НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №2

з дисципліни **«**Комп’ютерне моделювання**»**

Виконав:

студент 3 курсу

ФІОТ гр. ІО-21

Кузьменко Володимир

Перевірив:

Марковський О.П.

Київ – 2014 р.

**Завдання**: Виконати імітаційне моделювання СМО різних типів. Визначити значення цільової функції.

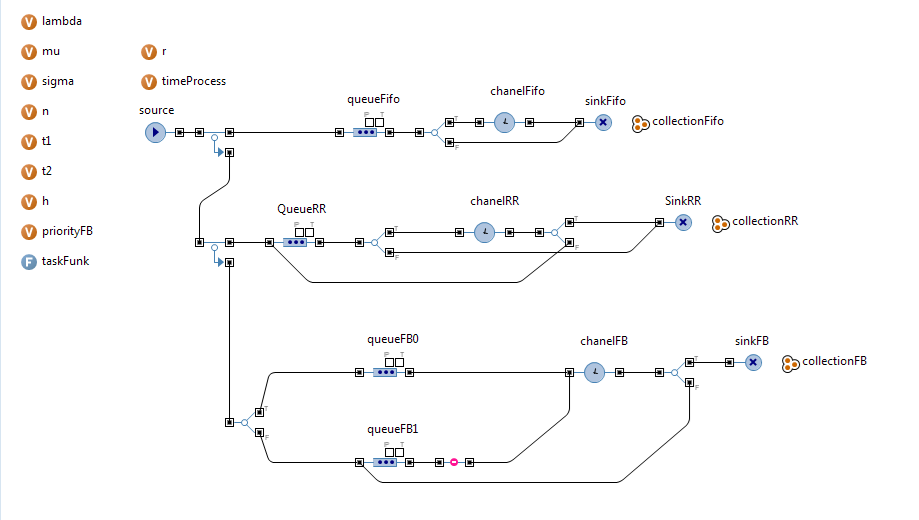
Типи СМО: FIFO, RR,FB

Канал Н(2)

Вагові коефіцієнти критеріїв ефективності k1=-1, k2=-1, k3=0, k4=1, k5=8

Параметри актуальності задачі: t1=3, t2=5

**Модель**



**Лістинг коду програми**

package cmlab2;

public class Main extends Agent

{

// Параметры

// Простые переменные

public

double

timeProcess;

public static

double

lambda =

2.5

;

public static

double

mu =

3

;

public

double

sigma;

public static

double

n =

5000.0

;

public

Random

r;

public static

double

t1 =

3

;

public static

double

t2 =

5

;

public static

double

h =

0.02

;

public static

double

priorityFB =

0.5

;

public static

double

longQueue =

0

;

public static

double

queueIterat =

0

;

// Коллекции

public static

java.util.ArrayList <

Task > collectionFifo = new java.util.ArrayList<Task>();

public static

java.util.ArrayList <

Task > collectionRR = new java.util.ArrayList<Task>();

public static

java.util.ArrayList <

Task > collectionFB = new java.util.ArrayList<Task>();

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final short \_STATECHART\_ELEMENT\_NEXT\_ID\_xjal = 0;

// Вложенные объекты

public com.xj.anylogic.libraries.processmodeling.Source<

Task > source;

public com.xj.anylogic.libraries.processmodeling.Queue<

Task > queueFifo;

public com.xj.anylogic.libraries.processmodeling.Delay<

Task > chanelFifo;

public com.xj.anylogic.libraries.processmodeling.Sink<

Task > sinkFifo;

public com.xj.anylogic.libraries.processmodeling.SelectOutput<

Task > selectOutput1;

public com.xj.anylogic.libraries.processmodeling.Split<

Task, Task > split;

public com.xj.anylogic.libraries.processmodeling.Queue<

Task > QueueRR;

public com.xj.anylogic.libraries.processmodeling.Delay<

Task > chanelRR;

public com.xj.anylogic.libraries.processmodeling.Sink<

Task > SinkRR;

public com.xj.anylogic.libraries.processmodeling.SelectOutput<

Task > selectOutput4;

public com.xj.anylogic.libraries.processmodeling.SelectOutput<

Task > selectOutput;

public com.xj.anylogic.libraries.processmodeling.Split<

Task, Task > split1;

public com.xj.anylogic.libraries.processmodeling.Delay<

Task > chanelFB;

public com.xj.anylogic.libraries.processmodeling.Sink<

Task > sinkFB;

public com.xj.anylogic.libraries.processmodeling.SelectOutput<

Task > selectOutput3;

public com.xj.anylogic.libraries.processmodeling.Queue<

Task > queueFB0;

public com.xj.anylogic.libraries.processmodeling.Queue<

Task > queueFB1;

public com.xj.anylogic.libraries.processmodeling.SelectOutput<

Task > selectOutput6;

public com.xj.anylogic.libraries.processmodeling.Hold<

Task > holdFB;

public String getNameOf( Agent ao ) {

if ( ao == source ) return "source";

if ( ao == queueFifo ) return "queueFifo";

if ( ao == chanelFifo ) return "chanelFifo";

if ( ao == sinkFifo ) return "sinkFifo";

if ( ao == selectOutput1 ) return "selectOutput1";

if ( ao == split ) return "split";

if ( ao == QueueRR ) return "QueueRR";

if ( ao == chanelRR ) return "chanelRR";

if ( ao == SinkRR ) return "SinkRR";

if ( ao == selectOutput4 ) return "selectOutput4";

if ( ao == selectOutput ) return "selectOutput";

if ( ao == split1 ) return "split1";

if ( ao == chanelFB ) return "chanelFB";

if ( ao == sinkFB ) return "sinkFB";

if ( ao == selectOutput3 ) return "selectOutput3";

if ( ao == queueFB0 ) return "queueFB0";

if ( ao == queueFB1 ) return "queueFB1";

if ( ao == selectOutput6 ) return "selectOutput6";

if ( ao == holdFB ) return "holdFB";

return super.getNameOf( ao );

}

public String getNameOf( AgentList<?> aolist ) {

return super.getNameOf( aolist );

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Source<Task> instantiate\_source\_xjal() {

com.xj.anylogic.libraries.processmodeling.Source<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Source<Task>( getEngine(), this, null ) {

@Override

public double interarrivalTime( ) {

return \_source\_interarrivalTime\_xjal( this );

}

@Override

public Agent newEntity( ) {

return \_source\_newEntity\_xjal( this );

}

@Override

public void onExit( Task entity ) {

\_source\_onExit\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_source\_xjal( final com.xj.anylogic.libraries.processmodeling.Source<Task> self ) {

self.arrivalType =

self.INTERARRIVAL\_TIME

;

self.rate =

(-1/lambda)\*Math.log(r.nextDouble())

;

self.rateSchedule = self.\_rateSchedule\_DefaultValue\_xjal();

self.modifyRate = self.\_modifyRate\_DefaultValue\_xjal();

self.arrivalSchedule = self.\_arrivalSchedule\_DefaultValue\_xjal();

self.multipleEntitiesPerArrival = self.\_multipleEntitiesPerArrival\_DefaultValue\_xjal();

self.limitArrivals =

true

;

self.maxArrivals =

(long)n

;

self.locationType = self.\_locationType\_DefaultValue\_xjal();

self.locationXYZInNetwork = self.\_locationXYZInNetwork\_DefaultValue\_xjal();

self.enableCustomStartTime = self.\_enableCustomStartTime\_DefaultValue\_xjal();

self.startTime = self.\_startTime\_DefaultValue\_xjal();

self.addToCustomPopulation = self.\_addToCustomPopulation\_DefaultValue\_xjal();

self.pushProtocol = self.\_pushProtocol\_DefaultValue\_xjal();

self.discardHangingEntities = self.\_discardHangingEntities\_DefaultValue\_xjal();

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_source\_xjal(com.xj.anylogic.libraries.processmodeling.Source<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Queue<Task> instantiate\_queueFifo\_xjal() {

com.xj.anylogic.libraries.processmodeling.Queue<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Queue<Task>( getEngine(), this, null ) {

@Override

public void onEnter( Task entity ) {

\_queueFifo\_onEnter\_xjal( this, entity );

}

@Override

public void onAtExit( Task entity ) {

\_queueFifo\_onAtExit\_xjal( this, entity );

}

@Override

public void onExit( Task entity ) {

\_queueFifo\_onExit\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_queueFifo\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self ) {

self.capacity = self.\_capacity\_DefaultValue\_xjal();

self.maximumCapacity =

true

;

self.entityLocation = self.\_entityLocation\_DefaultValue\_xjal();

self.queuing = self.\_queuing\_DefaultValue\_xjal();

self.enableTimeout = self.\_enableTimeout\_DefaultValue\_xjal();

self.enablePreemption = self.\_enablePreemption\_DefaultValue\_xjal();

self.restoreEntityLocationOnExit =

false

;

self.forceStatisticsCollection =

true

;

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_queueFifo\_xjal(com.xj.anylogic.libraries.processmodeling.Queue<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Delay<Task> instantiate\_chanelFifo\_xjal() {

com.xj.anylogic.libraries.processmodeling.Delay<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Delay<Task>( getEngine(), this, null ) {

@Override

public double delayTime( Task entity ) {

return \_chanelFifo\_delayTime\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_chanelFifo\_xjal( final com.xj.anylogic.libraries.processmodeling.Delay<Task> self ) {

self.type = self.\_type\_DefaultValue\_xjal();

self.capacity = self.\_capacity\_DefaultValue\_xjal();

self.maximumCapacity = self.\_maximumCapacity\_DefaultValue\_xjal();

self.entityLocation = self.\_entityLocation\_DefaultValue\_xjal();

self.pushProtocol = self.\_pushProtocol\_DefaultValue\_xjal();

self.restoreEntityLocationOnExit =

false

;

self.forceStatisticsCollection = self.\_forceStatisticsCollection\_DefaultValue\_xjal();

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_chanelFifo\_xjal(com.xj.anylogic.libraries.processmodeling.Delay<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Sink<Task> instantiate\_sinkFifo\_xjal() {

com.xj.anylogic.libraries.processmodeling.Sink<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Sink<Task>( getEngine(), this, null ) {

@Override

public void onEnter( Task entity ) {

\_sinkFifo\_onEnter\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_sinkFifo\_xjal( final com.xj.anylogic.libraries.processmodeling.Sink<Task> self ) {

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_sinkFifo\_xjal(com.xj.anylogic.libraries.processmodeling.Sink<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> instantiate\_selectOutput1\_xjal() {

com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.SelectOutput<Task>( getEngine(), this, null ) {

