

**ВСЕРОССИЙСКИЙ ДЕТСКИЙ КОНКУРС НАУЧНО-  
ИССЛЕДОВАТЕЛЬСКИХ И ТВОРЧЕСКИХ РАБОТ  
«ПЕРВЫЕ ШАГИ В НАУКЕ»**

---

**Направление:** ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, МАТЕМАТИКА

**Тема:** Основные принципы теории вероятности и методы получения случайных чисел

**Соискатель:** Пешков Кирилл Александрович

**Научный руководитель:** Малиничева Юлия Валентиновна

**Место выполнения работы:** МАОУ "Гимназия №11 "Гармония"

## Оглавление

1. Введение.....	3
2. Основы теории вероятности и её использование.....	5
2.1. Понятие вероятности .....	5
2.2. Закон больших чисел.....	5
2.3. Генераторы псевдослучайных чисел .....	5
2.4. Генераторы случайных чисел .....	6
3. Орел и Решка .....	7
3.1. Подбрасывание монетки .....	7
3.2. Генератор псевдослучайных чисел на Python.....	8
3.2. Генератор случайных чисел на Arduino .....	9
Заключение .....	12
Список литературы .....	14

## 1. Введение

Современная наука и техника широко опираются на методы, позволяющие описывать и анализировать явления, исход которых невозможно предсказать заранее [1,4]. Подобные явления называют случайными, а их изучением занимается теория вероятностей [2,3]. Ситуации неопределённости окружают человека повсюду: от простого подбрасывания монеты до сложных физических или экономических процессов. Понимание природы случайности и умение количественно оценивать вероятность событий необходимы как в научной деятельности, так и в практических задачах моделирования, инженерии, программирования и анализа данных. Именно поэтому изучение основных принципов теории вероятностей и методов получения случайных чисел является актуальной задачей.

Особую значимость приобретает сравнение теоретических закономерностей случайных процессов с их практическим проявлением. С одной стороны, теория вероятностей предлагает строгий математический аппарат для вычисления вероятностей событий, моделирования случайных величин и анализа поведения больших выборок. С другой стороны, в реальных экспериментах результаты не всегда совпадают с ожидаемыми, что делает важным изучение закона больших чисел и методов генерации случайных величин.

Кроме того, в современной вычислительной технике широко используются генераторы псевдослучайных чисел, основанные на алгоритмах, а также физические генераторы случайных чисел, использующие реальные природные процессы. Сравнение их работы, точности и поведения при моделировании случайных событий позволяет глубже понять природу случайности и возможности её практического получения.

**Объект исследования** — случайные события и их вероятностные характеристики.

**Предмет исследования** — особенности генерации случайных чисел различными методами и их применение при моделировании подбрасывания монеты.

**Цель исследования** — изучить свойства случайных событий и сравнить результаты моделирования подбрасывания монеты, полученные при помощи псевдослучайных и физических генераторов случайных чисел, с теоретическими ожиданиями.

Для достижения цели были поставлены следующие задачи:

1. Рассмотреть основные понятия и законы теории вероятностей.
2. Описать принципы работы генераторов псевдослучайных чисел и их применение в языке Python.
3. Изучить работу физических генераторов случайных чисел на базе Arduino.
4. Провести серию экспериментов по моделированию подбрасывания монеты различными методами.
5. Сравнить экспериментальные данные с теоретической вероятностью и проанализировать их с точки зрения закона больших чисел.

**Гипотеза исследования** состоит в предположении, что при увеличении количества испытаний относительные частоты выпадения «орла» и «решки», полученные как при реальном подбрасывании монеты, так и при использовании псевдослучайных и физических генераторов, будут стремиться к теоретической вероятности 0,5.

В работе использовались следующие методы исследования:

- теоретический анализ литературы по теории вероятностей;
- экспериментальные методы (серии подбрасываний монет разных номиналов);

- компьютерное моделирование на Python;
- практическая работа с Arduino и библиотекой TrueRandom;
- сравнение и статистическая обработка результатов экспериментов.

