МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«Ймовірнісні основи програмної інженерії»

Лабораторна робота № 1
«Центральні тенденції та міра дисперсії»

Виконав:	Сирота Ангеліна Олександрівна	**Перевірила**:	Вечерковська Анастасія Сергіївна
Група	ІПЗ-21	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		

2022

Мета – навчитись використовувати на практиці набуті знання про центральні тенденції та міри.

Хід роботи

Постановка задачі:

Написати програму, що зчитує дані з файла і виконує наступні функції:

- Побудувати таблицю частот та сукупних частот для переглянутих фільмів. Визначити фільм, який був переглянутий частіше за інші.
- Знайти Моду та Медіану заданої вибірки.
- Порахувати Дисперсію та Середнє квадратичне відхилення розподілу.
- Побудувати гістограму частот для даного розподілу.

Усі результати записувати в окремий текстовий файл

Побудова математичної моделі:

Частота: кількість переглядів фільму, міститься у вхідному файлі

Сукупна частота: сума попередніх значень частот

Мода: значення, що має найбільшу частоту

- Якщо частоти усіх елементів дорівнюють 1, то моди немає.
- Якщо декілька елементів мають найбільшу частоту, то модами будуть ці елементи

Медіана: центральне значення. Знаходиться за формулою:

- Якщо парна кількість елементів у вибірці: $\frac{x_{\frac{n}{2}}+x_{\left(\frac{n}{2}+1\right)}}{2}$
- Якщо непарна кількість елементів у вибірці: $x_{\frac{n+1}{2}}$

  де n – кількість елементів у вибірці

Середнє $\bar{x}$ обчислюється за формулою: $\bar{x} = \frac{x_1 f_1 + x_2 f_2 + \ldots + x_n f_n}{f_1 + f_2 + \ldots + f_n}$

Дисперсія рахується за формулою: $Var(x) = \frac{\sum_{x \in x} f_x (x - \bar{x})^2}{\sum_{x \in x} f_x}$

Середнє квадратичне відхилення обчислюється за формулою: $\sigma = \sqrt{Var(x)}$

Гістограма складається з прямокутників, де ширина є фільмом / інтервалом, у якому розташовані фільми, а висота є кількістю переглядів. У випадку з інтервалами кількість переглядів є сумарною для кожного фільма з цього інтервалу.

Псевдокод алгоритму:

Заповнення масиву значеннями елементів і їх частот:

```
data = [for item in f] // записуємо дані з файла у масив
   for el in data:
      flag = False                  // якщо елемент ще не повторювався
      if arrR != 0:            // перевірити, чи міститься поточний елемент у масиві
         for i in arrR:             // повторюваних значень
            if el == i:
               flag = True
      if flag == False:           // якщо не міститься
         for el1 in data:
            if el == el1:
               k += 1             // частота
         arrfF.extend([[el, k, 0]])
         arrR.append(el)       // додати оброблений елемент
      flag == False              // у масив повторюваних значень
   arrfF = sorted(arrfF)       // відсортувати масив
   Cf(arrfF)                   // знайти сукупну частоту
   return arrfF
```

Знаходження сукупної частоти:

```
cf = 0                         // сукупна частота
   for i in range(arrfF):
      cf += arrfF[i][1]       // додавати частоту
      arrfF[i][2] = cf
```

Визначення фільму, який був переглянутий частіше за інших:

```
for i in range(arrfF):
      if arrfF[i][1] > max:    // max - максимальна частота
         max = arrfF[i][1]
         index = i             // індекс фільму
   print(index + 1)
```

Визначення моди:

```
for i in range(arrfF):
      if arrfF[i][1] > frmax:          // frmax - максимальна частота
         frmax = arrfF[i][1]
   for i in range(arrfF):
      if arrfF[i][1] == frmax:
         fmax.append(i + 1) 0       // fmax – масив елементів з максимальними
частотами
   if frmax == 1:
      print("Моди немає")
      fileOutput.write("Моди немає")
   else:
      for i in range(fmax):
         print(fmax[i])
```