@Override

public boolean condition( Task entity ) {

return \_selectOutput1\_condition\_xjal( this, entity );

}

@Override

public void onExitFalse( Task entity ) {

\_selectOutput1\_onExitFalse\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_selectOutput1\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self ) {

self.conditionIsProbabilistic =

false

;

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_selectOutput1\_xjal(com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Split<Task, Task> instantiate\_split\_xjal() {

com.xj.anylogic.libraries.processmodeling.Split<Task, Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Split<Task, Task>( getEngine(), this, null ) {

@Override

public Agent newEntity( Task original, int index ) {

return \_split\_newEntity\_xjal( this, original, index );

}

@Override

public void onExitCopy( Task entity, int index, Task original ) {

\_split\_onExitCopy\_xjal( this, entity, index, original );

}

@Override

public void onExit( Task entity ) {

\_split\_onExit\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_split\_xjal( final com.xj.anylogic.libraries.processmodeling.Split<Task, Task> self ) {

self.locationType = self.\_locationType\_DefaultValue\_xjal();

self.locationXYZInNetwork = self.\_locationXYZInNetwork\_DefaultValue\_xjal();

self.addToCustomPopulation = self.\_addToCustomPopulation\_DefaultValue\_xjal();

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_split\_xjal(com.xj.anylogic.libraries.processmodeling.Split<Task, Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Queue<Task> instantiate\_QueueRR\_xjal() {

com.xj.anylogic.libraries.processmodeling.Queue<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Queue<Task>( getEngine(), this, null ) {

@Override

public void onEnter( Task entity ) {

\_QueueRR\_onEnter\_xjal( this, entity );

}

@Override

public void onExit( Task entity ) {

\_QueueRR\_onExit\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_QueueRR\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self ) {

self.capacity = self.\_capacity\_DefaultValue\_xjal();

self.maximumCapacity =

true

;

self.entityLocation = self.\_entityLocation\_DefaultValue\_xjal();

self.queuing = self.\_queuing\_DefaultValue\_xjal();

self.enableTimeout = self.\_enableTimeout\_DefaultValue\_xjal();

self.enablePreemption = self.\_enablePreemption\_DefaultValue\_xjal();

self.restoreEntityLocationOnExit =

false

;

self.forceStatisticsCollection =

true

;

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_QueueRR\_xjal(com.xj.anylogic.libraries.processmodeling.Queue<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Delay<Task> instantiate\_chanelRR\_xjal() {

com.xj.anylogic.libraries.processmodeling.Delay<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Delay<Task>( getEngine(), this, null ) {

@Override

public double delayTime( Task entity ) {

return \_chanelRR\_delayTime\_xjal( this, entity );

}

@Override

public void onExit( Task entity ) {

\_chanelRR\_onExit\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_chanelRR\_xjal( final com.xj.anylogic.libraries.processmodeling.Delay<Task> self ) {

self.type = self.\_type\_DefaultValue\_xjal();

self.capacity = self.\_capacity\_DefaultValue\_xjal();

self.maximumCapacity = self.\_maximumCapacity\_DefaultValue\_xjal();

self.entityLocation = self.\_entityLocation\_DefaultValue\_xjal();

self.pushProtocol = self.\_pushProtocol\_DefaultValue\_xjal();

self.restoreEntityLocationOnExit =

false

;

self.forceStatisticsCollection =

true

;

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_chanelRR\_xjal(com.xj.anylogic.libraries.processmodeling.Delay<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Sink<Task> instantiate\_SinkRR\_xjal() {

com.xj.anylogic.libraries.processmodeling.Sink<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Sink<Task>( getEngine(), this, null ) {

@Override

public void onEnter( Task entity ) {

\_SinkRR\_onEnter\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_SinkRR\_xjal( final com.xj.anylogic.libraries.processmodeling.Sink<Task> self ) {

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_SinkRR\_xjal(com.xj.anylogic.libraries.processmodeling.Sink<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> instantiate\_selectOutput4\_xjal() {

com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.SelectOutput<Task>( getEngine(), this, null ) {

@Override

public boolean condition( Task entity ) {

return \_selectOutput4\_condition\_xjal( this, entity );

}

@Override

public void onExitTrue( Task entity ) {

\_selectOutput4\_onExitTrue\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_selectOutput4\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self ) {

self.conditionIsProbabilistic =

false

;

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_selectOutput4\_xjal(com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> instantiate\_selectOutput\_xjal() {

com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.SelectOutput<Task>( getEngine(), this, null ) {

@Override

public boolean condition( Task entity ) {

return \_selectOutput\_condition\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_selectOutput\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self ) {

self.conditionIsProbabilistic =

false

;

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_selectOutput\_xjal(com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Split<Task, Task> instantiate\_split1\_xjal() {

com.xj.anylogic.libraries.processmodeling.Split<Task, Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Split<Task, Task>( getEngine(), this, null ) {

@Override

public Agent newEntity( Task original, int index ) {

return \_split1\_newEntity\_xjal( this, original, index );

}

@Override

public void onExitCopy( Task entity, int index, Task original ) {

\_split1\_onExitCopy\_xjal( this, entity, index, original );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_split1\_xjal( final com.xj.anylogic.libraries.processmodeling.Split<Task, Task> self ) {

self.locationType = self.\_locationType\_DefaultValue\_xjal();

self.locationXYZInNetwork = self.\_locationXYZInNetwork\_DefaultValue\_xjal();

self.addToCustomPopulation = self.\_addToCustomPopulation\_DefaultValue\_xjal();

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_split1\_xjal(com.xj.anylogic.libraries.processmodeling.Split<Task, Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Delay<Task> instantiate\_chanelFB\_xjal() {

com.xj.anylogic.libraries.processmodeling.Delay<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Delay<Task>( getEngine(), this, null ) {

@Override

public double delayTime( Task entity ) {

return \_chanelFB\_delayTime\_xjal( this, entity );

}

@Override

public void onExit( Task entity ) {

\_chanelFB\_onExit\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_chanelFB\_xjal( final com.xj.anylogic.libraries.processmodeling.Delay<Task> self ) {

self.type = self.\_type\_DefaultValue\_xjal();

self.capacity = self.\_capacity\_DefaultValue\_xjal();

self.maximumCapacity = self.\_maximumCapacity\_DefaultValue\_xjal();

self.entityLocation = self.\_entityLocation\_DefaultValue\_xjal();

self.pushProtocol = self.\_pushProtocol\_DefaultValue\_xjal();

self.restoreEntityLocationOnExit =

false

;

self.forceStatisticsCollection = self.\_forceStatisticsCollection\_DefaultValue\_xjal();

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_chanelFB\_xjal(com.xj.anylogic.libraries.processmodeling.Delay<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Sink<Task> instantiate\_sinkFB\_xjal() {

com.xj.anylogic.libraries.processmodeling.Sink<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Sink<Task>( getEngine(), this, null ) {

@Override

public void onEnter( Task entity ) {

\_sinkFB\_onEnter\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_sinkFB\_xjal( final com.xj.anylogic.libraries.processmodeling.Sink<Task> self ) {

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_sinkFB\_xjal(com.xj.anylogic.libraries.processmodeling.Sink<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> instantiate\_selectOutput3\_xjal() {

com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.SelectOutput<Task>( getEngine(), this, null ) {

@Override

public boolean condition( Task entity ) {

return \_selectOutput3\_condition\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_selectOutput3\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self ) {

self.conditionIsProbabilistic =

false

;

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_selectOutput3\_xjal(com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Queue<Task> instantiate\_queueFB0\_xjal() {

com.xj.anylogic.libraries.processmodeling.Queue<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Queue<Task>( getEngine(), this, null ) {

@Override

public void onEnter( Task entity ) {

\_queueFB0\_onEnter\_xjal( this, entity );

}

@Override

public void onExit( Task entity ) {

\_queueFB0\_onExit\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_queueFB0\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self ) {

self.capacity = self.\_capacity\_DefaultValue\_xjal();

self.maximumCapacity =

true

;

self.entityLocation = self.\_entityLocation\_DefaultValue\_xjal();

self.queuing = self.\_queuing\_DefaultValue\_xjal();

self.enableTimeout = self.\_enableTimeout\_DefaultValue\_xjal();

self.enablePreemption = self.\_enablePreemption\_DefaultValue\_xjal();

self.restoreEntityLocationOnExit =

false

;

self.forceStatisticsCollection =

true

;

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_queueFB0\_xjal(com.xj.anylogic.libraries.processmodeling.Queue<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Queue<Task> instantiate\_queueFB1\_xjal() {

com.xj.anylogic.libraries.processmodeling.Queue<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Queue<Task>( getEngine(), this, null ) {

@Override

public void onEnter( Task entity ) {

\_queueFB1\_onEnter\_xjal( this, entity );

}

@Override

public void onExit( Task entity ) {

\_queueFB1\_onExit\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_queueFB1\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self ) {

self.capacity = self.\_capacity\_DefaultValue\_xjal();

self.maximumCapacity =

true

;

self.entityLocation = self.\_entityLocation\_DefaultValue\_xjal();

self.queuing = self.\_queuing\_DefaultValue\_xjal();

self.enableTimeout = self.\_enableTimeout\_DefaultValue\_xjal();

self.enablePreemption = self.\_enablePreemption\_DefaultValue\_xjal();

self.restoreEntityLocationOnExit =

false

;

self.forceStatisticsCollection =

true

;

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_queueFB1\_xjal(com.xj.anylogic.libraries.processmodeling.Queue<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> instantiate\_selectOutput6\_xjal() {

com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.SelectOutput<Task>( getEngine(), this, null ) {

@Override

public double probability( Task entity ) {

return \_selectOutput6\_probability\_xjal( this, entity );

}

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_selectOutput6\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self ) {

self.conditionIsProbabilistic = self.\_conditionIsProbabilistic\_DefaultValue\_xjal();

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_selectOutput6\_xjal(com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self ) {

self.create();

}

/\*\*

\* Создает экземпляр вложенного объекта<br>

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

protected com.xj.anylogic.libraries.processmodeling.Hold<Task> instantiate\_holdFB\_xjal() {

com.xj.anylogic.libraries.processmodeling.Hold<Task> \_result\_xjal = new com.xj.anylogic.libraries.processmodeling.Hold<Task>( getEngine(), this, null ) {

};

return \_result\_xjal;

