**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

**Кафедра компьютерных технологий и систем**

Отчет по лабораторным работам по курсу

“Имитационное и статистическое моделирование”

Вариант 4

**Выполнил**

Дунаев Виктор

4 курс 6 группа

**Преподаватель**

Дернакова О.В.

*Минск 2018*

*Лабораторная работа №1*

**Постановка задачи.**

Используя метод Маклерена-Марсальи построить датчик БСВ (1 датчик должен быть мультипликативно конгруентный, второй – на выбор). Исследовать точность построенной БСВ.

1. Осуществить моделирование *n* = 1000 реализаций БСВ с помощью мультипликативного конгруэнтного метода (МКМ) с параметрами *a*0, β, *M* = 231 .
2. Осуществить моделирование *n* = 1000 реализаций БСВ с помощью метода Макларена-Марсальи (один датчик должен быть мультипликативно конгруентный (п. 1), второй – на выбор).*K* – объем вспомогательной таблицы.
3. Проверить точность моделирования обоих датчиков (п. 1 и п. 2) с помощью критерия согласия Колмогорова и χ2-критерия Пирсона с уровнем значимости ε = 0.05.

С заданными параметрами a1 = 296454621; c1 = 48840859;

a2 = 302711857; c2 = 37330745;

K = 64.

**Теория.**

**1)Мультипликативный конгруэнтный метод.**

Согласно этому методу псевдослучайная последовательность реализаций  БСВ определяется по рекуррентным формулам :

****

где -- параметры датчика (натуральные числа) : -- множитель (<M), M – модуль, -- стартовое значение (нечётное число).

Операция  означает вычет числа по модулю М :



где [] – целая часть числа .

Период датчика Т; коэффициент использования БСВ =1. значения параметров  определяются из условия максимума периода Т. значение М зависит от способа представления целых чисел в ЭВМ.

**2) Метод Макларена-Марсальи.**

Метод основан на комбинировании двух простейших программных датчиков БСВ. Пусть {*bi*}, {*ci*} (*i* = 0, 1, 2, ...) — псевдослучайные последовательности, порождаемые независимо работающими программными датчиками Д1 и Д2 соответственно; V = {V(0), ..., V(K - 1)} — вспомогательная таблица К чисел. Вначале V-таблица заполняется К членами последовательности *{bi}:*

http://ok-t.ru/studopediaru/baza3/65136985853.files/image943.gif

Через *{ai}* будем обозначать комбинированную псевдослучайную последовательность. Член *аi* является результатом последовательности операций

s = [*сi, К*],*ai* = V(s), V(s) = *bi+K (i* = 0, 1, 2, ...).

Отсюда видно, что Д2 осуществляет случайный выбор из V-таблицы, а также ее случайное заполнение числами, порожденными Д1. Этот метод позволяет ослабить зависимость между членами *ai* и получать "невероятно" большие периоды, если периоды T1**,** Т2 последовательностей {bi}, {ci} — взаимно простые числа.

**3)Критерий Колмогорова и χ2-критерия Пирсона**

**Критерий согласия Колмогорова** предназначен для проверки гипотезы о принадлежности выборки некоторому закону распределения, то есть проверки того, что эмпирическое распределение соответствует предполагаемой модели.

Эмпирическая функция распределения по случайной выборке реализаций СВ  определяется по формуле:

.

Введём статистику



называемую расстоянием Колмогорова между  и .

Критерий согласия Колмогорова представляет собой следующее решающее правило:

принимается гипотеза 

Порог  - квантиль уровня  распределения Колмогорова,  - задаваемый пользователем уровень значимости.

Критерий χ2-критерия Пирсона широко используется в задачах статистического анализа данных для проверки соответствия экспериментальных данных заданному модельному непрерывному или дискретному закону распределения, определяемому функцией распределения .

Пусть как и при построении гистограммы вычислены частоты  попадания выборочных значений  в  ячеек гистограммы. Гипотетические вероятности попадания значений  в ячейки гистограммы при истинной гипотезе  и полностью заданной функции  равны:

,

где - границы ячеек гистограммы. Статистика критерия проверки гипотез имеет вид:



и характеризует взвешенную сумму квадратов уклонений частот  от гипотетических значений . Чем больше , тем “сильнее” выборка  не согласуется с .

