

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Рязанский государственный радиотехнический университет
Имени В. Ф. Уткина»

Факультет вычислительной техники
Кафедра вычислительной и прикладной математики

Отчёт по практической работе №6

по дисциплине:
«Моделирование»

по теме:
«Моделирование методом Монте-Карло»

Выполнил: студент гр. 242

Фокин А.М.

Проверил: Анастасьев А. А.

Рязань 2025

Цель работы:

Составить программу решения задачи, определенной в соответствии с вариантом задания, с помощью машинного моделирования (метод Монте-Карло). Построить доверительный интервал для полученных оценок, покрывающий точное значение оцениваемых вероятностей с надежностью $\beta=0,95$. Правильность результатов проверить аналитическим решением задачи.

19. Завод изготавливает изделия, каждое из которых с вероятностью r (независимо от других) оказывается дефектным. При осмотре дефект, если он имеется, обнаруживается с вероятностью p . Для контроля из продукции завода выбирается n изделий. Оценить вероятность следующих событий: \

A – ни в одном из изделий не будет обнаружено дефекта;
B – среди n изделий ровно в двух будет обнаружен дефект;
C – среди n изделий не менее чем в двух будет обнаружен дефект.

Программа реализует следующие операции:

– **Моделирование методом Монте-Карло:** для каждого испытания создаются случайные изделия, где вероятность брака и его обнаружения задаются как

```
defects = np.random.rand(n) < r  
detected = defects & (np.random.rand(n) < p)
```

– **Подсчёт количества обнаруженных дефектов и классификация событий:**

```
k = np.sum(detected)  
(при  $k == 0$  — событие A,  $k == 2$  — событие B,  $k \geq 2$  — событие C)
```

– **Оценка вероятностей по результатам моделирования:**

```
prob_A = results_A / N  
prob_B = results_B / N  
prob_C = results_C / N
```

– **Построение доверительных интервалов для оценок вероятностей:**

```
d = 1.96 * math.sqrt(prob * (1 - prob) / N)  
(интервал:  $\text{prob} \pm d$  при доверии  $\beta = 0.95$ )
```

– **Проверка правильности результатов аналитическим решением:**

используются биномиальные формулы:

```
true_A = (1 - p_detect) ** n  
true_B = math.comb(n, 2) * (p_detect ** 2) * ((1 - p_detect) **  
(n - 2))  
true_C = 1 - (...)
```

полный код программы приведен в приложении 1

Результат работы программы:

===== RESTART: D:/мусорка/учеба/моделирования/lab6-Fokin-242.py =====

Оценка вероятности события A: 0.43402

Доверительный интервал: 0.4309480687984798 0.43709193120152023

Оценка вероятности события B: 0.14996

Доверительный интервал: 0.1477470914409912 0.1521729085590088

Оценка вероятности события C: 0.1908

Доверительный интервал: 0.18836458378305473 0.19323541621694526

Аналитическое значение A: 0.4343884542236318

Аналитическое значение B: 0.14780703546361773

Аналитическое значение C: 0.18788245514712304

|

Оценка вероятности события A: 0.43402

Доверительный интервал: 0.4309480687984798 0.43709193120152023

Оценка вероятности события B: 0.14996

Доверительный интервал: 0.1477470914409912 0.1521729085590088

Оценка вероятности события C: 0.1908

Доверительный интервал: 0.18836458378305473 0.19323541621694526

Аналитическое значение A: 0.4343884542236318

Аналитическое значение B: 0.14780703546361773

Аналитическое значение C: 0.18788245514712304

Ответ на вопрос

8. Какие события называются зависимыми? Как их моделировать?

Случайные события А и В называются зависимыми, если появление одного из них изменяет вероятность появления другого события – В

Difference between	
Dependent Events	Independent Events
<ul style="list-style-type: none">▶ Events where the outcome of one event affects the probability of the other.▶ Ex - Drawing cards without replacement.▶ Formula : $P(A \text{ and } B) = P(A) \times P(B A)$	<ul style="list-style-type: none">▶ Events where the outcome of one event does not affect the probability of the other.▶ Ex - Tossing a coin and rolling a die.▶ Formula : $P(A \text{ and } B) = P(A) \times P(B)$

Источник:

<https://www.geeksforgeeks.org/maths/dependent-and-independent-events-probability/>

Приложение 1 - код программы

```
import numpy as np
import math

r = 0.1
p = 0.8
n = 10
N = 100000
beta = 0.95

results_A = 0
results_B = 0
results_C = 0

for _ in range(N):
    defects = np.random.rand(n) < r
    detected = defects & (np.random.rand(n) < p)
    k = np.sum(detected)
    if k == 0:
        results_A += 1
    if k == 2:
        results_B += 1
    if k >= 2:
        results_C += 1

prob_A = results_A / N
prob_B = results_B / N
prob_C = results_C / N

def conf_interval(prob):
    z = 1.96
    d = z * math.sqrt(prob * (1 - prob) / N)
    return prob - d, prob + d

a_low, a_high = conf_interval(prob_A)
b_low, b_high = conf_interval(prob_B)
c_low, c_high = conf_interval(prob_C)

print("Оценка вероятности события A:", prob_A)
print("Доверительный интервал:", a_low, a_high)
print("Оценка вероятности события B:", prob_B)
print("Доверительный интервал:", b_low, b_high)
print("Оценка вероятности события C:", prob_C)
```

```
print("Доверительный интервал:", c_low, c_high)

p_detect = r * p
true_A = (1 - p_detect) ** n
true_B = math.comb(n, 2) * (p_detect ** 2) * ((1 - p_detect) **
(n - 2))
true_C = 1 - ((1 - p_detect) ** n) - n * p_detect * ((1 -
p_detect) ** (n - 1))

print("Аналитическое значение A:", true_A)
print("Аналитическое значение B:", true_B)
print("Аналитическое значение C:", true_C)
```