}

/\*\*

\* Инициализация параметров экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void setupParameters\_holdFB\_xjal( final com.xj.anylogic.libraries.processmodeling.Hold<Task> self ) {

self.mode = self.\_mode\_DefaultValue\_xjal();

self.nEntitiesForSelfBlock = self.\_nEntitiesForSelfBlock\_DefaultValue\_xjal();

self.initiallyBlocked = self.\_initiallyBlocked\_DefaultValue\_xjal();

}

/\*\*

\* Инициализация экземпляра вложенного объекта<br>

\* Пользователь не должен вызывать этот метод

\*/

private void create\_holdFB\_xjal(com.xj.anylogic.libraries.processmodeling.Hold<Task> self ) {

self.create();

}

private double \_source\_interarrivalTime\_xjal( final com.xj.anylogic.libraries.processmodeling.Source<Task> self ) {

return

(-1/lambda)\*Math.log(r.nextDouble())

;

}

private Agent \_source\_newEntity\_xjal( final com.xj.anylogic.libraries.processmodeling.Source<Task> self ) {

return

new Task()

;

}

private void \_source\_onExit\_xjal( final com.xj.anylogic.libraries.processmodeling.Source<Task> self, Task entity ) {

timeProcess = Math.abs(1/mu+sigma\*r.nextGaussian());

;

}

private void \_queueFifo\_onEnter\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self, Task entity ) {

entity.timeEntered = time();

;

}

private void \_queueFifo\_onAtExit\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self, Task entity ) {

entity.timeResponse = time() - entity.timeEntered;

entity.actuality( time() - entity.timeEntered);

if (entity.actuality==0.0)

entity.fifoFlag = false;

else entity.fifoFlag = true;

;

}

private void \_queueFifo\_onExit\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self, Task entity ) {

queueIterat+=1;

longQueue+=queueFifo.size();

;

}

private double \_chanelFifo\_delayTime\_xjal( final com.xj.anylogic.libraries.processmodeling.Delay<Task> self, Task entity ) {

return

poisson(mu)

;

}

private void \_sinkFifo\_onEnter\_xjal( final com.xj.anylogic.libraries.processmodeling.Sink<Task> self, Task entity ) {

entity.timeSystem = time() - entity.timeEntered;

collectionFifo.add(entity);

;

}

private boolean \_selectOutput1\_condition\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self, Task entity ) {

return

entity.fifoFlag==true

;

}

private void \_selectOutput1\_onExitFalse\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self, Task entity ) {

entity.timeSystem = time() - entity.timeEntered;

;

}

private Agent \_split\_newEntity\_xjal( final com.xj.anylogic.libraries.processmodeling.Split<Task, Task> self, Task original, int index ) {

return

new Task()

;

}

private void \_split\_onExitCopy\_xjal( final com.xj.anylogic.libraries.processmodeling.Split<Task, Task> self, Task entity, int index, Task original ) {

entity.timeProcess = timeProcess;

entity.timeEntered = time();

;

}

private void \_split\_onExit\_xjal( final com.xj.anylogic.libraries.processmodeling.Split<Task, Task> self, Task entity ) {

entity.timeProcess =timeProcess;

;

}

private void \_QueueRR\_onEnter\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self, Task entity ) {

entity.buf = time();

;

}

private void \_QueueRR\_onExit\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self, Task entity ) {

if (entity.flagReact){

entity.timeResponse = time() - entity.timeEntered;

entity.flagReact=false;

}

entity.timeSystem+=time() - entity.buf;

entity.actuality(entity.timeSystem);

if (entity.actuality==0.0)

entity.rrFlag = false;

else entity.rrFlag = true;

;

}

private double \_chanelRR\_delayTime\_xjal( final com.xj.anylogic.libraries.processmodeling.Delay<Task> self, Task entity ) {

return

entity.timeWork

;

}

private void \_chanelRR\_onExit\_xjal( final com.xj.anylogic.libraries.processmodeling.Delay<Task> self, Task entity ) {

if (entity.timeProcess ==0.0){

entity.flagRR=true;

}else{

entity.flagRR=false;

}

entity.timeSystem+=entity.timeWork;

;

}

private void \_SinkRR\_onEnter\_xjal( final com.xj.anylogic.libraries.processmodeling.Sink<Task> self, Task entity ) {

collectionRR.add(entity);

;

}

private boolean \_selectOutput4\_condition\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self, Task entity ) {

return

entity.rrFlag==true

;

}

private void \_selectOutput4\_onExitTrue\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self, Task entity ) {

if (entity.timeProcess<h){

entity.timeWork = entity.timeProcess;

entity.timeProcess -= entity.timeWork;

}else{

entity.timeWork = h;

entity.timeProcess-=h;

}

;

}

private boolean \_selectOutput\_condition\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self, Task entity ) {

return

entity.flagRR==true

;

}

private Agent \_split1\_newEntity\_xjal( final com.xj.anylogic.libraries.processmodeling.Split<Task, Task> self, Task original, int index ) {

return

new Task()

;

}

private void \_split1\_onExitCopy\_xjal( final com.xj.anylogic.libraries.processmodeling.Split<Task, Task> self, Task entity, int index, Task original ) {

entity.timeProcess = timeProcess;

entity.timeEntered = time();

;

}

private double \_chanelFB\_delayTime\_xjal( final com.xj.anylogic.libraries.processmodeling.Delay<Task> self, Task entity ) {

return

entity.timeWork

;

}

private void \_chanelFB\_onExit\_xjal( final com.xj.anylogic.libraries.processmodeling.Delay<Task> self, Task entity ) {

if (entity.timeProcess ==0.0){

entity.flagFB=true;

}else{

entity.flagFB=false;

}

entity.timeSystem+=entity.timeWork;

;

}

private void \_sinkFB\_onEnter\_xjal( final com.xj.anylogic.libraries.processmodeling.Sink<Task> self, Task entity ) {

collectionFB.add(entity);

;

}

private boolean \_selectOutput3\_condition\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self, Task entity ) {

return

entity.flagFB==true

;

}

private void \_queueFB0\_onEnter\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self, Task entity ) {

entity.buf = time();

holdFB.setBlocked(true);

;

}

private void \_queueFB0\_onExit\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self, Task entity ) {

if (entity.flagReact){

entity.timeResponse = time() - entity.timeEntered;

entity.flagReact=false;

}

entity.timeSystem+=time() - entity.buf;

entity.actuality(entity.timeSystem);

if (entity.actuality==0.0){

entity.timeProcess =0.0;

}

else {

if (entity.timeProcess<h){

entity.timeWork = entity.timeProcess;

entity.timeProcess -= entity.timeWork;

}else{

entity.timeWork = h;

entity.timeProcess-=h;

}

}

holdFB.setBlocked(false);

;

}

private void \_queueFB1\_onEnter\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self, Task entity ) {

entity.buf = time();

;

}

private void \_queueFB1\_onExit\_xjal( final com.xj.anylogic.libraries.processmodeling.Queue<Task> self, Task entity ) {

if (entity.flagReact){

entity.timeResponse = time() - entity.timeEntered;

entity.flagReact=false;

}

entity.timeSystem+=time() - entity.buf;

entity.actuality(entity.timeSystem);

if (entity.actuality==0.0){

entity.timeProcess =0.0;

}

else {

if (entity.timeProcess<h){

entity.timeWork = entity.timeProcess;

entity.timeProcess -= entity.timeWork;

}else{

entity.timeWork = h;

entity.timeProcess-=h;

}

}

;

}

private double \_selectOutput6\_probability\_xjal( final com.xj.anylogic.libraries.processmodeling.SelectOutput<Task> self, Task entity ) {

return

priorityFB

;

}

// Функции

static

double

taskFunk( double x1, double x2, double x3, double x4, double x5 ) {

return -1\*x1-1\*x2+0\*x3+x4+8\*x5;

}

// Области

public ViewArea \_origin\_VA = new ViewArea( this, "[Нач. координат]", 0, 0, ViewArea.TOP\_LEFT, ViewArea.SPECIFIED\_ZOOM, 1, 100, 100 );

@Override

@AnyLogicInternalCodegenAPI

public int getViewAreas(Map<String, ViewArea> \_output) {

if ( \_output != null ) {

\_output.put( "\_origin\_VA", this.\_origin\_VA );

}

return 1 + super.getViewAreas( \_output );

}

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_SHAPE\_NEXT\_ID\_xjal = 1;

/\*\*

\* Идентификатор группы presentation верхнего уровня

\*/

@AnyLogicInternalCodegenAPI

protected static final int \_presentation = 0;

@AnyLogicInternalCodegenAPI

public boolean isPublicPresentationDefined() {

return false;

}

@AnyLogicInternalCodegenAPI

public boolean isEmbeddedAgentPresentationVisible( Agent \_a ) {

return super.isEmbeddedAgentPresentationVisible( \_a );

}

/\*\*

\* Идентификатор группы icon верхнего уровня

\*/

@AnyLogicInternalCodegenAPI

protected static final int \_icon = -1;

protected ShapeTopLevelPresentationGroup presentation;

protected ShapeGroup icon;

@Override

@AnyLogicInternalCodegenAPI

public Object getPersistentShape( int \_shape ) {

switch (\_shape) {

case \_presentation: return presentation;

case \_icon: return icon;

default: return super.getPersistentShape( \_shape );

}

}

@Override

@AnyLogicInternalCodegenAPI

public String getNameOfShape\_xjal( Object \_shape ) {

try {

if ( \_shape == null ) return null;

String \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, presentation, "presentation" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, icon, "icon" ); if (\_name\_xjal != null) return \_name\_xjal;

} catch (Exception e) {

return null;

}

return super.getNameOfShape\_xjal( \_shape );

}

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector2\_pointsX\_xjal = {

580, 530 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector2\_pointsY\_xjal = {

130,130 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector1\_pointsX\_xjal = {

420, 390 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector1\_pointsY\_xjal = {

140,140 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector3\_pointsX\_xjal = {

450, 480 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector3\_pointsY\_xjal = {

130,130 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector4\_pointsX\_xjal = {

450, 560, 580 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector4\_pointsY\_xjal = {

150,150, 130 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector\_pointsX\_xjal = {

200, 180 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector\_pointsY\_xjal = {

140,140 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector5\_pointsX\_xjal = {

230, 340 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector5\_pointsY\_xjal = {

140,140 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector18\_pointsX\_xjal = {

360, 320 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector18\_pointsY\_xjal = {

250,250 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector19\_pointsX\_xjal = {

390, 460 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector19\_pointsY\_xjal = {

240,240 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector20\_pointsX\_xjal = {

