

# **Modelovanie a simulácia**

## **Dynamická simulácia**

**AnyLogic**

**3. cvičenie**

# Ciele cvičenia

- Použitie objektov modelu v AnyLogic-u pre modelovanie:
  - Pridelovania vlastností (atribútov) zákazníkov a ich využívanie pre sledovanie výstupných údajov
  - Empirického rozdelenia pravdepodobnosti
  - Zdrojov (obsluhujúcich entít)
  - Pridelenie zdrojov na obsluhu
  - Priority pri čakaní vo fronte

# Zadanie

- Modelovaný systém – doplnenie a úprava modelu z cvičenia 2
  - Príchod návštevníkov do múzea – 1 až 4 návštevníci pri jednom vstupe
  - Atribúty návštevníkov (vlastnosti) – čas začiatku čakania, ŤZP, typ expozície
  - Nákup vstupeniek – vrátenie predbiehania vo fronte
  - Obsluha – dve zamestnankyne
- Návštevník príde do múzea, kúpi si vstupenku (ak musí, tak počká v rade pred pokladňou; ŤZP majú pri čakaní prednosť), odchádza na prehliadku múzea

# Zadanie

- Vstupné údaje
  - Počet návštevníkov na jeden vstup – 1 – 25%, 2 – 30%, 3 – 35%, 4 – 10%
  - 5% návštevníkov – ŤZP
- Výstupné údaje – do modelu pridať alebo prispôbiť
  - Priemerné časy čakania – všetci, ŤZP, zdraví
  - Maximálna a priemerná dĺžka frontu
  - Histogram pre časy čakania návštevníkov

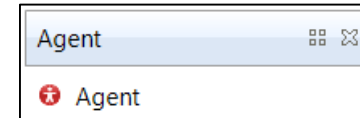
# Nové prvky simulačného modelu

- Empirické rozdelenie pravdepodobnosti – *customDistributionArrivals*
- Atribút s informáciou o zdraví návštevníka – *parDisability*
- Zdroje (obsluhujúce entity) – *resPoolPersonnel*
- Modelovanie obsluhy iným objektom – *serviceSell*
- Štatistika pre dĺžku frontu pred pokladňou – *statQueueLength*
- Štatistiky pre časy čakania – *statWaitingTimeDisability*, *statWaitingTimeOthers*
- Údaje pre histogram časov čakania – *hdWaitingTime*

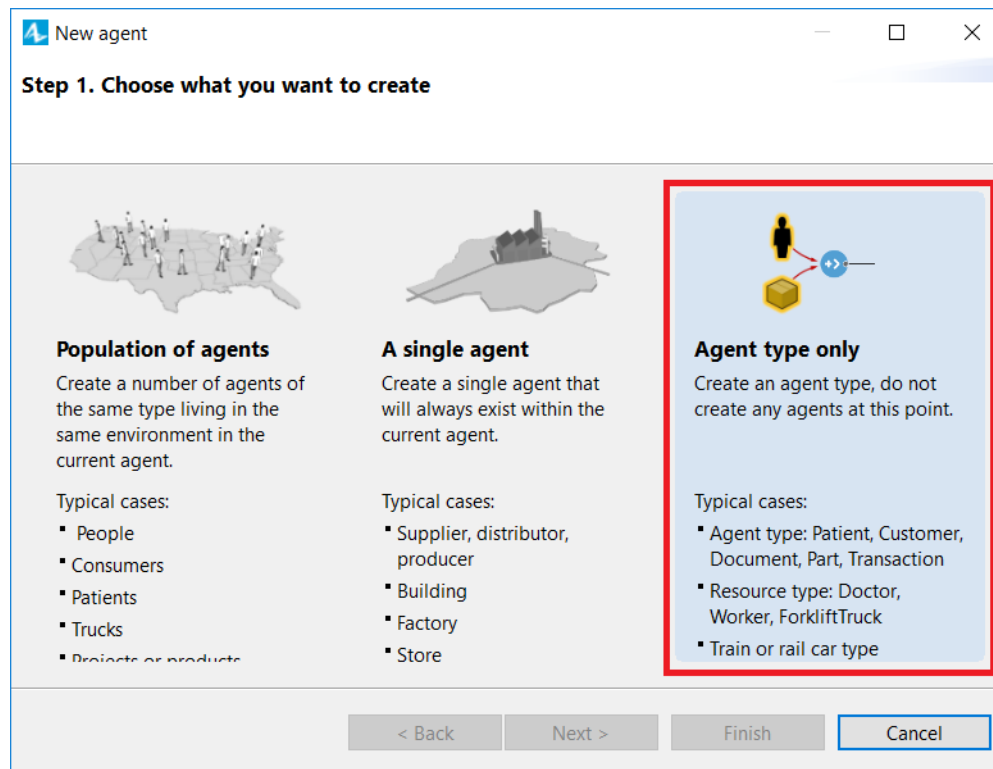
# Nová trieda - návštevník

- Objekt **Agent** – knižnica **Agent**

- Chytiť a premiestniť na ľubovoľné miesto na pracovnej ploche v rámci triedy **Main**



1.



# Nová trieda - návštevník

- Objekt **Agent** – knižnica **Agent**

2.

The screenshot shows a software window titled "New agent" with standard Windows window controls (minimize, maximize, close). The window is at "Step 2. Creating new agent type".

Inside the window, there is a text field labeled "Agent type name:" containing the word "Visitor".

Below the text field are two radio button options:

- ☒ Create the agent type "from scratch"
- ☐ Use database table

Under the "Use database table" option, there is a smaller text label: "I have agent data stored in a database".