Визначення медіани:

```
if n % 2 == 0:                          // парна кількість елементів
    index = n / 2 - 1
    median = (arr[index] + arr[index + 1]) / 2

else:
    index = (n + 1) // 2 - 1
    median = arr[index]
```

Визначення середнього значення:

```
average(arrfF):
    for i in range(arrfF):
        numerator += arrfF[i][0] * arrfF[i][1] // чисельник
        denominator += arrfF[i][1]             // знаменник
    Xave = numerator / denominator             // середнє значення
    return Xave
```

Визначення дисперсії та середнього квадратичного відхилення:

```
def dispersion(arrfF):
    Xave = average(arrfF)
    for i in range(arrfF):
        numerator += arrfF[i][1] * math.pow(arrfF[i][0] - Xave, 2) // чисельник
        denominator += arrfF[i][1]                                 // знаменник
    dis = numerator / denominator                                  // дисперсія
    print(dis)

    msd = sqrt(dis)                                                // середнє квадратичне відхилення
    print(msd)
```

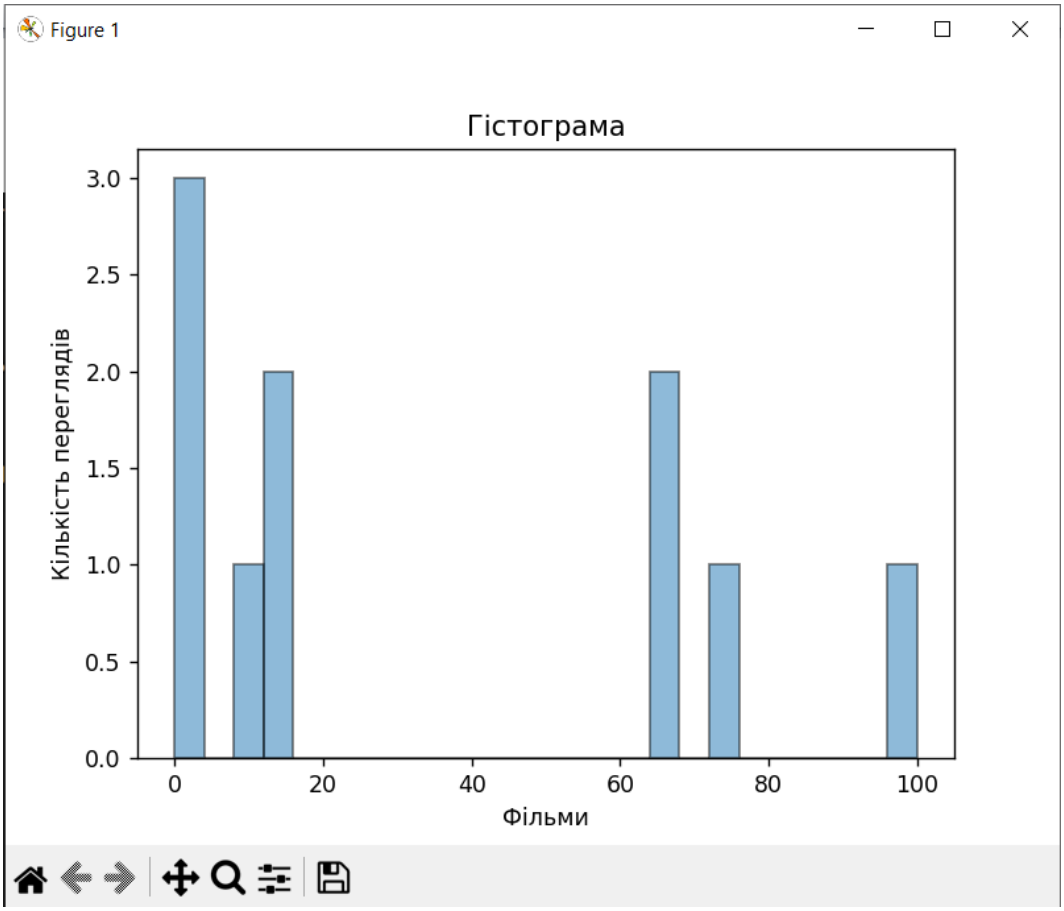
Побудова гістограми:

```
p = 0                                       // параметр для інтервалів
interval = int(round(MaxEl(arrfF) / 25))

# інтервали
while (arrInt <= MaxEl(arrfF) / interval + 1):
    arrInt.append(p)
    p += interval


pyplot.hist(arr, arrInt, edgecolor = 'k', alpha = 0.5)
```