390, 430, 630, 660 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector20\_pointsY\_xjal = {

260,260, 260, 230 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector7\_pointsX\_xjal = {

540, 510 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector7\_pointsY\_xjal = {

240,240 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector8\_pointsX\_xjal = {

570, 660 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector8\_pointsY\_xjal = {

230,230 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector9\_pointsX\_xjal = {

570, 520, 310, 270 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector9\_pointsY\_xjal = {

250,290, 290, 250 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector6\_pointsX\_xjal = {

230, 270 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector6\_pointsY\_xjal = {

250,250 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector10\_pointsX\_xjal = {

230, 230, 200, 200 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector10\_pointsY\_xjal = {

160,190, 210, 250 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector15\_pointsX\_xjal = {

660, 620 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector15\_pointsY\_xjal = {

380,380 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector16\_pointsX\_xjal = {

690, 730 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector16\_pointsY\_xjal = {

370,370 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector25\_pointsX\_xjal = {

260, 270, 360 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector25\_pointsY\_xjal = {

440,470, 470 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector26\_pointsX\_xjal = {

440, 410 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector26\_pointsY\_xjal = {

470,470 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector14\_pointsX\_xjal = {

360, 340, 270, 260 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector14\_pointsY\_xjal = {

380,380, 380, 420 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector17\_pointsX\_xjal = {

230, 230 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector17\_pointsY\_xjal = {

430,270 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector11\_pointsX\_xjal = {

410, 570 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector11\_pointsY\_xjal = {

380,380 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector12\_pointsX\_xjal = {

470, 510, 570, 570 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector12\_pointsY\_xjal = {

470,470, 420, 380 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector13\_pointsX\_xjal = {

690, 690, 620, 380, 360 };

@AnyLogicInternalCodegenAPI

protected static final int[] \_connector13\_pointsY\_xjal = {

390,430, 490, 490, 470 };

@AnyLogicInternalCodegenAPI

private void drawModelElements\_PlainVariables\_xjal(Panel \_panel, Graphics2D \_g, boolean \_publicOnly, boolean \_isSuperClass ) {

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 30, 30, 10, 0, "lambda", lambda, false );

}

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 30, 60, 10, 0, "mu", mu, false );

}

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 30, 90, 10, 0, "sigma", sigma, false );

}

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 30, 120, 10, 0, "n", n, false );

}

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 150, 60, 10, 0, "r", r, false );

}

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 30, 150, 10, 0, "t1", t1, false );

}

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 30, 180, 10, 0, "t2", t2, false );

}

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 30, 210, 10, 0, "h", h, false );

}

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 30, 240, 10, 0, "priorityFB", priorityFB, false );

}

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 394, 41, 10, 0, "longQueue", longQueue, false );

}

if (!\_publicOnly) {

drawPlainVariable( \_panel, \_g, 393, 70, 10, 0, "queueIterat", queueIterat, false );

}

}

@AnyLogicInternalCodegenAPI

private void drawModelElements\_CollectionVariables\_xjal(Panel \_panel, Graphics2D \_g, boolean \_publicOnly, boolean \_isSuperClass ) {

if (!\_publicOnly) {

drawCollection( \_panel, \_g, 640, 130, 10, 0, "collectionFifo", collectionFifo );

}

if (!\_publicOnly) {

drawCollection( \_panel, \_g, 720, 230, 10, 0, "collectionRR", collectionRR );

}

if (!\_publicOnly) {

drawCollection( \_panel, \_g, 790, 370, 10, 0, "collectionFB", collectionFB );

}

}

@AnyLogicInternalCodegenAPI

private void drawModelElements\_Functions\_xjal(Panel \_panel, Graphics2D \_g, boolean \_publicOnly, boolean \_isSuperClass ) {

if (!\_publicOnly) {

drawFunction( \_panel, \_g, 30, 270, 10, 0, "taskFunk");

}

}

@AnyLogicInternalCodegenAPI

private void drawModelElements\_EmbeddeObjects\_xjal(Panel \_panel, Graphics2D \_g, boolean \_publicOnly, boolean \_isSuperClass ) {

// Embedded object "source"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 130 , 90 , 7, 29, "source", this.source );

}

// Embedded object "queueFifo"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 290 , 90 , 55, 18, "queueFifo", this.queueFifo );

}

// Embedded object "chanelFifo"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 430 , 80 , 58, 20, "chanelFifo", this.chanelFifo );

}

// Embedded object "sinkFifo"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 530 , 80 , 53, 30, "sinkFifo", this.sinkFifo );

}

// Embedded object "selectOutput1"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 370 , 90 , 25, 17, null, this.selectOutput1 );

}

// Embedded object "split"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 150 , 90 , 49, 27, null, this.split );

}

// Embedded object "QueueRR"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 220 , 200 , 55, 18, "QueueRR", this.QueueRR );

}

// Embedded object "chanelRR"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 410 , 190 , 58, 20, "chanelRR", this.chanelRR );

}

// Embedded object "SinkRR"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 610 , 180 , 53, 30, "SinkRR", this.SinkRR );

}

// Embedded object "selectOutput4"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 310 , 200 , 25, 17, null, this.selectOutput4 );

}

// Embedded object "selectOutput"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 490 , 190 , 25, 17, null, this.selectOutput );

}

// Embedded object "split1"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 150 , 200 , 49, 27, null, this.split1 );

}

// Embedded object "chanelFB"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 520 , 330 , 58, 20, "chanelFB", this.chanelFB );

}

// Embedded object "sinkFB"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 680 , 320 , 53, 30, "sinkFB", this.sinkFB );

}

// Embedded object "selectOutput3"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 610 , 330 , 25, 17, null, this.selectOutput3 );

}

// Embedded object "queueFB0"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 310 , 330 , 55, 18, "queueFB0", this.queueFB0 );

}

// Embedded object "queueFB1"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 310 , 420 , 55, 18, "queueFB1", this.queueFB1 );

}

// Embedded object "selectOutput6"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 180 , 380 , 25, 17, null, this.selectOutput6 );

}

// Embedded object "holdFB"

if (!\_publicOnly) {

drawEmbeddedObjectModel( \_panel, \_g, 390 , 420 , 50, 30, null, this.holdFB );

}

}

@AnyLogicInternalCodegenAPI

private void drawModelElements\_Connectors\_xjal(Panel \_panel, Graphics2D \_g, boolean \_publicOnly, boolean \_isSuperClass ) {

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector2\_pointsX\_xjal, \_connector2\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector1\_pointsX\_xjal, \_connector1\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector3\_pointsX\_xjal, \_connector3\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector4\_pointsX\_xjal, \_connector4\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector\_pointsX\_xjal, \_connector\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector5\_pointsX\_xjal, \_connector5\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector18\_pointsX\_xjal, \_connector18\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector19\_pointsX\_xjal, \_connector19\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector20\_pointsX\_xjal, \_connector20\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector7\_pointsX\_xjal, \_connector7\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector8\_pointsX\_xjal, \_connector8\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector9\_pointsX\_xjal, \_connector9\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector6\_pointsX\_xjal, \_connector6\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector10\_pointsX\_xjal, \_connector10\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector15\_pointsX\_xjal, \_connector15\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector16\_pointsX\_xjal, \_connector16\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector25\_pointsX\_xjal, \_connector25\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector26\_pointsX\_xjal, \_connector26\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector14\_pointsX\_xjal, \_connector14\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector17\_pointsX\_xjal, \_connector17\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector11\_pointsX\_xjal, \_connector11\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector12\_pointsX\_xjal, \_connector12\_pointsY\_xjal, false );

}

if (!\_publicOnly) {

drawConnector( \_panel, \_g, \_connector13\_pointsX\_xjal, \_connector13\_pointsY\_xjal, false );

}

}

@AnyLogicInternalCodegenAPI

private void drawModelElements\_AgentLinks\_xjal(Panel \_panel, Graphics2D \_g, boolean \_publicOnly, boolean \_isSuperClass ) {

if (\_publicOnly) { return; }

drawLinkToAgent( \_panel, \_g, 50, -50, 15, 0, "connections", true, connections );

}

@Override

@AnyLogicInternalCodegenAPI

public void drawModelElements( Panel \_panel, Graphics2D \_g, boolean \_publicOnly, boolean \_isSuperClass ) {

super.drawModelElements( \_panel, \_g, \_publicOnly, true );

drawModelElements\_PlainVariables\_xjal( \_panel, \_g, \_publicOnly, \_isSuperClass );

drawModelElements\_CollectionVariables\_xjal( \_panel, \_g, \_publicOnly, \_isSuperClass );

drawModelElements\_Functions\_xjal( \_panel, \_g, \_publicOnly, \_isSuperClass );

drawModelElements\_EmbeddeObjects\_xjal( \_panel, \_g, \_publicOnly, \_isSuperClass );

drawModelElements\_Connectors\_xjal( \_panel, \_g, \_publicOnly, \_isSuperClass );

drawModelElements\_AgentLinks\_xjal( \_panel, \_g, \_publicOnly, \_isSuperClass );

}

@AnyLogicInternalCodegenAPI

private boolean onClickModelAt\_EmbeddedObjects\_xjal( Panel \_panel, double \_x, double \_y, int \_clickCount, boolean \_publicOnly, boolean \_isSuperClass ) {

if ( source.onClickIconAt( \_x - 130, \_y - 90, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "source" );

} else {

\_panel.addInspect( \_x, \_y, this, "source" );

}

return true;

}

if ( queueFifo.onClickIconAt( \_x - 290, \_y - 90, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "queueFifo" );

} else {

\_panel.addInspect( \_x, \_y, this, "queueFifo" );

}

return true;

}

if ( chanelFifo.onClickIconAt( \_x - 430, \_y - 80, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "chanelFifo" );

} else {

\_panel.addInspect( \_x, \_y, this, "chanelFifo" );

}

return true;

}

if ( sinkFifo.onClickIconAt( \_x - 530, \_y - 80, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "sinkFifo" );

} else {

\_panel.addInspect( \_x, \_y, this, "sinkFifo" );

}

return true;

}

if ( selectOutput1.onClickIconAt( \_x - 370, \_y - 90, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "selectOutput1" );

} else {

\_panel.addInspect( \_x, \_y, this, "selectOutput1" );

}

return true;

}

if ( split.onClickIconAt( \_x - 150, \_y - 90, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "split" );

} else {

\_panel.addInspect( \_x, \_y, this, "split" );