Статистика имеет, в предположении, что гипотеза  верна,  - распределение с  степенями свободы (см. соответствующий раздел).

 - критерий Пирсона основан на (16) и имеет вид:

принимается гипотеза 

где порог критерия  находится из ограничения на ошибку первого рода:  и имеет вид: , здесь  - функция распределения статистики (16).

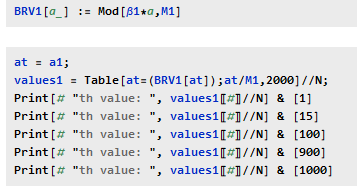
Если  - дискретная СВ с функцией распределения - выборка реализаций СВ , тогда - соответственно частоты и гипотетические вероятности значений .

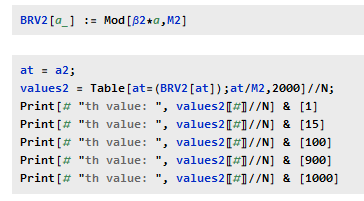
**Выводы:**

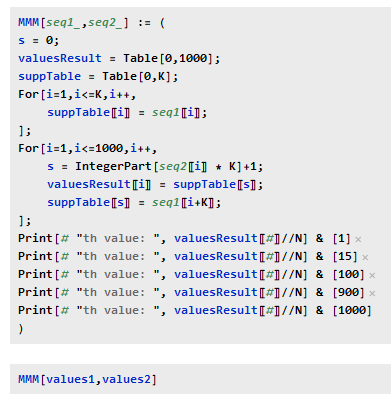
Смоделированы 3 псевдослучайные последовательности : две для мультипликативно- конгруентного метода, а также для метода Макларена- Марсальи с заданными параметрами.

С помощью критерия согласия Колмогорова и χ2-критерия Пирсона проверены точности моделирования для двух датчиков, и оба эти критерия для обоих датчиков выполнены.

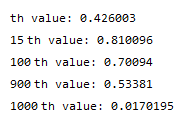
**Код:**

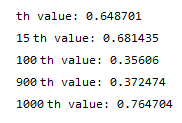


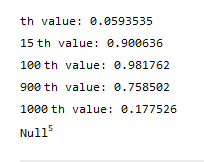




**Результат:**

****

****

****

*Лабораторная работа №2*

**Постановка задачи.**

Смоделировать дискретную случайную величину. Исследовать точность моделирования.

1) Осуществить моделирование n = 1000 реализаций СВ из заданных дискретных распределений.

2) Вывести на экран несмещенные оценки математического ожидания и дисперсии, сравнить их с истинными значениями.

3) Для каждой из случайных величин построить свой χ2-критерием Пирсона с уровнем значимость ε=0.05. Проверить, что вероятность ошибки I рода стремится к 0.05.

4) Осуществить проверку каждой из сгенерированных выборок каждым из построенных критериев.

Бернулли – Bi(1,p), p = 0.3; Биномиальное – Bi(m,p), m = 4, p = 0.2.

**Теория.**

* **Распределение Биномиальное [B(N,p)]:**

Случайная величина  принимает только целые неотрицательные значения, причем

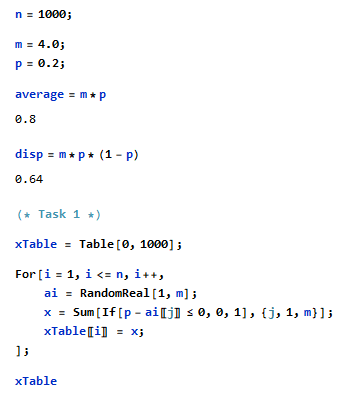
В данной работе, сначала моделировалась последовательность БСВ, а потом по каждой БСВ строился соответствующий элемент выборки распределения Биномиального: отрезок [0;1] разбивался на интервалы , где , проверялось, в какой интервал попадает элемент последовательности БСВ.

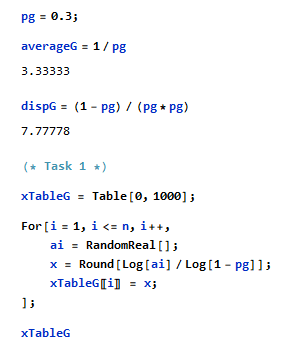
* **Распределение Бернулли**

ДСВ ξ имеет распределение Бернулли Bi(1,p), если: ξ ∈{0,1}, P{ξ = 1}= p, P{ξ = 0}= 1− p, где p ε (0,1) – параметр распределения.

