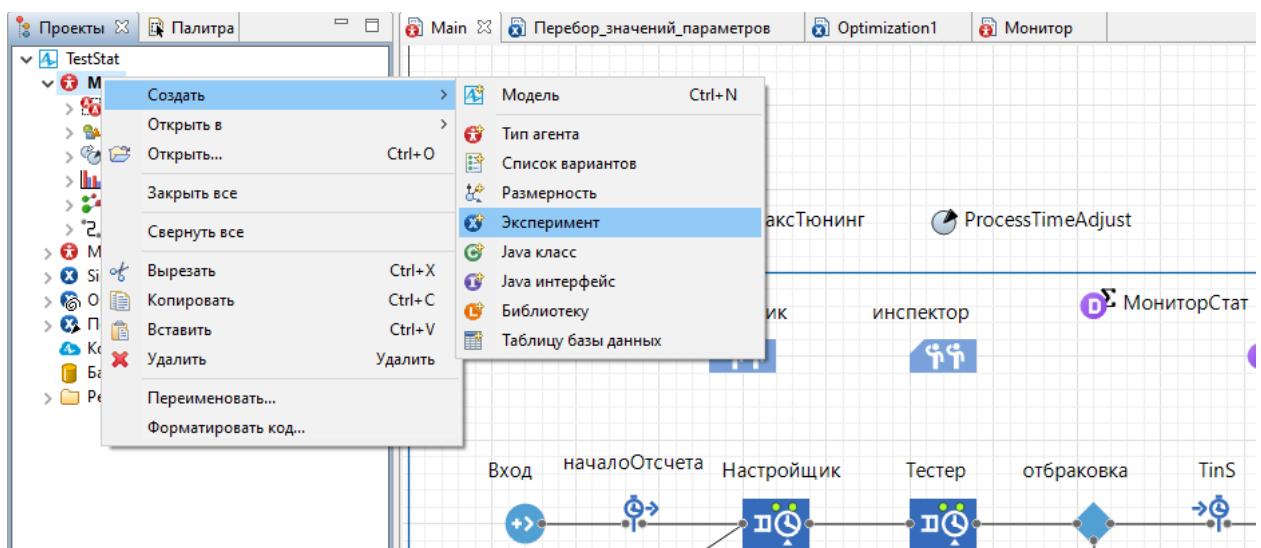
Настройщик - Service

Имя:	<input type="text" value="Настройщик"/>	<input checked="" type="checkbox"/> Отображать имя	<input type="checkbox"/> Исключить
Захватить:	<input type="radio"/> (альтернативный) набор ресурсов <input checked="" type="radio"/> ресурсы одного типа		
Тип ресурсов:	<input type="radio"/> настройщик		
Количество ресурсов:	<input type="text" value="1"/>		
Вместимость очереди:	<input type="text" value="200"/>		
Максимальная вместимость:	<input type="radio"/>		
Время задержки:	<input type="text" value="exponential(1/ProcessTimeAdjust)"/> минуты		

Теперь можно добавить новый эксперимент.

В панели Проект щелкните правой кнопкой мыши на модели Main и из меню выберите Создать, и далее - Эксперимент. В появившемся окне Тип эксперимента выберите – Варьирование параметров.

В поле Имя введите имя (без пробелов), например, Перебор_значений_параметров_1. Нажмите Готово.



Теперь в свойствах эксперимента установите такие настройки:

Перебор_значений_параметров - Эксперимент варьирования параметров

Имя: Исключить

Агент верхнего уровня:

Максимальный размер памяти: Мб

Параметры

Параметры: Варьировать в диапазоне Произвольно

Кол-во "прогонов":

Параметр	Тип	Значение		
		Мин.	Макс.	Шаг
МаксТюнинг	Диапазон	1	5	1
ВремяНастройки	Диапазон	1	3	0.3

Модельное время

Остановить:

Начальное время: Конечное время:

Начальная дата: Конечная дата:

Случайность

Генератор случайных чисел:

Случайное начальное число (уникальные "прогоны")

Фиксированное начальное число (воспроизводимые "прогоны") Начало

Нестандартный генератор (подкласс класса Random):

Репликации

Использовать репликации

Фиксированное количество репликаций

Кол-во репликаций за итерацию:

После этого нажмите кнопку «Создать интерфейс».