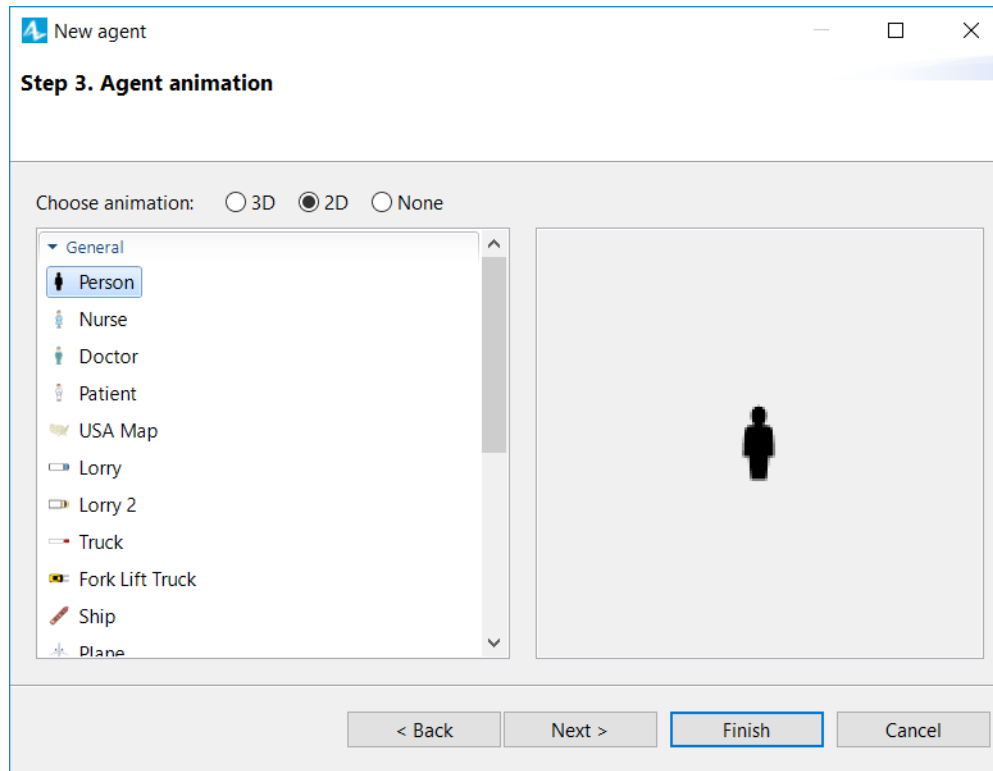
Below these options is a checked checkbox labeled "Agent will be used in flowcharts as:". To its right is a dropdown menu. The dropdown menu is open, showing a list of options: "Agent" (highlighted in blue), "Resource unit", "Pedestrian", "Rail Car", "Train", and "Car".

At the bottom of the window, there are four buttons: "< Back", "Next >", "Finish" (which is highlighted with a blue border), and "Cancel".

# Nová trieda - návštevník

- Objekt **Agent** – knižnica **Agent**

3.





# Nový atribút – začiatok čakania

- Objekt **Agent** – knižnica **Agent**

4.

New agent

Step 4. Agent parameters

Please fix the parameters you want to see in your Visitor:

Parameters
parWaitingStartTime
<add new...>

Parameter: parWaitingStartTime

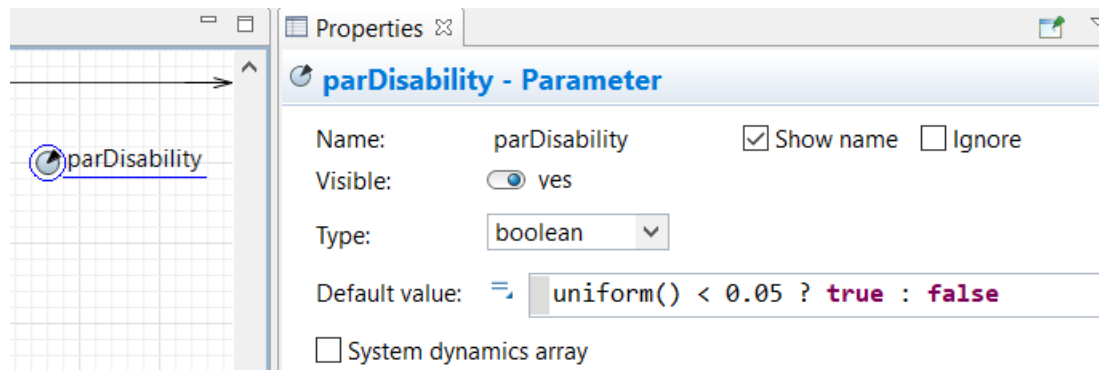
Type: double

Default value: 0

< Back   Next >   **Finish**   Cancel

# Ďalší atribút – zdravotný stav

- Objekt **Parameter** – knižnica **Agent**  **Parameter**
  - Chytiť a premiestniť na ľubovoľné miesto na pracovnej ploche v rámci triedy **Visitor**



# Modelovanie rôzneho počtu zákazníkov pri príchode

- Objekt **Custom Distribution** – knižnica **Agent**

The screenshot shows the 'Properties' window for a 'customDistributionArrivals' object in NetLogo. The window is titled 'customDistributionArrivals - Custom Distribution'. The settings are as follows:

- Name: customDistributionArr (with checkboxes for 'Show name' and 'Ignore')
- Visible: ☒ yes
- Type: ☐ Continuous, ☒ Discrete, ☐ Options
- Define using: ☒ Ranges, ☐ Frequency table, ☐ Observed samples

Below the settings is a 'Data' section with a checkbox 'Loaded from database' (unchecked). It contains a table with the following data:

Interval start	Interval end	Number of observati...
1	1	25.0
2	2	30.0
3	3	35.0
4	4	10.0

# Modelovanie rôzneho počtu zákazníkov pri príchode

- Úprava objektu **sourceArrival**

Properties

**sourceArrival - Source**

Name: sourceArrival ☒ Show name ☐ Ignore

Arrivals defined by: Interarrival time

Interarrival time: exponential((double)1/100)

Set agent parameters from DB: ☐

Multiple agents per arrival: ☒

Agents per arrival: customDistributionArrivals.getInt()

Limited number of arrivals: ☐

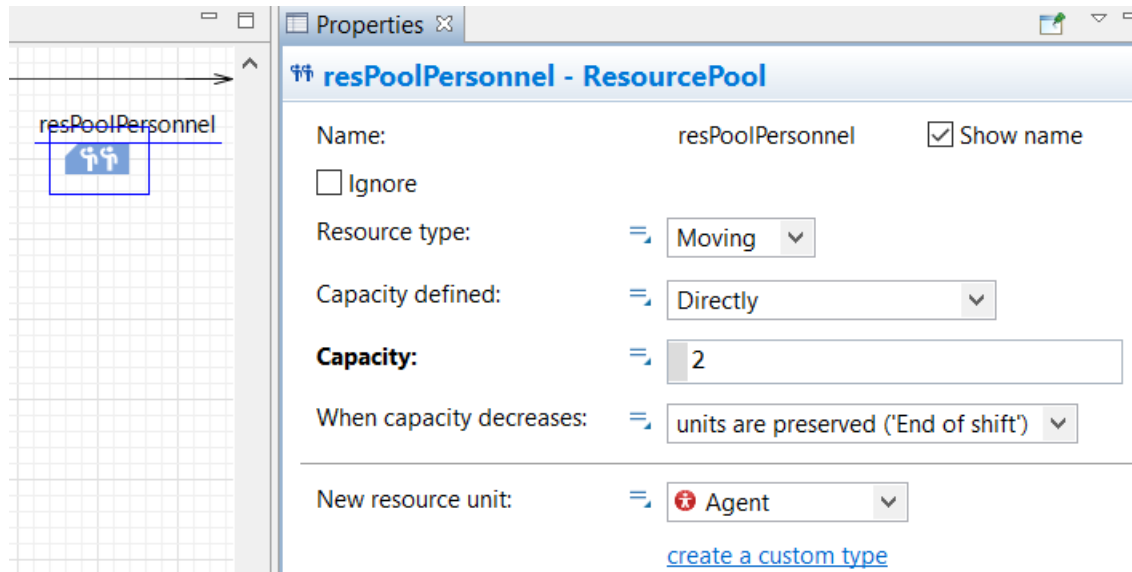
---

New agent: Visitor

Location of arrival: Not specified

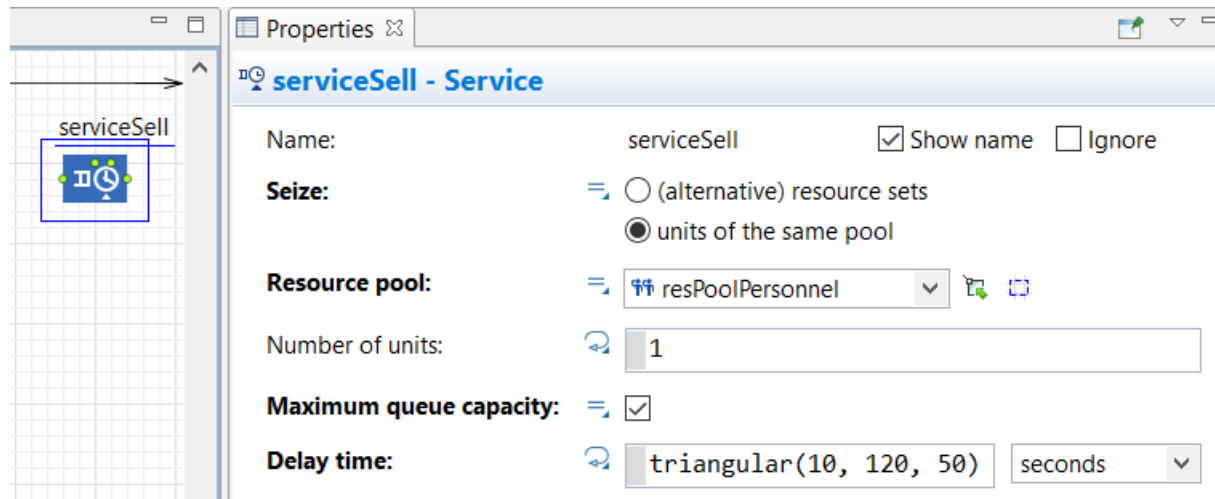
# Modelovanie zdrojov

- Objekt **Resource Pool** – knižnica **Process Modeling Library**



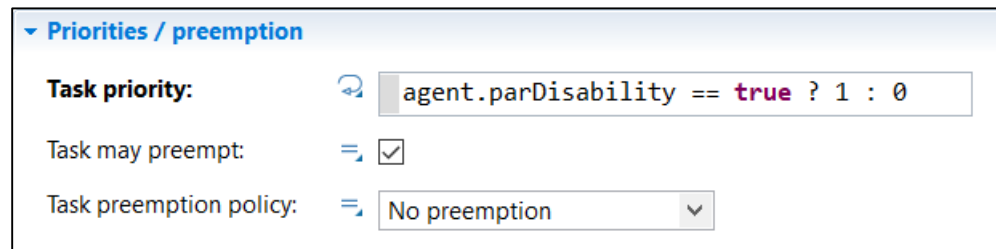
# Modelovanie pridelenia zdrojov pre obsluhu

- Objekt **Service** – knižnica **Process Modeling Library**
- Nahradenie objektov **Queue** a **Delay** (vplyv na objekty štatistiky!!!)



