НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО”

КАФЕДРА АВТОМАТИЗОВАНИХ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ І УПРАВЛІННЯ

Комп’ютерний практикум № 11

з дисципліни

“Моделювання систем”

Виконала:

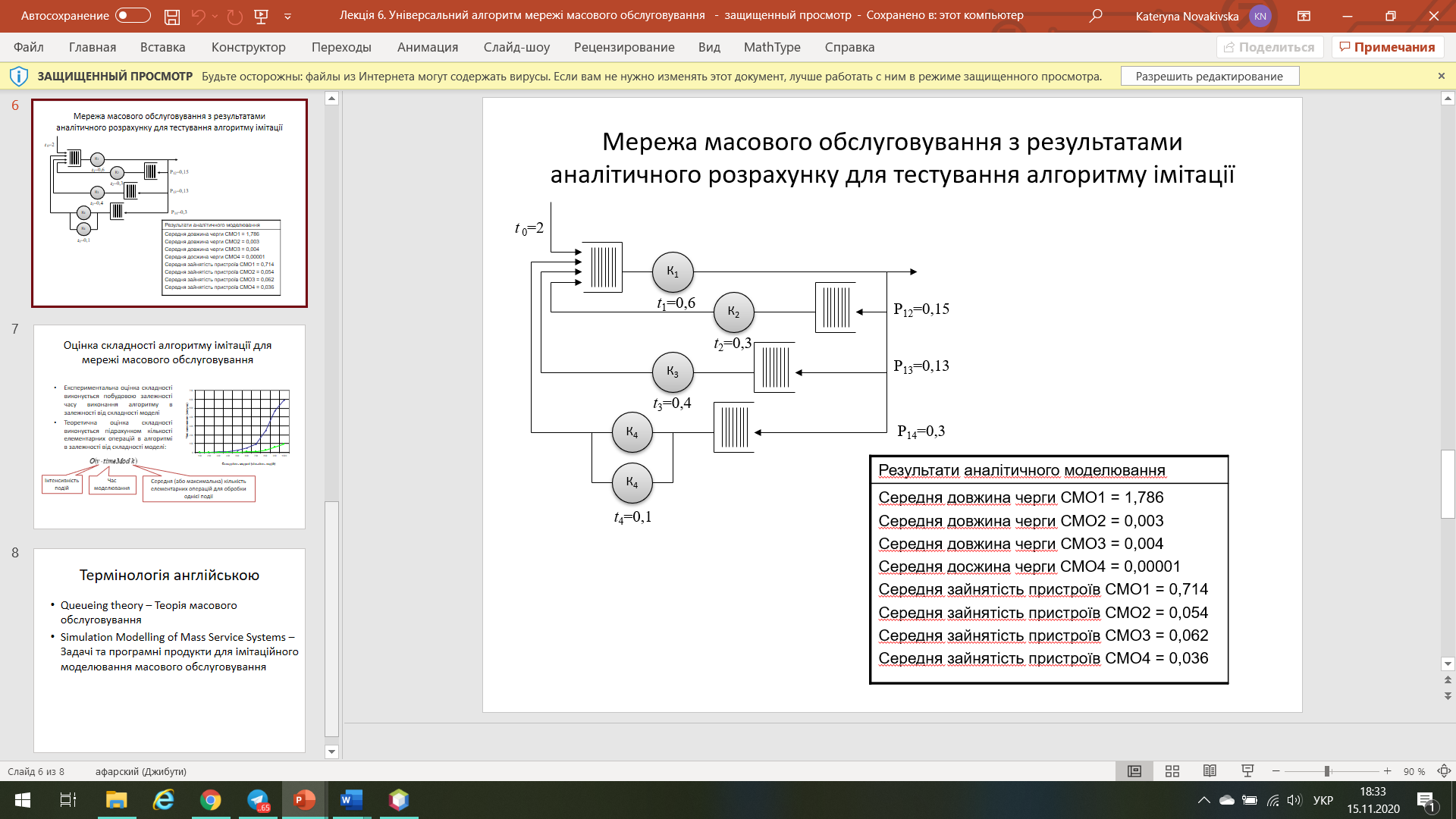
студентка групи ІС-71

Алпаєва Ю.С.