Випробування алгоритму:

Набір з 10 фільмів:

```
Введіть значення кількості елементів у вхідному файлі (10/100/1000):
10
------------------------------------------------------------

Фільм з максимальною кількістю переглядів( 3 ) : 1

--------- Таблиця частот ----------
Елемент | Частота | Сукупна частота
--------+---------+----------------
1       | 3       | 3
10      | 1       | 4
12      | 2       | 6
66      | 2       | 8
75      | 1       | 9
97      | 1       | 10

Мода: 1
Медіана: 49

Дисперсія: 1251
Середнє квадратичне відхилення: 35
```

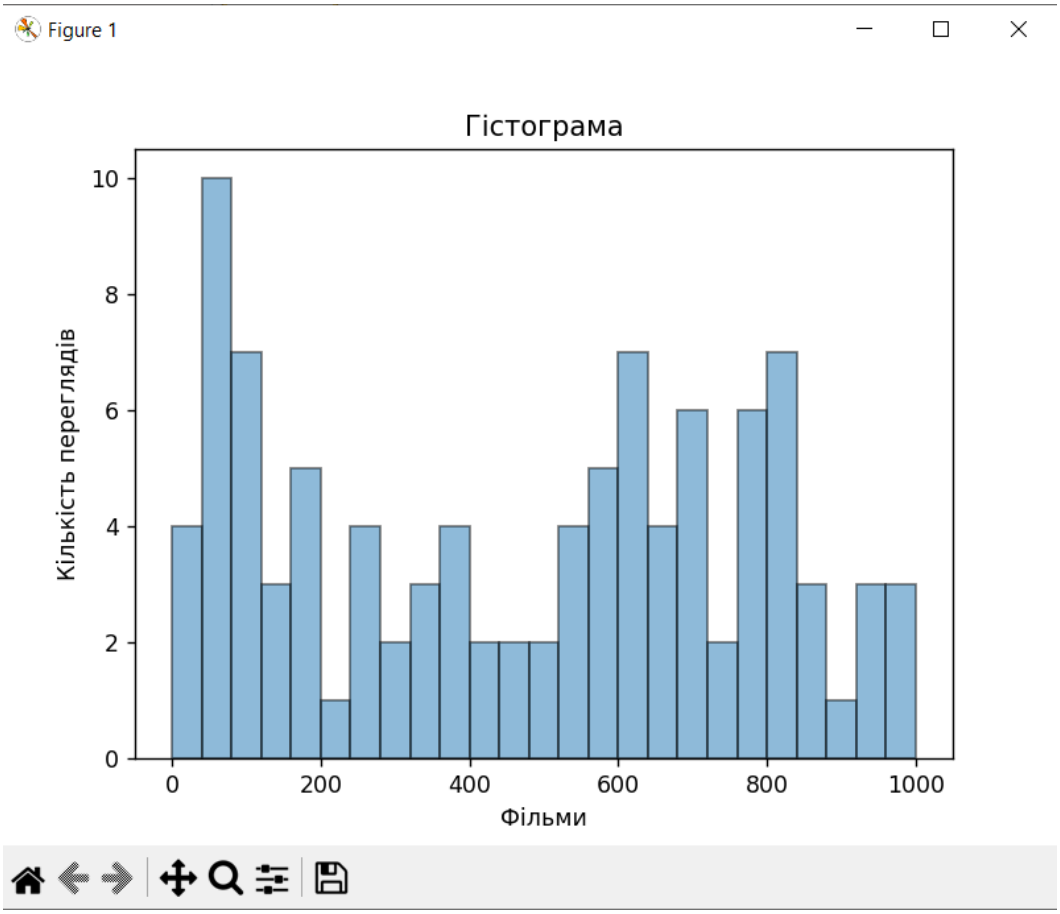


Набір зі 100 фільмів:

```
Введіть значення кількості елементів у вхідному файлі (10/100/1000):
100
-----------------------------------------------------------

Фільм з максимальною кількістю переглядів( 4 ) : 22

--------- Таблиця частот ----------
Елемент | Частота | Сукупна частота
--------+---------+----------------
22      | 4       | 4
46      | 2       | 6
47      | 1       | 7
51      | 3       | 10
71      | 1       | 11
77      | 1       | 12
79      | 2       | 14
80      | 1       | 15
91      | 1       | 16
97      | 1       | 17
99      | 1       | 18
103     | 1       | 19
119     | 2       | 21
146     | 1       | 22
147     | 1       | 23
154     | 1       | 24
162     | 2       | 26
168     | 1       | 27
193     | 1       | 28
198     | 1       | 29
225     | 1       | 30
250     | 1       | 31
251     | 1       | 32
255     | 1       | 33
269     | 1       | 34
```

```
269     | 1       | 34
288     | 1       | 35
