package sample;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import sample.lab3.generator.Generator;

import java.util.HashMap;

import java.util.Map;

import java.util.stream.Collectors;

public class Controller {

@FXML

public Label label;

@FXML

public Button button;

@FXML

public Label P0000;

@FXML

public Label P0010;

@FXML

public Label P0110;

@FXML

public Label P0001;

@FXML

public Label P0011;

@FXML

public Label P0210;

@FXML

public Label P0111;

@FXML

public Label P1210;

@FXML

public Label P0211;

@FXML

public Label P1211;

@FXML

public TextField p;

@FXML

public TextField p1;

@FXML

public TextField p2;

@FXML

public Label mediumLength;

@FXML

public Label mediumTimeInQueue;

@FXML

public Label relativeCapacity;

@FXML

private void initialize(){

}

@FXML

private void execute(ActionEvent event){

Double p = Double.valueOf(this.p.getText());

Double p1 = 1 - Double.valueOf(this.p1.getText());

Double p2 = 1 - Double.valueOf(this.p2.getText());

Generator generator = new Generator(p,p1,p2);

generator.startEmulation(200000);

SystemInf systemInf = generator.getSystemInf();

Map<String,Double> map = new HashMap<>();

for(Map.Entry<String,Integer> entry:generator.getSystemInf().getStateMap().entrySet()){

map.put(entry.getKey(),( (double)entry.getValue()/200000));

}

P0000.setText(map.get("0000").toString());

P0010.setText(map.get("0010").toString());

P0001.setText(map.get("0001").toString());

P0011.setText(map.get("0011").toString());

P0210.setText(map.get("0210").toString());

P0111.setText(map.get("0111").toString());

P1210.setText(map.get("1210").toString());

P0211.setText(map.get("0211").toString());

P1211.setText(map.get("1211").toString());

P0110.setText(map.get("0110").toString());

mediumLength.setText(String.valueOf(systemInf.getLengthOfQuery().stream().mapToInt(a -> a.intValue()).average().getAsDouble()));

mediumTimeInQueue.setText(String.valueOf(systemInf.getTimeInQuery().stream().mapToInt(a -> a).average().getAsDouble()));

double relative = (double)systemInf.getCountWeddRequests()/(systemInf.getCountExecutedRequests()+systemInf.getCountWeddRequests());

relative = 1- relative;

Integer a = systemInf.getCountWeddRequests()+systemInf.getCountExecutedRequests();

relativeCapacity.setText(String.valueOf(relative));

}

}

package sample;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.stage.Stage;

public class Main extends Application {

@Override

public void start(Stage primaryStage) throws Exception{

Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));

primaryStage.setTitle("Hello World");

primaryStage.setScene(new Scene(root, 600, 300));

primaryStage.show();

}

public static void main(String[] args) {

launch(args);

}

}

package sample.lab3.state;

import sample.lab3.entity.Request;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import java.util.Queue;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public interface BaseState {

void calcNextState(Queue<Request> requests, State state, SystemInf systemInf);

}

package sample.lab3.state;

import sample.lab3.entity.Request;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import java.util.Queue;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class NRequestFirstNSecond implements BaseState {

@Override

public void calcNextState(Queue<Request> requests, State state, SystemInf systemInf) {

if(state.getSecondHandler()== 0){

state.setSecondHandler(state.getFirstHandler());

state.setFirstHandler((byte)0);

}else {

systemInf.addCountWeddRequests();

state.setFirstHandler((byte)0);

}

if(state.getStateQuery()>0){

state.setFirstHandler((byte)1);

state.deleteFromQuery();

systemInf.addTimeInQuery(requests.poll().getTime()+1);

}

requests.forEach(Request::addTact);

if(state.getStateSource()==1){

state.setStateSource((byte)0);

state.addToQuery();

requests.add(new Request());

}

systemInf.addlengthOfQuery(state.getStateQuery());

systemInf.addState(state.toString());

}

}

package sample.lab3.state;

import sample.lab3.entity.Request;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import java.util.Queue;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class NRequestFirstSecond implements BaseState {

@Override

public void calcNextState(Queue<Request> requests, State state, SystemInf systemInf) {

systemInf.addcountExecutedRequests();

state.setFirstHandler((byte)0);

if(state.getStateQuery()>0){

state.setFirstHandler((byte)1);

systemInf.addTimeInQuery(requests.poll().getTime()+1);

state.deleteFromQuery();

}

requests.forEach(Request::addTact);

if(state.getStateSource()==1){

state.setStateSource((byte)0);

state.addToQuery();

requests.add(new Request());

}

systemInf.addState(state.toString());

systemInf.addlengthOfQuery(state.getStateQuery());