Появится прототип интерфейса примерно такого вида:

TestStat : перебор значений-1

Итерация: ?

Параметры

МаксТюнинг ?
Время настройки ?

Добавьте в интерфейс эксперимента компоненты визуализации результатов.

Из палитры Статистика добавьте объект Данные гистограммы, и задайте ему такие свойства:

 итогNIS - Данные гистограммы

Имя: Отображать имя Исключить

Видимость: да

Значение:

Кол-во интервалов:

Считать CDF

Вычислять процентили: Нижний: Верхний:

 Вести журнал в базе данных
[Вести журнал выполнения модели](#)

▼ Диапазон значений

Выбирается автоматически
 Фиксированный
Нач. размер интервала:

▼ Обновление данных

Обновлять данные автоматически
 Не обновлять данные автоматически

В этой гистограмме будет отображаться количество мониторов в системе.

Из палитры Статистика добавьте еще объект Данные гистограммы, и задайте ему такие свойства:

итогНастройки - Данные гистограммы

Имя:	итогНастройки	<input checked="" type="checkbox"/> Отображать имя	<input type="checkbox"/> Исключить
Видимость:	<input checked="" type="radio"/> да		
Значение:	1		
Кол-во интервалов:	10		
<input checked="" type="checkbox"/> Считать CDF <input type="checkbox"/> Вычислять процентили: Нижний: <input type="text" value="10"/> Верхний: <input type="text" value="10"/> ! <input checked="" type="checkbox"/> Вести журнал в базе данных Вести журнал выполнения модели			
▼ Диапазон значений <input checked="" type="radio"/> Выбирается автоматически <input type="radio"/> Фиксированный Нач. размер интервала: <input type="text" value="0.1"/>			
▼ Обновление данных <input type="radio"/> Обновлять данные автоматически <input checked="" type="radio"/> Не обновлять данные автоматически			

В этой гистограмме будет отображаться количество выполненных настроек мониторов.

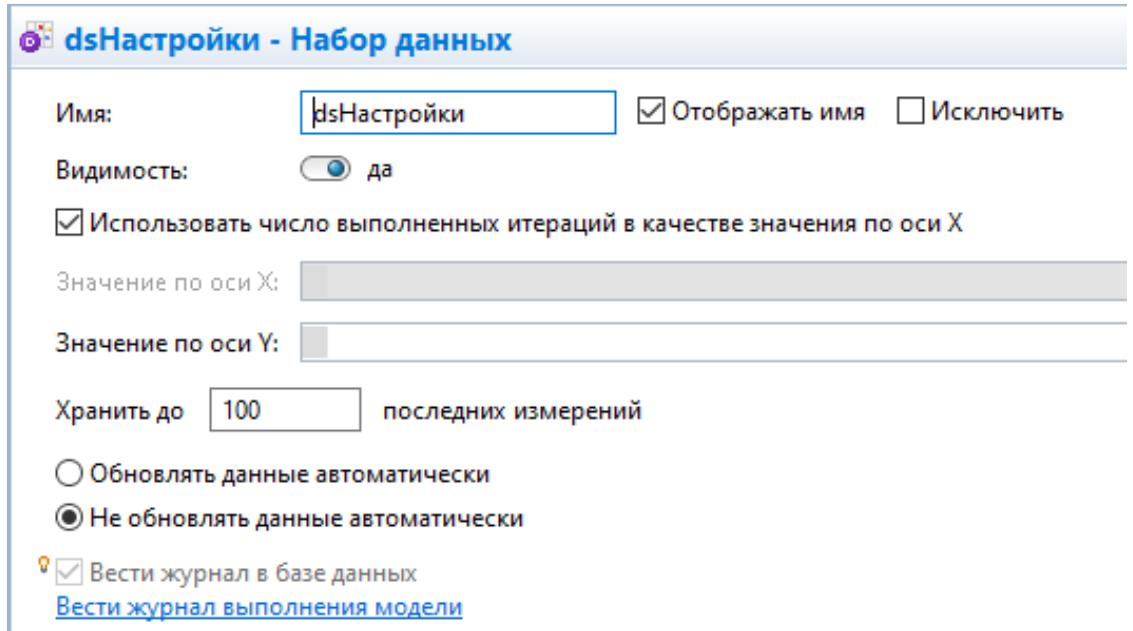
Из палитры Статистика добавьте еще объект Данные гистограммы, и задайте ему такие свойства:

итогTinS - Данные гистограммы

Имя:	итогTinS	<input checked="" type="checkbox"/> Отображать имя	<input type="checkbox"/> Исключить
Видимость:	<input checked="" type="radio"/> да		
Значение:			
Кол-во интервалов:	10		
<input checked="" type="checkbox"/> Считать CDF <input type="checkbox"/> Вычислять процентили: Нижний: <input type="text" value="10"/> Верхний: <input type="text" value="10"/> ! <input checked="" type="checkbox"/> Вести журнал в базе данных Вести журнал выполнения модели			
▼ Диапазон значений <input checked="" type="radio"/> Выбирается автоматически <input type="radio"/> Фиксированный Нач. размер интервала: <input type="text" value="0.1"/>			
▼ Обновление данных <input type="radio"/> Обновлять данные автоматически <input checked="" type="radio"/> Не обновлять данные автоматически			

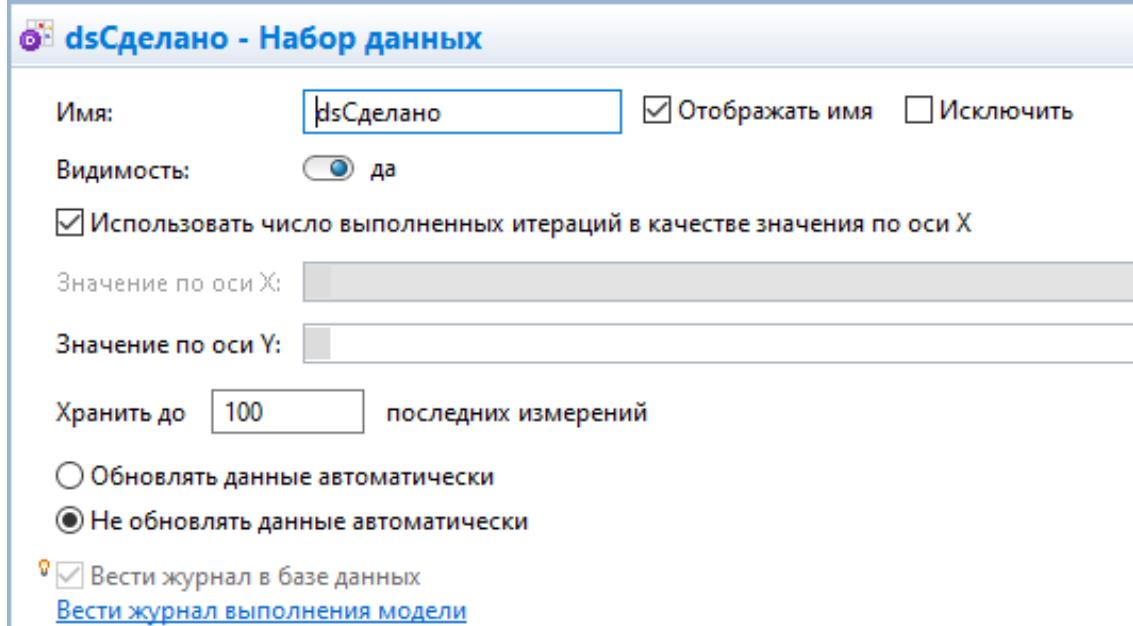
В этой гистограмме будет отображаться время выполнения настройки мониторов.

Из палитры Статистика добавьте объект Набор данных, и задайте ему такие свойства:



В этом массиве будут собираться данные о количестве выполненных настроек мониторов.

Из палитры Статистика добавьте еще объект Набор данных, и задайте ему такие свойства:



В этом массиве будут собираться данные о количестве годных мониторов.