317     | 1       | 36
354     | 1       | 37
355     | 1       | 38
359     | 1       | 39
361     | 1       | 40
362     | 1       | 41
382     | 1       | 42
384     | 1       | 43
414     | 1       | 44
429     | 1       | 45
447     | 1       | 46
450     | 1       | 47
498     | 1       | 48
503     | 1       | 49
529     | 1       | 50
535     | 1       | 51
548     | 1       | 52
553     | 1       | 53
566     | 1       | 54
569     | 1       | 55
571     | 1       | 56
587     | 1       | 57
589     | 1       | 58
607     | 2       | 60
612     | 2       | 62
613     | 1       | 63
615     | 1       | 64
636     | 1       | 65
642     | 1       | 66
657     | 1       | 67
660     | 1       | 68
676     | 1       | 69
685     | 1       | 70
687     | 1       | 71
688     | 1       | 72
694     | 1       | 73
702     | 1       | 74
707     | 1       | 75
736     | 1       | 76
```

```
738     | 1       | 77
763     | 1       | 78
768     | 1       | 79
775     | 1       | 80
777     | 1       | 81
782     | 1       | 82
784     | 1       | 83
813     | 1       | 84
817     | 1       | 85
820     | 1       | 86
821     | 1       | 87
824     | 1       | 88
832     | 1       | 89
834     | 1       | 90
858     | 1       | 91
878     | 1       | 92
879     | 1       | 93
880     | 1       | 94
923     | 1       | 95
928     | 1       | 96
945     | 1       | 97
976     | 1       | 98
984     | 1       | 99
999     | 1       | 100

Мода: 1
Медіана: 510

Дисперсія: 89013
Середнє квадратичне відхилення: 298
```

