Лабораторная работа №10.1. Моделирование нестационарного пуассоновского потока

Задание: обеспечить определение на числовом отрезке [0,100] случайное положение точек – моментов событий нестационарного пуассоновского потока с заданной функцией, представляющей собой интенсивность потока. Дать графическую иллюстрацию. Варианты заданий приведены в таблице.

Листинг 1. Пример программной реализации

```
package com.company;
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.scene.Scene;
import javafx.scene.chart.Axis;
import javafx.scene.chart.LineChart;
import javafx.scene.chart.NumberAxis;
import javafx.scene.chart.XYChart;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
public class Main extends Application {
 @Override
 public void start(Stage primaryStage) {
 Pane root = new Pane();
 // Массив массивов
 ObservableList<XYChart.Series> seriesList =
FXCollections.observableArrayList();
 // Массив
 ObservableList<XYChart.Data> aList =
FXCollections.observableArrayList();
 List<Double> lambda = new ArrayList<>();
 int Tn=100;
 int t=0;
 int N = 0;
 while (t<=Tn) {
 double r = 0+Math.random()*1.0;;
 double temp = 1.1-Math.pow(t-50,2)/2500;
 lambda.add(temp);
 double ro = -1/temp*Math.log(r);
 t+=ro;
 N++;
 aList.add(new XYChart.Data(t,N));
 }
```

```
// Create axes
double fxMax = Collections.max(lambda);
double fxMin = Collections.min(lambda);
// Create axes
Axis yAxis = new NumberAxis("Count", 0, N, N/10);
Axis xAxis = new NumberAxis("t", 0, Tn, 10);
seriesList.add(new XYChart.Series("Graphic", aList));
LineChart chart = new LineChart(xAxis, yAxis, seriesList);
chart.setPrefHeight(768);
chart.setMinHeight(768);
chart.setMaxHeight(768);
chart.setPrefWidth(1024);
chart.setMinWidth(1024);
chart.setMaxWidth(1024);
chart.setPrefSize(1024, 768);
chart.setMinSize(1024, 768);
chart.setMaxSize(1024, 768);
root.getChildren().add(chart);
Scene scene = new Scene(root);
primaryStage.setScene(scene);
primaryStage.setTitle("Моделирование потока случайных событий");
primaryStage.show();
}
public static void main(String[] args) {
launch(args);
}
}
```

№ варианта	Интенсивность потока	№ варианта	Интенсивность потока
1	$\lambda(t)=\begin{cases}0{,}1, \textit{если } t\in[0,40]\\ 0{,}5, \textit{если } t\in(40,60)\\ 0{,}3, \textit{если } t\in[60,100]\end{cases}$	11	$\lambda(t)=0{,}1-\frac{(t-50)^2}{4000}$
2	$\lambda(t)=1-0{,}02\lvert t-50\rvert$	12	$\lambda(t)=0{,}2+0{,}01\lvert t-60\rvert$
3	$\lambda(t)=1{,}1-\frac{(t-50)^2}{2500}$	13	$\lambda(t)=\frac{(t-100)^2}{10000}$
4	$\lambda(t)=\begin{cases}0{,}02t, \textit{если } t\in[0,50[\\ 0{,}5, \textit{если } t\in[50,100]\end{cases}$	14	$\lambda(t)=0{,}8-0{,}004t$
5	$\lambda(t)=\begin{cases}0{,}6, \textit{если } t\in[0,30]\\ 0{,}1, \textit{если } t\in(30,70)\\ 0{,}5, \textit{если } t\in[70,100]\end{cases}$	15	$\lambda(t)=\begin{cases}0{,}3t, \textit{если } t\in[0,50]\\ 0{,}01(t-50), \textit{если } t\in]50,100]\end{cases}$
6	$\lambda(t)=0{,}1-0{,}004t$	16	$\lambda(t)=0{,}1\sqrt{t}$
7	$\lambda(t)=1-\frac{(t-50)^2}{3000}$	17	$\lambda(t)=0{,}15\sqrt[4]{t+50}$

8	$\lambda(t)=1-0{,}01\lvert t-70\rvert$	18	$\lambda(t)=0{,}8\cos\dfrac{t}{150}$
9	$\lambda(t)=0{,}2-0{,}003t$	19	$\lambda(t)=1-e^{-0{,}1t}$
10	$\lambda(t)=0{,}015\lvert t-50\rvert$	20	$\lambda(t)=\begin{cases}0{,}3, \text{если } t\in[0{,}30]\\ 0{,}2, \text{если } t\in(30{,}70)\\ 0{,}4, \text{если } t\in[70{,}100]\end{cases}$

