

А л г о р и т м и т а с т р у к т у р и д а н и х .

О с н о в и а л г о р и т м і з а ц і ї

Додаток 1

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації
і управління

Звіт

з лабораторної роботи №7
з дисципліни «Алгоритми та структури даних-1.
Основи алгоритмізації»
«Дослідження лінійного пошуку в послідовностях»
Варіант 8

Виконав
студент

ІП-13, Гончаров Євген Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірів

Наталія Вечерковська Сергіївна

(прізвище, ім'я, по батькові)

Алгоритми та структури даних.

Основи алгоритмізації

Лабораторна робота 7

Дослідження лінійного пошуку в послідовностях

Мета – дослідити методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набутти практичних навичок їх використання під час складання програмних специфікацій.

Індивідуальне завдання

Варіант 8

Завдання

| 8 | $45 + 2 * i$ | $61 - 2 * i$ | Середнє арифметичне елементів, коди яких більше 55

1. Постановка задачі

Спочатку за допомогою ф-ції `getMass1` та `getMass2` згенеруємо 2 масиви. Потім за допомогою ф-ції `getElementsFromMassAbove55` знайдемо всі елементи обох масивів що більші за 55, та середнє арифметичне між їх елементами за допомогою ф-ї `getAverageElementsFromMasses`, запишемо результат у 3-й масив та виведемо його.

2. Побудова математичної моделі.

Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Змінна <code>mass1</code>	Символьний[10]	<code>mass1</code>	Вхідні данні
Змінна <code>mass2</code>	Символьний[10]	<code>mass2</code>	Вхідні данні
Змінна <code>seed</code>	Цілочисельний	<code>seed</code>	Вхідні данні
Змінна <code>getAverageElementsFromMasses</code>	Функція(символьний[])	<code>getAverageElementsFromMasses(mass[1], mass2[])</code>	Обчислення
Змінна <code>getElementsFromMassAbove55</code>	Функція(символьний[])	<code>getElementsFromMassAbove55(mass[])</code>	Обчислення
Змінна <code>getMass1</code>	Функція(цілочисельний)	<code>getMass1(seed)</code>	Обчислення
Змінна <code>getMass2</code>	Функція(цілочисельний)	<code>getMass2(seed)</code>	Обчислення
Змінна <code>result</code>	Символьний[]	<code>result</code>	Кінцеві данні

Опишемо знаходження довжини масиву через стандартний метод `length`

Розв'язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначемо основні дії.

Крок 2. Введемо змінну `seed`

А л г о р и т м и т а с т р у к т у р и д а н и х .

О с н о в и а л г о р и т м і з а ц і ї

Крок 3. Обрахуємо значення масиву mass1 та mass2

Крок 4. Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Крок 5. Виведемо результат

П с е в д о к о д

Основна програма:

Крок 1.

Початок

Визначемо основні дії.

Введемо змінну seed

Обрахуємо значення масиву mass1 та mass2

Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Виведемо результат

кінець

Крок 2.

Початок

Визначемо основні дії.

Введемо змінну seed

Обрахуємо значення масиву mass1 та mass2

Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Виведемо результат

кінець

Крок 3.

Початок

Визначемо основні дії.

seed = 0

Обрахуємо значення масиву mass1 та mass2

Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Виведемо результат

кінець

Крок 4.

Початок

Визначемо основні дії.

seed = 0

mass1 = getMass1(seed)

mass2 = getMass2(seed)

Алгоритми та структури даних.

Основи алгоритмізації

Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Виведемо результат

кінець

Крок 5.

Початок

Визначемо основні дії.

seed = 0

mass1 = getMass1(seed)

mass2 = getMass2(seed)

result = getAverageElementsFromMasses(mass1, mass2)

Виведемо результат

кінець

Крок 6.