}

return true;

}

if ( QueueRR.onClickIconAt( \_x - 220, \_y - 200, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "QueueRR" );

} else {

\_panel.addInspect( \_x, \_y, this, "QueueRR" );

}

return true;

}

if ( chanelRR.onClickIconAt( \_x - 410, \_y - 190, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "chanelRR" );

} else {

\_panel.addInspect( \_x, \_y, this, "chanelRR" );

}

return true;

}

if ( SinkRR.onClickIconAt( \_x - 610, \_y - 180, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "SinkRR" );

} else {

\_panel.addInspect( \_x, \_y, this, "SinkRR" );

}

return true;

}

if ( selectOutput4.onClickIconAt( \_x - 310, \_y - 200, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "selectOutput4" );

} else {

\_panel.addInspect( \_x, \_y, this, "selectOutput4" );

}

return true;

}

if ( selectOutput.onClickIconAt( \_x - 490, \_y - 190, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "selectOutput" );

} else {

\_panel.addInspect( \_x, \_y, this, "selectOutput" );

}

return true;

}

if ( split1.onClickIconAt( \_x - 150, \_y - 200, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "split1" );

} else {

\_panel.addInspect( \_x, \_y, this, "split1" );

}

return true;

}

if ( chanelFB.onClickIconAt( \_x - 520, \_y - 330, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "chanelFB" );

} else {

\_panel.addInspect( \_x, \_y, this, "chanelFB" );

}

return true;

}

if ( sinkFB.onClickIconAt( \_x - 680, \_y - 320, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "sinkFB" );

} else {

\_panel.addInspect( \_x, \_y, this, "sinkFB" );

}

return true;

}

if ( selectOutput3.onClickIconAt( \_x - 610, \_y - 330, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "selectOutput3" );

} else {

\_panel.addInspect( \_x, \_y, this, "selectOutput3" );

}

return true;

}

if ( queueFB0.onClickIconAt( \_x - 310, \_y - 330, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "queueFB0" );

} else {

\_panel.addInspect( \_x, \_y, this, "queueFB0" );

}

return true;

}

if ( queueFB1.onClickIconAt( \_x - 310, \_y - 420, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "queueFB1" );

} else {

\_panel.addInspect( \_x, \_y, this, "queueFB1" );

}

return true;

}

if ( selectOutput6.onClickIconAt( \_x - 180, \_y - 380, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "selectOutput6" );

} else {

\_panel.addInspect( \_x, \_y, this, "selectOutput6" );

}

return true;

}

if ( holdFB.onClickIconAt( \_x - 390, \_y - 420, true ) ) {

if ( \_clickCount == 2 ) {

\_panel.browseAgent\_xjal( \_x, \_y, this, "holdFB" );

} else {

\_panel.addInspect( \_x, \_y, this, "holdFB" );

}

return true;

}

return false;

}

@AnyLogicInternalCodegenAPI

private boolean onClickModelAt\_AgentLinks\_xjal( Panel \_panel, double \_x, double \_y, int \_clickCount, boolean \_publicOnly, boolean \_isSuperClass ) {

if ( modelElementContains(\_x, \_y, 50, -50) ) {

\_panel.addInspect\_xjal( 50, -50, this, "connections", Panel.INSPECT\_CONNECTIONS\_xjal );

return true;

}

return false;

}

@AnyLogicInternalCodegenAPI

private boolean onClickModelAt\_PlainVariables\_xjal( Panel \_panel, double \_x, double \_y, int \_clickCount, boolean \_publicOnly, boolean \_isSuperClass ) {

if( !\_publicOnly && modelElementContains(\_x, \_y, 30, 30) ) {

\_panel.addInspect( 30, 30, this, "lambda" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 30, 60) ) {

\_panel.addInspect( 30, 60, this, "mu" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 30, 90) ) {

\_panel.addInspect( 30, 90, this, "sigma" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 30, 120) ) {

\_panel.addInspect( 30, 120, this, "n" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 150, 60) ) {

\_panel.addInspect( 150, 60, this, "r" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 30, 150) ) {

\_panel.addInspect( 30, 150, this, "t1" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 30, 180) ) {

\_panel.addInspect( 30, 180, this, "t2" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 30, 210) ) {

\_panel.addInspect( 30, 210, this, "h" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 30, 240) ) {

\_panel.addInspect( 30, 240, this, "priorityFB" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 394, 41) ) {

\_panel.addInspect( 394, 41, this, "longQueue" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 393, 70) ) {

\_panel.addInspect( 393, 70, this, "queueIterat" );

return true;

}

return false;

}

@AnyLogicInternalCodegenAPI

private boolean onClickModelAt\_CollectionVariables\_xjal( Panel \_panel, double \_x, double \_y, int \_clickCount, boolean \_publicOnly, boolean \_isSuperClass ) {

if( !\_publicOnly && modelElementContains(\_x, \_y, 640, 130) ) {

\_panel.addInspect( 640, 130, this, "collectionFifo" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 720, 230) ) {

\_panel.addInspect( 720, 230, this, "collectionRR" );

return true;

}

if( !\_publicOnly && modelElementContains(\_x, \_y, 790, 370) ) {

\_panel.addInspect( 790, 370, this, "collectionFB" );

return true;

}

return false;

}

@Override

@AnyLogicInternalCodegenAPI

public boolean onClickModelAt( Panel \_panel, double \_x, double \_y, int \_clickCount, boolean \_publicOnly, boolean \_isSuperClass ) {

if ( onClickModelAt\_EmbeddedObjects\_xjal( \_panel, \_x, \_y, \_clickCount, \_publicOnly, \_isSuperClass ) ) { return true; }

if ( onClickModelAt\_AgentLinks\_xjal( \_panel, \_x, \_y, \_clickCount, \_publicOnly, \_isSuperClass ) ) { return true; }

if ( onClickModelAt\_PlainVariables\_xjal( \_panel, \_x, \_y, \_clickCount, \_publicOnly, \_isSuperClass ) ) { return true; }

if ( onClickModelAt\_CollectionVariables\_xjal( \_panel, \_x, \_y, \_clickCount, \_publicOnly, \_isSuperClass ) ) { return true; }

return super.onClickModelAt( \_panel, \_x, \_y, \_clickCount, \_publicOnly, true );

}

/\*\*

\* Конструктор

\*/

public Main( Engine engine, Agent owner, AgentList<? extends Main> collection ) {

super( engine, owner, collection );

if (isTopLevelClass\_xjal( Main.class )) {

instantiateBaseStructure\_xjal();

}

}

@AnyLogicInternalCodegenAPI

public void onOwnerChanged\_xjal() {

super.onOwnerChanged\_xjal();

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

public void instantiateBaseStructure\_xjal() {

super.instantiateBaseStructure\_xjal();

source = instantiate\_source\_xjal();

queueFifo = instantiate\_queueFifo\_xjal();

chanelFifo = instantiate\_chanelFifo\_xjal();

sinkFifo = instantiate\_sinkFifo\_xjal();

selectOutput1 = instantiate\_selectOutput1\_xjal();

split = instantiate\_split\_xjal();

QueueRR = instantiate\_QueueRR\_xjal();

chanelRR = instantiate\_chanelRR\_xjal();

SinkRR = instantiate\_SinkRR\_xjal();

selectOutput4 = instantiate\_selectOutput4\_xjal();

selectOutput = instantiate\_selectOutput\_xjal();

split1 = instantiate\_split1\_xjal();

chanelFB = instantiate\_chanelFB\_xjal();

sinkFB = instantiate\_sinkFB\_xjal();

selectOutput3 = instantiate\_selectOutput3\_xjal();

queueFB0 = instantiate\_queueFB0\_xjal();

queueFB1 = instantiate\_queueFB1\_xjal();

selectOutput6 = instantiate\_selectOutput6\_xjal();

holdFB = instantiate\_holdFB\_xjal();

setupReferences\_xjal();

}

@AnyLogicInternalCodegenAPI

private void setupReferences\_xjal() {

}

/\*\*

\* Simple constructor. Please add created agent to some population by calling goToPopulation() function

\*/

public Main() {

}

@Override

@AnyLogicInternalCodegenAPI

public void create() {

super.create();

// Создание экземпляров вложенных объектов

// Присвоение начальных значений простым переменным

setupPlainVariables\_Main\_xjal();

// Динамическая инициализация элементов, у которых разрешено программное управление

presentation = new ShapeTopLevelPresentationGroup( Main.this, true, 0, 0, 0, 0 );

icon = new ShapeGroup( Main.this, true, 0, 0, 0 );

// Создание нереплицированных вложенных объектов

setupParameters\_source\_xjal( source );

create\_source\_xjal( source );

setupParameters\_queueFifo\_xjal( queueFifo );

create\_queueFifo\_xjal( queueFifo );

setupParameters\_chanelFifo\_xjal( chanelFifo );

create\_chanelFifo\_xjal( chanelFifo );

setupParameters\_sinkFifo\_xjal( sinkFifo );

create\_sinkFifo\_xjal( sinkFifo );

setupParameters\_selectOutput1\_xjal( selectOutput1 );

create\_selectOutput1\_xjal( selectOutput1 );

setupParameters\_split\_xjal( split );

create\_split\_xjal( split );

setupParameters\_QueueRR\_xjal( QueueRR );

create\_QueueRR\_xjal( QueueRR );

setupParameters\_chanelRR\_xjal( chanelRR );

create\_chanelRR\_xjal( chanelRR );

setupParameters\_SinkRR\_xjal( SinkRR );

create\_SinkRR\_xjal( SinkRR );

setupParameters\_selectOutput4\_xjal( selectOutput4 );

create\_selectOutput4\_xjal( selectOutput4 );

setupParameters\_selectOutput\_xjal( selectOutput );

create\_selectOutput\_xjal( selectOutput );

setupParameters\_split1\_xjal( split1 );

create\_split1\_xjal( split1 );

setupParameters\_chanelFB\_xjal( chanelFB );

create\_chanelFB\_xjal( chanelFB );

setupParameters\_sinkFB\_xjal( sinkFB );

create\_sinkFB\_xjal( sinkFB );

setupParameters\_selectOutput3\_xjal( selectOutput3 );

create\_selectOutput3\_xjal( selectOutput3 );

setupParameters\_queueFB0\_xjal( queueFB0 );

create\_queueFB0\_xjal( queueFB0 );

setupParameters\_queueFB1\_xjal( queueFB1 );

create\_queueFB1\_xjal( queueFB1 );

setupParameters\_selectOutput6\_xjal( selectOutput6 );

create\_selectOutput6\_xjal( selectOutput6 );

setupParameters\_holdFB\_xjal( holdFB );

create\_holdFB\_xjal( holdFB );