## 2. Основы теории вероятности и её использование

### 2.1. Понятие вероятности

Вероятность — это число, показывающее, насколько возможно, что произойдёт какое-то событие [1,4].

Для нахождения вероятности используют классическую формулу вероятности:

$$P(A) = \frac{m}{n} \quad (1)$$

где:

- $P(A)$  — вероятность события  $A$
- $m$  — количество благоприятных исходов
- $n$  — общее количество всех исходов

Вероятность всегда описывается  $0 \leq P(A) \leq 1$ , где 0 это событие невозможно, а 1 это событие, которое обязательно произойдет [].

Например:

- При подбрасывании монетки вероятность выпадения «орла» равна 1 из 2, то есть 0,5 (или 50%).
- При броске кубика вероятность выпадения каждой стороны равна 1 из 6.

Понятие вероятности помогает численно описывать случайность. Чем больше вероятность, тем более возможно событие.

### 2.2. Закон больших чисел

Результат одного эксперимента может сильно отличаться от того, что мы видим в теории, но, если повторять опыт много раз, результаты начинают выравниваться. Именно это объясняет закон больших чисел.

Закон больших чисел утверждает. Чем больше раз мы повторяем случайный эксперимент, тем ближе относительная частота наступления события к его теоретической вероятности [1,3,8]. То есть если  $n$  стремится к бесконечности, то  $\frac{m}{n}$  стремится к  $P(A)$ .

Данный закон связывает реальные результаты с теорией вероятности. И демонстрирует, что случайность не означает хаос, а может быть предсказуемой.

### 2.3. Генераторы псевдослучайных чисел

Компьютер, являясь электронной вычислительной машиной, работающей на алгоритмах, не может создавать полностью случайные числа [6,7]. Всё, что он генерирует, называется псевдослучайными числами.

Псевдослучайные числа — это числа, которые выглядят случайными и ведут себя как случайные в большинстве экспериментов, но на самом деле вычисляются по определённой формуле [6,7].

Они зависят от начального значения, которое называется зерно (seed).

Если использовать одно и то же зерно, последовательность чисел будет одинаковой, что удобно для проверки и повторяемости экспериментов.

Псевдослучайные числа широко используют в:

1. Моделировании случайных процессов.
2. Игровых программах и симуляциях.
3. Криптографии и защите информации.
4. Научных и инженерных расчетах, где нужен большой объём случайных данных.

#### **2.4. Генераторы случайных чисел**

Генераторы случайных чисел — это устройства или системы, которые создают последовательность чисел, каждое из которых невозможно заранее предсказать. Такие числа необходимы для экспериментов, моделирования процессов и проверки законов теории вероятности [6,7].

Физические генераторы случайных чисел используют естественные процессы, в которых результат определяется множеством мельчайших случайных факторов.

Примеры таких процессов:

- электрический шум в цепях,
- колебания температуры или света,
- атмосферные шумы и другие природные явления.

Использование этих процессов позволяет получать настоящие случайные числа, которые невозможно предсказать или повторить. Так же активно используются в тех же сферах, что и псевдослучайные числа.

### 3. Орел и Решка

#### 3.1. Подбрасывание монетки

Подбрасывание монетки — один из самых простых и наглядных примеров случайного события. Монета имеет две стороны: орёл и решка. При каждом подбрасывании невозможно заранее сказать, какая сторона выпадет.

Результат одного подбрасывания не зависит от предыдущих (например, если вчера выпал орёл, сегодня вероятность выпадения орла остается той же).