Початок

Визначемо основні дії.

seed = 0

mass1 = getMass1(seed)

mass2 = getMass2(seed)

result = getAverageElementsFromMasses(mass1, mass2)

Вивід result

кінець

Підпрограми:

```
getMass1(seed)
    символний mass1[10]
    повторити для i від 0 до 9:
        mass1[i] = символний(2*(seed+i) + 45)
    все повторити
    повернути mass1
кінець getMass1
```

```
getMass2(seed)
    символний mass2[10]
    повторити для i від 0 до 9:
        mass2[i] = символний(61 - 2*(seed+i))
    все повторити
    повернути mass2
кінець getMass2
```

Алгоритми та структури даних.

Основи алгоритмізації

getElementsFromMassAbove55(mass[])

цілочисельний count = 0

повторити для i від 1 до mass.length-1:

якщо mass[i] > 55 то:

count++

інакше:

mass[i] = 0

все інакше

все повторити

символьний result[count]

повторити для i від 1 до mass.length-1:

якщо mass[i] != 0 то:

result[result.length-count] = mass[i]

count--

все якщо

все повторити

повернути result

кінець **getElementsFromMassAbove55**

getAverageElementsFromMasses(mass1, mass2)

символьний average1[] =

getElementsFromMassAbove55(mass1);

символьний average2[] =

getElementsFromMassAbove55(mass2);

цілочисельний bigSize = 0

якщо average2.length > average1.length то:

bigSize = average2.length

інакше:

bigSize = average1.length

все інакше

символьний result[bigSize]

повторити для i від 0 до bigSize-1:

якщо i > average1.length то:

result[i] = average2[i]/2

інакше якщо i > average2.length то:

result[i] = average1[i]/2

Алгоритми та структури даних.

Основи алгоритмізації

```
інакше:  
    result[i] = символний((average1[i] +  
        average2[i])/2)  
все інакше  
все повторити  
повернути result[]  
кінець getAverageElementsFromMasses
```

Алгоритми та структури даних.

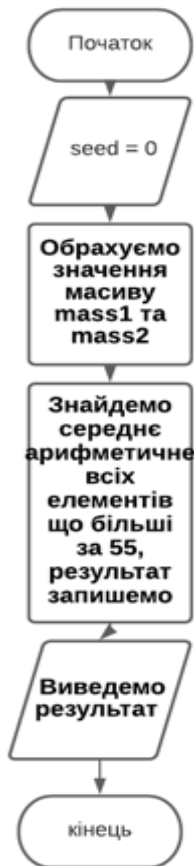
Основи алгоритмізації

Основна програма:

Крок 1, 2



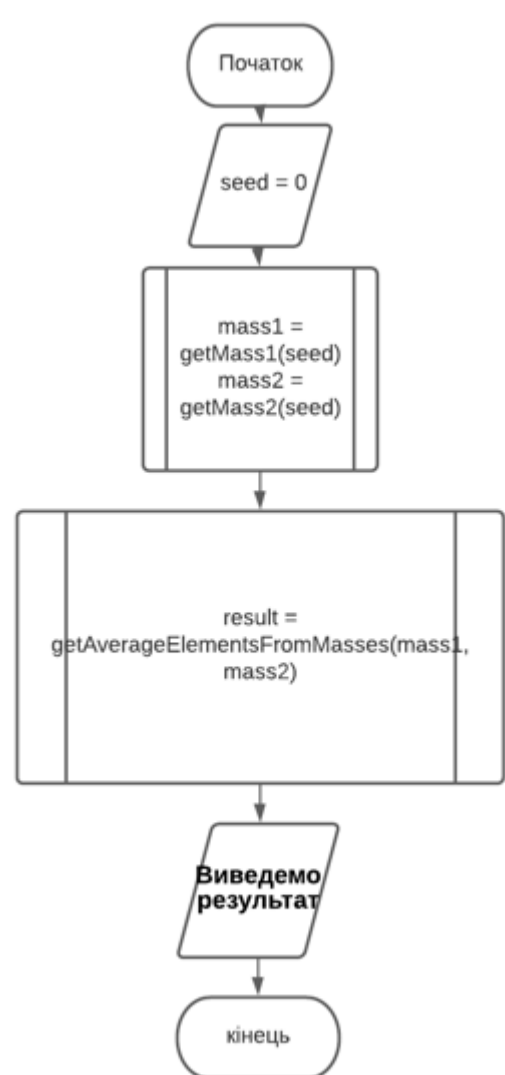
Крок 3



Крок 4

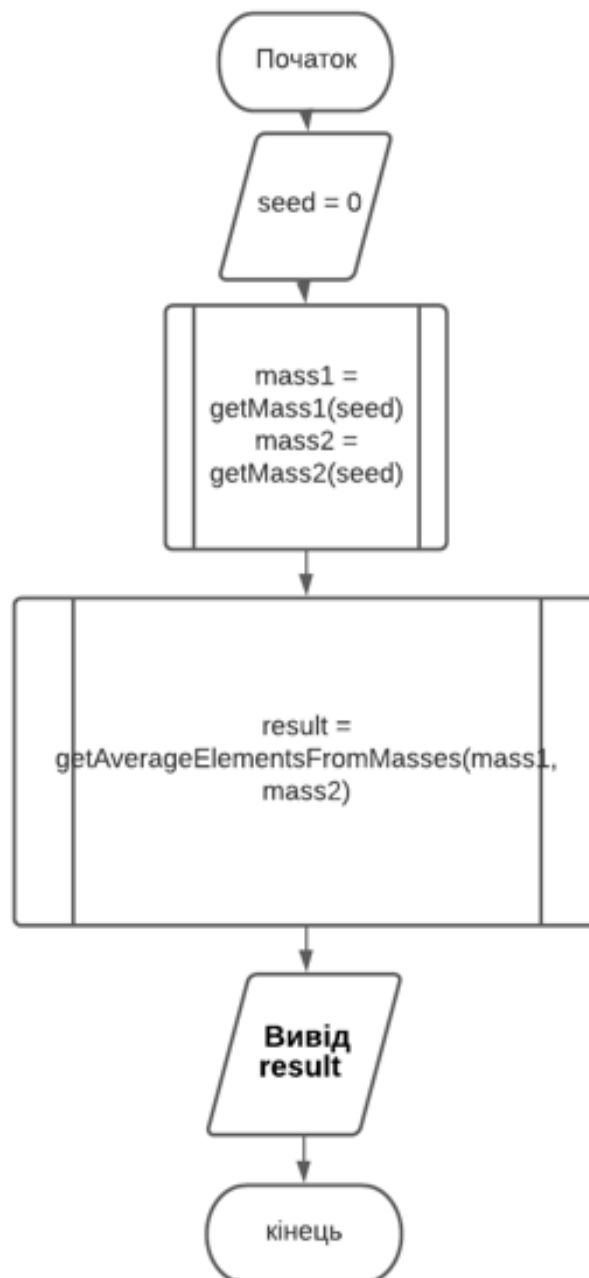


Крок 5



Основи алгоритмізації