# Použitie priority pri čakaní vo fronte

- V objekte **Service**, časť **Priorities / preemption**
- Odkaz na atribút *parDisability* – kontrola jeho hodnoty

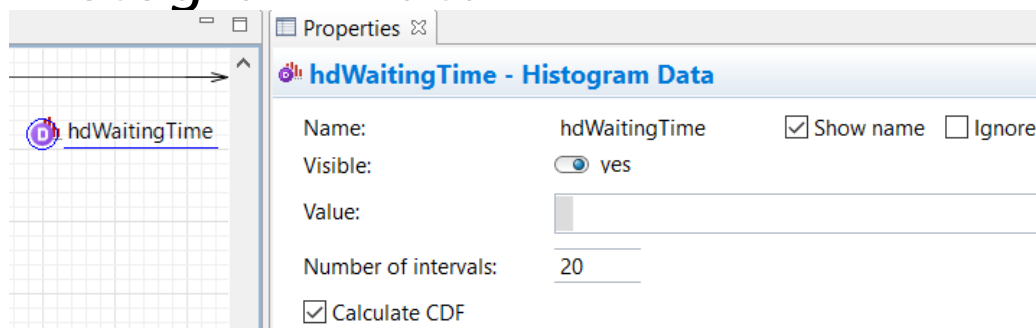


The screenshot shows a configuration window titled "Priorities / preemption". It contains three settings:

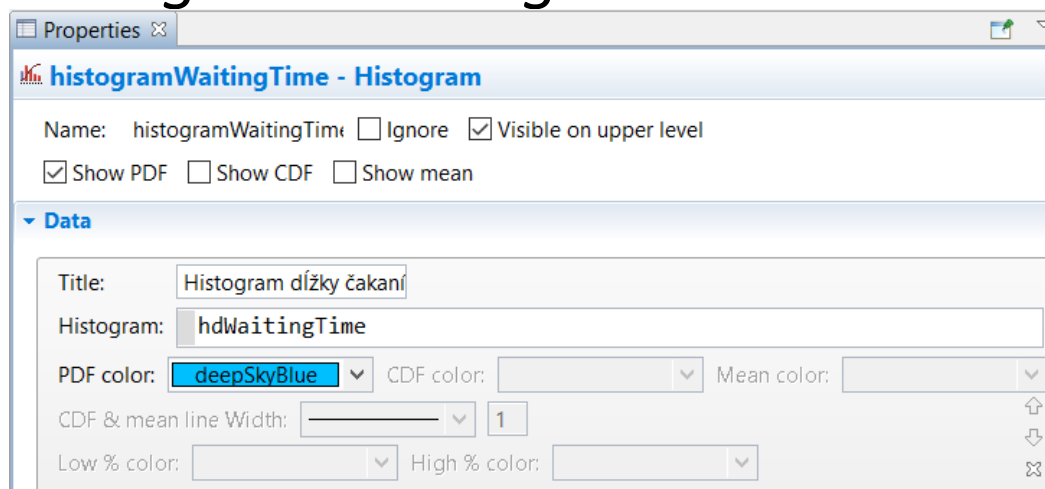
- Task priority:** A text input field containing the expression `agent.parDisability == true ? 1 : 0`. A small icon of a document with a circular arrow is to the left of the field.
- Task may preempt:** A checkbox that is currently checked. A small icon of a document with a circular arrow is to the left of the checkbox.
- Task preemption policy:** A dropdown menu currently showing "No preemption". A small icon of a document with a circular arrow is to the left of the dropdown.

# Histogram – čas čakania

- Objekt *Histogram Data*



- Objekt *histogramWaitingTime*






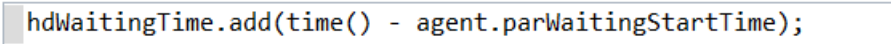










# Histogram – čas čakania

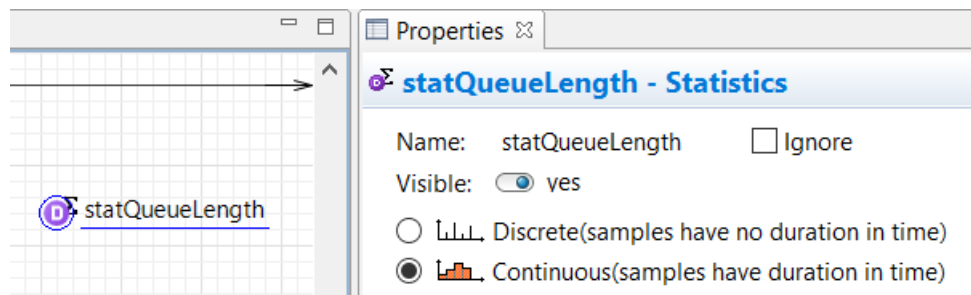
- Objekt *serviceSell*

**Actions**

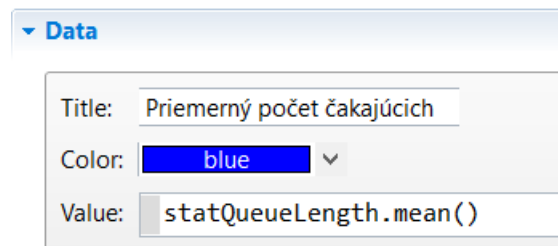
<b>On enter:</b>		
On seize unit:		
<b>On enter delay:</b>		 <pre>hdWaitingTime.add(time() - agent.parWaitingStartTime);</pre>
On at exit:		
On exit:		
On remove:		