Как мы знаем из главы «2.1 Понятие вероятности», вероятности выпадения орла или решки равны по 50%

После проведения серии экспериментов в различные дни и с различными монетами получили следующие данные:

Количество бросков	Орел	Решка	P(Орел)	P(Решка)
10	7	3	0,7	0,3
50	23	27	0,46	0,54
100	48	52	0,48	0,52
150	73	77	0,49	0,51

*Таблица 3.1. Результат подбрасывания 10 рублевой монеты*

Количество бросков	Орел	Решка	P(Орел)	P(Решка)
10	3	7	0,3	0,7
50	23	27	0,46	0,54
100	47	53	0,47	0,53
150	72	78	0,48	0,52

*Таблица 3.2. Результат подбрасывания 5 рублевой монеты*

Количество бросков	Орел	Решка	P(Орел)	P(Решка)
10	6	4	0,6	0,4
50	20	30	0,4	0,6
100	47	57	0,47	0,53
150	75	75	0,5	0,5

*Таблица 3.3. Результат подбрасывания 1 рублевой монеты*

Количество бросков	Орел	Решка	P(Орел)	P(Решка)
10	5	5	0,5	0,5
50	27	23	0,54	0,46
100	51	49	0,51	0,49
150	69	81	0,46	0,54

Таблица 3.4. Результат подбрасывания монеты номиналом 1 копейка

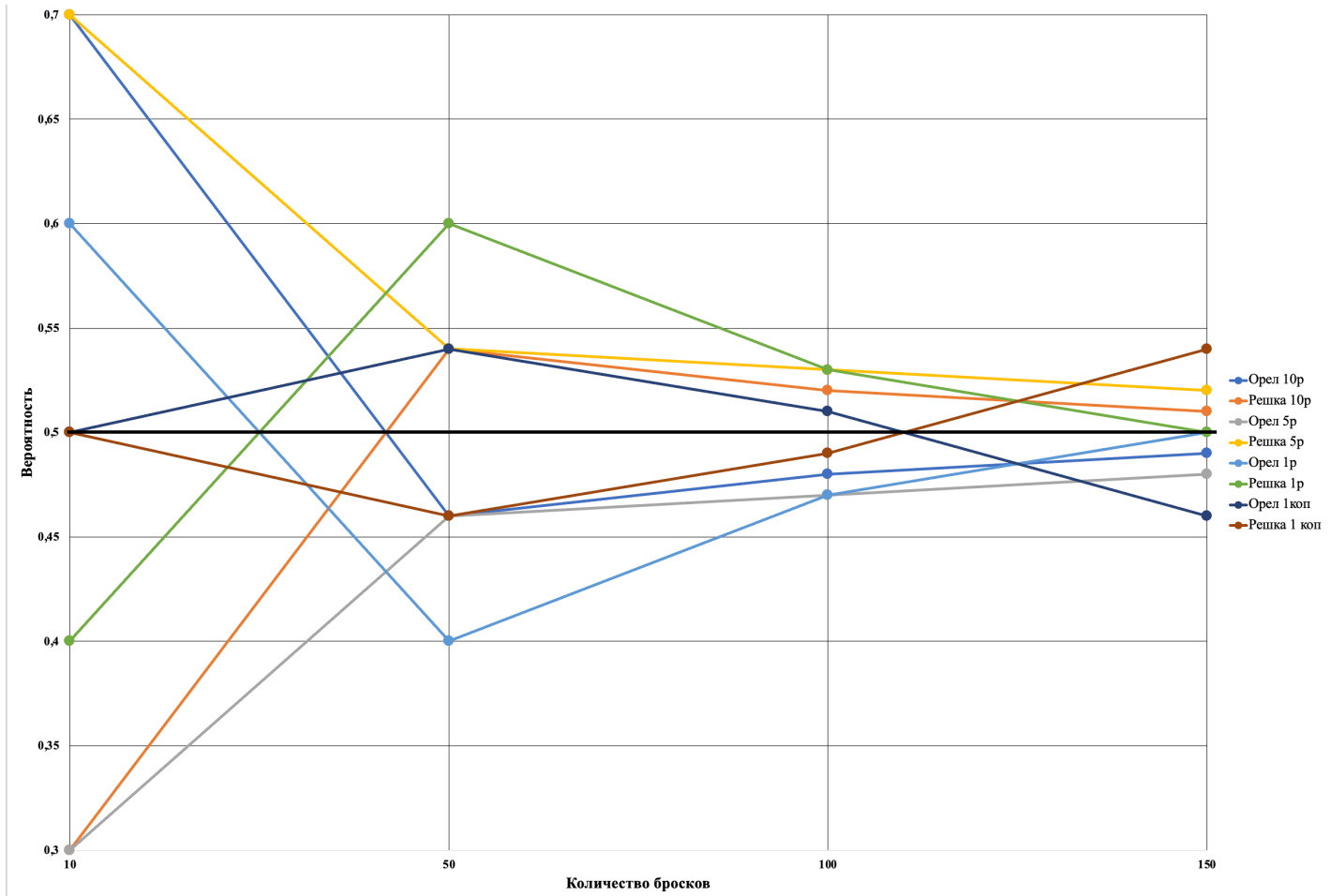


Рисунок 3.1. График зависимости вероятности выпадение орла или решки в зависимости от количества бросков

Подбрасывание монетки на практике демонстрирует, что теоретическая вероятность и экспериментальные результаты не всегда совпадают при малом числе испытаний, что видно в таблицах 3.1, 3.2 и 3.3, но при увеличении количества бросков наблюдается стремление к теоретической вероятности, как видно на рис 3.1., что подтверждает закон больших чисел.

### 3.2. Генератор псевдослучайных чисел на Python

Для моделирования подбрасывания монеты был создан простой генератор псевдослучайных чисел на языке Python [9,10]. В основе алгоритма лежит стандартный модуль random, который предоставляет функции для генерации случайных чисел различных типов.

В качестве начального значения генератора (seed) использовалось текущее системное время, что обеспечивает различный результат при каждом запуске программы. Такой подход является стандартным для большинства реализаций псевдослучайных генераторов.

Код генератора приведён в Приложении 1.

Количество бросков	Орел	Решка	P(Орел)	P(Решка)
10	3	7	0,3	0,7
50	21	29	0,42	0,58
100	46	54	0,46	0,54
150	70	80	0,47	0,53
500	253	247	0,51	0,49
1000	486	514	0,49	0,51

Таблица 3.5. Результаты моделирования подбрасывания монеты в среде Python.