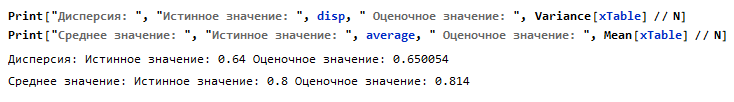
Распределение Бернулли описывает случайный эксперимент (испытание Бернулли) с двумя исходами: успех (ξ = 1) и неудача (ξ = 0), причем вероятность успеха равна p.

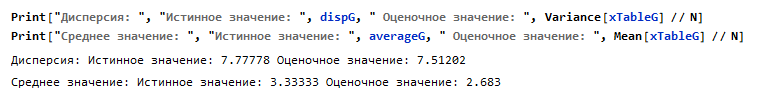
**Код:**

****

****

**Результаты:**

****

****

*Лабораторная работа №3*

**Постановка задачи.**

Смоделировать непрерывную случайную величину. Исследовать точность моделирования.

1) Осуществить моделирование n = 1000 реализаций СВ из нормального закона распределения N(m, s2) с заданными параметрами. Вычислить несмещенные оценки математического ожидания и дисперсии, сравнить их с истинными.

2) Смоделировать n = 1000 СВ из заданных абсолютно непрерывных распределений. Вычислить несмещенные оценки математического ожидания и дисперсии, сравнить их с истинными значениями (если это возможно).

3) Для каждой из случайных величин построить свой критерий Колмогорова с уровнем значимость ε=0.05. Проверить, что вероятность ошибки I рода стремится к 0.05.

4) Для каждой из случайных величин построить свой χ2-критерий Пирсона с уровнем значимость ε=0.05. Проверить, что вероятность ошибки I рода стремится к 0.05.

5) Осуществить проверку каждой из сгенерированных выборок каждым из построенных критериев.

m = -0.8, s 2 = 3; Экспоненциальное Е(a), a = 0.25; Логистическое LG(a, b), a = 0, b = 1.5.

**Теория.**

#### Понятие НСВ.

Непрерывной случайной величиной (НСВ) называется случайная величина имеющая абсолютно-непрерывное распределение вероятности, описываемое функцией распределения



и плотностью распределения



Основными числовыми характеристиками НСВ  являются:

среднее значение ;

дисперсия .

На практике для описания НСВ используются модельные непрерывные законы распределения с функциональными характеристиками, заданными в параметрическом виде:

,

где  - параметр распределения.

#### Метод обратной функции.

Метод обратной функции является одним из универсальных методов моделирования НСВ  с заданной плотностью  и функцией распределения . Приведем математическое обоснование метода и сформулируем моделирующий алгоритм.

Пусть  - строго монотонная возрастающая функция. Найдем обратную функцию , решая относительно  следующее уравнение:

.

Известно, что, если  - БСВ, то СВ , определяемая выражением:

,

имеет заданную плотность  (функцию распределения ).

Таким образом, имеет место следующий алгоритм моделирования:

Моделируется реализация БСВ .

Принимается решение о том, что реализацией СВ  является величина , определяемая в соответствии с (14) по формуле:

.

Коэффициент использования БСВ .

#### Распределение Стьюдента

НСВ  с плотностью распределения

p(x)=

Имеет распределение Стьюдента (t-распределение) t(m) с m степенями свободы(m>0 -натуральное число, параметр распределения).

Среднее значение и дисперсия ~ t (m) равны: .

#### Связь с другими распределениями

СВ ~ t (m) связана с независимыми случайными величинами

~ x(1), ~ x(m), следующими соотношениями:

1.  ~F(1,m);
2. ;
3. .

#### Алгоритм моделирования

- определяется формулой (2).

#### Распределение Лапласа

НСВ  с плотностью распределения

,(27)

имеет распределение Лапласа , где  параметр распределения. Распределение  иногда называется двусторонним экспоненциальным распределением, поскольку график функции (27) можно рассматривать как результат “склеивания” графика плотности распределении со своим зеркальным относительно вертикальной оси отражением (с учетом необходимой нормировки). Согласно (27) функция распределения имеет вид:

 (28)

Среднее значение и дисперсия СВ : .