# Oprava barChartAverageQueue

- Úprava maxima pre zobrazovanie – z 1 na 3
- Objekt *queueWaiting* z pred. verzie modelu je preč
- Pridanie nového objektu **Statistics**



- Odkaz na výstupy z objektu – *statDlзкаFrontu.mean()*



# Oprava barChartAverageQueue

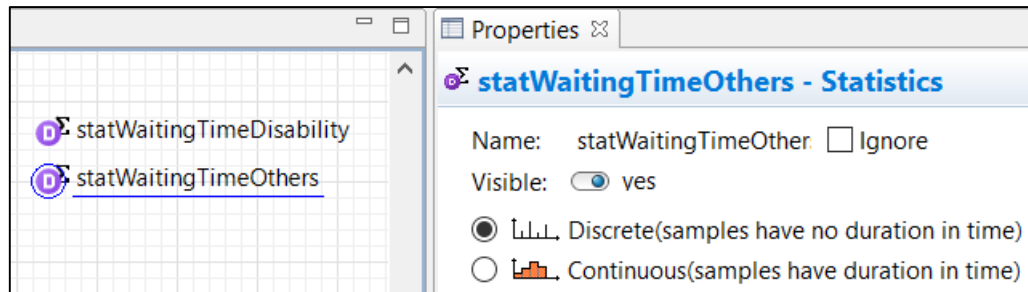
- Objekt *serviceSell*

**Actions**

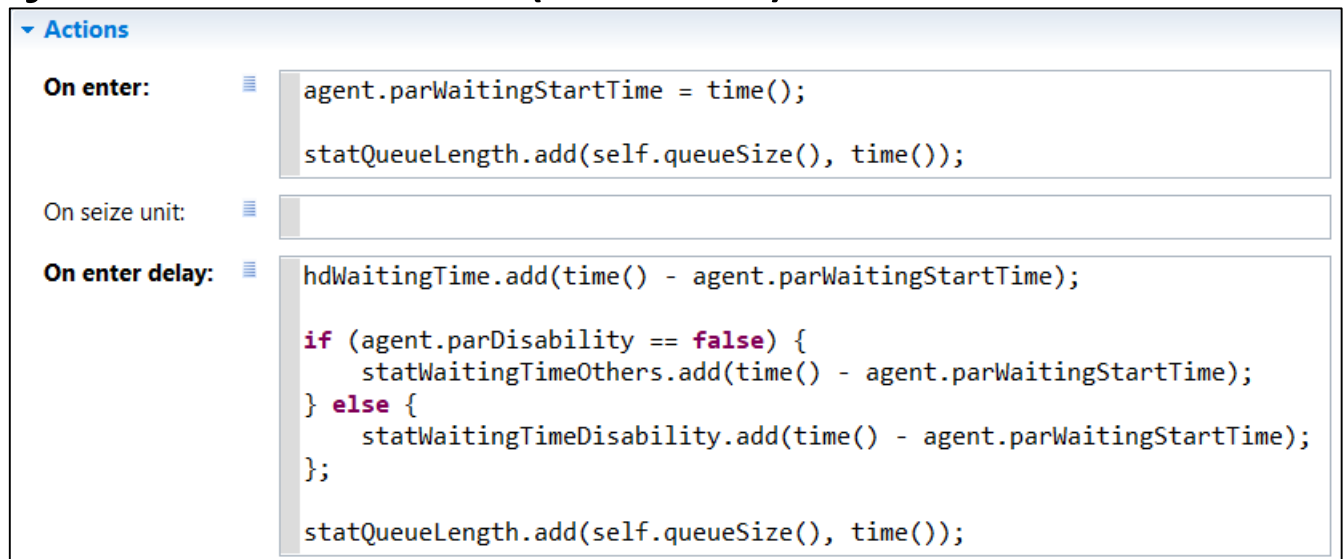
<b>On enter:</b>	<pre>statQueueLength.add(self.queueSize(), time());</pre>
On seize unit:	
<b>On enter delay:</b>	<pre>hdWaitingTime.add(time() - agent.parWaitingStartTime); statQueueLength.add(self.queueSize(), time());</pre>

# Štatistiky pre časy čakania

- Nové objekty **Statistics** (rovnaké nastavenie)

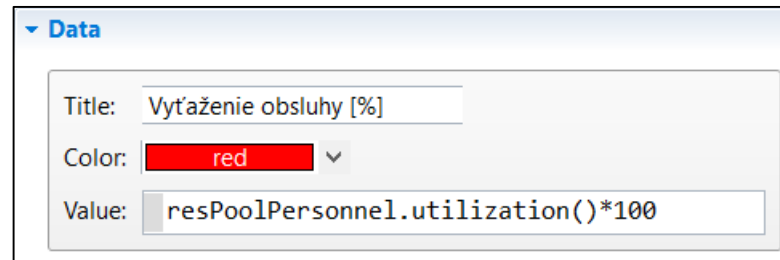


- Kód v objekte *serviceSell* (**Service**)



# Oprava barChartSellUtilization

- Objekt *serviceSell* (pôvodne ako objekt **Delay**) z predchádzajúcej verzie modelu je preč
- Odkaz na výstupy z objektu **Resource Pool** –

