Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота № 4**

з дисципліни «Комп’ютерне моделювання»

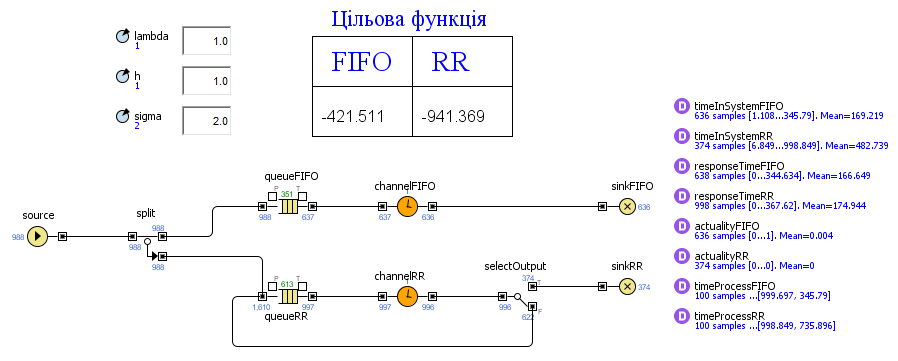
Виконав студент групи ІО-01

*Редько Олександр*

# Завдання

Виконати імітаційне моделювання СМО типів: FIFO, RR. Визначити значення цільової функції.

# Модель



# Програма

## Лістинг коду

/\*\*

\* Task

\*/

**public** **class** Task **extends** Entity **implements** java.io.Serializable {

**double** timeProcess;

**boolean** flagRR;

**double** enteredSystemFIFO;

**double** enteredSystemRR;

**double** processFIFO;

**double** processRR;

**double** timeInChannelRR;

**double** responseTimeFIFO;

**double** responseTimeRR;

**double** actualityFIFO;

**double** actualityRR;

/\*\*

\* Default constructor

\*/

**public** Task(){

}

**public** Task(**double** timeProcess){

setTimeProcess( timeProcess );

}

/\*\*

\* Constructor initializing the fields

\*/

**public** Task(**double** enteredSystem, **double** processRR,

**double** enteredSystemFIFO, **double** enteredSystemRR, **double** timeInChannelRR){

**this**.enteredSystemFIFO = enteredSystemFIFO;

**this**.enteredSystemRR = enteredSystemRR;

**this**.processRR = processRR;

**this**.timeInChannelRR = timeInChannelRR;

}

@Override

**public** String toString() {

**return**

"enteredSystemFIFO = " + enteredSystemFIFO +" " +

"enteredSystemRR = " + enteredSystemRR +" " +

"processRR = " + processRR +" ";

}

**public** **void** setTimeProcess(**double** timeProcess) {

**this**.timeProcess = timeProcess;

}

**public** **double** getTimeProcess() {

**return** **this**.timeProcess;

}

**public** **void** setFlagRR(**boolean** flagRR) {

**this**.flagRR = flagRR;

}

/\*\*

\* This number is here for model snapshot storing purpose<br>

\* It needs to be changed when this class gets changed

\*/

**private** **static** **final** **long** serialVersionUID = 1L;

}}

**package** lab4;

**import** java.sql.Connection;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Calendar;

**import** java.util.Collection;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.Currency;

**import** java.util.Date;

**import** java.util.Enumeration;

**import** java.util.HashMap;

**import** java.util.HashSet;

**import** java.util.Hashtable;

**import** java.util.Iterator;

**import** java.util.LinkedList;

**import** java.util.List;

**import** java.util.ListIterator;

**import** java.util.Locale;

**import** java.util.Map;

**import** java.util.Random;

**import** java.util.Set;

**import** java.util.SortedMap;

**import** java.util.SortedSet;

**import** java.util.Stack;

**import** java.util.Timer;

**import** java.util.TreeMap;

**import** java.util.TreeSet;

**import** java.util.Vector;

**import** java.awt.Color;

**import** java.awt.Font;

**import** java.awt.Graphics2D;

**import** java.awt.geom.AffineTransform;

**import** **static** java.lang.Math.\*;

**import** **static** com.xj.anylogic.engine.presentation.UtilitiesColor.\*;

**import** **static** com.xj.anylogic.engine.presentation.UtilitiesDrawing.\*;

**import** **static** com.xj.anylogic.engine.HyperArray.\*;

**import** com.xj.anylogic.engine.\*;

**import** com.xj.anylogic.engine.analysis.\*;

**import** com.xj.anylogic.engine.connectivity.\*;

**import** com.xj.anylogic.engine.connectivity.ResultSet;

**import** com.xj.anylogic.engine.connectivity.Statement;

**import** com.xj.anylogic.engine.presentation.\*;

**import** com.xj.anylogic.libraries.enterprise.\*;

**import** java.awt.geom.Arc2D;

**public** **class** QueueingModel **extends** ActiveObject

{

// Parameters

**public**

**double** h;

/\*\*

\* Returns default value for parameter <code>h</code>.

\* <i>This method should not be called by user</i>

\*/

**public**

**double** \_h\_DefaultValue\_xjal() {

**return**

1

;

}

**public** **void** set\_h(

**double** h ) {

**if** (h == **this**.h) {

**return**;

}

**this**.h = h;

onChange\_h();

onChange();

}

**void** onChange\_h() {

}

**public**

**double** sigma;

/\*\*

\* Returns default value for parameter <code>sigma</code>.

\* <i>This method should not be called by user</i>

\*/

**public**

**double** \_sigma\_DefaultValue\_xjal() {

**return**

2

;

}

**public** **void** set\_sigma(

**double** sigma ) {

**if** (sigma == **this**.sigma) {

**return**;

}

**this**.sigma = sigma;

onChange\_sigma();

onChange();

}

**void** onChange\_sigma() {

}

**public**

**double** lambda;

/\*\*

\* Returns default value for parameter <code>lambda</code>.

\* <i>This method should not be called by user</i>

\*/

**public**

**double** \_lambda\_DefaultValue\_xjal() {

**return**

1

;

}

**public** **void** set\_lambda(

**double** lambda ) {

**if** (lambda == **this**.lambda) {

**return**;

}

**this**.lambda = lambda;

onChange\_lambda();

onChange();

}

**void** onChange\_lambda() {

**int** index;

source.set\_rate( lambda );

}