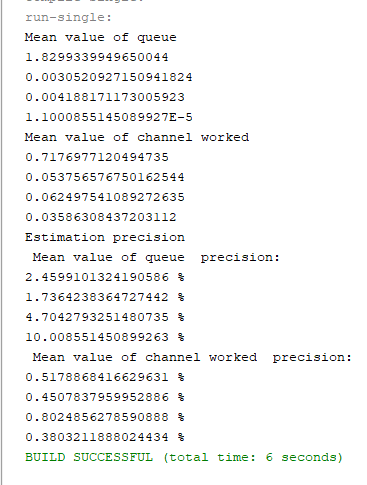
Перевірив:

асистент

ас. Дифучин А. Ю.

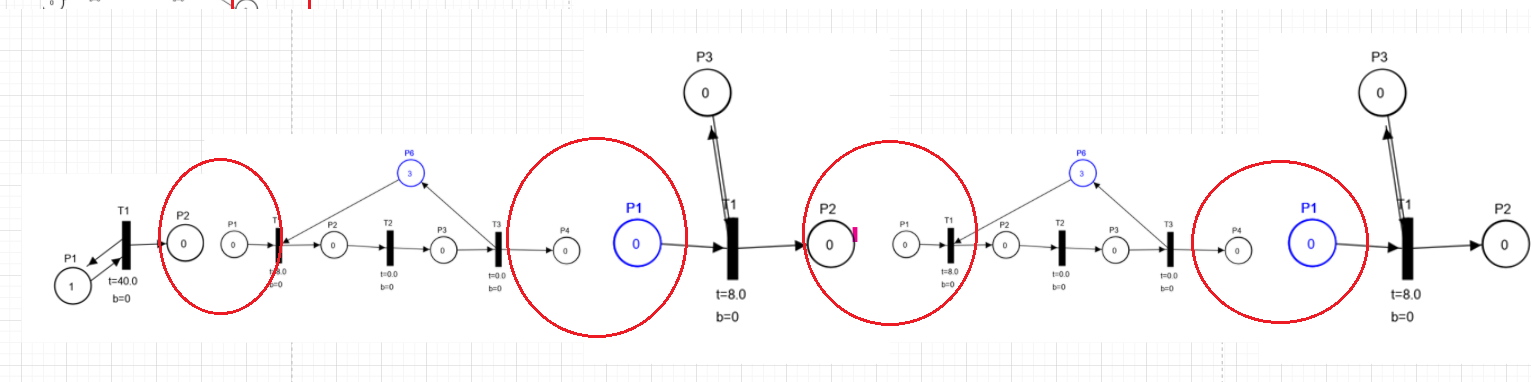


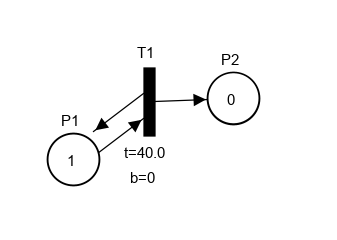
Результат роботи:



Результати статистичних характеристик отримані під час експерименту відхиляються незначно, отже алгоритм працює правильно

Завдання 2





Надходження

public static PetriNet CreateNetNadchodT1() throws ExceptionInvalidNetStructure, ExceptionInvalidTimeDelay {

ArrayList<PetriP> d\_P = new ArrayList<>();

ArrayList<PetriT> d\_T = new ArrayList<>();

ArrayList<ArcIn> d\_In = new ArrayList<>();

ArrayList<ArcOut> d\_Out = new ArrayList<>();

d\_P.add(new PetriP("P1",1));

d\_P.add(new PetriP("P2",0));

d\_T.add(new PetriT("T1",40.0));

d\_In.add(new ArcIn(d\_P.get(0),d\_T.get(0),1));

d\_Out.add(new ArcOut(d\_T.get(0),d\_P.get(0),1));

d\_Out.add(new ArcOut(d\_T.get(0),d\_P.get(1),1));

PetriNet d\_Net = new PetriNet("NadchodT1",d\_P,d\_T,d\_In,d\_Out);

PetriP.initNext();