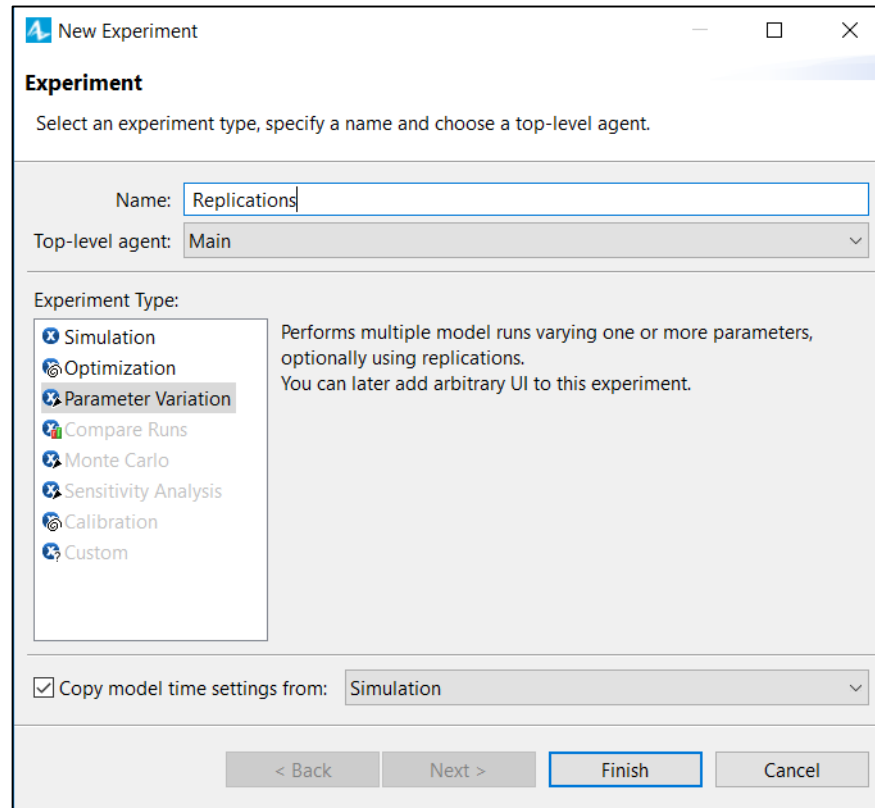


The screenshot shows a configuration window titled 'Data' with a blue header. Inside, there are three fields: 'Title' with the text 'Vyťaženie obsluhy [%]', 'Color' with a red color swatch and the text 'red', and 'Value' with the text 'resPoolPersonnel.utilization()\*100'.

Data	
Title:	Vyťaženie obsluhy [%]
Color:	red
Value:	resPoolPersonnel.utilization()*100

# Definovanie replikácií

- New – Experiment

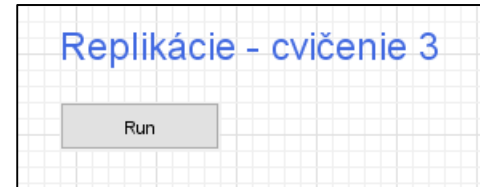


The screenshot shows a 'New Experiment' dialog box with the following elements:

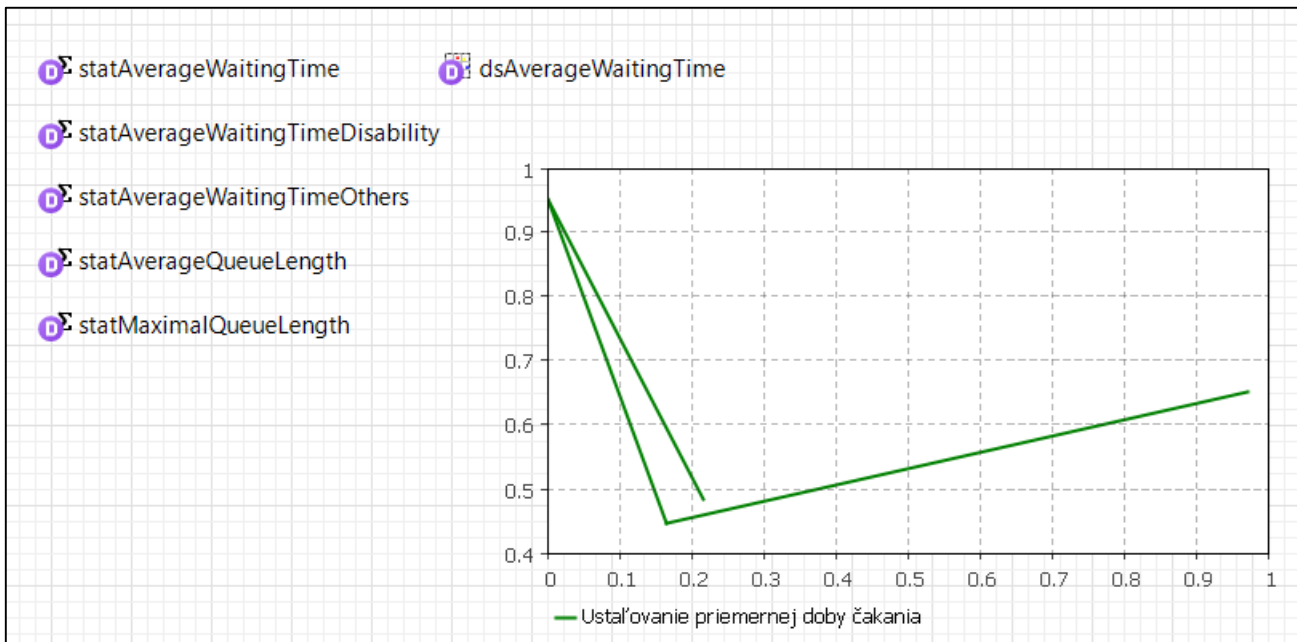
- Title Bar:** 'New Experiment' with standard window controls (minimize, maximize, close).
- Section Header:** 'Experiment'.
- Instruction:** 'Select an experiment type, specify a name and choose a top-level agent.'
- Name Field:** A text input field containing 'Replications'.
- Top-level agent:** A dropdown menu currently set to 'Main'.
- Experiment Type:** A list of options with icons:
  - Simulation (selected)
  - Optimization
  - Parameter Variation
  - Compare Runs
  - Monte Carlo
  - Sensitivity Analysis
  - Calibration
  - Custom
- Description:** Text explaining the selected type: 'Performs multiple model runs varying one or more parameters, optionally using replications. You can later add arbitrary UI to this experiment.'
- Copy settings:** A checked checkbox 'Copy model time settings from:' followed by a dropdown menu set to 'Simulation'.
- Buttons:** '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