Лабораторная работа № 10.2. Моделирование одноканальной системы массового обслуживания с отказами

Пример программной реализации. Одноканальная СМО без очереди с отказами

Листинг 2

```
#!/usr/bin/env python3

# Поток простейший. Очередь отсутствует.
data = {"lambda": 0.95, "tsred": 1, "mu": 1, "m": 0 }

def setRo(data):
    data["ro"] = data["lambda"] / data["mu"]

def setP0(data):
    ro = data["ro"]
    if ro == 1:
        result = (1 - ro) / (1 - ro ** (data["m"] + 2))
    else:
        result = 1 / (data["m"] + 2)
    data["p0"] = result

def getP(data, k):
    return k * data["p0"]

def setP(data, k):
    name = "p" + str(k)
    data[name] = getP(data, k)

def setPotkaz(data):
    data["potkaz"] = getP(data, 1)

def setQ(data):
    data["q"] = 1 - data["potkaz"]

def setPsystem(data):
    data["psystem"] = 1 - data["potkaz"]
```

```
def setA(data):
    data["A"] = data["lambda"] * data["q"]

def setNochered(data):
    m = data["m"]
    ro = data["ro"]
    p0 = data["p0"]
    if m == 1:
        chislitel =  ro * ro * (1 - (ro ** m) * (m + 1 - m * ro))
        znamenatel = (1 - ro ** (m - 2)) * (1 - ro)
        result = p0 * chislitel / znamenatel
    else:
        result = m * (m + 1) / (2 * m + 4)
    data["nochered"] = result

def setNobsl(data):
    data["nobsl"] = data["ro"] * data["q"]

def setNsystem(data):
    data["nsystem"] = data["nochered"] + data["nobsl"]

def setTochered(data):
    data["tochered"] = data["nochered"] / (data["lambda"] * data["psystem"])

def setTsystem(data):
    data["tsystem"] = data["nsystem"] / (data["lambda"] * data["psystem"])

def setAllParams(data):
    setRo(data)
    setP0(data)
    setPotkaz(data)
    setQ(data)
    setPsystem(data)
    setA(data)
    setNochered(data)
    setNobsl(data)
    setNsystem(data)
    setTochered(data)
    setTsystem(data)

def printParam(name, value):
    print("{} \t: {}".format(name, value))

def printAllParams(data):
    printParam("Коэффициент использования объекта        ", data["ro"])
    printParam("Вероятность того, что линия свободна     ", data["p0"])
    printParam("Вероятность отказа в обслуживании        ", data["potkaz"])
```

```
    printParam("Вероятность принятия заявки в систему    ", data["q"])
    printParam("Относительная пропускная способность     ", data["psystem"])
    printParam("Абсолютная пропусная способность         ", data["A"])
    printParam("Среднее число заявок в очереди           ", data["nochered"])
    printParam("Среднее число заявок в обслуживании      ", data["nobsl"])
    printParam("Среднее число заявок в СМО               ", data["nsystem"])
    printParam("Среднее время ожидания заявки в очереди  ", data["tochered"])
    printParam("Среднее время пребывания заявки в системе", data["tsystem"])

# Исполнение
setAllParams(data)
printAllParams(data)
```

Листинг 3

```
#!/usr/bin/env python3

import random
from statistics import mean

# Интенсивность поступления звонков в минуту
prob = 0.95
# Среднее время обслуживания в минутах
handle_time = 1
# Величина интервала моделирования в минутах
minutes_for_model = 60 * 24 * 365  # год

# Массив со временами звонков в минутах отсчитывамых с нуля
rings = []
last_ring_time = 0.0
for minute in range(0, minutes_for_model - 1):
    rnd = random.expovariate(prob)
    if(rnd <= 0.0):
        print("rnd = ", str(rnd), " < 0")
    last_ring_time += rnd
    rings.append(last_ring_time)

rings.sort()  # Выставим звонки в порядке возрастания

handles = []
lambd = 1.0 / handle_time  # Интенсивность потока для времени разговора
for ring in rings:
    rnd = random.expovariate(lambd)
    handles.append(rnd)

reject_count = 0
for ring in range(0, len(rings) - 2):
```

```
    handle_end = rings[ring] + handles[ring]  # Время окончания обслуживания
    # Если заявка поступила во время обслуживания предыдущей
    if rings[ring + 1] < handle_end:
        reject_count += 1  # то ей отказано в обслуживании

reject_prob = reject_count / len(rings)
print("Вероятность отказа в обслуживании = " + str(reject_prob))

all_time_of_work = (rings[-1] + handles[-1])

work_percent = sum(handles) / all_time_of_work
print("Нагрузка устройства обслуживания = " + str(work_percent))

# Среднее время пребывания заявки в системе равно времени обслуживания
# (заявки начинают обслуживаться немедленно,
#  заявки недообслуживаются и очередь отсутствует)
avg_time = mean(handles)
print("Среднее время пребывания заявки в системе = {} минут".format(avg_time))

print("==================================")
print("Минимальное время обслуживания в минутах " + str(min(handles)))
print("Максимальное время обслуживания в минутах " + str(max(handles)))
```