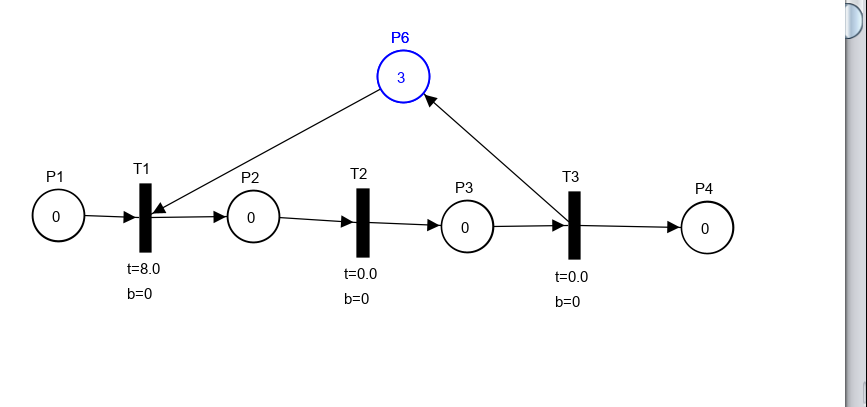
PetriT.initNext();

ArcIn.initNext();

ArcOut.initNext();

return d\_Net;

}



Надходження – захват - переміщення

public static PetriNet CreateNetAct1() throws ExceptionInvalidNetStructure, ExceptionInvalidTimeDelay {

ArrayList<PetriP> d\_P = new ArrayList<>();

ArrayList<PetriT> d\_T = new ArrayList<>();

ArrayList<ArcIn> d\_In = new ArrayList<>();

ArrayList<ArcOut> d\_Out = new ArrayList<>();

d\_P.add(new PetriP("P1",0));

d\_P.add(new PetriP("P2",0));

d\_P.add(new PetriP("P3",0));

d\_P.add(new PetriP("P4",0));

d\_P.add(new PetriP("P6",3));

d\_T.add(new PetriT("T1",8.0));

d\_T.add(new PetriT("T2",0.0));

d\_T.add(new PetriT("T3",0.0));

d\_In.add(new ArcIn(d\_P.get(0),d\_T.get(0),1));

d\_In.add(new ArcIn(d\_P.get(1),d\_T.get(1),1));

d\_In.add(new ArcIn(d\_P.get(2),d\_T.get(2),1));

d\_In.add(new ArcIn(d\_P.get(4),d\_T.get(0),1));

d\_Out.add(new ArcOut(d\_T.get(0),d\_P.get(1),1));

d\_Out.add(new ArcOut(d\_T.get(1),d\_P.get(2),1));

d\_Out.add(new ArcOut(d\_T.get(2),d\_P.get(3),1));

d\_Out.add(new ArcOut(d\_T.get(2),d\_P.get(4),1));

PetriNet d\_Net = new PetriNet("Act1",d\_P,d\_T,d\_In,d\_Out);

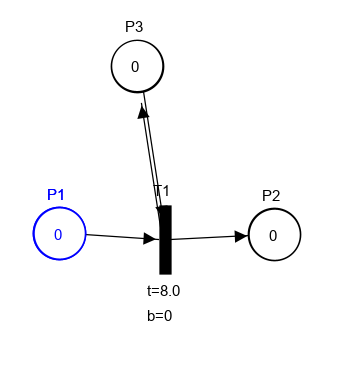
PetriP.initNext();

PetriT.initNext();

ArcIn.initNext();

ArcOut.initNext();

return d\_Net;

}  обробка деталі

public static PetriNet CreateNetProcessing() throws ExceptionInvalidNetStructure, ExceptionInvalidTimeDelay {

ArrayList<PetriP> d\_P = new ArrayList<>();

ArrayList<PetriT> d\_T = new ArrayList<>();

ArrayList<ArcIn> d\_In = new ArrayList<>();

ArrayList<ArcOut> d\_Out = new ArrayList<>();

d\_P.add(new PetriP("P1",0));

d\_P.add(new PetriP("P2",0));

d\_P.add(new PetriP("P3",0));

d\_T.add(new PetriT("T1",8.0));

d\_In.add(new ArcIn(d\_P.get(0),d\_T.get(0),1));

d\_In.add(new ArcIn(d\_P.get(2),d\_T.get(0),1));

d\_Out.add(new ArcOut(d\_T.get(0),d\_P.get(1),1));

d\_Out.add(new ArcOut(d\_T.get(0),d\_P.get(2),1));