Для заполнения этих наборов данных дополним действия в свойствах эксперимента:

▼ Действия Java

Код инициализации эксперимента:

```
[REDACTED]
```

Действие перед запуском каждого эксперимента:

```
[REDACTED]
```

Действие перед "прогоном" модели:

```
[REDACTED]
```

Действие после "прогона" модели:

```
итогNIS.add(root.МониторСтат.max());  
итогНастройки.add(root.Настройки.getYMax());  
итогTinS.add(root.TinS.distribution.mean());  
dsНастройки.add(root.Настройки.getYMax());  
dsСделано.add(root.СделаноДС.max());
```

Действие после итерации:

```
[REDACTED]
```

В разделе свойств Случайность установите Случайное начальное число (уникальные прогоны).

Для проверки настроек запустите созданный эксперимент и убедитесь, что наборы данных собирают значения.

Добавим диаграммы на панель эксперимента.

Из палитры Статистика добавьте объект Гистограмма, и задайте ей такие свойства:

chart01 - Гистограмма

Имя: Исключить Блокировать
 Отображать плотность вер-ти Отображать ф-ю распределения Отображать среднее

▼ Данные

Заголовок:

Данные:

Цвет плотности вер-ти: Цвет линии ф. распред.:

Толщина линии ф-ии распред. и среднего: 1

Цвет нижнего %: Цвет верхнего %:

Добавить данные

► Обновление данных

▼ Внешний вид

Относительная ширина столбцов: 70%

Метки по оси Y:

Цвет фона:

Цвет границы:

Цвет меток:

Цвет сетки:

► Местоположение и размер

▼ Легенда

Отображать легенду

Ширина:

Цвет текста:

Расположение:

Из палитры Статистика добавьте еще объект Гистограмма, и задайте ей такие свойства:

chart02 - Гистограмма

Имя: Исключить Блокировать

Отображать плотность вер-ти Отображать ф-ю распределения Отображать среднее

▼ Данные

Заголовок:

Данные:

Цвет плотности вер-ти: Цвет линии ф. распред.: Цвет линии среднего:

Толщина линии ф-ии распред. и среднего: 1

Цвет нижнего %: Цвет верхнего %:

► Обновление данных

▼ Внешний вид

Относительная ширина столбцов: 80%

Метки по оси Y:

Цвет фона:

Цвет границы:

Цвет меток:

Цвет сетки:

► Местоположение и размер

▼ Легенда

Отображать легенду

Высота:

Цвет текста:

Расположение:

Из палитры Статистика добавьте еще объект Гистограмма, и задайте ей такие свойства:

chart - Гистограмма

Имя: Исключить Блокировать

Отображать плотность вер-ти Отображать ф-ю распределения Отображать среднее

▼ Данные

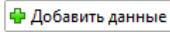
Заголовок:

Данные:

Цвет плотности вер-ти: Цвет линии ф. распред.: Цвет линии среднего:

Толщина линии ф-ии распред. и среднего: 1

Цвет нижнего %: Цвет верхнего %:



► Обновление данных

▼ Внешний вид

Относительная ширина столбцов: 69%

Метки по оси Y:

Цвет фона:

Цвет границы:

Цвет меток:

Цвет сетки:

► Местоположение и размер

▼ Легенда

Отображать легенду

Ширина:

Цвет текста:

Расположение:     

Из палитры Статистика добавьте объект График, и задайте ему такие свойства:

plot01 - График

Имя: Исключить Блокировать

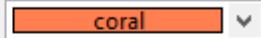
▼ Данные

Значение Набор данных

Заголовок:

Набор данных:

Стиль маркера: 

Цвет:  coral



▼ Обновление данных

Обновлять данные автоматически

Не обновлять данные автоматически

Отображать до последних значений (для данных типа "Значение")

► Масштаб

▼ Внешний вид

Метки по оси X:

Метки по оси Y:

Цвет фона:

Цвет границы:

Цвет меток:

Цвет сетки:

Рисовать линию

► Местоположение и размер

▼ Легенда

Отображать легенду

Высота:

Цвет текста:

Расположение: 

Из палитры Статистика добавьте еще объект График, и задайте ему такие свойства:

plot - График

Имя: Исключить Блокировать

▼ Данные

Значение Набор данных

Заголовок:

Набор данных:

Стиль маркера: 

Цвет:  green



▼ Обновление данных

Обновлять данные автоматически

Не обновлять данные автоматически

Отображать до последних значений (для данных типа "Значение")

