

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Моделирование

ОТЧЕТ
по лабораторной работе № 8
на тему
МЕТОД МОНТЕ-КАРЛО
ВАРИАНТ № 1, 1, 1

Студент:

П.В. Сякачѐв

Проверила:

Ю.О. Герман

МИНСК 2022

1. Цель работы

Научиться имитировать случайные события методом Монте-Карло.

2. Задание

2.1 Имитация случайных событий

- разработать алгоритм имитации для решения задачи на основе метода Монте-Карло.
- выполнить два испытания алгоритма, используя случайные числа из учебного пособия.
- разработать имитационную модель для решения задачи.

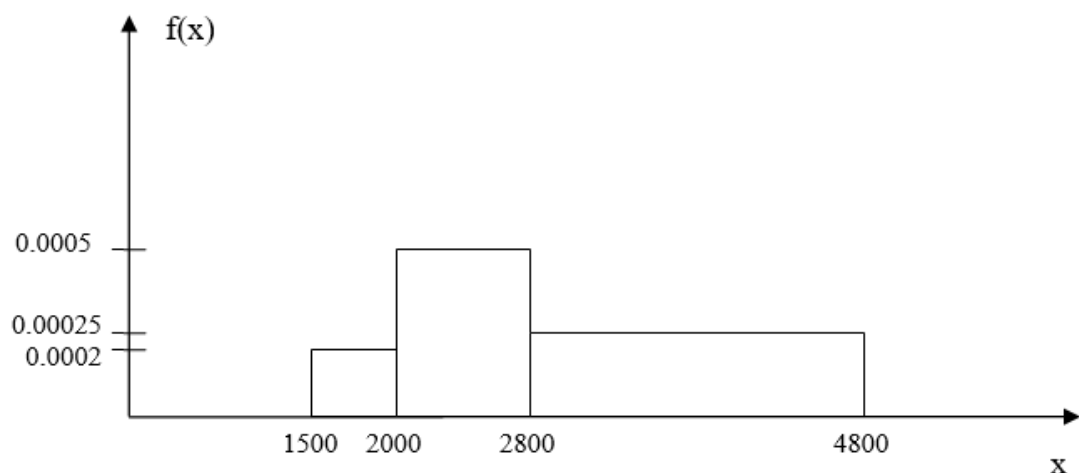
Предприятие выпускает некоторые изделия. Себестоимость изделия – 10 ден.ед. Изделия продаются по цене 15 ден.ед.

Изделие может иметь дефекты трех типов: А, В и С. Дефекты не связаны друг с другом, т.е. они возможны в любых комбинациях. Дефект А имеется у 6% изделий, дефект В – у 4%, дефект С – у 9%. Если изделие имеет один дефект, то он устраняется. Затраты на устранение дефекта А, В и С составляют, соответственно, 4, 6 и 2 ден.ед. Если изделие имеет два и более дефектов, то оно бракуется (т.е. в этом случае убытки от его выпуска составляют 10 ден.ед.).

Найти: а) среднюю прибыль от одного изделия; б) вероятность того, что изделие не будет забраковано.

2.2 Имитация случайных величин: метод обратных функций, метод исключений

Некоторая непрерывная случайная величина X задана плотностью распределения.



Требуется:

- 1) решить задачу имитации заданной случайной величины методом обратных функций. Для этого:
 - разработать алгоритм имитации заданной случайной величины на основе метода обратных функций;
 - используя случайные числа из учебного пособия (приложение 1), выполнить имитацию двух значений заданной случайной величины;
 - выполнить проверку разработанного алгоритма, используя построение гистограммы;
- 2) решить задачу имитации заданной случайной величины методом исключений, выполнив те же действия, что и в пункте 1);
- 3) используя один из разработанных алгоритмов, выполнить имитацию заданной случайной величины и найти для нее: а) среднее значение; б) вероятность того, что величина X превысит некоторое заданное значение (выбрать это значение самостоятельно).