#### Алгоритм моделирования

Алгоритм моделирования основан на методе обратной функции. Обратная для  функция согласно (28) имеет вид:

(29)

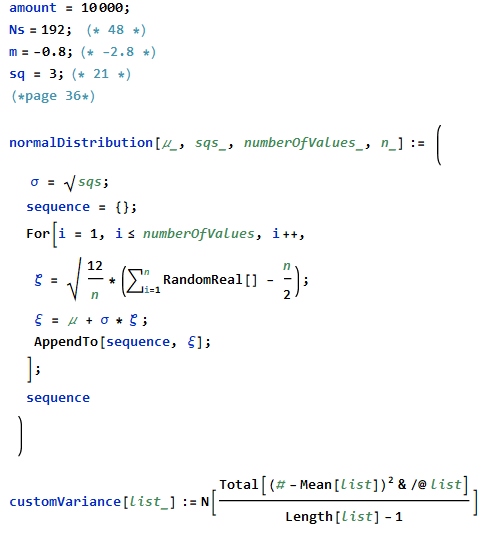
.(30)

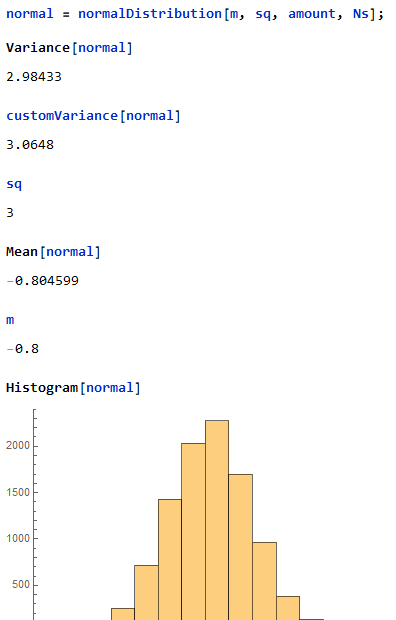
Для моделирования реализации x СВ выполняются следующие действия:

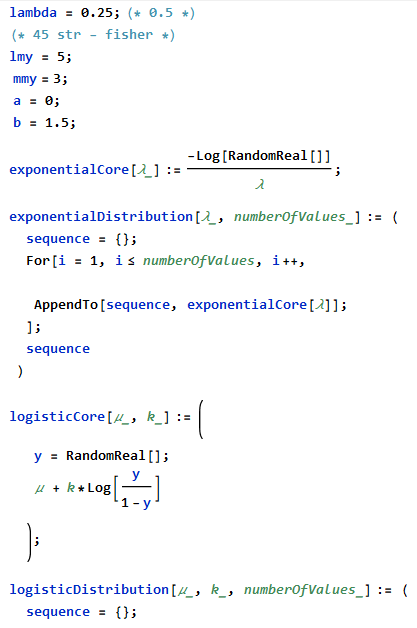
1. Моделируется реализация y БСВ.
2. Принимается решение о том, что реализацией СВ является величина x, вычисляемая по формуле (29), если y<0.5; но формуле (30), если .

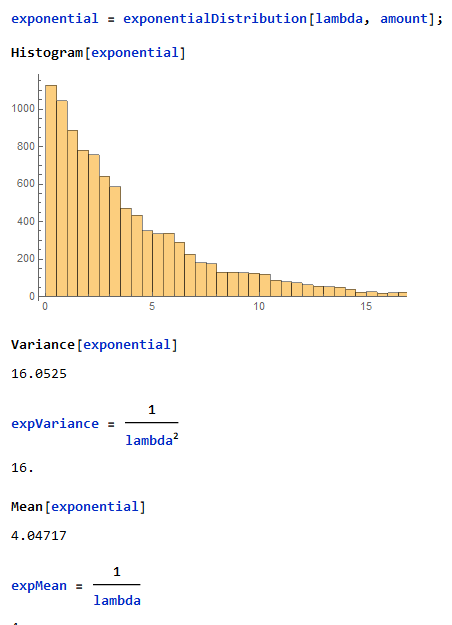
Коэффициент использования БСВ к=1.

**Код и результаты:**









*Лабораторная работа №4*

**Постановка задачи.**

По методу Монте-Карло вычислить приближенное значение определенного

интеграла. Параметр числа итераций n выбрать большим 1000. Сравнить полученное значение

либо с точным значением (если его получится вычислить), либо с приближенным, полученным в

каком-либо математическом пакете.