► Масштаб

▼ Внешний вид

Метки по оси X:

Метки по оси Y:

Цвет фона:

Цвет границы:

Цвет меток:

Цвет сетки:

Рисовать линию

► Местоположение и размер

▼ Легенда

Отображать легенду

Высота:

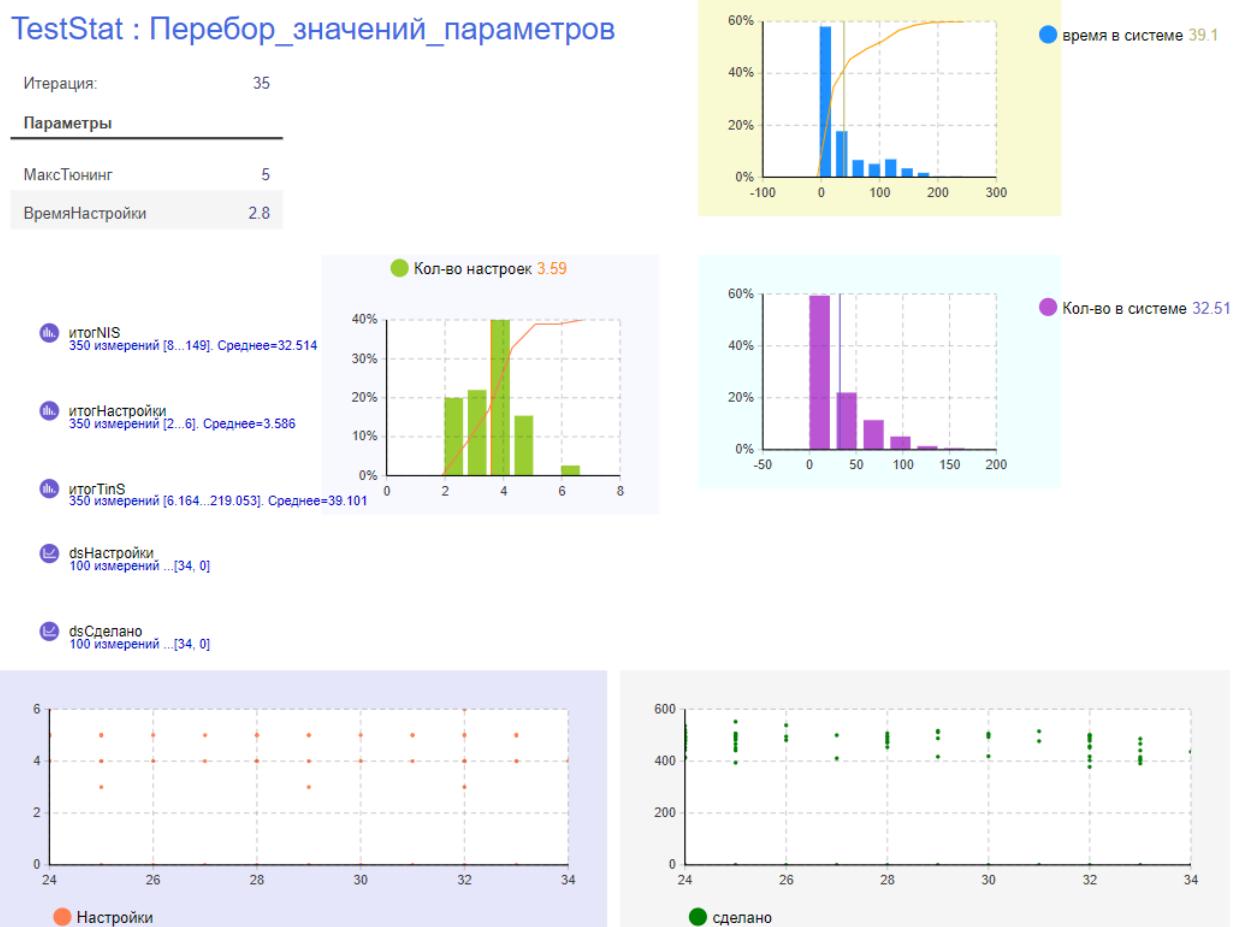
Цвет текста:

Расположение: 

Расположите на панели эксперимента графики и диаграммы примерно в таком виде:



Запустите созданный эксперимент и проанализируйте полученные результаты:

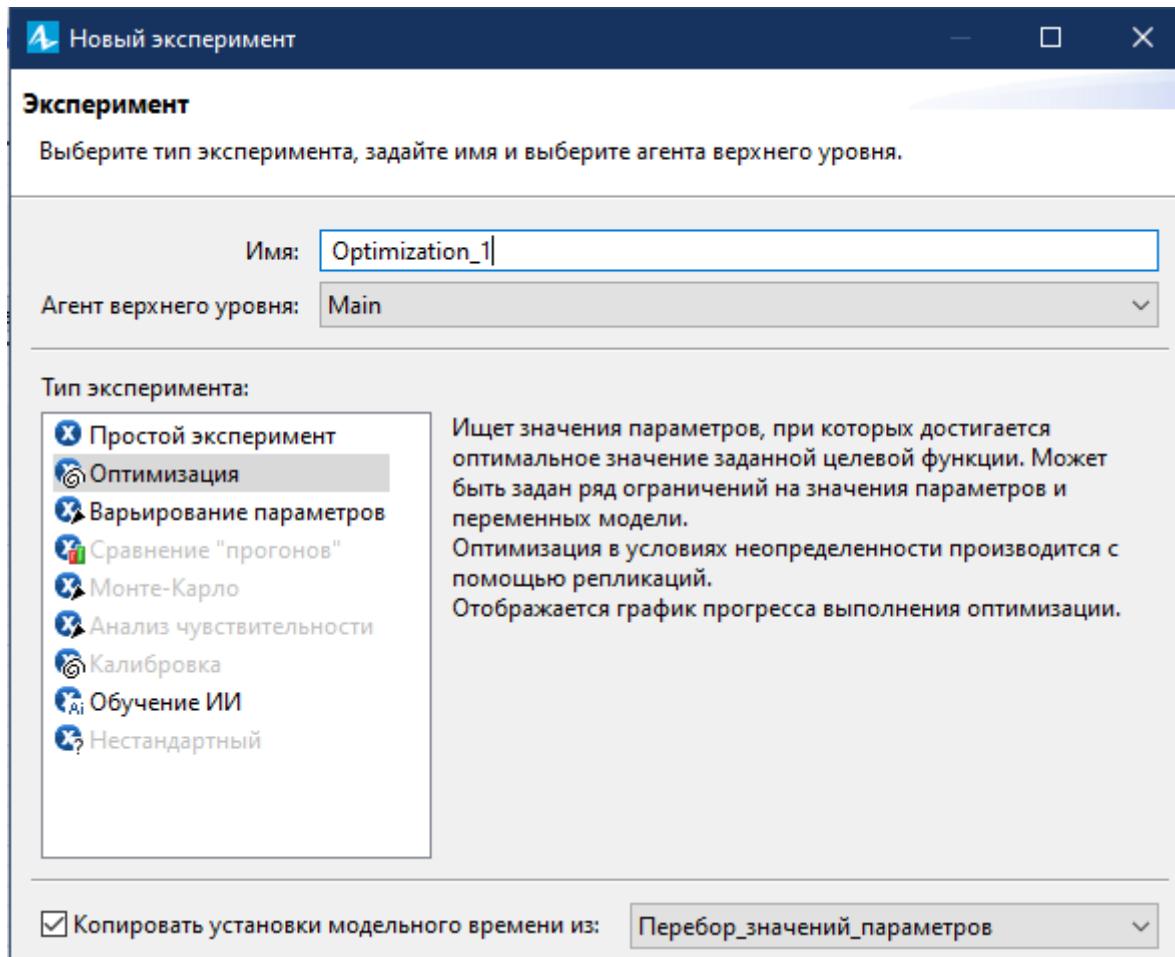


Проанализируйте полученные результаты эксперимента и сделайте выводы, отразив их в отчете.

Фаза 5. Эксперимент «Оптимизация»

Теперь добавим новый эксперимент другого типа.

В панели Проект щелкните правой кнопкой мыши на модели Main и из меню выберите Создать, и далее - Эксперимент. В появившемся окне Тип эксперимента выберите – Оптимизация и задайте такие настройки:



Нажмите Готово.