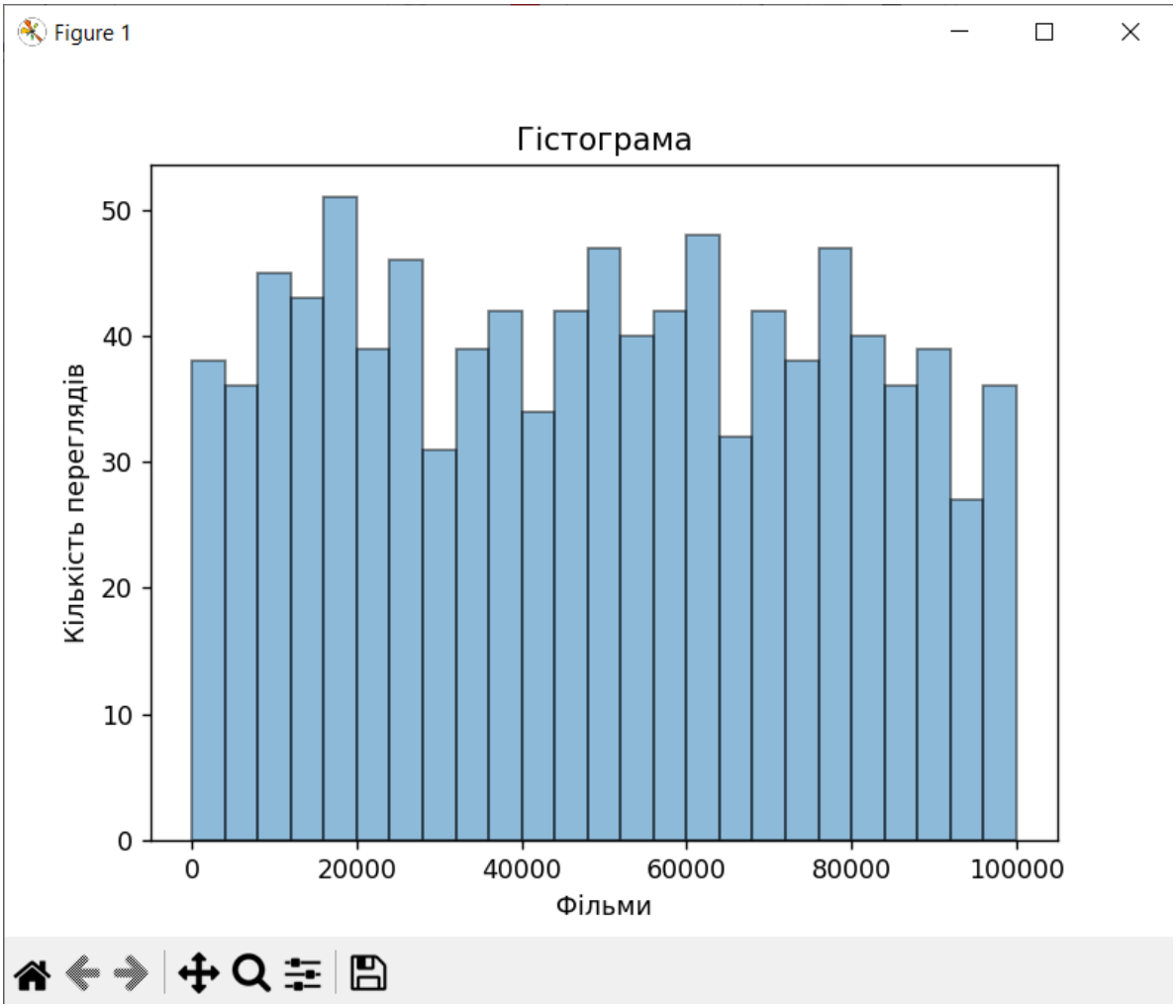


Набір із 1000 фільмів:

```
Фільм з максимальною кількістю переглядів( 2 ) : 14023

--------- Таблиця частот ----------
Елемент | Частота | Сукупна частота
--------+---------+----------------
48      | 1       | 1
80      | 1       | 2
113     | 1       | 3
129     | 1       | 4
300     | 1       | 5
480     | 1       | 6
645     | 1       | 7
729     | 1       | 8
820     | 1       | 9
896     | 1       | 10
930     | 1       | 11
981     | 1       | 12
1152    | 1       | 13
1210    | 1       | 14
1568    | 1       | 15
1723    | 1       | 16
1941    | 1       | 17
1975    | 1       | 18
2178    | 1       | 19
2237    | 1       | 20
2317    | 1       | 21
2352    | 1       | 22
2374    | 1       | 23
2505    | 1       | 24
2626    | 1       | 25
2694    | 1       | 26
2722    | 1       | 27
2723    | 1       | 28
2876    | 1       | 29
3008    | 1       | 30
```