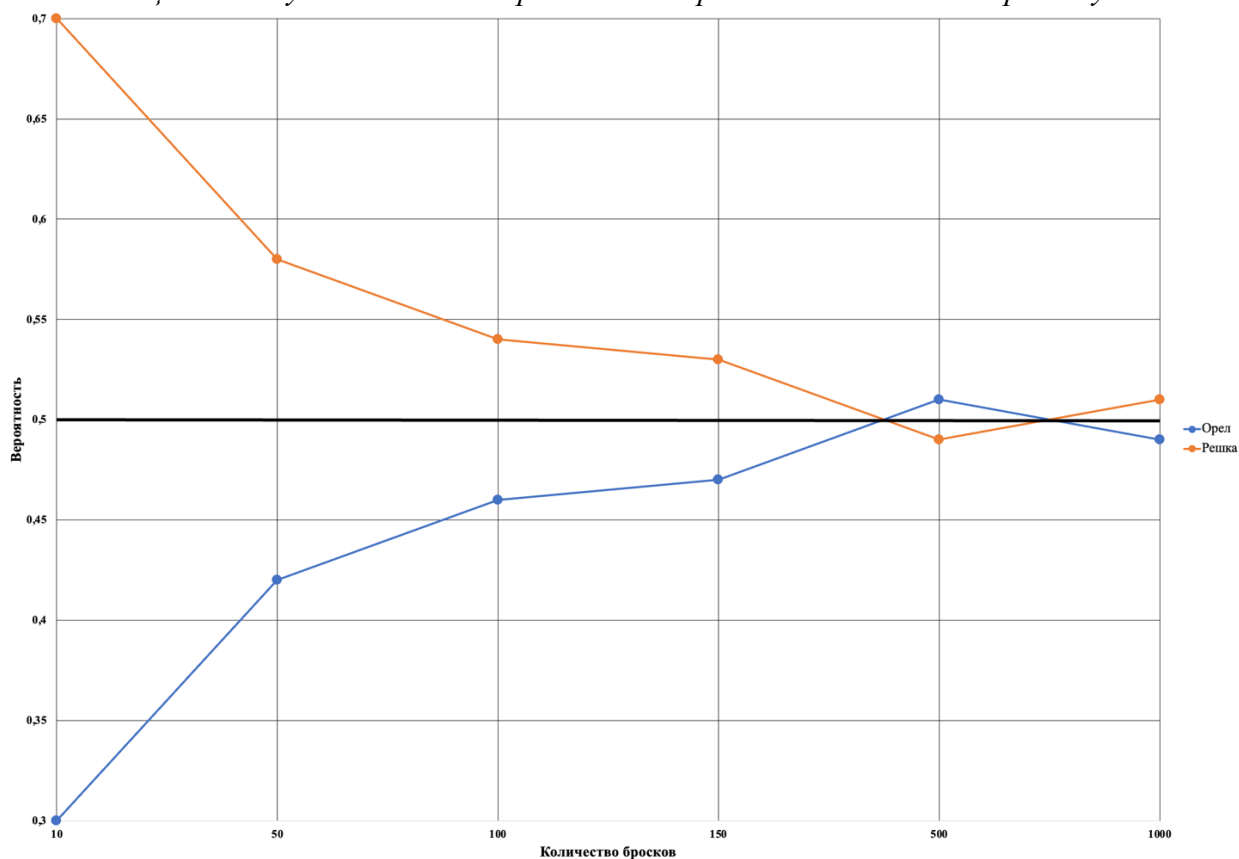


Рисунок 3.2. График зависимости вероятности выпадения орла или решки в зависимости от количества бросков, смоделированного в Python

Из полученных данных видно, что с увеличением числа экспериментов вероятность выпадения «орла» и «решки» стремится к 0,5, что подтверждает корректность работы генератора и его соответствие закону больших чисел.

### 3.3. Генератор случайных чисел на Arduino

В отличие от компьютеров, которые используют псевдослучайные числа, Arduino позволяет получать реально случайные числа с помощью физических процессов. В нашем случае

используется библиотека TrueRandom, которая генерирует случайные числа на основе электрического шума в микроконтроллере [12,13,14].

Каждое считывание шума превращается в 0 или 1, что имитирует подбрасывание монеты:

- 0 — Орёл
- 1 — Решка

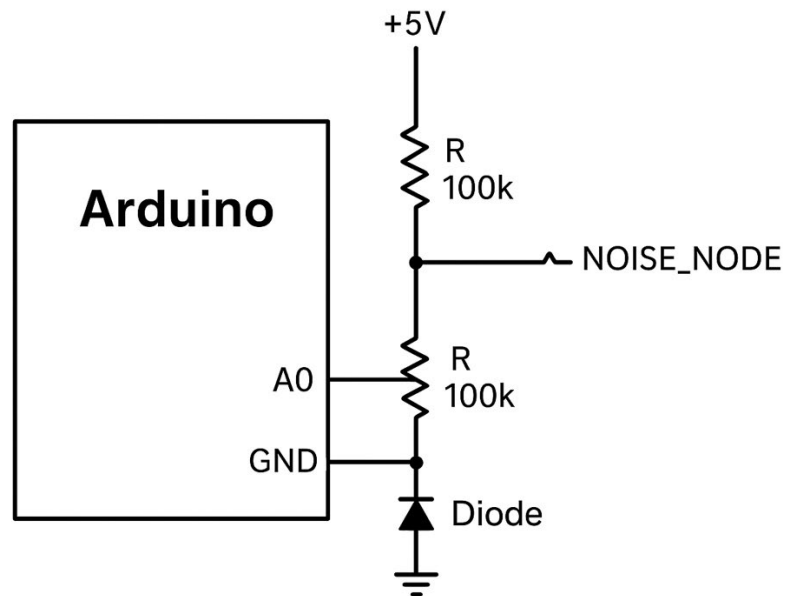


Рисунок 3.3. Электрическая схема для получения шума

Количество бросков	Орел	Решка	P(Орел)	P(Решка)
2300	1600	700	0,7	0,3
3500	2100	1400	0,6	0,4
4200	2300	1900	0,55	0,45
5000	2500	2500	0,5	0,5