2.3 Решение задач моделирования объектов на основе имитации случайных величин

- разработать алгоритм имитации для решения задач на основе метода Монте-Карло.
- выполнить одно испытание алгоритма, используя случайные числа из учебного пособия (приложение 1).
- разработать имитационную модель на любом языке для решения задачи.

2.3.1 Часть 1

Предприятие выпускает комплекты инструментов по заказам. Количество инструментов в комплекте – от 3 до 6. Инструменты могут быть стандартными или высокоточными; комплект может содержать стандартные и высокоточные инструменты в любой комбинации. Время изготовления одного стандартного инструмента составляет от 30 мин до одного часа, высокоточного – от 40 мин до двух часов. Время испытания инструмента – экспоненциальная случайная величина со средним значением 10 мин для стандартных инструментов, 15 мин – для высокоточных.

Найти: а) среднее время изготовления и испытания комплекта; б) вероятность того, что комплект будет состоять только из высокоточных инструментов; в) вероятность того, что время изготовления и испытания комплекта превысит 24 часа.

2.3.2 Часть 2

В мастерской выполняется проверка и ремонт некоторых приборов. Прибор состоит из пяти схем. Вероятность неисправности схемы - 0,1. Работа с прибором включает следующие операции:

- проверка: от 2 до 5 мин на каждую схему;
- замена неисправных схем: время замены одной схемы – гауссовская случайная величина со средним значением 5 мин и стандартным отклонением 0,5 мин;
- настройка. Время настройки – экспоненциальная случайная величина со средним значением 6 мин, если ни одна схема не заменялась, и 10 мин – если потребовалась замена хотя бы одной схемы.

Найти: а) среднее время работы с прибором; б) вероятность того, что в приборе потребуется замена хотя бы одной схемы; в) среднее количество схем, заменяемых в одном приборе.

3. Ход работы

3.1 Общая часть выполнения

Для решения поставленных задач был выбран язык C#. Для получения наборов случайных чисел был написан класс Random_functions с следующими методами:

- Get_System_Random() – получение случайных чисел, используя средства языка;
- Get_Lehmer_Random() – получение случайных чисел при помощи метода Лемера;
- Get_Congruent_Random() – получение случайных чисел при помощи конгруэнтного метода;
- Get_Prepared1() – получение заранее заготовленных чисел из приложения 1;
- Get_Prepared2() – получение заранее заготовленных чисел из приложения 1;
- Get_Prepared3() – получение заранее заготовленных чисел из приложения 1;

3.2 Задание 1

Для выполнения задания 1 был написан класс Part_1, вот основной его метод:

```
public void Run(double[] chances_A, double[] chances_B, double[] chances_C)
{
    int cost = 10;
    int price = 25;
```

```

int cost_A = 4, cost_B = 6, cost_C = 2;
double chance_A = 0.06, chance_B = 0.04, chance_C = 0.09;
for (int i = 0; i < 100; i++)
{
    money -= cost;
    bool is_A = false;
    bool is_B = false;
    bool is_C = false;
    if (chances_A[i] <= chance_A) is_A = true;
    if (chances_B[i] <= chance_B) is_B = true;
    if (chances_C[i] <= chance_C) is_C = true;
    if (is_A || is_B || is_C)
        if (!(is_A && is_B && is_C) && (is_A ^ is_B ^ is_C))
        {
            if (is_A) money -= cost_A;
            else if (is_B) money -= cost_B;
            else if (is_C) money -= cost_C;
        }
    else
    {
        rejected++;
        continue;
    }
    money += price;
}
count++;
}

```

Результат выполнения этой части программы:

```

----- Часть первая. Вариант 1 -----

Используя заранее заготовленные числа (100 итераций):
Средний общий доход:                1386
Средняя прибыль от изделия:          13,86
Вероятность того, что изделие не забракут: 98%

Используя средства языка (1000x100 итераций):
Средний общий доход:                1415,34
Средняя прибыль от изделия:          14,15
Вероятность того, что изделие не забракут: 98,9%