PetriNet d\_Net = new PetriNet("Processing",d\_P,d\_T,d\_In,d\_Out);

PetriP.initNext();

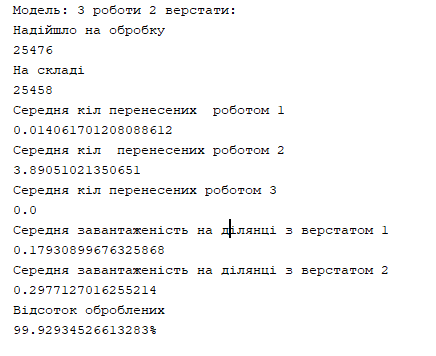
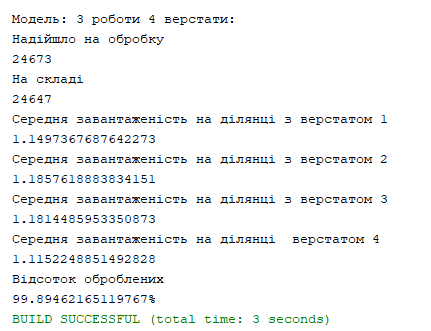
PetriT.initNext();

ArcIn.initNext();

ArcOut.initNext();

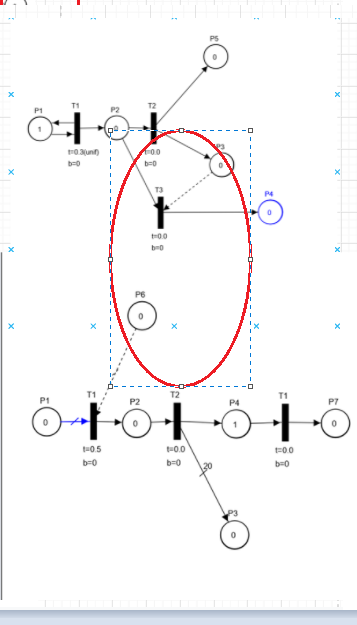
return d\_Net;

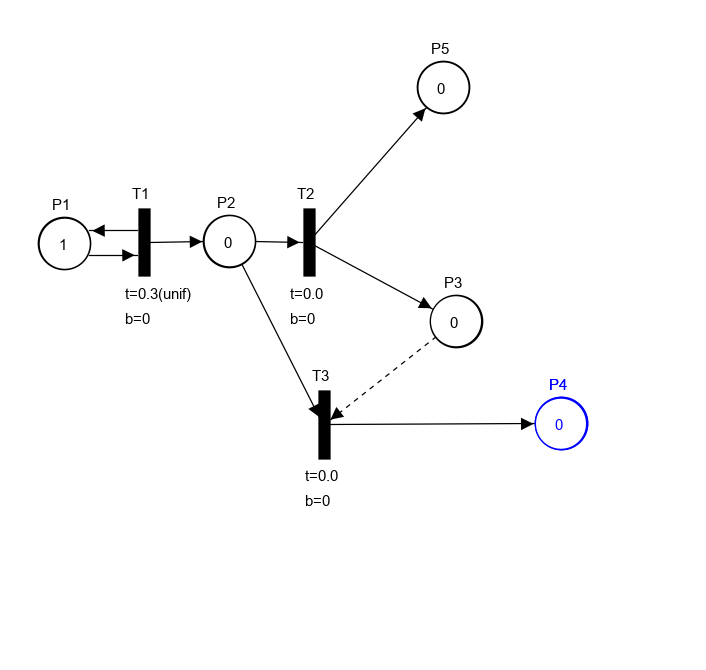
}

Висновок 2: Як бачимо зменшилась середня кількість оброблених деталей, оскільки кількість роботів не збільшилась а кіл конвеєрів збільшилвсь Відсоток оброблених деталей, що надійшли, став меншим через те, що роботів по відношенню до кількості пристроїв у першому випадку було більше, ніж у другому.