}

}

package sample.lab3.state;

import sample.lab3.entity.Request;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import java.util.Queue;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class NRequestNFirstNSecond implements BaseState {

@Override

public void calcNextState(Queue<Request> requests, State state, SystemInf systemInf) {

systemInf.addState(state.toString());

systemInf.addlengthOfQuery(state.getStateQuery());

requests.forEach(Request::addTact);

}

}

package sample.lab3.state;

import sample.lab3.entity.Request;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import java.util.Queue;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class NRequestNFirstSecond implements BaseState {

@Override

public void calcNextState(Queue<Request> requests, State state, SystemInf systemInf) {

state.setSecondHandler((byte)0);

systemInf.addState(state.toString());

systemInf.addcountExecutedRequests();

systemInf.addlengthOfQuery(state.getStateQuery());

requests.forEach(Request::addTact);

}

}

package sample.lab3.state;

import sample.lab3.entity.Request;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import java.util.Queue;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class RequestFirstNSecond implements BaseState {

@Override

public void calcNextState(Queue<Request> requests, State state, SystemInf systemInf) {

if(state.getSecondHandler()==0){

state.setSecondHandler((byte)1);

}else {

systemInf.addCountWeddRequests();

}

if(state.getStateQuery()>0){

state.setFirstHandler((byte)1);

systemInf.addTimeInQuery(requests.poll().getTime()+1);

requests.forEach(Request::addTact);

requests.add(new Request());

}else {

state.setFirstHandler((byte)1);

systemInf.addTimeInQuery(0);

}

systemInf.addState(state.toString());

systemInf.addlengthOfQuery(state.getStateQuery());

}

}

package sample.lab3.state;

import sample.lab3.entity.Request;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import java.util.Queue;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class RequestFirstSecond implements BaseState {

@Override

public void calcNextState(Queue<Request> requests, State state, SystemInf systemInf) {

systemInf.addcountExecutedRequests();

state.setFirstHandler((byte)0);

if(state.getStateQuery()>0){

state.setFirstHandler((byte)1);

systemInf.addTimeInQuery(requests.poll().getTime()+1);

requests.forEach(Request::addTact);

requests.add(new Request());

}else {

state.setFirstHandler((byte)1);

systemInf.addTimeInQuery(0);

}

systemInf.addlengthOfQuery(state.getStateQuery());

systemInf.addState(state.toString());

}

}

package sample.lab3.state;

import sample.lab3.entity.Request;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import java.util.Queue;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class RequestNFirstNSecond implements BaseState {

@Override

public void calcNextState(Queue<Request> requests, State state, SystemInf systemInf) {

requests.forEach(Request::addTact);

if(state.getStateQuery()==2){

state.setStateSource((byte)1);

}else if(state.getFirstHandler()==0){

state.setFirstHandler((byte)1);

systemInf.addTimeInQuery(0);

}else {

state.addToQuery();

requests.add(new Request());

}

systemInf.addlengthOfQuery(state.getStateQuery());

systemInf.addState(state.toString());

}

}

package sample.lab3.state;

import sample.lab3.entity.Request;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import java.util.Queue;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class RequestNFirstSecond implements BaseState {

@Override

public void calcNextState(Queue<Request> requests, State state, SystemInf systemInf) {

systemInf.addcountExecutedRequests();

state.setSecondHandler((byte) 0);

if(state.getFirstHandler() == 0){

if(state.getStateQuery()>0){

state.setFirstHandler((byte)1);

systemInf.addTimeInQuery(requests.poll().getTime()+1);

requests.forEach(Request::addTact);

requests.add(new Request());

}else {

state.setFirstHandler((byte)1);

systemInf.addTimeInQuery(0);

}

}else {

if (state.getStateQuery() == 2) {

state.setStateSource((byte) 1);

requests.forEach(Request::addTact);

}else {

requests.forEach(Request::addTact);

state.addToQuery();

requests.add(new Request());

}

}

systemInf.addlengthOfQuery(state.getStateQuery());

systemInf.addState(state.toString());

}

}

package sample.lab3.generator;

import sample.lab3.entity.Request;

import sample.lab3.entity.State;

import sample.lab3.entity.SystemInf;

import sample.lab3.logger.CustomLogger;

import sample.lab3.state.BaseState;

import sample.lab3.state.\*;

import java.util.\*;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class Generator {

private Map<String, BaseState> stateMap = new HashMap<>();

private State state = new State();

private SystemInf systemInf = new SystemInf();

private Queue<Request> requests = new LinkedList<>();

private Double pi;

private Double p1;

private Double p2;

private int countSource = 0;

private Random random = new Random();

public Generator(Double pi, Double p1, Double p2) {

this.p1 = p1;

this.p2 = p2;

this.pi = pi;

stateMap.put("000", new NRequestNFirstNSecond());

stateMap.put("100", new RequestNFirstNSecond());

stateMap.put("010", new NRequestFirstNSecond());

stateMap.put("110", new RequestFirstNSecond());

stateMap.put("001", new NRequestNFirstSecond());