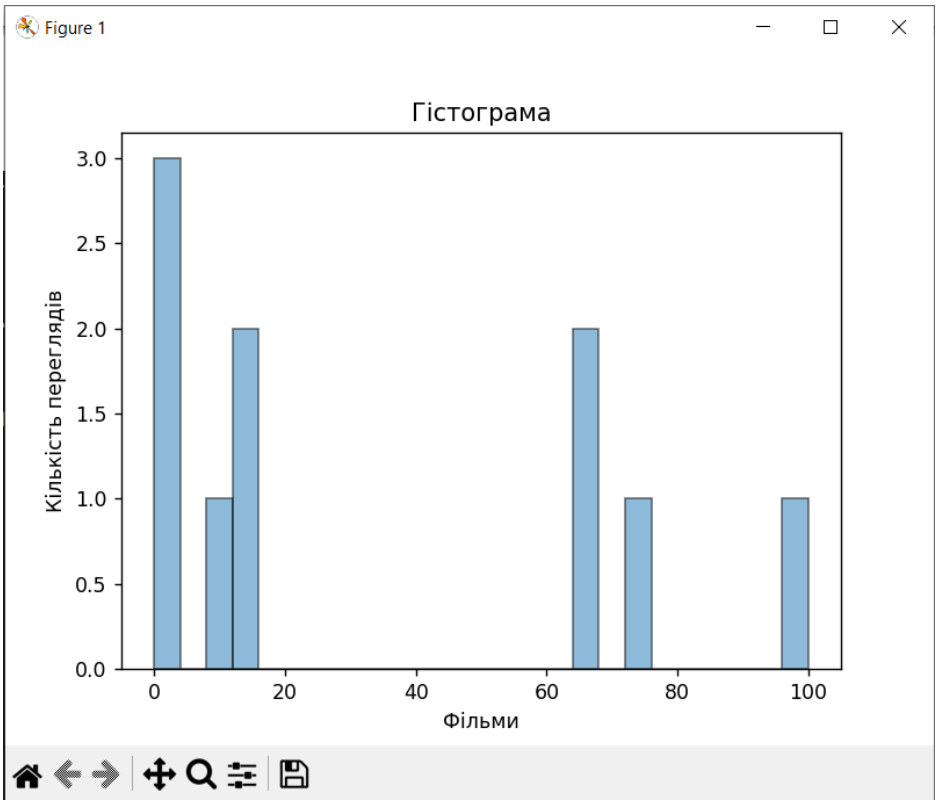
```
39125   | 1       | 402
39172   | 1       | 403
39311   | 1       | 404
39425   | 1       | 405
39549   | 1       | 406
39651   | 1       | 407
39709   | 1       | 408
39724   | 1       | 409
39725   | 1       | 410
40177   | 1       | 411
40218   | 1       | 412
40345   | 1       | 413
40363   | 1       | 414
40425   | 1       | 415
40507   | 1       | 416
40519   | 1       | 417
40617   | 2       | 419
40655   | 1       | 420
40782   | 1       | 421
40814   | 1       | 422
40913   | 1       | 423
41043   | 1       | 424
41293   | 1       | 425
41392   | 1       | 426
41402   | 1       | 427
41562   | 1       | 428
41770   | 1       | 429
41856   | 1       | 430
41886   | 1       | 431
41976   | 1       | 432
42043   | 1       | 433
42076   | 1       | 434
42220   | 1       | 435
42226   | 1       | 436
42778   | 1       | 437
43264   | 1       | 438
43306   | 1       | 439
43483   | 1       | 440
43730   | 1       | 441
43888   | 1       | 442
43913   | 1       | 443
```

