

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

Виконав:	Коваленко Владислав Олександрович	Перевірила:	Марцафей А. С.
Група	ІПЗ-22(2)	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

Назва роботи: Центральні тенденції та міра дисперсії

Мета: Навчитись використовувати на практиці набуті знання про центральні тенденції та міри.

Завдання №1:

Постановка задачі:

Побудувати таблицю частот та сукупних частот для переглянутих фільмів.
Визначити фільм, який був переглянутий частіше за інші.

Побудова математичної моделі:

Для розв'язання даної задачі нам потрібно порахувати кількість фільмів, для кожного числа переглядів, для цього скористуємося циклом і запишемо дані в двумірний масив, одночасно в цьому циклі ми шукаємо найбільшу кількість переглядів, порівнюючи на кожній ітерації значення максимального елемента з поточним і якщо поточний елемент більший за попередній максимальний, змінюємо значення максимального елемента на поточний елемент в масиві. Виконавши ці дії запишемо наші дані в таблицю частот, а також порахуємо сукупні частоти, додаючи до переднього значення частоти наступну і так далі. І в кінці запишемо індекс і кількість переглядів фільму, який найчастіше переглядали.

Псевдокод алгоритму:

```
74 static void Task1(int[] mas, StreamWriter print)
75 {
76     print.WriteLine("=====Task 1=====");
77     print.WriteLine("          Table of frequencies          ");
78     print.WriteLine("=====");
79     print.WriteLine("n" + "\t=" + '\t' + "Value" + "\t=" + '\t' + "Frequency" + "\t=" + '\t' + "Cumulative Frequency" + "\t=");
80     int counter1 = 1;
81     int counter2 = 1;
82     int counter3 = 1;
83     int counter4 = 0;
84     int counter5 = 0;
85     int maxelement = 0;
86     int indexmaxelement = 0;
87
88     for (int i = 0; i < mas.Length; i++)
89     {
90         if (mas[i] > maxelement)
91         {
92             maxelement = mas[i];
93             indexmaxelement = i;
94         }
95         for (int j = 1; j < mas.Length; j++)
96         {
97             if (mas[i] == mas[j])
98             {
99                 counter1++;
100             }
101         }
102         counter5 += counter1;
103         if (i > 0)
104         {
105             for (int h = 0; h < counter2 - 1; h++)
106             {
107                 if (mas[i] == mas[h])
108                 {
```

```

102 counter5 += counter1;
103 if (i > 0)
104 {
105     for (int h = 0; h < counter2 - 1; h++)
106     {
107         if (mas[i] == mas[h])
108         {
109             counter4++;
110         }
111     }
112 }
113
114
115
116 if (counter4 == 0)
117 {
118     print.WriteLine(counter3 + ") " + "\t=" + "\t" + mas[i] + "\t=" + "\t" + counter1 + "\t=" + "\t" + "\t" + counter5 + "\t=");
119 }
120 else
121 {
122     counter3--;
123 }
124
125 else
126 {
127     print.WriteLine(counter3 + ") " + "\t=" + "\t" + mas[i] + "\t=" + "\t" + counter1 + "\t=" + "\t" + "\t" + counter5 + "\t=");
128 }
129 counter1 = 0;
130 counter2++;
131 counter3++;
132 counter4 = 0;
133 }
134 print.WriteLine("Найчастіше перегладали фільм під індексом - " + (indexmaxelement+1) + " ,а саме " + maxelement + " разів");
135 print.WriteLine("=====");
136 }
137

```

Випробування алгоритму:

=====Task 1=====

Table of frequencies

=====							
№	=	Value	=	Frequency	=	Cumulative Frequency	=
1)	=	1	=	3	=	3	=
2)	=	66	=	2	=	5	=
3)	=	75	=	1	=	6	=
4)	=	12	=	2	=	12	=
5)	=	10	=	1	=	13	=
6)	=	97	=	1	=	14	=
Найчастіше перегладали фільм під індексом - 7 ,а саме 97 разів							
=====							

Завдання 2:

Постановка задачі:

Знайти Моду та Медіану заданої вибірки.