Таблица 3.6. Результаты моделирования подбрасывания монеты с помощью генератора случайных чисел собранного на Arduino

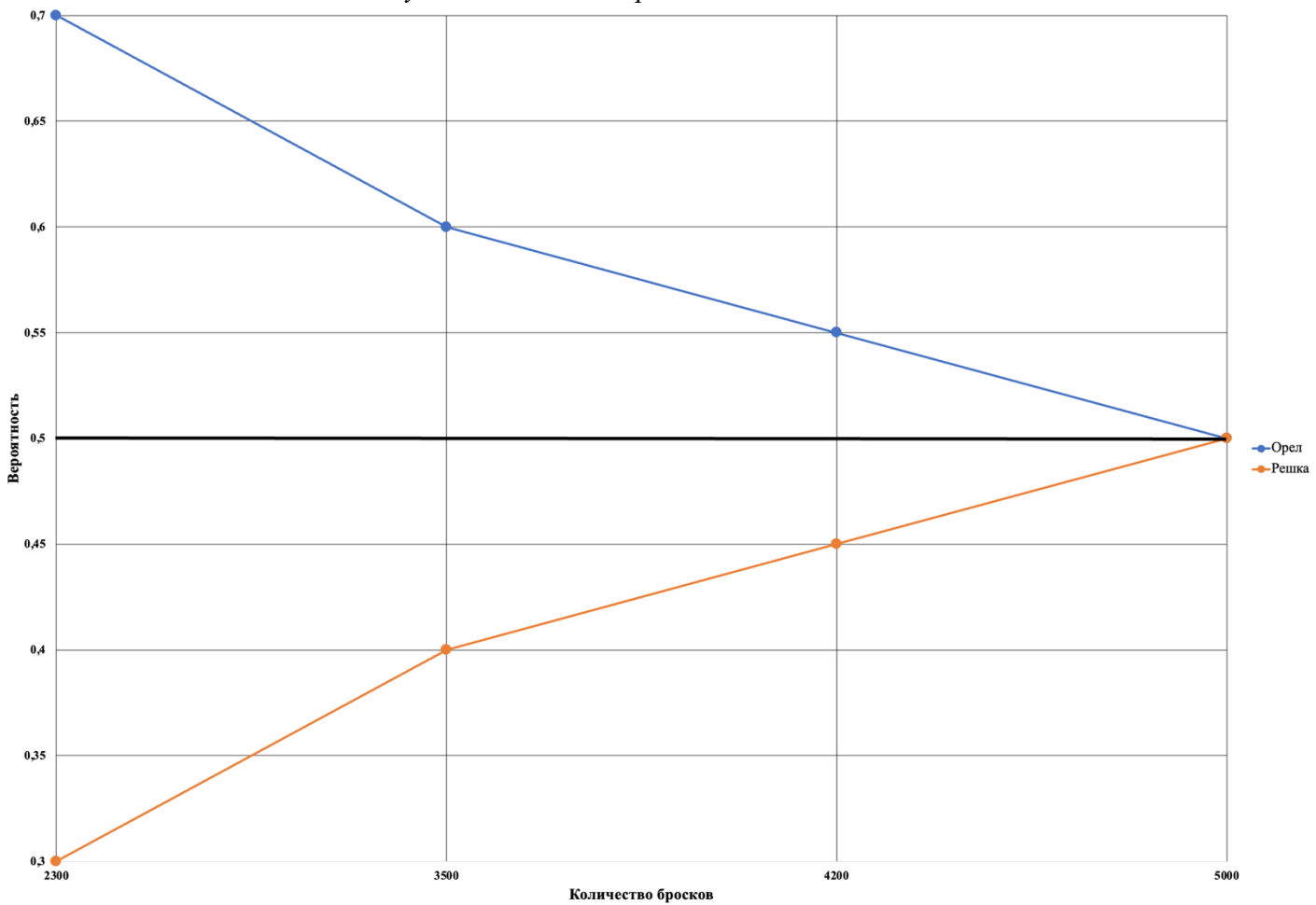


Рисунок 3.2. График зависимости вероятности выпадение орла или решки в зависимости от количества бросков, смоделированного с помощью генератора случайных чисел собранного на Arduino.