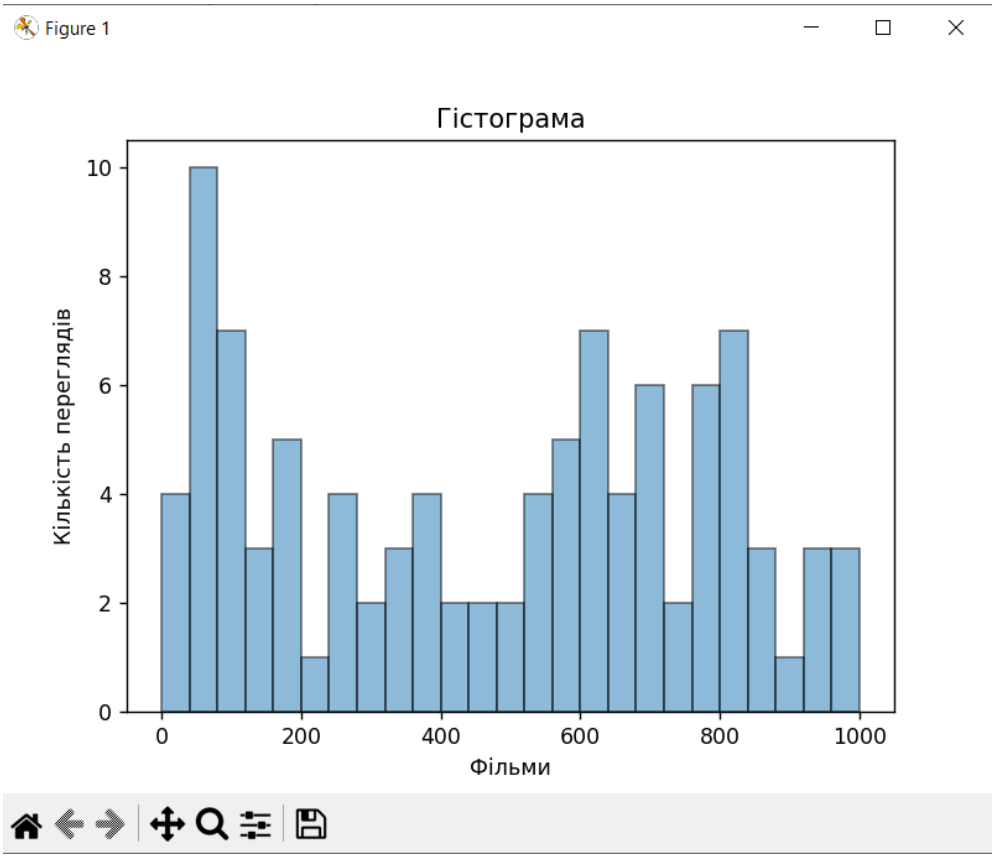
```
96701   | 1         | 972
97399   | 1         | 973
97523   | 1         | 974
97871   | 1         | 975
97990   | 1         | 976
97991   | 1         | 977
98046   | 1         | 978
98228   | 1         | 979
98249   | 1         | 980
98345   | 1         | 981
98419   | 1         | 982
98521   | 1         | 983
98668   | 1         | 984
98728   | 1         | 985
98790   | 1         | 986
99024   | 1         | 987
99172   | 1         | 988
99189   | 1         | 989
99193   | 1         | 990
99246   | 1         | 991
99256   | 1         | 992
99272   | 1         | 993
99403   | 1         | 994
99575   | 1         | 995
99696   | 1         | 996
99808   | 1         | 997
99820   | 1         | 998
99968   | 1         | 999
99970   | 1         | 1000

Мода: 14023 40617 93548
Медіана: 50009

Дисперсія: 801811587
Середнє квадратичне відхилення: 28316
```