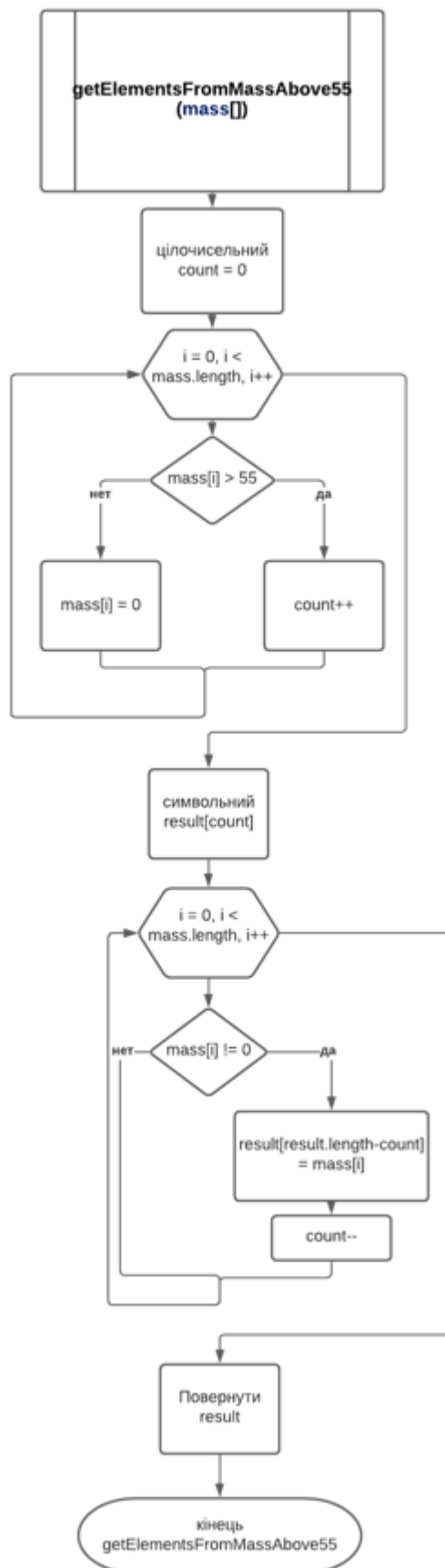
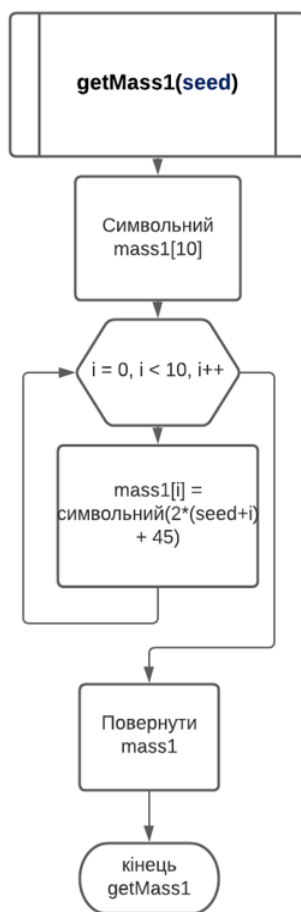
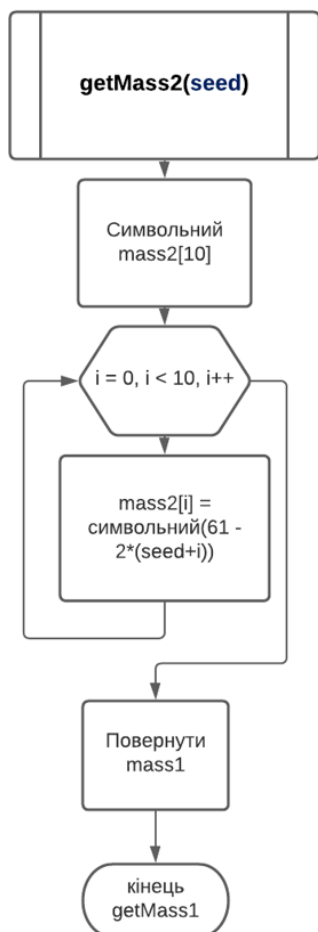
К р о к 6