Физический генератор случайных чисел на Arduino позволяет на практике наблюдать настоящую случайность.

Эксперимент подтверждает, что с увеличением количества бросков относительные частоты орла и решки приближаются к теоретической вероятности 50%.

## Заключение

В ходе выполненного исследования были рассмотрены основные принципы теории вероятностей, изучены особенности генерации случайных и псевдослучайных чисел, а также проведены практические эксперименты, направленные на моделирование подбрасывания монеты. Полученные результаты позволяют сделать следующие выводы:

1. Теоретическая вероятность и экспериментальные данные совпадают при большом числе испытаний.

Практические эксперименты с реальными монетами, моделирование на Python и с генератором случайных чисел построенном на Arduino показали, что относительные частоты выпадения «орла» и «решки» стремятся к значению 0,5 при увеличении количества бросков. Это подтверждает действие закона больших чисел и демонстрирует его фундаментальную роль в статистике случайных процессов.

2. Псевдослучайные числа корректно моделируют случайные события.

Использование стандартного модуля random в Python дало результаты, аналогичные экспериментальному подбрасыванию монеты. Полученные данные показали высокую устойчивость алгоритма к статистическим отклонениям и соответствие теоретическим ожиданиям, что подтверждает его применимость для моделирования простых случайных процессов.

3. Физический генератор случайных чисел обеспечивает истинную случайность.

Генератор на базе Arduino, использующий электрический шум, продемонстрировал возможность получения непредсказуемых значений, основанных на природных процессах. Подобный подход важен для задач, требующих высокой степени энтропии (например, криптография, моделирование физических процессов).

4. Сравнение разных методов генерации случайных данных показало их общую тенденцию к теоретическому распределению.

Независимо от используемого способа — реальные броски монеты, генерация в Python или физический шум — при большом числе попыток распределение стремится к равновероятному, что подтверждает универсальность статистических законов.

5. Цель и задачи исследования достигнуты полностью.

В работе были рассмотрены теоретические основы, изучены два разных подхода к генерации случайных чисел, проведены серии экспериментов и выполнено сравнение результатов. Анализ подтвердил гипотезу о стремлении относительных частот к теоретической вероятности 0,5.

Личный вклад автора заключается в проведении экспериментальных измерений с монетами разных номиналов, написании и тестировании программы моделирования на Python, сборе данных с Arduino, их обработке, сравнительном анализе и формулировке выводов по результатам исследования.

Перспективы дальнейшей работы могут включать расширение количества экспериментов, сравнение различных алгоритмов генерации псевдослучайных чисел, исследование криптографических генераторов, а также моделирование более сложных случайных процессов.