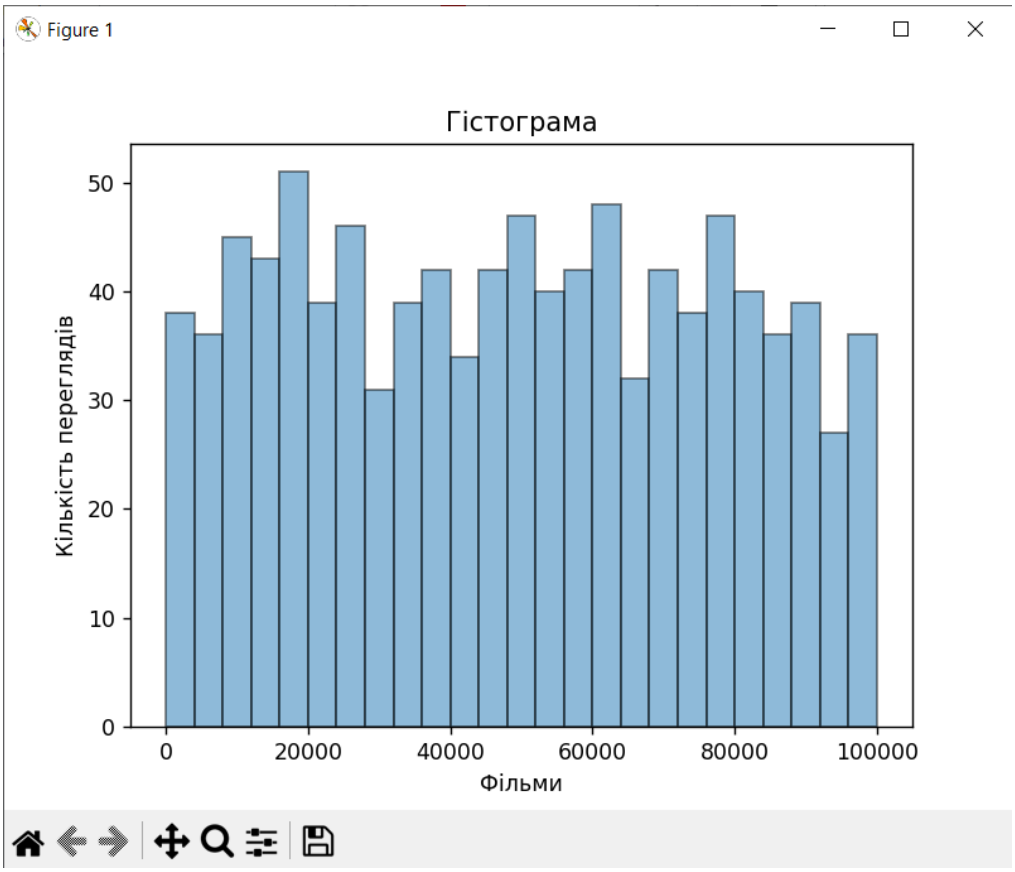
Висновки з вигляду гістограми, про закон розподілу:

Дана гістограма не є симетричною. Складається з 5 кластерів і 4 прогалин. Даний результат зумовлений розбиттям на інтервали в залежності від значень фільмів.





Дана гістограма не є симетричною. Складається з одного кластеру. Даний результат зумовлений розбиттям на інтервали в залежності від значень фільмів.



Дана гістограма не є симетричною. Складається з одного кластеру.

Висновок: в ході цієї лабораторної роботи було побудовано таблицю частот та сукупних частот для переглянутих фільмів, визначено фільм, який був переглянутий частіше за інші, знайдено моду та медіану заданої вибірки, пораховано дисперсію та середнє квадратичне відхилення, побудовано гістограму частот за допомогою бібліотеки matplotlib і проаналізовано отримані діаграми. Вони не є симетричними і складаються з одного кластеру, крім гістограми на 10 фільмів. Вона складається з 5 кластерів і 4 прогалин. Такі результати зумовлені розбиттям на інтервали в залежності від значень фільмів. Отримані результати внесено у новий текстовий файл.