```

3.3 Задание 2

Запишем уравнения плотности распределения:

$$f(x) = \begin{cases} 0, & x < 1500 \\ 0,0002, & 1500 \leq x < 2000 \\ 0,0005, & 2000 \leq x < 2800 \\ 0,00025, & 2800 \leq x < 4800 \\ 0, & x \geq 4800 \end{cases}$$

Найдем выражения для функции распределения:

На интервале $x < 1500$: $\int_{-\infty}^x 0dx = 0$

На интервале $1500 \leq x < 2000$: $\int_{-\infty}^{1500} 0dx + \int_{1500}^x 0,0002dx = 0,0002x - 0,3$

На интервале $2000 \leq x < 2800$: $\int_{-\infty}^{1500} 0dx + \int_{1500}^{2000} 0,0002dx + \int_{2000}^x 0,0005dx = 0,0005x - 0,9$

На интервале $2800 \leq x < 4800$: $\int_{-\infty}^{1500} 0dx + \int_{1500}^{2000} 0,0002dx + \int_{2000}^{2800} 0,0005dx + \int_{2800}^x 0,00025dx = 0,00025x - 0,2$

На интервале $x \geq 4800$: $\int_{-\infty}^{1500} 0dx + \int_{1500}^{2000} 0,0002dx + \int_{2000}^{2800} 0,0005dx + \int_{2800}^{4800} 0,00025dx = 1$

Таким образом функция распределения времени внесения добавок имеет вид:

$$F(x) = \begin{cases} 0, & x < 1500 \\ 0,0002x - 0,3, & 1500 \leq x < 2000 \\ 0,0005x - 0,9, & 2000 \leq x < 2800 \\ 0,00025x - 0,2, & 2800 \leq x < 4800 \\ 1, & x \geq 4800 \end{cases}$$

Находим выражение $X = F^{-1}(R)$:

$$X = \begin{cases} \frac{(R + 0,3)}{0,0002} & 0 \leq R < 0,1 \\ \frac{(R + 0,9)}{0,0005} & 0,1 \leq R < 0,5 \\ \frac{(R + 0,2)}{0,00025} & 0,5 \leq R < 1 \end{cases}$$

Для выполнения задания 2 был написан класс Part_2, вот два основных его метода:

```
public void Run_1(double[] R)
{
    double temp;
    for (int i = 0; i < 100; i++)
    {
        if (R[i] < 0.1) temp = (R[i] + 0.3) / 0.0002;
        else if (R[i] < 0.5) temp = (R[i] + 0.9) / 0.0005;
        else temp = (R[i] + 0.2) / 0.00025;
        X += temp;
        if (temp > value) exceeding++;
        count++;
    }
}
```

```

public void Run_2(double[] R1, double[] R2)
{
    int a = 1500;
    int b = 4800;
    double fmax = 0.0005;
    double Rs1, Rs2, FR;
    for (int i = 0; i < 100; i++)
    {
        Rs1 = a + (b - a) * R1[i];
        Rs2 = fmax * R2[i];
        if (Rs2 >= 1500 && Rs2 <= 2000) FR = 0.0002;
        else if (Rs2 > 2000 && Rs2 <= 2800) FR = 0.0005;
        else FR = 0.00025;
        if (Rs2 <= FR)
            X += Rs1;
        else continue;
        if (Rs1 > value) exceeding++;
        count++;
    }
}

```

Результат выполнения этой части программы:

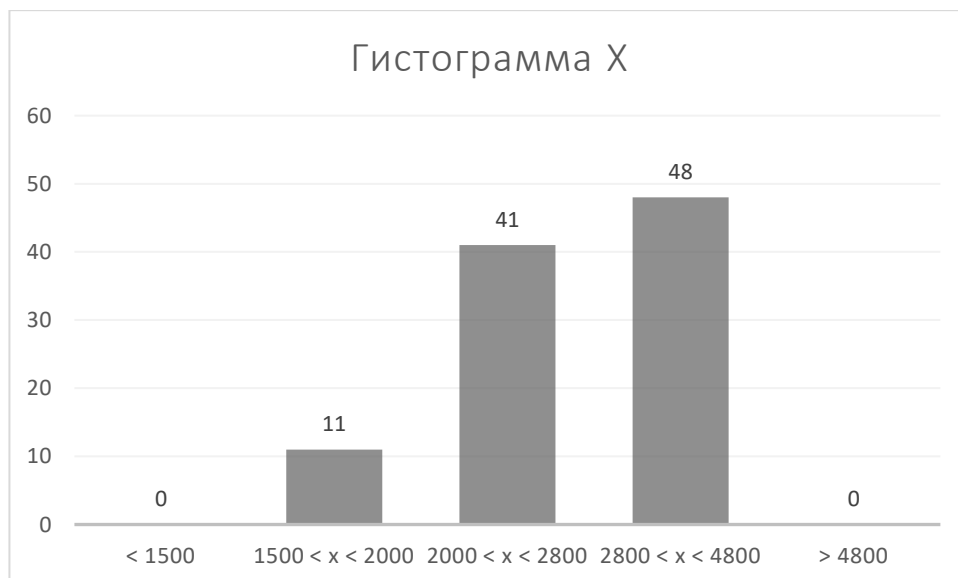
```

----- Часть вторая. Вариант 1 -----
Используя метод обратных функций:
Среднее значение X:                      2939,68
Вероятность того, что X превысит значение 0 : 100%

Используя метод исключений:
Среднее значение X:                      2892,73
Вероятность того, что X превысит значение 0 : 100%

```

Гистограмма значений X для метода обратных функций:



3.4 Задание 3

3.4.1 Часть 1

Для выполнения первой части третьего задания был написан класс Part_3_1, вот основной его метод:

```
public void Run(double[] Rand)
{
    int t = 0;
    for (int i = 0; i < 5; i++)
    {
        double cur_time = 0;
        bool is_precise = true;
        int number;
        double r = Rand[++t];
        if (r < 0.25) number = 3;
        else if (r < 0.5) number = 4;
        else if (r < 0.75) number = 5;
        else number = 6;
        for (int j = 0; j < number; j++)
        {
            if (Rand[++t] < 0.5)
            {
                is_precise = false;
                cur_time += standard_time(new double[2] { Rand[++t], Rand[++t] });
            }
            else cur_time += standard_time(new double[2] { Rand[++t], Rand[++t] });
        }
        time += cur_time;
        if (is_precise) precise_count++;
        if (cur_time > value) exceeding++;
        count++;
    }
}
```

Результат выполнения этой части программы

```
Первая часть:
Среднее время изготовления комплекта:      246,04 минут
Вероятность того, комплект состоит только из высокоточных инструментов: 5,96%
Вероятность того, что среднее время превысит 24 часа: 0%
```

3.4.1 Часть 1

Для выполнения первой части третьего задания был написан класс Part_3_2, вот основной его метод:

```
public void Run(double[] Rand)
{
    int t = 0;
    double r_sum;
    double cur_time;
    bool is_serviceable;
    for (int i = 0; i < 2; i++)
    {
        cur_time = 0;
```



```

is_serviceable = true;
for (int j = 0; j < 5; j++)
{
    cur_time += 2 + (5 - 2) * Rand[+t];
    if (Rand[+t] <= 0.1)
    {
        r_sum = 0;
        for (int k = 0; k < 6; k++)
            r_sum += Rand[+t];
        cur_time += 5 + 0.5 * Math.Sqrt(2) * (r_sum - 3);
        is_serviceable = false;
        subst_number++;
    }
}
if (is_serviceable)
    cur_time = -6 * Math.Log(1 - Rand[+t]);
else
{
    cur_time = -10 * Math.Log(1 - Rand[+t]);
    subst_count++;
}
time += cur_time;
count++;
}
}

```

Результат выполнения этой части программы:

```

Вторая часть:
Среднее время работы с прибором:          7,65 минут
Вероятность того, потребуется хотя бы одна замена: 42,7%
Среднее количество заменяемых схем:        0,52

```

Вывод

В ходе выполнения лабораторной работы научились имитировать случайные события методом Монте-Карло.