# Definovanie replikácií

- Properties – *Create default UI*

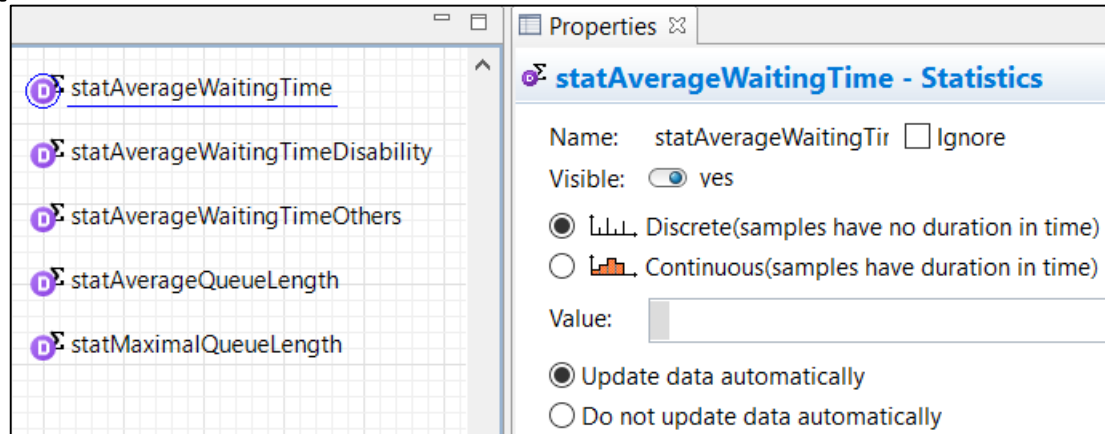


- Objekty **Statistics**, **Data Set** a **Plot**

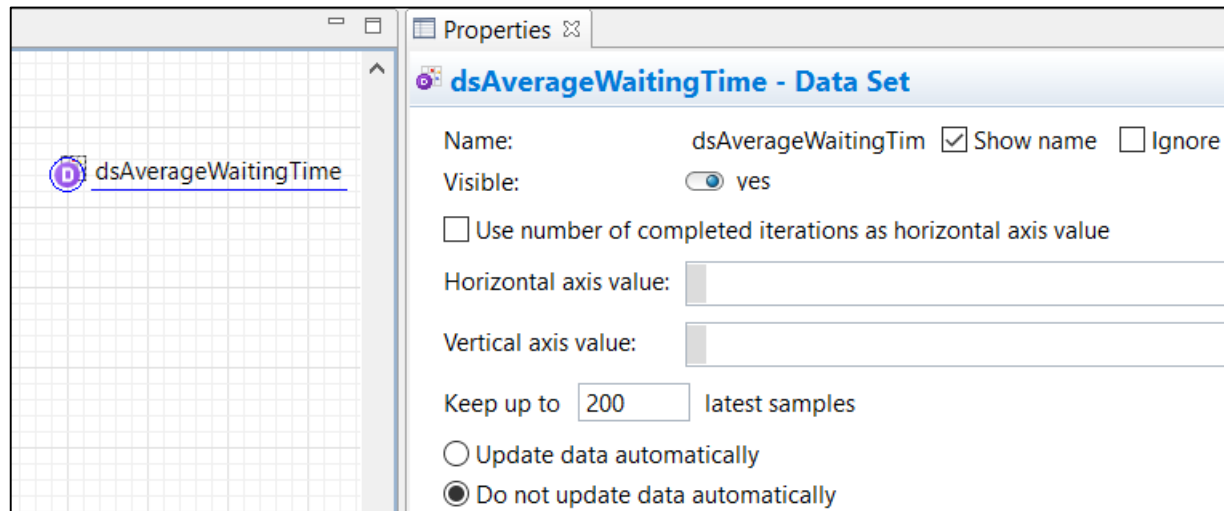


# Definovanie replikácií

- Objekty **Statistics**



- Objekt **Data Set**





# Definovanie replikácií

- Objekt **Plot**

Properties

**plot - Plot**

Name: plot ☐ Ignore

**Data**

☐ Value ☒ Data set

Title: Ustáľovanie priemernej doby

Data set: dsAverageWaitingTime

Point style: ———— ▾

Line width: ———— ▾ 2 pt

Color: green ▾

+ × ↑ ↓

**Data update**

☒ Update data automatically

☐ Do not update data automatically

Display up to 100 latest samples (applies to "Value" data)

# Definovanie replikácií

Properties

## Replications - Parameter Variation Experiment

Name: Replications ☐ Ignore

Top-level agent: Main

Maximum available memory: 256 Mb

Create default UI

### Parameters

#### Model time

Stop: Stop at specified date

Start time: 0 Stop time: 32400

Start date: 10/10/2017 Stop date: 10/10/2017

9:00:00 AM 6:00:00 PM

Additional experiment stop conditions:

Enabled	Expression

#### Randomness

Random number generation:

☒ Random seed (unique simulation runs)

☐ Fixed seed (reproducible simulation runs) Seed value: 1

☐ Custom generator (subclass of Random): new Random()