## Список литературы

1. Гмурман, В. Е. Теория вероятностей и математическая статистика: учебное пособие для вузов / В. Е. Гмурман. — 10-е изд., перераб. — Москва: Высшая школа, 2003. — 479 с.
2. Колмогоров, А. Н. Основные понятия теории вероятностей / А. Н. Колмогоров. — 2-е изд. — Москва: Наука, 1974. — 120 с.
3. Феллер, В. Введение в теорию вероятностей и её приложения. В 2 т. Т. 1 / В. Феллер; пер. с англ. Ю. В. Прохоров. — 2-е изд. — Москва: Мир, 1967. — 498 с.
4. Чистяков, В. П. Курс теории вероятностей: учебник / В. П. Чистяков. — 6-е изд., изм. — Москва: ФИЗМАТЛИТ, 2003. — 240 с.
5. Боровков, А. А. Теория вероятностей: учебное пособие / А. А. Боровков. — 3-е изд., испр. — Москва: УРСС, 2003. — 472 с.
6. Кнут, Д. Э. Искусство программирования. Т. 2. Получисленные алгоритмы / Д. Э. Кнут; пер. с англ. И. В. Красильщик. — 3-е изд. — Москва: Вильямс, 2000. — 832 с.
7. Маршалл, В. Л. Теория и приложения генераторов случайных чисел / В. Л. Маршалл, Д. Д. Браун. — Москва: Наука, 1983. — 212 с.
8. Вентцель, Е. С. Теория вероятностей: учебник для вузов / Е. С. Вентцель. — 10-е изд., стер. — Москва: Высшая школа, 2001. — 575 с.
9. Лутц, М. Изучаем Python / М. Лутц; пер. с англ. А. М. Киселев. — 5-е изд. — Москва: О'Рейли, 2019. — 1216 с.
10. Свейгарт, Э. Автоматизация рутинных задач с помощью Python / Э. Свейгарт ; пер. с англ. А. Н. Киселев. — 2-е изд. — Москва: Вильямс, 2015. — 592 с.
11. Бхаргава, А. Грокаем алгоритмы: иллюстрированное пособие для программистов и любопытствующих / А. Бхаргава ; пер. с англ. А. А. Слинкин. — Санкт-Петербург: Питер, 2017. — 432 с.
12. Монк, С. Программирование Arduino. Второе издание / С. Монк; пер. с англ. — Москва : Альтшер, 2015. — 560 с.
13. Блюм, Д. Исследуем Arduino / Д. Блюм; пер. с англ. О. В. Полулях. — Санкт-Петербург

```
1 import random
2 import matplotlib.pyplot as plt
3
4 # Ввод количества подбрасываний
5 n = int(input("Сколько раз подбросить монету? "))
6
7 orel = 0
8 reshka = 0
9 p_orel_list = [] # для графика
10 p_reshka_list = []
11
12 for i in range(1, n+1):
13     result = random.choice(["орёл", "решка"])
14     if result == "орёл":
15         orel += 1
16     else:
17         reshka += 1
18     # Считаем относительные частоты после каждого броска
19     p_orel_list.append(orel/i)
20     p_reshka_list.append(reshka/i)
21
22 print(f"Орёл выпал {orel} раз, решка {reshka} раз.")
23 print(f"Доля орлов: {orel/n:.2f}, доля решек: {reshka/n:.2f}")
24
25 # Строим график
26 plt.plot(range(1, n+1), p_orel_list, label="Доля орлов")
27 plt.plot(range(1, n+1), p_reshka_list, label="Доля решек")
28 plt.axhline(0.5, color='gray', linestyle='--', label="Теоретическая вероятность")
29 plt.xlabel("Номер броска")
30 plt.ylabel("Доля выпадений")
31 plt.title("Моделирование подбрасывания монеты")
32 plt.legend()
33 plt.show()
```

Рис.1. Код псевдослучайного генератора в среде Python

```

1  #include <TrueRandom.h>
2
3  unsigned long countTotal = 0; // Общее количество бросков
4  unsigned long countHeads = 0; // Орёл
5  unsigned long countTails = 0; // Решка
6
7  void setup() {
8      Serial.begin(9600);
9      Serial.println("Генератор случайных бросков монетки (на основе TrueRandom)");
10     Serial.println("-----");
11     Serial.println("Орёл / Решка / P(Орёл) / P(Решка)");
12 }
13
14 void loop() {
15     // Генерируем случайное число 0 или 1
16     int coin = TrueRandom.random(0, 2);
17
18     countTotal++;
19
20     if (coin == 0) {
21         countHeads++;
22         Serial.print("Орёл");
23     } else {
24         countTails++;
25         Serial.print("Решка");
26     }
27
28     // Вычисляем вероятности
29     float pHeads = (float)countHeads / countTotal;
30     float pTails = (float)countTails / countTotal;
31
32     Serial.print(" | ");
33     Serial.print(countHeads);
34     Serial.print(" / ");
35     Serial.print(countTails);
36     Serial.print(" | P(Орёл) = ");
37     Serial.print(pHeads, 2);
38     Serial.print(" | P(Решка) = ");
39     Serial.println(pTails, 2);
40
41     delay(500); // Пауза между бросками
42 }

```

Рис.1. Код для анализа данных с генератора случайных чисел на базе Arduino в среде C++