**Теория**

Задача приближенного вычисления интеграла



где - подмножество из  При n=1 имеем определенный интеграл вида



Заметим, что схема вычислений как многомерных , так и одномерных интегралов , абсолютно аналогична.

Пусть η – произвольная случайная величина с плотностью распределения вероятностей 

Предполагается только, что существуют моменты случайных величин, встречающиеся ниже. Рассмотрим случайную величину, являющуюся функциональным преобразованием случайной величины η.



Можно показать, что  Поэтому в качестве приближенного значения интеграла можно использовать статистическую оценку , построенную в выборке из n независимых случайных величин 



**Код:**

import random

import math

import numpy as np

def integral\_1(x):

return x \* math.atan(x)

def integral\_2(x):

return (x \*\* 4 - 3 \* x \*\* 2 + 1) \* math.exp(-0.5 \* x \*\* 2)

def integral\_3(x, y):

return math.sqrt((y + 1) \* y + math.sin(x) \*\* 2) / (y - 0.7 \* x \* x + 0.3)

def monte\_carlo\_1d(func, a, b, n=10000):

return (b - a) / n \* np.sum([func(random.uniform(a, b)) for \_ in np.arange(n)])

def area\_g(x, y):

if math.fabs(x) < 2 and 4 >= y >= x \*\* 2:

return 1

return -1

def monte\_carlo\_2d(f, g, x0, x1, y0, y1, n):

x = np.random.uniform(x0, x1, n)

y = np.random.uniform(y0, y1, n)

f\_mean = 0

num\_inside = 0

for i in np.arange(n):

for j in np.arange(n):

if g(x[i], y[j]) >= 0:

num\_inside += 1

f\_mean += f(x[i], y[j])

f\_mean = f\_mean / float(num\_inside)

area = num\_inside / float(n \*\* 2) \* (x1 - x0) \* (y1 - y0)

return area \* f\_mean

if \_\_name\_\_ == "\_\_main\_\_":

mc\_1 = monte\_carlo\_1d(integral\_1, 0, math.sqrt(3))

print("Monte Carlo: {}; Wolfram Alpha: {}".format(mc\_1, 1.2284))

mc\_2 = monte\_carlo\_1d(integral\_2, -50, 50)

print("Monte Carlo: {}; Wolfram Alpha: {}".format(mc\_2, math.sqrt(2 \* math.pi)))

mc\_3 = monte\_carlo\_2d(integral\_3, area\_g, -2, 2, 0, 4, 1000)

print("Monte Carlo: {}; Wolfram Alpha: {}".format(mc\_3, 15.6951))

**Результаты:**

Monte Carlo: 1.2329668413871777; Wolfram Alpha: 1.2284  
Monte Carlo: 2.4463694886705984; Wolfram Alpha: 2.5066282746310002  
Monte Carlo: 15.538448925273352; Wolfram Alpha: 15.6951

*Лабораторная работа №4(доп)*

**Постановка задачи.**

Решить систему линейных алгебраических уравнений fAxx методом Монте-

Карло. Сравнить с решением данного уравнения, полученным произвольным численным методом

или решением в произвольном математическом пакете. В качестве матрицы А взять матрицу P из

своего варианта лабораторной работы номер 2 и все ее элементы умножить на 0.9. В качестве

вектора f выбрать вектор

из той же лабораторной работы. Если система получается

несовместной или имеет не одно решение, то разрешается изменить матрицу А, домножив

некоторые ее элементы на -1.

**Теория.**

Необходимо решить систему, представленную в виде , где , собственные значения *A* по модулю меньше 1.

Наша цель – вычислить скалярное произведение вектора решения  с некоторым вектором .

Рассмотрим цепь Маркова с параметрами  такими что





 если 

 если 

Положим



Выберем некоторое натуральное *N* и рассмотрим случайную величину



Где 🡪🡪…🡪 - траекторая цепи Маркова.

*Qm* опряделяется как:



Тогда скалярное произведение вектором *h* и *x* приблизительно равно .

Можем найти *x*, скалярно умножая его на векторы *h* у которых в одной позиции стоит 1, а в остальных – 0.