Задание: обеспечить определение на числовом отрезке [0,100] случайное положение точек – моментов изменения состояния одноканальной СМО с отказами, для которой известны характеристики: $\lambda(t)$ – интенсивность потока заявок, $\mu(t)$ – интенсивность обслуживания. Должны подводиться итоги: количество обслуженных заявок, количество отказов. Варианты вида функции $\lambda(t)$ заимствуются из лабораторной № 10.1. Варианты задания функции интенсивности обслуживания приведены в таблице.

№ варианта	Интенсивность обслуживания	№ варианта	Интенсивность обслуживания
1	$\mu(t) = 0{,}015\lvert t-50\rvert$	11	$\mu(t) = \begin{cases} 0{,}8t, \textit{если } t \in [0{,}30[\\ 1{,}02, \textit{если } t \in [30{,}100] \end{cases}$
2	$\mu(t) = 0{,}02\sqrt{0{,}1+(t-50)^2}$	12	$\mu(t) = \begin{cases} 0{,}7t, \textit{если } t \in [0{,}30[\\ 0{,}95, \textit{если } t \in [30{,}100] \end{cases}$
3	$\mu(t) = \begin{cases} 0{,}7t, \textit{если } t \in [0{,}40] \\ 0{,}9, \textit{если } t \in]40{,}100] \end{cases}$	13	$\mu(t) = 0{,}2-0{,}01\lvert t-50\rvert$
4	$\mu(t) = \dfrac{(t-100)^2}{10000}$	14	$\mu(t) = 0{,}009(100-t)$
5	$\mu(t) = \begin{cases} 0{,}8t, \textit{если } t \in [0{,}50] \\ 0{,}01, \textit{если } t \in]50{,}100] \end{cases}$	15	$\mu(t) = \begin{cases} 1{,}05t, \textit{если } t \in [0{,}20[\\ 0{,}8, \textit{если } t \in [20{,}100] \end{cases}$

6	$\mu(t)=\begin{cases}0{,}6, \textit{если } t\in[0{,}40[\\0{,}8, \textit{если } t\in[40{,}60]\\0{,}7, \textit{если } t\in]60{,}100]\end{cases}$	16	$\mu(t)=0{,}1-0{,}01\lvert t-70\rvert$
7	$\mu(t)=1-0{,}017\lvert t-50\rvert$	17	$\mu(t)=\begin{cases}0{,}4t, \textit{если } t\in[0{,}30[\\1{,}02, \textit{если } t\in[30{,}100]\end{cases}$
8	$\mu(t)=0{,}1-0{,}012\lvert t-50\rvert$	18	$\mu(t)=\begin{cases}0{,}5t, \textit{если } t\in[0{,}20[\\0{,}95, \textit{если } t\in[20{,}100]\end{cases}$
9	$\mu(t)=\begin{cases}0{,}2, \textit{если } t\in[0{,}20[\\0{,}9, \textit{если } t\in[20{,}80]\\0{,}4, \textit{если } t\in]80{,}100]\end{cases}$	19	$\mu(t)=\begin{cases}1, \textit{если } t\in[0{,}30[\\0{,}7, \textit{если } t\in[30{,}80]\\0{,}9, \textit{если } t\in]80{,}100]\end{cases}$
10	$\mu(t)=1{,}05-0{,}018\lvert t-50\rvert$	20	$\mu(t)=0{,}15-0{,}01\lvert t-60\rvert$

Лабораторная работа № 10.3 Моделирование одноканальной системы массового обслуживания с отказами (очная форма обучения)

Задание: обеспечить определение на числовом отрезке [0,100] случайное положение точек – моментов изменения состояния одноканальной СМО с ожиданием, для которой известны характеристики: $\lambda(t)$ – интенсивность потока заявок, $\mu(t)$ – интенсивность обслуживания. На моменты изменения состояния системы должно определяться количество заявок в накопителе (Блок 2 «уметь, блок 3 «владеть»). Варианты вида функции $\lambda(t)$ и $\mu(t)$ заимствуются из лабораторной № 10.1 и № 10.2.