Міністерство освіти і науки, молоді та спорту України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №2**

З дисципліни «Комп’ютерне моделювання»

Виконав: Перевірив:

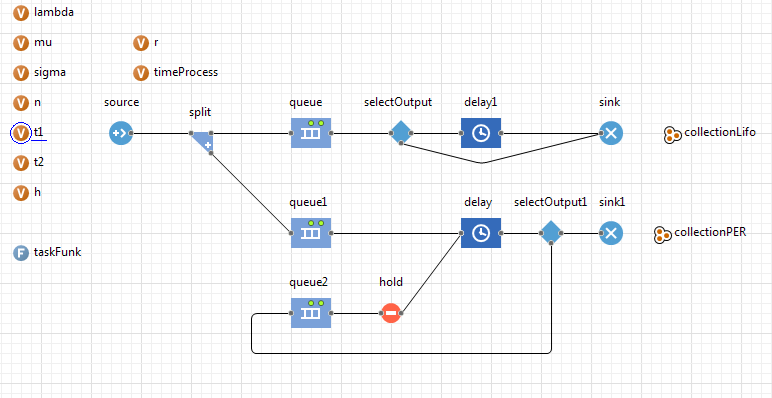
Студент групи ІО-21 доц. Марковський О.П.

Коноз А.О.

Дата здачі\_\_\_\_\_\_\_\_\_\_\_\_\_

Захищено з балом\_\_\_\_\_

Київ 2014

****

**Лістинг коду**

/\*\*

\* Command

\*/

**public** **class** Task **extends** Agent **implements** Serializable {

**public** **double** enteringtime = 0.0;

**public** **double** responsingtime = 0.0;

**double** systemtime = 0.0;

**double** actuality = 0.0;

**double** buf = 0.0;

**double** worktime = 0.0;

**double** processingtime = 0.0;

**boolean** lifointel = **false**;

**boolean** perintel;

**boolean** reactintel = **true**;

/\*\*

\* Конструктор по умолчанию

\*/

**public** Task() {

}

/\*\*

\* Конструктор, инициализирующий поля

\*/

**public** **void** actuality(**double** time) {

// System.out.println(time);

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

actuality=0.0;

lifointel = **false**;

// System.out.println(0.0);

}

**else** **if** (time<=Main.*t1*){

actuality=1.0;

lifointel=**true**;

}

**else**{

actuality =(time-Main.*t1*)/Main.*t2*;

lifointel=**true**;

}

}

**public** Task(**double** appeartime, **double** finishtime) {

**this**.enteringtime = appeartime;

**this**.responsingtime = finishtime;

}

@Override

**public** String toString() {

**return**

"appeartime = " + enteringtime +" " +

"finishtime = " + responsingtime +" ";

}

/\*\*

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

\* Его рекомендуется изменить в случае изменения класса

\*/

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

}

**double** meanLifo =0;

**double** meanResponseLifo=0;

**double** notDelTaskLifo=0;

**double** actualityLifo=0;

**double** tasksLifo=0;

ArrayList<Task> collLifo = Main.collectionLifo;

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

**while** (iterator.hasNext()){

Task t = iterator.next();

meanLifo+=t.systemtime;

meanResponseLifo+=t.responsingtime;

actualityLifo+=t.actuality;

**if** (t.actuality!=0)

notDelTaskLifo+=1;

}

**double** duspLifo=0;

iterator = collLifo.iterator();

meanLifo =meanLifo/Main.n;

**while** (iterator.hasNext()){

Task t = iterator.next();

duspLifo+=Math.pow(t.systemtime - meanLifo, 2);

}

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

meanResponseLifo = meanResponseLifo/Main.n;

tasksLifo = notDelTaskLifo/Main.n;

actualityLifo = actualityLifo/Main.n;

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

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

System.out.println();

System.out.println("Дисперсія середнього часу в системі Lifo ");

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

System.out.println();

System.out.println("Середній час реакції Lifo ");

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

System.out.println();

System.out.println("Відсоток оброблених задач ");

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

System.out.println();

System.out.println("Сумарна оцінка актуальності ");

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

System.out.println();

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

System.out.printf("%3.5f",Main.taskFunk(meanLifo, duspLifo, meanResponseLifo, tasksLifo, actualityLifo));

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

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

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

**double** meanPER =0;

**double** meanResponsePER=0;

**double** notDelTaskPER=0;

**double** actualityPER=0;

**double** tasksPER=0;

ArrayList<Task> collPER = Main.collectionPER;

iterator = collPER.iterator();

**while** (iterator.hasNext()){

Task t = iterator.next();

meanPER+=t.systemtime;

meanResponsePER+=t.responsingtime;

actualityPER+=t.actuality;

**if** (t.actuality!=0)

notDelTaskPER+=1;

}

**double** duspPER=0;

iterator = collPER.iterator();

meanPER = meanPER/Main.n;

**while** (iterator.hasNext()){

Task t = iterator.next();

duspPER+=Math.pow(meanPER-t.systemtime, 2);

}

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

meanResponsePER = meanResponsePER/Main.n;

tasksPER = notDelTaskPER/Main.n;

actualityPER = actualityPER/Main.n;

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

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

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

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

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

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

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

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

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

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

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

System.out.printf("%3.5f",Main.taskFunk(meanPER, duspPER, meanResponsePER, tasksPER, actualityPER));

Main.collectionLifo.clear();

Main.collectionPER.clear();