**Код:**

console.log = function(str){

document.body.innerHTML+=str+'<br>';

}

// let x = 0;

// let y = 0;

let A = [

[0.6,-0.3],

[0.4,0.2]

];

let f = [1,-2];

pi = [0.5,0.5];

p = [

[0.5,0.5],

[0.5,0.5]

];

const N = 100;

const m = 10000;

function solve(N,m,p,pi,A,f,h){

let i1 = [];

let Q = [];

let ksi = new Array(m);

for(let i = 0; i < m; i += 1){

ksi[i] = 0;

}

for(let i = 0; i < m; i += 1){

const mca = ((amount)=>{

const array =[];

for(let i = 0; i< amount; i +=1 ){

array.push(Math.random());

}

return array;

})(N+1);

i1[0] = mca[0] < pi[0] ? 0 : 1;

for(let k = 1; k < N; k += 1){

i1[k] = mca[k] < 0.5 ? 0 : 1;

}

Q[0] = pi[i1[0]] > 0 ? h[i1[0]] / pi[i1[0]] : 0;

for(let k = 1; k < N; k += 1){

if(p[i1[k-1]][i1[k]]){

Q[k] = (Q[k - 1] \* A[i1[k - 1]][i1[k]]) / p[i1[k - 1]][i1[k]];

}

else{

Q[k] = 0;

}

}

for(let k = 0; k < N; k += 1){

ksi[i] += Q[k] \* f[i1[k]];

}

}

return ksi.reduce((acc,cur)=>{acc+=cur; return acc},0)/m;

}

console.log('x: '+solve(N,m,p,pi,A,f,[1,0]));

console.log('y: '+solve(N,m,p,pi,A,f,[0,1]));

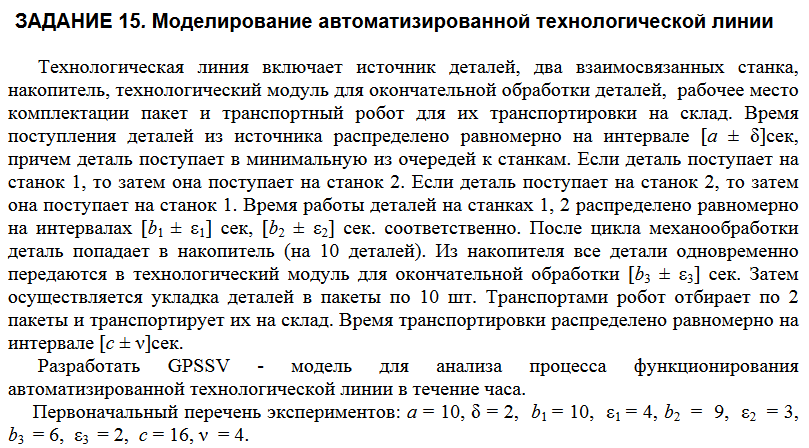
console.log('correct answer: x = 3.18182, y =0.909091');

**Результаты:**

x: 3.1289309777998118  
y: -0.9259226442634948  
correct answer: x = 3.18182, y =0.909091

*Лабораторная работа №5.1*

**Постановка задачи.**

****

**Код:**

INITIAL X$NUM 0; ;Количество в накопителе

INITIAL X$PACKAGE 0; ;Количество упакованных пакетов

PROC STORAGE 2 ;память PROC ёмкостью 2

START 1 ;количество транзактов (количество смен k)

GENERATE 3600 ;1 час в секундах

TERMINATE 1 ;значение счётчика транзактов уменьшается на 1

GENERATE 400 ;поступление детали на обработку

QUEUE ADMISSION\_QUEUE ;очередь поступивших деталей

GATE SNF PROC ;если память PROC не заполнена, входим в блок

DEPART ADMISSION\_QUEUE ;деталь покидает очередь поступивших

ENTER PROC ;войти в память PROC

ADVANCE 10,2

TRANSFER BOTH,START\_MACHINE\_1,START\_MACHINE\_2 ;попытка войти в один из блоков

START\_MACHINE\_1 SEIZE MACHINE\_1 ;занять первый станок

ADVANCE 10,4 ;обработать деталь

RELEASE MACHINE\_1 ;освободить 1й станок

TRANSFER ,AFTER ;переход в блок AFTER

START\_MACHINE\_2 SEIZE MACHINE\_2 ;занять второй станок

ADVANCE 9,3 ;обработать деталь

RELEASE MACHINE\_2

TRANSFER ,AFTER ;переход в блок AFTER

AFTER SAVEVALUE NUM+,1;