Будет создан прототип эксперимента, создайте его интерфейс с графиком результатов итераций.

Предположим, что нам надо определить при каком варианте конфигурации параметров в системе будут получены максимальные результаты, т.е. сделано наибольшее количество мониторов, но с учетом ограничений загрузки ресурсов, например, для настройщика не более 75%, и для инспектора не более 70%.

Для этого в свойствах эксперимента установите настройки как на рисунке ниже.

Выполните эксперимент, сохраните его результаты в отчет.

Optimization_1 - Оптимизационный эксперимент

Имя: Optimization_1 Исключить

Агент верхнего уровня: Main

Оптимизатор: OptQuest DEMO

Целевая функция: минимизировать максимизировать

`root.Выход.in.count()`

Количество итераций

Фиксированное: 750

Бесконечное

Автоматическая остановка

Максимальный размер памяти: 512 Мб

Параметры

Параметры:

Параметр	Тип	Значение			
		Мин.	Макс.	Шаг	Начальное
МаксТюнинг	дискретный	2	5	1	2
ВремяНастройки	дискретный	1.5	3.5	0.2	1.5

Модельное время

Ограничения

Требования

Требования (проверяются после "прогона" для определения того, допустимо ли найденное решение):

Вкл.	Выражение	Тип	Граница
<input checked="" type="checkbox"/>	root.настройщик.utilization()*100	<=	75.0
<input checked="" type="checkbox"/>	root.инспектор.utilization()*100	<=	70.0

Случайность

Генератор случайных чисел:

Случайное начальное число (уникальные "прогоны")

Репликации

Использовать репликации

Эксперимент настроен на воспроизводимые прогоны. См. секцию свойств эксперимента "Случайность".

Фиксированное количество репликаций

Кол-во репликаций за итерацию: 10

Заметьте, что в настройках действий Java есть операторы для сброса и обновления результатов прогонов в эксперименте.

▼ Действия Java

Код инициализации эксперимента:

```
[ ]
```

Действие перед запуском каждого эксперимента:

```
datasetCurrentObjective.reset();
datasetBestInfeasibleObjective.reset();
datasetBestFeasibleObjective.reset();
```

Действие перед "прогоном" модели:

```
[ ]
```

Действие после "прогона" модели:

```
[ ]
```

Действие после итерации:

```
if (isBestSolutionFeasible()) {
    datasetBestFeasibleObjective.update();
}
if (!isCurrentSolutionFeasible()) {
    bestInfeasibleObjective = min( bestInfeasibleObjective, getCurrentObjectiveValue() );
}
if (bestInfeasibleObjective != Double.POSITIVE_INFINITY) {
    datasetBestInfeasibleObjective.update();
}
```

Действие по окончании эксперимента:

```
[ ]
```

TestStat : Optimization_1

	Текущее	Лучшее
Итерация:	44 недопуст.	18
Функционал	442.5	489
Параметры		Copy best
МаксТюнинг	5	5
ВремяНастройки	2.7	2.7



Измените в настройках эксперимента оптимизатор на – Генетический.

Выполните эксперимент, сохраните его результаты в отчет.

TestStat : Optimization_1

	Текущее	Лучшее
Итерация:	22 недопуст.	21
Функционал	495	474.7
Параметры	Copy best	
МаксТюнинг	5	5
ВремяНастройки	1.9	1.9



После выполнения эксперимента можно в панели разработчика увидеть количество выполненных вычислений операторов модели для оценки сложности оптимизационной задачи:

Соб/сек: 0 Кадр/сек: 40 Шаг: 2,918,280

После завершения настроек запустите этот эксперимент, получите результат, скопируйте лучшие значения в буфер нажатием кнопки «Скопировать лучшее» (Copy best).

Перейдите к Простому эксперименту, вставьте эти значения из буфера в Простой эксперимент, нажав на кнопку, запустите его, получите результаты и графики.

Проанализируйте полученные результаты простого и оптимизационного эксперимента, сделайте выводы, отразив их в отчете.

Сделайте отчет в формате docx, приложите к отчету модель и пришлите на почту преподавателю.