Завдання 3





Очікування у черзі-завантаження

public static PetriNet CreateNetIncomming() throws ExceptionInvalidNetStructure, ExceptionInvalidTimeDelay {

ArrayList<PetriP> d\_P = new ArrayList<>();

ArrayList<PetriT> d\_T = new ArrayList<>();

ArrayList<ArcIn> d\_In = new ArrayList<>();

ArrayList<ArcOut> d\_Out = new ArrayList<>();

d\_P.add(new PetriP("P1",1));

d\_P.add(new PetriP("P2",0));

d\_P.add(new PetriP("P3",0));

d\_P.add(new PetriP("P4",0));

d\_P.add(new PetriP("P5",0));

d\_T.add(new PetriT("T1",0.3));

d\_T.get(0).setDistribution("unif", d\_T.get(0).getTimeServ());

d\_T.get(0).setParamDeviation(0.0);

d\_T.add(new PetriT("T2",0.0));

d\_T.add(new PetriT("T3",0.0));

d\_In.add(new ArcIn(d\_P.get(0),d\_T.get(0),1));

d\_In.add(new ArcIn(d\_P.get(1),d\_T.get(1),1));

d\_In.add(new ArcIn(d\_P.get(1),d\_T.get(2),1));

d\_In.add(new ArcIn(d\_P.get(2),d\_T.get(2),1));

d\_In.get(3).setInf(true);

d\_Out.add(new ArcOut(d\_T.get(0),d\_P.get(0),1));

d\_Out.add(new ArcOut(d\_T.get(0),d\_P.get(1),1));

d\_Out.add(new ArcOut(d\_T.get(1),d\_P.get(2),1));

d\_Out.add(new ArcOut(d\_T.get(2),d\_P.get(3),1));

d\_Out.add(new ArcOut(d\_T.get(1),d\_P.get(4),1));

PetriNet d\_Net = new PetriNet("Incomming",d\_P,d\_T,d\_In,d\_Out);

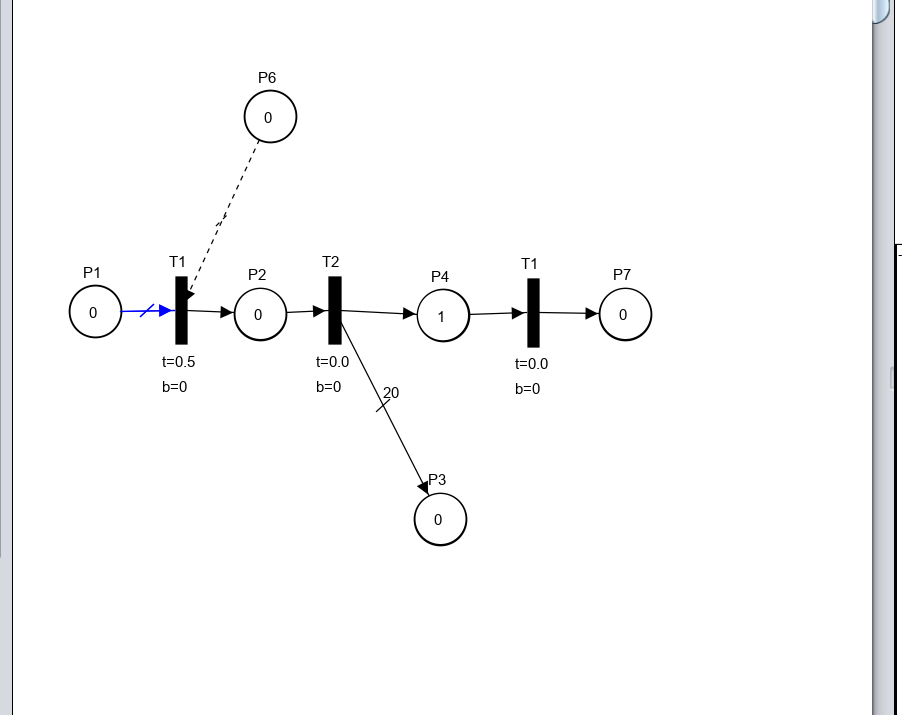
PetriP.initNext();

PetriT.initNext();