TEST E X$NUM 10,MAKE\_PACKAGE

MAKE\_PACKAGE SEIZE PACKAGE

ADVANCE 6,2

RELEASE PACKAGE

SAVEVALUE NUM,0;

SAVEVALUE PACKAGE+,1;

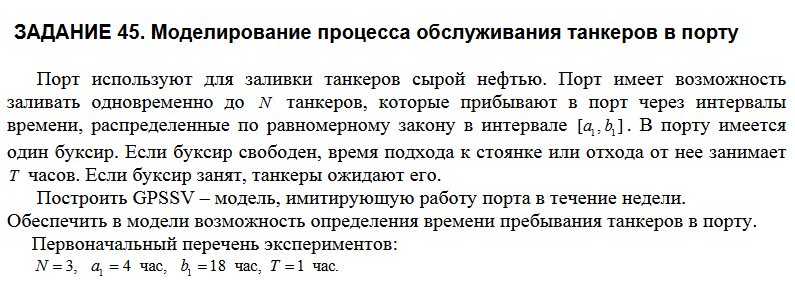
TEST E X$PACKAGE 2,TRANSPORT\_PACKAGE

TRANSPORT\_PACKAGE ADVANCE 16,4

SAVEVALUE PACKAGE,0

*Лабораторная работа №5.2*

**Постановка задачи.**

****

**Код:**

SIMULATE;

\*ОПРЕДЕЛЕНИЕ ФУНКЦИЙ

MEAN FUNCTION P1,L3

1,18/2,24/3,36 ;ОПИСЫВАЕТ СРЕДНЕЕ ВРЕМЯ ЗАЛИВКИ В ЗАВИСИМОСТИ ОТ ТИПА ТАНКЕРА

SPRED FUNCTION P1,L3

1,2/2,3/3,4 ;ОПИСЫВАЕТ ПОЛОВИНУ РАЗМАХА ВРЕМЕНИ ЗАЛИВКИ В ЗАВИСИМОСТИ ОТ ТИПА ТАНКЕРА

TYPE FUNCTION RN1,D3

.25,1/.8,2/1,3 ;(ПЕРВЫЙ 0.25, ВТОРОЙ 0.25+0.55=>0.8, ТРЕТИЙ 0,8+0,2=>1)

XPDIS FUNCTION RN1,C24

0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38

.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99

.96,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8 ;МОДЕЛИРОВАНИЕ ЭКСПОНЕНЦИАЛЬНО РАСПР СЛ.ВЕЛИЧИНЫ ПРИ СР.ЗНАЧ=1

PRICHAL STORAGE 3

\*ОПРЕДЕЛЕНИЕ ТАБЛИЦЫ

1 TABLE M1,20,10,9 ;ВРЕМЯ ПРИБЫВАНИЯ В ПОРТУ ТАНКЕРА 1 ТИПА

2 TABLE M1,20,10,9 ;ВРЕМЯ ПРИБЫВАНИЯ В ПОРТУ ТАНКЕРА 2 ТИПА

3 TABLE M1,40,10,9 ;ВРЕМЯ ПРИБЫВАНИЯ В ПОРТУ ТАНКЕРА 3 ТИПА

\*ОПРЕДЕЛЕНИЕ ПЕРЕМЕННЫХ

GOIN BVARIABLE (SNF$PRICHAL\*FNU$BUKSIR\*LR$STORM); ПРИНИМАЕТ ЗН-Е ИСТИНА ПРИ ВЫПОЛНЕНИИ УСЛ-ИЙ,ОБЕСП ПОДХОД К СТОЯНКЕ

GOOUT BVARIABLE FNU$BUKSIR\*LR$STORM ; ПРИНИМАЕТ ЗН-Е ИСТИНА ПРИ ВЫПОЛНЕНИИ УСЛ-ИЙ,ОБЕСП ОТХОД ОТ СТОЯНКИ

SPACE VARIABLE 48\*N$HERE; ПРИНИМАЕТ ЗН-ИЕ 0,48,96,144, 192

\*ГЕНЕРАЦИЯ ШТОРМА

GENERATE ,,,1; ГЕНЕРАЦИЯ ШТОРМА

NEXT ADVANCE 48,FN$XPDIS;ШТОРМ НА ПУТИ

LOGIC S STORM; ШТОРМ НАСТУПАЕТ, ФЛАГ - ВКЛЮЧИТЬ(S)