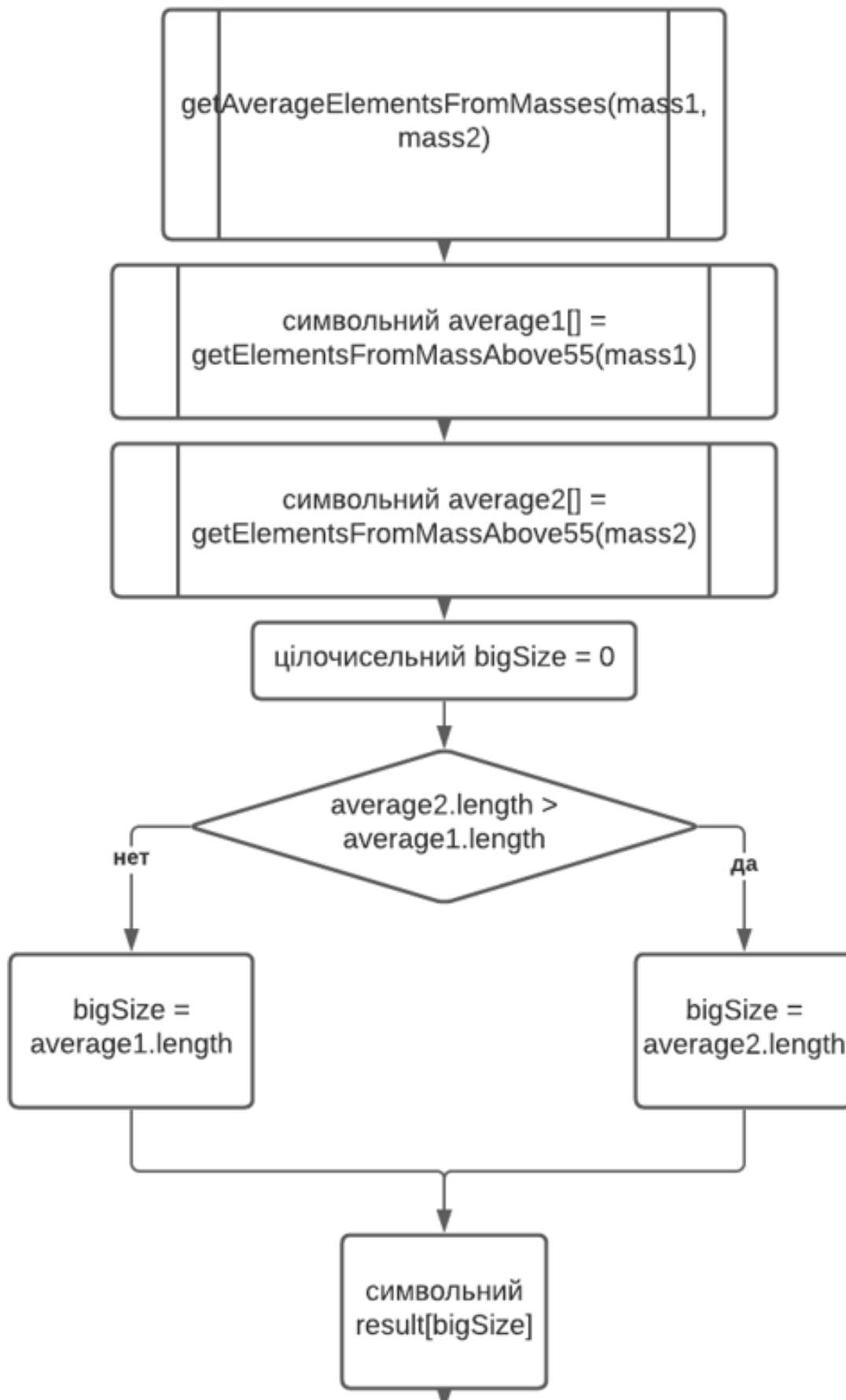


Алгоритми та структури даних.