// Соединители с нереплицированными объектами

sinkFifo.in.connect( chanelFifo.out ); // connector2

selectOutput1.in.connect( queueFifo.out ); // connector1

selectOutput1.outT.connect( chanelFifo.in ); // connector3

selectOutput1.outF.connect( sinkFifo.in ); // connector4

split.in.connect( source.out ); // connector

split.out.connect( queueFifo.in ); // connector5

selectOutput4.in.connect( QueueRR.out ); // connector18

selectOutput4.outT.connect( chanelRR.in ); // connector19

selectOutput4.outF.connect( SinkRR.in ); // connector20

selectOutput.in.connect( chanelRR.out ); // connector7

selectOutput.outT.connect( SinkRR.in ); // connector8

selectOutput.outF.connect( QueueRR.in ); // connector9

split1.out.connect( QueueRR.in ); // connector6

split.outCopy.connect( split1.in ); // connector10

selectOutput3.in.connect( chanelFB.out ); // connector15

selectOutput3.outT.connect( sinkFB.in ); // connector16

selectOutput6.outF.connect( queueFB1.in ); // connector25

holdFB.in.connect( queueFB1.out ); // connector26

queueFB0.in.connect( selectOutput6.outT ); // connector14

selectOutput6.in.connect( split1.outCopy ); // connector17

queueFB0.out.connect( chanelFB.in ); // connector11

holdFB.out.connect( chanelFB.in ); // connector12

selectOutput3.outF.connect( queueFB1.in ); // connector13

// Создание реплицированных вложенных объектов

setupInitialConditions\_xjal( Main.class );

if (isTopLevelClass\_xjal( Main.class )) {

onCreate();

}

}

@Override

@AnyLogicInternalCodegenAPI

public void start() {

super.start();

source.start();

queueFifo.start();

chanelFifo.start();

sinkFifo.start();

selectOutput1.start();

split.start();

QueueRR.start();

chanelRR.start();

SinkRR.start();

selectOutput4.start();

selectOutput.start();

split1.start();

chanelFB.start();

sinkFB.start();

selectOutput3.start();

queueFB0.start();

queueFB1.start();

selectOutput6.start();

holdFB.start();

if (isTopLevelClass\_xjal( Main.class )) {

onStartup();

}

}

/\*\*

\* Присвоение начальных значений простым переменным<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

public void setupPlainVariables\_xjal() {

setupPlainVariables\_Main\_xjal();

}

/\*\*

\* Присвоение начальных значений простым переменным<br>

\* <em>This method isn't designed to be called by user and may be removed in future releases.</em>

\*/

@AnyLogicInternalCodegenAPI

private void setupPlainVariables\_Main\_xjal() {

sigma =

2

;

r =

new Random();

;

}

// API пользователя -----------------------------------------------------

@AnyLogicInternalCodegenAPI

static LinkToAgentAnimationSettings \_connections\_commonAnimationSettings\_xjal = new LinkToAgentAnimationSettingsImpl( false, black, 1.0, LINE\_STYLE\_SOLID, ARROW\_NONE, 0.0 );

public LinkToAgentCollection<Agent, Agent> connections = new LinkToAgentStandardImpl<Agent, Agent>(this, \_connections\_commonAnimationSettings\_xjal);

@Override

public LinkToAgentCollection<? extends Agent, ? extends Agent> getLinkToAgentStandard\_xjal() {

return connections;

}

@AnyLogicInternalCodegenAPI

public void drawLinksToAgents(boolean \_underAgents\_xjal, LinkToAgentAnimator \_animator\_xjal) {

super.drawLinksToAgents(\_underAgents\_xjal, \_animator\_xjal);

if ( \_underAgents\_xjal ) {

\_animator\_xjal.drawLink( this, connections, true, true );

}

}

public List<Object> getEmbeddedObjects() {

List<Object> list = super.getEmbeddedObjects();

if (list == null) {

list = new LinkedList<Object>();

}

list.add( source );

list.add( queueFifo );

list.add( chanelFifo );

list.add( sinkFifo );

list.add( selectOutput1 );

list.add( split );

list.add( QueueRR );

list.add( chanelRR );

list.add( SinkRR );

list.add( selectOutput4 );

list.add( selectOutput );

list.add( split1 );

list.add( chanelFB );

list.add( sinkFB );

list.add( selectOutput3 );

list.add( queueFB0 );

list.add( queueFB1 );

list.add( selectOutput6 );

list.add( holdFB );

return list;

}

public AgentList<? extends Main> getPopulation() {

return (AgentList<? extends Main>) super.getPopulation();

}

public List<? extends Main> agentsInRange( double distance ) {

return (List<? extends Main>) super.agentsInRange( distance );

}

@AnyLogicInternalCodegenAPI

public void onDestroy() {

source.onDestroy();

queueFifo.onDestroy();

chanelFifo.onDestroy();

sinkFifo.onDestroy();

selectOutput1.onDestroy();

split.onDestroy();

QueueRR.onDestroy();

chanelRR.onDestroy();

SinkRR.onDestroy();

selectOutput4.onDestroy();

selectOutput.onDestroy();

split1.onDestroy();

chanelFB.onDestroy();

sinkFB.onDestroy();

selectOutput3.onDestroy();

queueFB0.onDestroy();

queueFB1.onDestroy();

selectOutput6.onDestroy();

holdFB.onDestroy();

super.onDestroy();

}

}

/\*\*

\* Task

\*/

public class Task extends Agent implements Serializable {

double buf=0;

double timeProcess=0;

double timeWork=0;

double timeSystem=0;

double timeEntered=0;

double timeResponse=0;

double actuality=0;

boolean fifoFlag=true;

boolean rrFlag = true;

boolean flagRR;

boolean fbFlag;

boolean flagFB;

boolean flagReact=true;

public void actuality(double time) {

if (time>=Main.t2+Main.t1){

actuality=0;

fifoFlag = false;

}

else if (time<=Main.t1){

actuality=1;

fifoFlag=true;

}

else{

actuality =-0.2\*time+1.6;

fifoFlag=true;

}

}

/\*\*

\* Конструктор по умолчанию

\*/

public Task() {

}

@Override

public String toString() {

return super.toString();

}

/\*\*

\* Это число используется при сохранении состояния модели<br>

\* Его рекомендуется изменить в случае изменения класса

\*/

private static final long serialVersionUID = 1L;

} \* Его рекомендуется изменить в случае изменения класса

\*/

private static final long serialVersionUID = 1L;

}double meanFifo =0;

double meanResponseFifo=0;

double notDelTaskFifo=0;

double actualityFifo=0;

double tasksFifo=0;

ArrayList<Task> collFifo = Main.collectionFifo;

Iterator<Task> iterator = collFifo.iterator();

while (iterator.hasNext()){

Task t = iterator.next();

meanFifo+=t.timeSystem;

meanResponseFifo+=t.timeResponse;

actualityFifo+=t.actuality;

if (t.actuality!=0)

notDelTaskFifo+=1;

}

double duspFifo=0;

iterator = collFifo.iterator();

meanFifo =meanFifo/new Double (collFifo.size());

while (iterator.hasNext()){

Task t = iterator.next();

duspFifo+=Math.pow(t.timeSystem - meanFifo, 2);

}

duspFifo = duspFifo/new Double(collFifo.size()-1);

meanResponseFifo = meanResponseFifo/new Double (collFifo.size());

tasksFifo = notDelTaskFifo/new Double (collFifo.size());

actualityFifo = actualityFifo/new Double(collFifo.size());

System.out.println("Середній час задачі в системі FIFO ");

System.out.printf("%3.5f",meanFifo);

System.out.println("\nДисперсія середнього часу в системі FIFO ");

System.out.printf("%3.5f", duspFifo);

System.out.println("\nСередній час реакції FIFO ");

System.out.printf("%3.5f",meanResponseFifo);

System.out.println("\nВідсоток оброблених задач ");

System.out.printf("%3.5f", tasksFifo);

System.out.println("\nСумарна оцінка актуальності ");

System.out.printf("%3.5f",actualityFifo);

System.out.println("\n-----------------------------------");

double meanRR =0;

double meanResponseRR=0;

double notDelTaskRR=0;

double actualityRR=0;

double tasksRR=0;

ArrayList<Task> collRR = Main.collectionRR;

iterator = collRR.iterator();

while (iterator.hasNext()){

Task t = iterator.next();

meanRR+=t.timeSystem;

meanResponseRR+=t.timeResponse;

actualityRR+=t.actuality;

if (t.actuality!=0)

notDelTaskRR+=1;

}

double duspRR=0;

iterator = collFifo.iterator();

meanRR =meanRR/new Double (collRR.size());

while (iterator.hasNext()){

Task t = iterator.next();

duspRR+=Math.pow(t.timeSystem-meanRR, 2);

}

duspRR = duspRR/new Double(collRR.size());

meanResponseRR = meanResponseRR/new Double (collRR.size());

tasksRR = notDelTaskRR/new Double (collRR.size());

actualityRR = actualityRR/new Double(collRR.size());

System.out.println("\nСередній час задачі в системі RR ");

System.out.printf("%3.5f",meanRR);

System.out.println("\nДисперсія середнього часу в системі RR ");

System.out.printf("%3.5f", duspRR);

System.out.println("\nСередній час реакції RR ");

System.out.printf("%3.5f",meanResponseRR);

System.out.println("\nВідсоток оброблених задач ");

System.out.printf("%3.5f", tasksRR);

System.out.println("\nСумарна оцінка актуальності ");

System.out.printf("%3.5f",actualityRR);

System.out.println("\nЗначення цільової функції");

System.out.printf("%3.5f",Main.taskFunk(meanRR, duspRR, meanResponseRR, tasksRR, actualityRR));

System.out.println("\n-----------------------------------");

double meanFB =0;

double meanResponseFB=0;

double notDelTaskFB=0;

double actualityFB=0;

double tasksFB=0;

ArrayList<Task> collFB = Main.collectionFB;

iterator = collFB.iterator();

while (iterator.hasNext()){

Task t = iterator.next();

meanFB+=t.timeSystem;

meanResponseFB+=t.timeResponse;

actualityFB+=t.actuality;

if (t.actuality!=0)

notDelTaskFB+=1;

}

double duspFB=0;

iterator = collFifo.iterator();

meanFB = meanFB/new Double (collFB.size());

while (iterator.hasNext()){

Task t = iterator.next();

duspFB+=Math.pow(meanFB-t.timeSystem, 2);