ADVANCE 4,2; ШТОРМ ИДЕТ

LOGIC R STORM; ОКОНЧАНИЕ ШТОРМА, ФЛАГ-ВЫКЛЮЧИТЬ(R)

TRANSFER ,NEXT; ЖДАТЬ СЛЕДУЮЩЕГО ШТОРМА

\*ИНИЦИАЛИЗАЦИЯ ПРЕДПОЛАГАЕМЫХ ТАНКЕРОВ

GENERATE ,,,5

ASSIGN 1,4; ПРИСВОИТЬ ЗНАЧЕНИЕ 4

HERE ADVANCE V$SPACE; РАСТЯНУТЬ ВРЕМЯ И ПОСЕТИТЬ В ПЕРВУЮ ОЧЕРЕДЬ ПОРТ

MARK 3; P3 ЗАПИСЫВАЕТ ТЕКУЩЕЕ ВРЕМЯ ТАЙМЕРА

TRANSFER ,PORT; ОТПРАВИТЬ В ПОРТ

\*ДЕЯТЕЛЬНОСТЬ ПОРТА

GENERATE 11,7;

ASSIGN 1,FN$TYPE ; ПРИСВАИВАЕТ ТИП ТАНКЕРА

PORT TEST E BV$GOIN,1; ЖДАТЬ ЧТО ВСЕ ОК ДЛЯ ДВИЖЕНИЯ В ПРИЧАЛ (BV БУЛЕВСКАЯ)

SEIZE BUKSIR; ЗАНЯТЬ БУКСИР

ENTER PRICHAL

ADVANCE 1; ПЕРЕМЕЩЕНИЕ ТАНКЕРА В ПРИЧАЛ

RELEASE BUKSIR; БОЛЬШЕ НЕ НУЖДАЕМСЯ В БУКСИРЕ

ASSIGN 2,FN$SPRED; ПАРАМЕТРУ P2 ПРИСВАЕВАЕМ СР РАЗМАХ ВРЕМЕНИ

ADVANCE FN$MEAN,P2 ;ТАНКЕР ЗАГРУЖАЕТСЯ

TEST E BV$GOOUT,1 ;ЖДАТЬ ЧТО ВСЕ ОК ДЛЯ ВЫХОДА ИЗ ПРИЧАЛА (BV БУЛЕВСКАЯ)

SEIZE BUKSIR ; ЗАНЯТЬ БУКСИР

ADVANCE 1 ; ПЕРЕМЕЩЕНИЕ ТАНКЕРА ИЗ ПРИЧАЛА

RELEASE BUKSIR; БУКСИР БОЛЬШЕ НЕ НУЖЕН

LEAVE PRICHAL; РИЧАЛ ДОСТУПЕН ДЛЯ ДР ТАНКЕРОВ

TABULATE P1; ЗАПИСАТЬ ВРЕМЯ ПРИБЫВАНИЯ В ПОРТУ

TEST NE P1,4,CYCLE; ОТСЕИВАТЬ ПРЕПОЛОГАЕМЫЕ ТАНКЕРЫ

TERMINATE;

CYCLE ADVANCE 240,24 ВРЕМЯ ПРИБЫВАНИЯ В ПУТИ ТУДА ОБРАТНО

MARK 3; В P3 ВРЕМЯ ВОЗВРАЩЕНИЯ В ПОРТ

TRANSFER .PORT ; ТАНКЕРЫ ВОЗВР-СЯ В ПОРТ

GENERATE 8760; 1 ГОД

TERMINATE 1;

**Литература**

1. Харин Ю.С., Малюгин В.И., Кирлица В.П., Лобач В.И., Хацкевич Г.А. Основы имитационного и статистического моделирования. Учебное пособие. Минск: ДизайнПРО, 1997 – 228 с.
2. Лобач В.И., Кирлица В.П., Малюгин В.И., Сталевская С.Н. Имитационное и статистическое моделирование. Практикум для студентов математических и экономических специальностей. Минск, БГУ, 2004 –189 с.