ArcIn.initNext();

ArcOut.initNext();

return d\_Net;



Переїзд

public static PetriNet CreateNetTransition(int , int ) throws ExceptionInvalidNetStructure, ExceptionInvalidTimeDelay {

ArrayList<PetriP> d\_P = new ArrayList<>();

ArrayList<PetriT> d\_T = new ArrayList<>();

ArrayList<ArcIn> d\_In = new ArrayList<>();

ArrayList<ArcOut> d\_Out = new ArrayList<>();

d\_P.add(new PetriP("P1",0));

d\_P.add(new PetriP("P2",0));

d\_P.add(new PetriP("P3",0));

d\_P.add(new PetriP("P4",1));

d\_P.add(new PetriP("P6",0));

d\_P.add(new PetriP("P7",0));

d\_T.add(new PetriT("T1",0.5));

d\_T.get(0).setProbability(0.9);

d\_T.add(new PetriT("T2",0.0));

d\_T.add(new PetriT("T1",0.0));

d\_In.add(new ArcIn(d\_P.get(0),d\_T.get(0),));

d\_In.add(new ArcIn(d\_P.get(1),d\_T.get(1),1));

d\_In.add(new ArcIn(d\_P.get(4),d\_T.get(0),));

d\_In.get(2).setInf(true);

d\_In.add(new ArcIn(d\_P.get(3),d\_T.get(2),1));

d\_Out.add(new ArcOut(d\_T.get(0),d\_P.get(1),1));

d\_Out.add(new ArcOut(d\_T.get(1),d\_P.get(2),20));

d\_Out.add(new ArcOut(d\_T.get(1),d\_P.get(3),1));

d\_Out.add(new ArcOut(d\_T.get(2),d\_P.get(5),1));

PetriNet d\_Net = new PetriNet("Transition",d\_P,d\_T,d\_In,d\_Out);

PetriP.initNext();

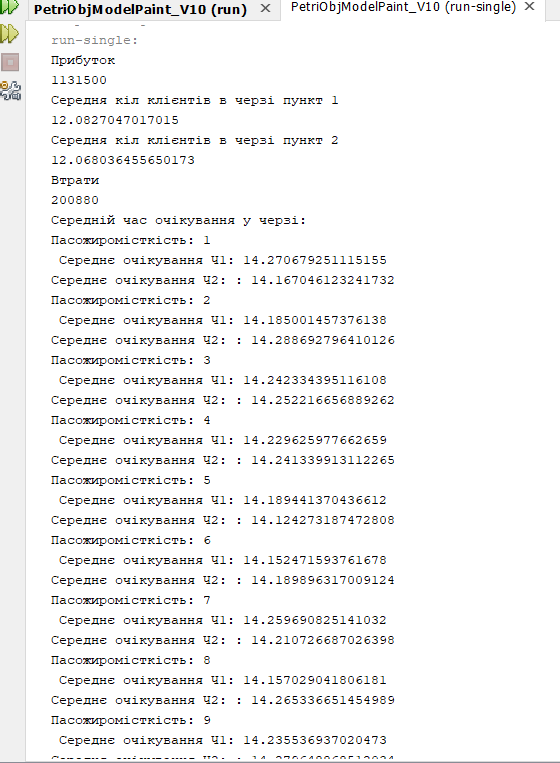
PetriT.initNext();

ArcIn.initNext();

ArcOut.initNext();

return d\_Net;

}

.

Висновок 3: Як бачимо середня довжина черги ,витрати і прибуток майже не змінилися, оскільки ми не змінили частоту надходження автобусів. В двічі зменшився час перебування в черзі, оскільки збільшилась ймовірність пови автобуса.

Переваги та недоліки використання технології Петрі-об’єктного моделювання

|  |  |
| --- | --- |
| + | - |
| декомпозиція | Складність зв’язування об’єктів |
| аналіз окремих ділянок | перенавантаження коду зв’язками |
| багаторазове створення однотипних ділянок | додаткова виділена пам’ять через дублювання маркування в спільних позиціях |

Висновок 4: отже цей спосіб підходить для моделювання складних систем, оскільки можна налагодити кожну частину, і розуміння системи стає легшим.