//stateMap.put("110", new RequestFirstNSecond());

stateMap.put("101", new RequestNFirstSecond());

stateMap.put("011", new NRequestFirstSecond());

stateMap.put("111", new RequestFirstSecond());

}

private String generateRandomInput() {

StringBuffer result = new StringBuffer();

double randNumber;

if (state.getStateSource() == 1) {

result.append(0);

} else {

randNumber = random.nextDouble();

if (Double.compare(randNumber, pi) <= 0) {

result.append(1);

countSource++;

} else {

result.append(0);

}

}

if (state.getFirstHandler() == 0) {

result.append(0);

} else {

randNumber = random.nextDouble();

if (Double.compare(randNumber, p1) <= 0) {

result.append(1);

} else {

result.append(0);

}

}

if (state.getSecondHandler() == 0) {

result.append(0);

} else {

randNumber = random.nextDouble();

if (Double.compare(randNumber, p1) <= 0) {

result.append(1);

} else {

result.append(0);

}

}

return result.toString();

}

public void startEmulation(int countTacts){

CustomLogger customLogger = new CustomLogger();

for(int i =0; i< countTacts; i++){

String input = generateRandomInput();

BaseState baseState = stateMap.get(input);

baseState.calcNextState(requests,state,systemInf);

customLogger.add(input,state.toString());

}

customLogger.writeToFile();

}

public State getState() {

return state;

}

public SystemInf getSystemInf(){

return systemInf;

}

}

package sample.lab3.entity;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class Request {

private Integer time = 0;

public void addTact(){

time++;

}

public Integer getTime() {

return time;

}

}

package sample.lab3.entity;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class State {

private Byte stateSource;

private Byte stateQuery;

private Byte firstHandler;

private Byte secondHandler;

public State(){

this.stateSource = 0;

this.stateQuery = 0;

this.firstHandler = 0;

this.secondHandler = 0;

}

public Byte getStateSource() {

return stateSource;

}

public void setStateSource(Byte stateSource) {

this.stateSource = stateSource;

}

public Byte getStateQuery() {

return stateQuery;

}

public void setStateQuery(Byte stateQuery) {

this.stateQuery = stateQuery;

}

public Byte getFirstHandler() {

return firstHandler;

}

public void setFirstHandler(Byte firstHandler) {

this.firstHandler = firstHandler;

}

public Byte getSecondHandler() {

return secondHandler;

}

public void setSecondHandler(Byte secondHandler) {

this.secondHandler = secondHandler;

}

@Override

public String toString() {

return stateSource.toString() + stateQuery.toString() + firstHandler.toString() + secondHandler.toString();

}

public void addToQuery(){

stateQuery++;

}

public void deleteFromQuery(){

stateQuery--;

}

}

package sample.lab3.entity;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

/\*\*

\* Created by Иван on 15.11.2016.

\*/

public class SystemInf {

private Integer countExecutedRequests = 0;

private Integer countWeddRequests = 0;

private List<Byte> lengthOfQuery = new ArrayList<>();

private Map<String,Integer> stateMap = new HashMap<>();

private List<Integer> timeInQuery = new ArrayList<>();

public SystemInf(){

stateMap.put("0000",0);

stateMap.put("0010",0);

stateMap.put("0110",0);

stateMap.put("0001",0);

stateMap.put("0011",0);

stateMap.put("0210",0);

stateMap.put("0111",0);

stateMap.put("1210",0);

stateMap.put("0211",0);

stateMap.put("1211",0);

}

public void addcountExecutedRequests(){

countExecutedRequests++;

}

public void addCountWeddRequests(){

countWeddRequests++;

}

public void addlengthOfQuery(Byte length){

lengthOfQuery.add(length);

}

public void addState(String state){

stateMap.put(state,stateMap.get(state)+1);

}

public void addTimeInQuery(Integer tact){

timeInQuery.add(tact);

}

public Integer getCountExecutedRequests() {

return countExecutedRequests;

}

public Integer getCountWeddRequests() {

return countWeddRequests;

}

public List<Byte> getLengthOfQuery() {

return lengthOfQuery;

}

public Map<String, Integer> getStateMap() {

return stateMap;

}

public List<Integer> getTimeInQuery() {return timeInQuery}}