Побудова математичної моделі:

Для того ,щоб знайти Моду вибірки потрібно знайти кількість переглядів,що зустрічаються найчастіше.Якщо частота всіх елементів дорівнює одиниці то моди немає,якщо частота два і більше то мода може бути як в єдиному екземплярі так і в багатьох. Дня цього знайдемо найбільшу частоту і якщо вона більше одиниці запишемо це значення до Моду ,а потім перевіримо чи є такі ж значення частот ще,якщо є допишемо до моди ці значення. Для того,щоб знайти медіану потрібно відсортувати значення в порядку зростання. Потім якщо кількість елементів не парна,то медіана дорівнює значенню під індексом $(n+1)/2$, де n-це кількість елементів. Якщо кількість елементів

парна ,то медіана дорівнює половині суми значень під індексами $n/2$ і $(n+1)/2$.

Псевдокод алгоритму:

```
138 static void Task2(int[] mas,ref int[,] mas1, StreamWriter print,ref int counter3)
139 {
140     print.WriteLine("=====Task 2=====");
141     int counter1 = 1;
142     int counter2 = 0;
143
144     int counter4 = 0;
145
146
147     for (int i = 0; i < mas.Length; i++)
148     {
149         for (int j = 1; j < mas.Length; j++)
150         {
151             if (mas[i] == mas[j])
152             {
153                 counter1++;
154             }
155         }
156         if (i > 0)
157         {
158             for (int h = 0; h < counter2; h++)
159             {
160                 if (mas[i] == mas[h])
161                 {
162                     counter4++;
163                 }
164             }
165
166             if (counter4 == 0)
167             {
168                 mas1[counter3, 0] = mas[i];
169                 mas1[counter3, 1] = counter1;
170             }
171             else
172             {
173                 counter3--;
174             }
175         }
176         else
177         {
178             mas1[counter3, 0] = mas[i];
179             mas1[counter3, 1] = counter1;
180         }
181         counter1 = 0;
182         counter2++;
183         counter3++;
184         counter4 = 0;
185     }
186
187     BubbleSort(ref mas1, counter3);
188
189
190     Console.WriteLine("=====");
191     print.WriteLine("Мода = " + mas1[0, 0] + " Кількість - " + mas1[0, 1]);
192     for (int i = 1; i < counter3; i++)
193     {
194         if (mas1[0, 1] == mas1[i, 1])
195         {
196             print.WriteLine("Мода = " + mas1[i, 0] + " Кількість - " + mas1[i, 1]);
197         }
198     }
199
200     mas = BubbleSort2(mas);
201     if (mas.Length % 2 == 0)
202     {
203         int temp = (mas.Length+1)/2;
204         print.WriteLine("Медіана = " + mas[temp]);
205     }
206 }
```

```

206         print.WriteLine("Медіана = " + mas[temp]);
207     }
208     else
209     {
210
211         print.WriteLine("Медіана = " + (mas[mas.Length / 2] + mas[(mas.Length / 2 + 1)]/2) ;
212     }
213 }
214

```

Випробування алгоритму:

=====Task 2=====

Мода = 1 Кількість - 3

Медіана = 12

Завдання 3:

Постановка задачі:

Порахувати Дисперсію та Середнє квадратичне відхилення розподілу.

Побудова математичної моделі:

$$\sigma_j^2 = \frac{\sum_{i=1}^n (x_i - \bar{x}_j)^2}{n}$$

Для обчислення Дисперсії ми скористаємося формулою $\sigma_j^2 = \frac{\sum_{i=1}^n (x_i - \bar{x}_j)^2}{n}$. Для цього потрібно обчислити за допомогою циклу середнє арифметичне та за допомогою циклу обчислити значення по формулі і поділити на кількість елементів. Для того ,щоб знайти Середнє квардратичне відхилення розподілу потрібно знайти корінь квадратний з дисперсії.