Selection mode for simultaneous events: LIFO (in the reverse order of scheduling)

### Replications

☒ Use replications

☒ Fixed number of replications

Replications per iteration: 500

☐ Varying number of replications (Stop after minimum re

# Definovanie replikácií

**Java actions**

Initial experiment setup:

Before each experiment run:

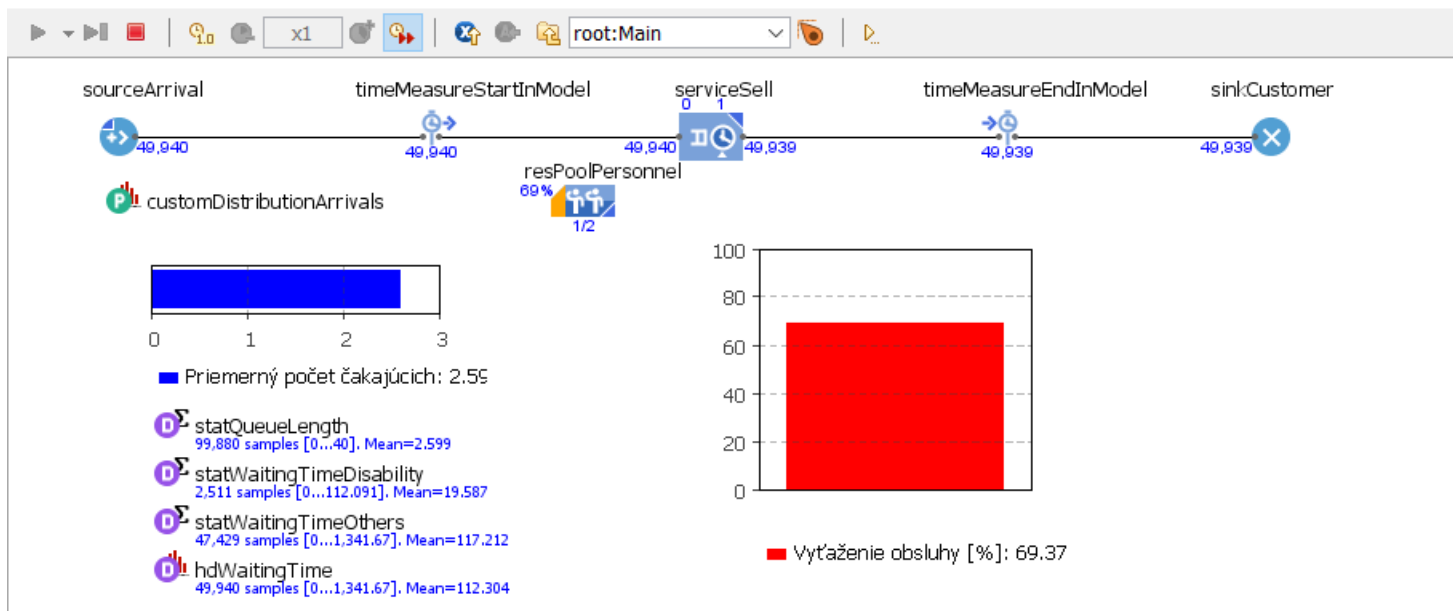
Before simulation run:

After simulation run:

```
statAverageWaitingTime.add(root.hdWaitingTime.mean());
statAverageWaitingTimeDisability.add(root.statWaitingTimeDisability.mean());
statAverageWaitingTimeOthers.add(root.statWaitingTimeOthers.mean());
statAverageQueueLength.add(root.statQueueLength.mean());
statMaximalQueueLength.add(root.statQueueLength.max());

dsAverageWaitingTime.add(statAverageWaitingTime.count(), statAverageWaitingTime.mean());
```

#### 4 Model : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]





## Replikácie - cvičenie 3

Run

 **statAverageWaitingTime**  
500 samples [54.066...222.177]. Mean=106.059

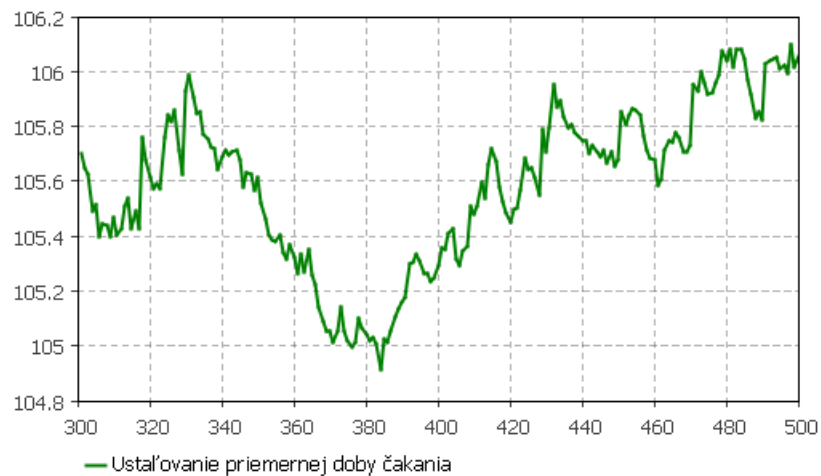
 **statAverageWaitingTimeDisability**  
500 samples [11.076...27.473]. Mean=19.371

 **statAverageWaitingTimeOthers**  
500 samples [56.059...231.302]. Mean=110.616

 **statAverageQueueLength**  
500 samples [1.088...5.413]. Mean=2.481

 **statMaximalQueueLength**  
500 samples [10...39]. Mean=18.968

 **dsAverageWaitingTime**  
200 samples ...[500, 106.059]



**Koniec**