Листинг программы:

**package** lab3;

**public** **class** SystemTest {

**private** **static** **int** *modelingTime* = 10;

**public** **static** **void** main(String[] args) {

**int** requestCompleteIntensity = 4;

**int** comeRequestsIntensity = 4;

Queue queue;

Processor processor;

**for** (**int** k = 0; k < 1; k++) {

comeRequestsIntensity += k;

queue = **new** Queue(requestCompleteIntensity, *modelingTime*);

processor = **new** Processor(comeRequestsIntensity, *modelingTime*,

queue);

// run system

**for** (**int** i = 0; i < *modelingTime*; i++) {

**if** (i >= processor.getCurNextReqTime()) {

processor.runNextRequest(i);

}

}

}

}

}

**package** lab3;

**public** **class** Processor {

**private** **int**[] comeRequestsTime;

**private** Queue queue;

**private** **int** curNextReqTime;

**private** **int** comeIntensity;

**private** **int** requestNumber = 0;

**public** Processor(**int** comeIntensity, **int** maxNumbOfrequests, Queue queue) {

**this**.comeRequestsTime = **new** **int**[maxNumbOfrequests];

**this**.queue = queue;

**this**.curNextReqTime = 0;

**this**.comeIntensity = comeIntensity;

generateComeRequestTime();

// printComeRequestTime();

queue.setAbsoluteComeTimes(comeRequestsTime);

queue.printRequests(comeRequestsTime);

}

**private** **void** generateComeRequestTime() {

**for** (**int** i = 0; i < comeRequestsTime.length; i++) {

comeRequestsTime[i] = (**int**) ((-10 / comeIntensity) \* Math.*log*(Math

.*random*()));

}

curNextReqTime = comeRequestsTime[0];

}

**public** **void** runNextRequest(**int** curTime) {

requestNumber = queue.getNeedRequest(curTime);

System.*out*.println("rn = " + requestNumber);

**if** (requestNumber > 0) {

curNextReqTime += queue.getRequestRunTime(requestNumber);

} **else** {

curNextReqTime++;

}

queue.deleteCompletedRequest(requestNumber);

}

**public** **int** getCurNextReqTime() {

**return** curNextReqTime;

}

**public** **void** printComeRequestTime() {

**for** (**int** i = 0; i < comeRequestsTime.length; i++) {

System.*out*.println("crt " + comeRequestsTime[i]);

}

}

}

**package** lab3;

**import** java.util.ArrayList;

**public** **class** Queue {

**private** ArrayList<Request> requests = **new** ArrayList<Request>();

**private** **int** intensity;

**private** **final** **int** maxLengthOfQueue;

**private** **int** numbOfRequests;

**public** Queue(**int** intensity, **int** maxNumbOfRequests) {

**this**.intensity = intensity;

**this**.maxLengthOfQueue = maxNumbOfRequests;

generateRequests();

}

**private** **void** generateRequests() {

**int** requestRunTime;

**for** (**int** i = 0; i < maxLengthOfQueue; i++) {

requestRunTime = (**int**) ((-10 / intensity) \* Math.*log*(Math.*random*()));

Request request = **new** Request(requestRunTime);

requests.add(request);

}

}

**public** **void** removeRequest(**int** systemTime) {

}

**public** **void** incNumbOfRequests() {

numbOfRequests++;

}

**public** **void** decNumbOfRequests() {

**if** (numbOfRequests > 0) {

numbOfRequests--;

}

}

**public** **void** setAbsoluteComeTimes(**int**[] comeTimes) {

**int** absoluteTime = 0;

**for** (**int** i = 0; i < requests.size(); i++) {

absoluteTime += comeTimes[i];

requests.get(i).setAbsoluteComeTime(absoluteTime);

}

}

**public** **int** getNeedRequest(**int** currentTime){

**for** (**int** i=0; i<requests.size(); i++){

**if** (requests.get(i).getAbsoluteTime()<=currentTime){

**return** i;

}

}

**return** (requests.size()-1);

}

**public** **int** getRequestRunTime(**int** requestNumb){

**return** requests.get(requestNumb).getCompleteTime();

}

**public** **void** deleteCompletedRequest(**int** requestNumb){

requests.remove(requestNumb);

}

**public** **void** printRequests(**int**[] ct) {

**int** i = 0;

**for** (Request request : requests) {

System.*out*.println("\* " + request.getAbsoluteTime() + " " + ct[i]+" "+request.getCompleteTime());

i++;

}

}

}

**package** lab3;

**public** **class** Request {

**private** **int** requestTime;

**private** **int** absoluteComeTime;

**public** Request(**int** requestTime){

**this**.requestTime = requestTime;

}

**public** **int** getCompleteTime(){

**return** **this**.requestTime;

}

**public** **void** setAbsoluteComeTime(**int** absoluteComeTime){

**this**.absoluteComeTime = absoluteComeTime;

}

**public** **int** getAbsoluteTime(){

**return** **this**.absoluteComeTime;

}

}