}

duspFB = duspFB/new Double(collFB.size()-1);

meanResponseFB = meanResponseFB/new Double (collFB.size());

tasksFB = notDelTaskFB/new Double (collRR.size());

actualityFB = actualityFB/new Double(collFB.size());

System.out.println("\nСередній час задачі в системі FB ");

System.out.printf("%3.5f",meanFB);

System.out.println("\nДисперсія середнього часу в системі FB ");

System.out.printf("%3.5f", duspFB);

System.out.println("\nСередній час реакції FB ");

System.out.printf("%3.5f",meanResponseFB);

System.out.println("\nВідсоток оброблених задач ");

System.out.printf("%3.5f", tasksFB);

System.out.println("\nСумарна оцінка актуальності ");

System.out.printf("%3.5f",actualityFB);

System.out.println("\nЗначення цільової функції");

System.out.printf("%3.5f",Main.taskFunk(meanFB, duspRR, meanResponseRR, tasksRR, actualityRR));

Main.collectionFifo.clear();

Main.collectionRR.clear();

Main.collectionFB.clear();

package cmlab2;

import java.io.Serializable;

import java.sql.Connection;

import java.sql.SQLException;

import java.util.ArrayDeque;

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.LinkedHashMap;

import java.util.LinkedHashSet;

import java.util.LinkedList;

import java.util.List;

import java.util.ListIterator;

import java.util.Locale;

import java.util.Map;

import java.util.PriorityQueue;

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

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

import com.xj.anylogic.engine.markup.Network;

import com.xj.anylogic.engine.Position;

import com.xj.anylogic.engine.markup.PedFlowStatistics;

import com.xj.anylogic.engine.markup.DensityMap;

import static java.lang.Math.\*;

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

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.markup.\*;

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

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

import javax.swing.JApplet;

public class Simulation extends ExperimentSimulation<Main> {

@AnyLogicInternalCodegenAPI

public static String[] COMMAND\_LINE\_ARGUMENTS\_xjal = new String[0];

{

setCommandLineArguments\_xjal( COMMAND\_LINE\_ARGUMENTS\_xjal );

}

// Области

@AnyLogicInternalCodegenAPI

protected static final Font \_button\_Font = new Font("Dialog", 0, 11 );

@AnyLogicInternalCodegenAPI

protected static final Font \_editbox\_Font = \_button\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_editbox1\_Font = \_button\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_editbox2\_Font = \_button\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_editbox3\_Font = \_button\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text\_Font = new Font("SansSerif", 0, 24 );

@AnyLogicInternalCodegenAPI

protected static final Font \_text1\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text2\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text3\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final Font \_text4\_Font = \_text\_Font;

@AnyLogicInternalCodegenAPI

protected static final int \_button = 1;

@AnyLogicInternalCodegenAPI

protected static final int \_editbox = 2;

@AnyLogicInternalCodegenAPI

protected static final int \_editbox1 = 3;

@AnyLogicInternalCodegenAPI

protected static final int \_editbox2 = 4;

@AnyLogicInternalCodegenAPI

protected static final int \_editbox3 = 5;

@AnyLogicInternalCodegenAPI

protected static final int \_text = 6;

@AnyLogicInternalCodegenAPI

protected static final int \_text1 = 7;

@AnyLogicInternalCodegenAPI

protected static final int \_text2 = 8;

@AnyLogicInternalCodegenAPI

protected static final int \_text3 = 9;

@AnyLogicInternalCodegenAPI

protected static final int \_text4 = 10;

/\*\* Internal constant, shouldn't be accessed by user \*/

@AnyLogicInternalCodegenAPI

protected static final int \_SHAPE\_NEXT\_ID\_xjal = 11;

/\*\*

\* Идентификатор группы presentation верхнего уровня

\*/

@AnyLogicInternalCodegenAPI

protected static final int \_presentation = 0;

/\*\*

\* Идентификатор группы icon верхнего уровня

\*/

@AnyLogicInternalCodegenAPI

protected static final int \_icon = -1;

@Override

@AnyLogicInternalCodegenAPI

public void executeShapeControlAction( int \_shape, int index ) {

switch( \_shape ) {

case \_button: {

ShapeButton self = this.button;

if ( getState() == IDLE )

run();

getPresentation().setPresentable( getEngine().getRoot() );

;}

break;

default:

super.executeShapeControlAction( \_shape, index );

break;

}

}

@Override

@AnyLogicInternalCodegenAPI

public void executeShapeControlAction( int \_shape, int index, String value ) {

switch( \_shape ) {

case \_editbox: {

ShapeTextField self = this.editbox;

Main.lambda = Double.parseDouble(editbox.getText());

;}

break;

case \_editbox1: {

ShapeTextField self = this.editbox1;

Main.mu =Double.parseDouble(editbox1.getText());

;}

break;

case \_editbox2: {

ShapeTextField self = this.editbox2;

Main.h = Double.parseDouble(editbox2.getText());

;}

break;

case \_editbox3: {

ShapeTextField self = this.editbox3;

Main.priorityFB = Double.parseDouble(editbox3.getText());

;}

break;

default:

super.executeShapeControlAction( \_shape, index, value );

break;

}

}

@Override

@AnyLogicInternalCodegenAPI

public String getShapeControlDefaultValueString( int \_shape, int index ) {

switch(\_shape) {

case \_editbox: return

Double.toString(Main.lambda)

;

case \_editbox1: return

Double.toString(Main.mu)

;

case \_editbox2: return

Double.toString(Main.h)

;

case \_editbox3: return

Double.toString(Main.priorityFB)

;

default: return super.getShapeControlDefaultValueString( \_shape, index );

}

}

/\*\*

\* <i>Пользователь не должен вызывать этот метод</i>

\*/

@AnyLogicInternalCodegenAPI

private void \_button\_SetDynamicParams\_xjal( ShapeButton shape ) {

shape.setText(

getState() == IDLE ?

"Запустить" :

"Агент верхнего уровня"

);

}

protected ShapeButton button;

protected ShapeTextField editbox;

protected ShapeTextField editbox1;

protected ShapeTextField editbox2;

protected ShapeTextField editbox3;

protected ShapeText text;

protected ShapeText text1;

protected ShapeText text2;

protected ShapeText text3;

protected ShapeText text4;

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsBP0\_xjal() {

button = new ShapeButton(

Simulation.this, true, 140.0, 250.0,

100.0, 30.0,

controlDefault, controlDefault,

\_button\_Font,

"Запустить" ) {

@Override

public void updateDynamicProperties(boolean publicOnly) {

\_button\_SetDynamicParams\_xjal( this );

super.updateDynamicProperties(publicOnly);

}

@Override

@AnyLogicInternalCodegenAPI

public void action(){

executeShapeControlAction( \_button, 0 );

}

};

editbox = new ShapeTextField(

Simulation.this, true,140.0, 90.0,

100.0, 30.0,

controlDefault, controlDefault, \_editbox\_Font ) {

@Override

@AnyLogicInternalCodegenAPI

public void action(){

executeShapeControlAction( \_editbox, 0, value );

}

@Override

public void setValueToDefault() {

setText( getShapeControlDefaultValueString( \_editbox, 0 ) );

}

};

editbox1 = new ShapeTextField(

Simulation.this, true,140.0, 130.0,

100.0, 30.0,

controlDefault, controlDefault, \_editbox1\_Font ) {

@Override

@AnyLogicInternalCodegenAPI

public void action(){

executeShapeControlAction( \_editbox1, 0, value );

}

@Override

public void setValueToDefault() {

setText( getShapeControlDefaultValueString( \_editbox1, 0 ) );

}

};

editbox2 = new ShapeTextField(

Simulation.this, true,140.0, 170.0,

100.0, 30.0,

controlDefault, controlDefault, \_editbox2\_Font ) {

@Override

@AnyLogicInternalCodegenAPI

public void action(){

executeShapeControlAction( \_editbox2, 0, value );

}

@Override

public void setValueToDefault() {

setText( getShapeControlDefaultValueString( \_editbox2, 0 ) );

}

};

editbox3 = new ShapeTextField(

Simulation.this, true,140.0, 210.0,

100.0, 30.0,

controlDefault, controlDefault, \_editbox3\_Font ) {

@Override

@AnyLogicInternalCodegenAPI

public void action(){

executeShapeControlAction( \_editbox3, 0, value );

}

@Override

public void setValueToDefault() {

setText( getShapeControlDefaultValueString( \_editbox3, 0 ) );

}

};

text = new ShapeText(

SHAPE\_DRAW\_2D, true,30.0, 30.0, 0.0, 0.0,

black,"Лабораторна робота №2 Моделювання СМО",

\_text\_Font, ALIGNMENT\_LEFT );

text1 = new ShapeText(

SHAPE\_DRAW\_2D, true,30.0, 90.0, 0.0, 0.0,

black,"Lambda",

\_text1\_Font, ALIGNMENT\_LEFT );

text2 = new ShapeText(

SHAPE\_DRAW\_2D, true,30.0, 130.0, 0.0, 0.0,

black,"Mu",

\_text2\_Font, ALIGNMENT\_LEFT );

text3 = new ShapeText(

SHAPE\_DRAW\_2D, true,30.0, 170.0, 0.0, 0.0,

black,"h",

\_text3\_Font, ALIGNMENT\_LEFT );

text4 = new ShapeText(

SHAPE\_DRAW\_2D, true,30.0, 210.0, 0.0, 0.0,

black,"priorityFB",

\_text4\_Font, ALIGNMENT\_LEFT );

}

@AnyLogicInternalCodegenAPI

private void \_createPersistentElementsAP0\_xjal() {

}

// Статическая инициализация элементов, у которых разрешено программное управление

{

\_createPersistentElementsBP0\_xjal();

}

protected ShapeTopLevelPresentationGroup presentation;

protected ShapeGroup icon;

@Override

@AnyLogicInternalCodegenAPI

public Object getPersistentShape( int \_shape ) {

switch (\_shape) {

case \_presentation: return presentation;

case \_icon: return icon;

case \_button: return button;

case \_editbox: return editbox;

case \_editbox1: return editbox1;

case \_editbox2: return editbox2;

case \_editbox3: return editbox3;

case \_text: return text;

case \_text1: return text1;

case \_text2: return text2;

case \_text3: return text3;

case \_text4: return text4;

default: return super.getPersistentShape( \_shape );

