Филиппова Екатерина ПИ21-7

**Тема**: Программная реализация игры Рулетка

**Задача**: Доказать, что казино будет всегда в выигрыше

**Команда**: Филиппова Екатерина, Николаенко Анастасия, Якубсон Татьяна

Для 100 итераций

import random  
  
initial\_amount = 1000 # Начальная сумма игрока  
total\_expected\_loss = 0  
bet = 1 # Фиксированный размер ставки  
  
for attempt in range(1, 101):  
 choice = random.randint(1, 3)  
  
 if choice == 1:  
 n = random.randint(1, 36)  
 winnings = 36 / n  
 probability\_win = 1 / n  
 probability\_loss = 1 - probability\_win  
  
 elif choice == 2:  
 n = 18  
 winnings = 36 / n  
 probability\_win = 1 / n  
 probability\_loss = 1 - probability\_win  
  
 elif choice == 3:  
 n = 12  
 r = random.randint(1, 2)  
 winnings = 36 / (n \* r)  
 probability\_win = 1 / (n \* r)  
 probability\_loss = 1 - probability\_win  
  
 # Ограничиваем максимальные потери текущего раунда оставшейся суммой  
 max\_loss = min(initial\_amount - total\_expected\_loss, bet)  
 expected\_loss = probability\_loss \* max\_loss  
 total\_expected\_loss += expected\_loss  
  
 # Рассчитываем остаток после каждой попытки  
 remaining\_amount = initial\_amount - total\_expected\_loss  
  
 print(f'На попытке {attempt}, размер ставки: {bet}, ожидаемые потери: {expected\_loss}, остаток: {remaining\_amount}, вероятность выигрыша: {probability\_win}, вероятность проигрыша: {probability\_loss}')  
  
print(f'Общие ожидаемые потери после 100 попыток: {total\_expected\_loss}')  
print(f'Остаток после 100 попыток: {remaining\_amount}')

Изображение выглядит как текст, снимок экрана, Шрифт, информация

Автоматически созданное описание

Для 1000 итераций

import random  
  
initial\_amount = 1000 # Начальная сумма игрока  
total\_expected\_loss = 0  
bet = 1 # Фиксированный размер ставки  
  
for attempt in range(1, 1001):  
 choice = random.randint(1, 3)  
  
 if choice == 1:  
 n = random.randint(1, 36)  
 winnings = 36 / n  
 probability\_win = 1 / n  
 probability\_loss = 1 - probability\_win  
  
 elif choice == 2:  
 n = 18  
 winnings = 36 / n  
 probability\_win = 1 / n  
 probability\_loss = 1 - probability\_win  
  
 elif choice == 3:  
 n = 12  
 r = random.randint(1, 2)  
 winnings = 36 / (n \* r)  
 probability\_win = 1 / (n \* r)  
 probability\_loss = 1 - probability\_win  
  
 # Ограничиваем максимальные потери текущего раунда оставшейся суммой  
 max\_loss = min(initial\_amount - total\_expected\_loss, bet)  
 expected\_loss = probability\_loss \* max\_loss  
 total\_expected\_loss += expected\_loss  
  
 # Рассчитываем остаток после каждой попытки  
 remaining\_amount = initial\_amount - total\_expected\_loss  
  
 print(f'На попытке {attempt}, размер ставки: {bet}, ожидаемые потери: {expected\_loss}, остаток: {remaining\_amount}, вероятность выигрыша: {probability\_win}, вероятность проигрыша: {probability\_loss}')  
  
print(f'Общие ожидаемые потери после 1000 попыток: {total\_expected\_loss}')  
print(f'Остаток после 1000 попыток: {remaining\_amount}')

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Для 10000 итераций

import random  
  
initial\_amount = 1000 # Начальная сумма игрока  
total\_expected\_loss = 0  
bet = 1 # Размер ставки  
  
for attempt in range(1, 10001):  
 choice = random.randint(1, 3)  
  
 if choice == 1:  
 n = random.randint(1, 36)  
 winnings = 36 / n  
 probability\_win = 1 / n  
 probability\_loss = 1 - probability\_win  
  
 elif choice == 2:  
 n = 18  
 winnings = 36 / n  
 probability\_win = 1 / n  
 probability\_loss = 1 - probability\_win  
  
 elif choice == 3:  
 n = 12  
 r = random.randint(1, 2)  
 winnings = 36 / (n \* r)  
 probability\_win = 1 / (n \* r)  
 probability\_loss = 1 - probability\_win  
  
 expected\_loss = probability\_loss \* bet  
 total\_expected\_loss += expected\_loss  
  
 # Рассчитываем остаток после каждой попытки  
 remaining\_amount = initial\_amount - total\_expected\_loss  
  
 print(f'На попытке {attempt}, размер ставки: {bet}, ожидаемые потери: {expected\_loss}, остаток: {remaining\_amount}')  
  
print(f'Общие ожидаемые потери после 10000 попыток: {total\_expected\_loss}')  
print(f'Остаток после 10000

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Для 100000 итераций

import random  
  
initial\_amount = 1000 # Начальная сумма игрока  
total\_expected\_loss = 0  
bet = 1 # Фиксированный размер ставки  
  
for attempt in range(1, 10001):  
 choice = random.randint(1, 3)  
  
 if choice == 1:  
 n = random.randint(1, 36)  
 winnings = 36 / n  
 probability\_win = 1 / n  
 probability\_loss = 1 - probability\_win  
  
 elif choice == 2:  
 n = 18  
 winnings = 36 / n  
 probability\_win = 1 / n  
 probability\_loss = 1 - probability\_win  
  
 elif choice == 3:  
 n = 12  
 r = random.randint(1, 2)  
 winnings = 36 / (n \* r)  
 probability\_win = 1 / (n \* r)  
 probability\_loss = 1 - probability\_win  
  
 # Ограничиваем максимальные потери текущего раунда оставшейся суммой  
 max\_loss = min(initial\_amount - total\_expected\_loss, bet)  
 expected\_loss = probability\_loss \* max\_loss  
 total\_expected\_loss += expected\_loss  
  
 # Рассчитываем остаток после каждой попытки  
 remaining\_amount = initial\_amount - total\_expected\_loss  
  
 print(f'На попытке {attempt}, размер ставки: {bet}, ожидаемые потери: {expected\_loss}, остаток: {remaining\_amount}, вероятность выигрыша: {probability\_win}, вероятность проигрыша: {probability\_loss}')  
  
print(f'Общие ожидаемые потери после 10000 попыток: {total\_expected\_loss}')  
print(f'Остаток после 10000 попыток: {remaining\_amount}')

Изображение выглядит как текст, снимок экрана, Шрифт, информация

Автоматически созданное описание

**Выводы**

* Для 100 шагов:

При начальной сумме в размере 1000, игрок не теряет деньги.

* Для 1000 шагов:

При увеличении числа попыток, общие ожидаемые потери увеличиваются, подтверждая, что игрок в среднем теряет деньги. Например, при начальной сумме в размере 1000 при 1000 шагов, то игрок в минусе уже будет на 108 попытке.

* Для 10000 шагов:

При увеличении числа попыток, общие ожидаемые потери также увеличиваются, подтверждая, что игрок в среднем теряет деньги. Например, при начальной сумме в размере 1000 при 10000 шагов, то игрок в минусе уже будет на 1091 попытке.

Игрок проигрывает в каждой из ситуаций в долгосрочной перспективе, а казино всегда остаётся в выигрыше.