Псевдокод алгоритму:

```

215     ссылка: 1
216     static void Task3(int[,] mas,int counter,StreamWriter print)
217     {
218         print.WriteLine("=====Task 3=====");
219         double avarage = 0;
220         int counter1 = 0;
221         for (int i = 0; i < counter; i++)
222         {
223             avarage += mas[i, 0] * mas[i, 1];
224             counter1 += mas[i, 1];
225         }
226         avarage = avarage / counter1;
227         double dispercison = 0;
228         for(int i = 0; i < counter; i++)
229         {
230             dispercison += Math.Pow(mas[i, 0],2) * mas[i, 1];
231         }
232         dispercison = (dispercison / counter1) - Math.Pow(avarage, 2);
233         print.WriteLine("Дисперсія:" + dispercison);
234         double deviation = Math.Sqrt(dispercison);
235         print.WriteLine("Середнє квадратичне відхилення розподілу:" + deviation);

```

Випробування алгоритму:

=====Task 3=====

Дисперсія:1250,89

Середнє квадратичне відхилення розподілу:35,3679233204326

|

Завдання 4:

Постановка задачі:

Побудувати гістограму частот для даного розподілу.

Побудова математичної моделі:

Для побудови гістограми частот потрібно визначити проміжки кількостей переглядів, знайти їхню частоту та ширину. Для побудови діаграми на мові Python скористаємося засобом для малювання фігур, а саме Matplotlib та збережемо нашу діаграму в файлі.

Псевдокод алгоритму:

```
import matplotlib as mpt
import matplotlib.pyplot as plt

def Task4():
    figure = plt.figure(figsize = (6,4))
    ax = figure.add_subplot()

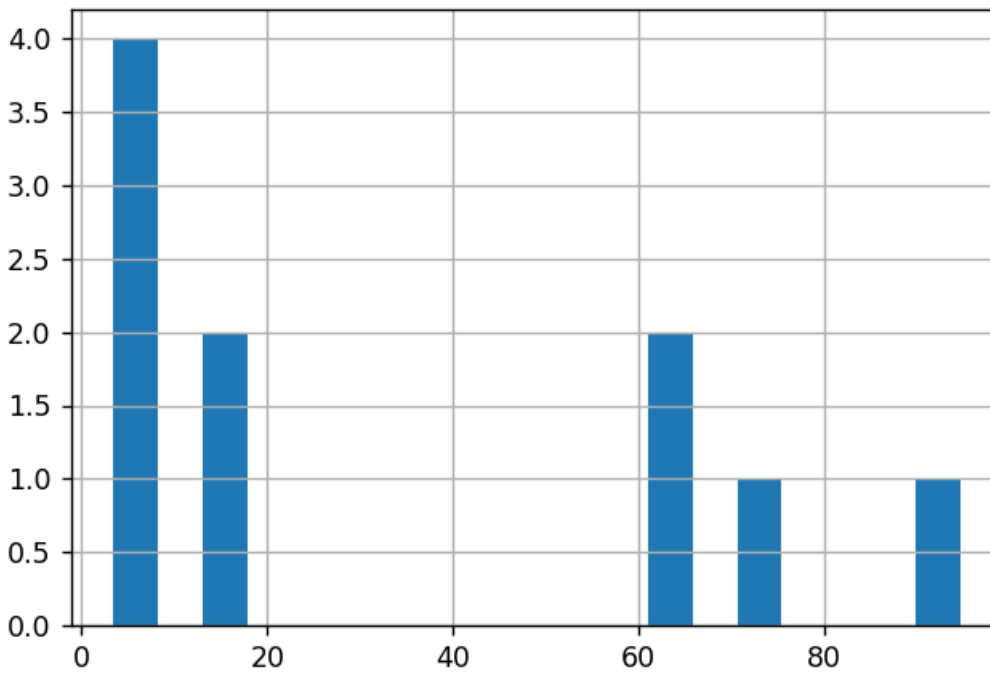
    File = input("Введіть ім'я файлу: ")

    Reader = open(File).read().splitlines()
    del Reader[0]
    Writer = open("Task4.txt", "w")

    Mas = [int(i) for i in Reader]

    plt.hist(Mas, rwidth = 0.5)
    plt.savefig("Task4")
    plt.grid();
    plt.show()
Task4()
```

Випробування алгоритму:

**Висновок:**

Виконавши цю лабораторну роботу, я навчився використовувати на практиці набуті знання про центральні тенденції та міри. Навчився знаходити моду, медіану, дисперсію та середнє квадратичне відхилення за допомогою мов програмування та різноманітних їх алгоритмів.