}

}

@Override

@AnyLogicInternalCodegenAPI

public String getNameOfShape\_xjal( Object \_shape ) {

try {

if ( \_shape == null ) return null;

String \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, presentation, "presentation" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, icon, "icon" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, button, "button" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, editbox, "editbox" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, editbox1, "editbox1" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, editbox2, "editbox2" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, editbox3, "editbox3" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, text, "text" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, text1, "text1" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, text2, "text2" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, text3, "text3" ); if (\_name\_xjal != null) return \_name\_xjal;

\_name\_xjal = checkNameOfShape\_xjal( \_shape, text4, "text4" ); if (\_name\_xjal != null) return \_name\_xjal;

} catch (Exception e) {

return null;

}

return super.getNameOfShape\_xjal( \_shape );

}

@Override

public int getWindowWidth() {

return 1000;

}

@Override

public int getWindowHeight() {

return 600;

}

@Override

@AnyLogicInternalCodegenAPI

public void onDestroy\_xjal() {

onDestroy();

super.onDestroy\_xjal();

}

/\*\*

\* Класс, который будет запущен при запуске эксперимента в виде апплета

\*/

@AnyLogicInternalCodegenAPI

public static class Applet extends JApplet {

@AnyLogicInternalCodegenAPI

Simulation ex;

@Override

public void init() {

ex = new Simulation();

ex.setup( this );

}

@Override

public void destroy() {

ex.close();

}

}

@Override

@AnyLogicInternalCodegenAPI

public void initDefaultRandomNumberGenerator(Engine \_e) {

\_e.getDefaultRandomGenerator().setSeed( 1 );

}

@Override

@AnyLogicInternalCodegenAPI

public Main createRoot( Engine engine ) {

// Создание корневого объекта

return new Main( engine, null, null );

}

@Override

@AnyLogicInternalCodegenAPI

public void setupRootParameters( final Main self, boolean callOnChangeActions ) {

final Main root = self; // for compatibility

}

/\*\*

\* Инициализация исполняющего модуля

\*/

@Override

@AnyLogicInternalCodegenAPI

public void setupEngine(Engine engine) {

engine.setATOL( 1.0E-5 );

engine.setRTOL( 1.0E-5 );

engine.setTTOL( 1.0E-5 );

engine.setHTOL( 0.0010 );

engine.setSolverODE( Engine.SOLVER\_ODE\_EULER );

engine.setSolverNAE( Engine.SOLVER\_NAE\_MODIFIED\_NEWTON );

engine.setSolverDAE( Engine.SOLVER\_DAE\_RK45\_NEWTON );

engine.setVMethods( 394292 );

engine.setSimultaneousEventsSelectionMode( Engine.EVENT\_SELECTION\_RANDOM );

engine.setStartTime( 0.0 );

engine.setTimeUnit( TIME\_UNIT\_MILLISECOND );

engine.setRealTimeMode( true );

engine.setRealTimeScale( 1.0 );

}

/\*\*

\* Инициализация эксперимента

\*/

@Override

@AnyLogicInternalCodegenAPI

public void setup( java.awt.Container container ) {

setName( "cmLab2 : Simulation" );

// Динамическая инициализация элементов, у которых разрешено программное управление

\_createPersistentElementsAP0\_xjal();

presentation = new ShapeTopLevelPresentationGroup( Simulation.this, true, 0, 0, 0, 0 , text, text1, text2, text3, text4, button, editbox, editbox1, editbox2, editbox3 );

icon = new ShapeGroup( Simulation.this, true, 0, 0, 0 );

editbox.setValueToDefault();

editbox1.setValueToDefault();

editbox2.setValueToDefault();

editbox3.setValueToDefault();

// Настройка презентации

Presentation \_p = new Presentation( this, container instanceof JApplet ? Presentation.MODE\_APPLET :

container != null ? Presentation.MODE\_COMPONENT :

Presentation.MODE\_APPLICATION, container );

\_p.start();

Panel \_panel = \_p.getPanel();

ToolBar \_tb = \_p.getToolBar();

StatusBar \_sb = \_p.getStatusBar();

\_panel.setFrameManagementBalance( 2.0 );

\_sb.setSectionVisible( StatusBar.DATE, true );

\_sb.setSectionVisible( StatusBar.EPS, false );

\_sb.setSectionVisible( StatusBar.EXPERIMENT, false );

\_sb.setSectionVisible( StatusBar.FPS, false );

\_sb.setSectionVisible( StatusBar.MEMORY, true );

\_sb.setSectionVisible( StatusBar.SECONDS, false );

\_sb.setSectionVisible( StatusBar.SIMULATION, true );

\_sb.setSectionVisible( StatusBar.STATUS, true );

\_sb.setSectionVisible( StatusBar.STEP, false );

\_sb.setSectionVisible( StatusBar.TIME, true );

\_tb.setSectionVisible( ToolBar.ANIMATION, false );

\_tb.setSectionVisible( ToolBar.EXECUTION, true );

\_tb.setSectionVisible( ToolBar.FILE, false );

\_tb.setSectionVisible( ToolBar.NAVIGATION, true );

\_tb.setSectionVisible( ToolBar.TIME\_SCALE, true );

\_tb.setSectionVisible( ToolBar.VIEW, false );

}

@Override

@AnyLogicInternalCodegenAPI

public void onEngineFinished() {

final Main root = (Main) getEngine().getRoot();

// Действие после каждого "прогона"

double meanFifo =0;

double meanResponseFifo=0;

double notDelTaskFifo=0;

double actualityFifo=0;

double tasksFifo=0;

ArrayList<Task> collFifo = Main.collectionFifo;

Iterator<Task> iterator = collFifo.iterator();

while (iterator.hasNext()){

Task t = iterator.next();

meanFifo+=t.timeSystem;

meanResponseFifo+=t.timeResponse;

actualityFifo+=t.actuality;

if (t.actuality!=0)

notDelTaskFifo+=1;

}

double duspFifo=0;

iterator = collFifo.iterator();

meanFifo =meanFifo/Main.n;

while (iterator.hasNext()){

Task t = iterator.next();

duspFifo+=Math.pow(t.timeSystem - meanFifo, 2);

}

duspFifo = duspFifo/(Main.n-1);

meanResponseFifo = meanResponseFifo/Main.n;

tasksFifo = notDelTaskFifo/Main.n;

actualityFifo = actualityFifo/Main.n;

System.out.println("Середній час задачі в системі FIFO ");

System.out.printf("%3.5f",meanFifo);

System.out.println("\nДисперсія середнього часу в системі FIFO ");

System.out.printf("%3.5f", duspFifo);

System.out.println("\nСередній час реакції FIFO ");

System.out.printf("%3.5f",meanResponseFifo);

System.out.println("\nВідсоток оброблених задач ");

System.out.printf("%3.5f", tasksFifo);

System.out.println("\nСумарна оцінка актуальності ");

System.out.printf("%3.5f",actualityFifo);

System.out.println("\n");

System.out.println("Цільова функція FiFO");

System.out.printf("%3.5f",Main.taskFunk(meanFifo, duspFifo, meanResponseFifo, tasksFifo, actualityFifo));

//double res =Main.longQueue/Main.queueIterat;

//System.out.printf("%3.5f",res);

System.out.println("\n-----------------------------------");

double meanRR =0;

double meanResponseRR=0;

double notDelTaskRR=0;

double actualityRR=0;

double tasksRR=0;

ArrayList<Task> collRR = Main.collectionRR;

iterator = collRR.iterator();

while (iterator.hasNext()){

Task t = iterator.next();

meanRR+=t.timeSystem;

meanResponseRR+=t.timeResponse;

actualityRR+=t.actuality;

if (t.actuality!=0)

notDelTaskRR+=1;

}

double duspRR=0;

iterator = collFifo.iterator();

meanRR =meanRR/Main.n;

while (iterator.hasNext()){

Task t = iterator.next();

duspRR+=Math.pow(t.timeSystem-meanRR, 2);

}

duspRR = duspRR/Main.n;

meanResponseRR = meanResponseRR/Main.n;

tasksRR = notDelTaskRR/Main.n;

actualityRR = actualityRR/Main.n;

System.out.println("\nСередній час задачі в системі RR ");

System.out.printf("%3.5f",meanRR);

System.out.println("\nДисперсія середнього часу в системі RR ");

System.out.printf("%3.5f", duspRR);

System.out.println("\nСередній час реакції RR ");

System.out.printf("%3.5f",meanResponseRR);

System.out.println("\nВідсоток оброблених задач ");

System.out.printf("%3.5f", tasksRR);

System.out.println("\nСумарна оцінка актуальності ");

System.out.printf("%3.5f",actualityRR);

System.out.println("\nЗначення цільової функції");

System.out.printf("%3.5f",Main.taskFunk(meanRR, duspRR, meanResponseRR, tasksRR, actualityRR));

System.out.println("\n-----------------------------------");

double meanFB =0;

double meanResponseFB=0;

double notDelTaskFB=0;

double actualityFB=0;

double tasksFB=0;

ArrayList<Task> collFB = Main.collectionFB;

iterator = collFB.iterator();

while (iterator.hasNext()){

Task t = iterator.next();

meanFB+=t.timeSystem;

meanResponseFB+=t.timeResponse;

actualityFB+=t.actuality;

if (t.actuality!=0)

notDelTaskFB+=1;

}

double duspFB=0;

iterator = collFifo.iterator();

meanFB = meanFB/Main.n;

while (iterator.hasNext()){

Task t = iterator.next();

duspFB+=Math.pow(meanFB-t.timeSystem, 2);

}

duspFB = duspFB/(Main.n-1);

meanResponseFB = meanResponseFB/Main.n;

tasksFB = notDelTaskFB/Main.n;

actualityFB = actualityFB/Main.n;

System.out.println("\nСередній час задачі в системі FB ");

System.out.printf("%3.5f",meanFB);

System.out.println("\nДисперсія середнього часу в системі FB ");

System.out.printf("%3.5f", duspFB);

System.out.println("\nСередній час реакції FB ");

System.out.printf("%3.5f",meanResponseFB);

System.out.println("\nВідсоток оброблених задач ");

System.out.printf("%3.5f", tasksFB);

System.out.println("\nСумарна оцінка актуальності ");

System.out.printf("%3.5f",actualityFB);

System.out.println("\nЗначення цільової функції");

System.out.printf("%3.5f",Main.taskFunk(meanFB, duspRR, meanResponseRR, tasksRR, actualityRR));

Main.collectionFifo.clear();

Main.collectionRR.clear();

Main.collectionFB.clear();

}

}