Основи алгоритмізації

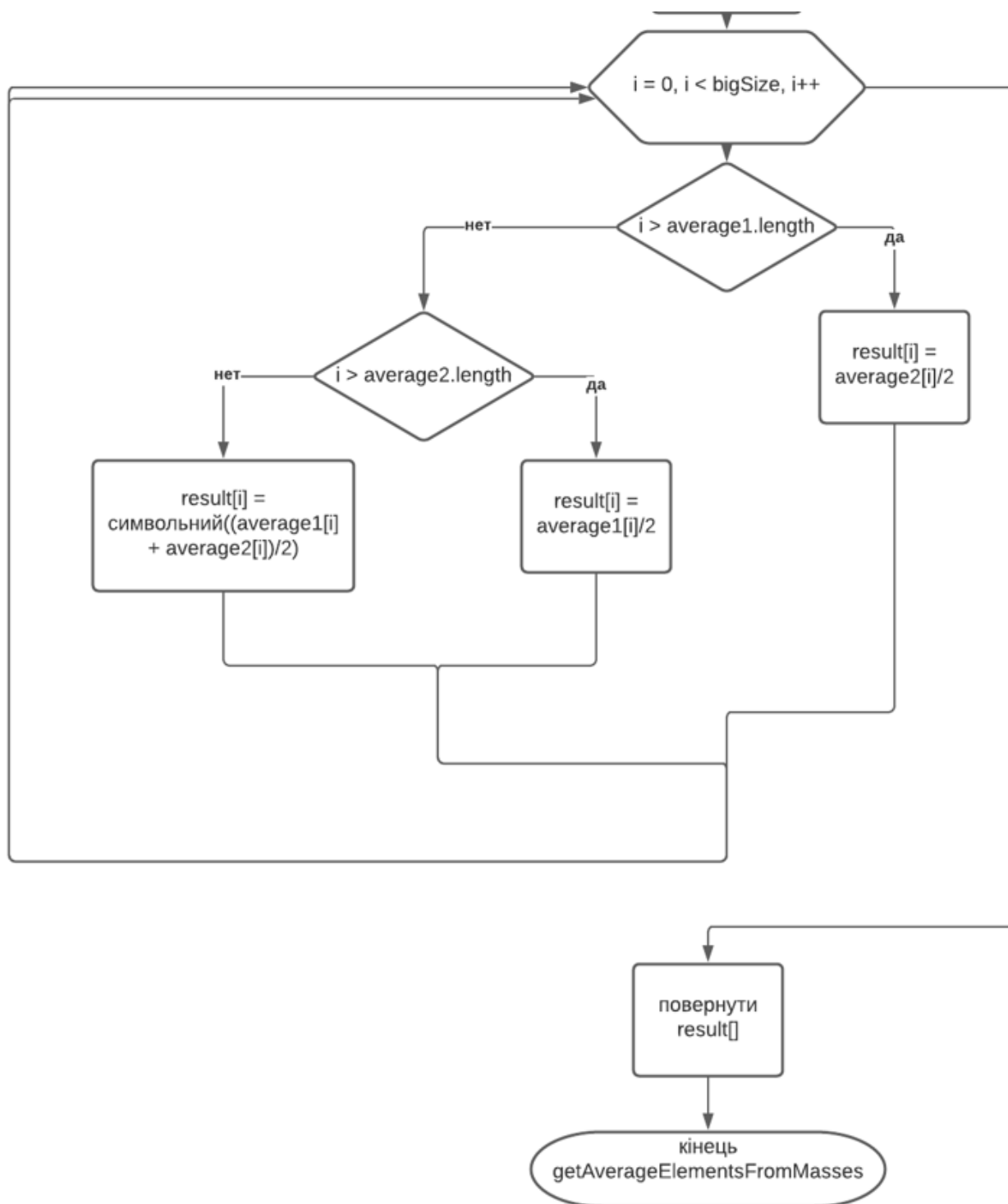
Підпрограми:





Алгоритми та структури даних.

Основи алгоритмізації



Алгоритми та структури даних.

Основи алгоритмізації

Тестування

Введіть насіння для створення масивів

0

Перший масив

0 0 0 0 0 0 57 59 61 63

Другий масив

61 59 57 0 0 0 0 0 0

Масив середніх значень елементів чий код більше за 55

59 59 59 31

Код

```
1  import java.util.Scanner;
2
3  public class Laba7ASD {
4      public static void main(String[] args) {
5          System.out.println("Введіть насіння для створення масивів");
6          Scanner in = new Scanner(System.in);
7          int seed = in.nextInt();
8          char[] mass1 = getMass1(seed);
9          char[] mass2 = getMass2(seed);
10
11          char[] solution = getAverageElementsFromMasses(mass1, mass2);
12
13          printTests(mass1, mass2, solution);
14      }
15  }
```

Алгоритми та структури даних.

Основи алгоритмізації

```
17 @ public static char[] getAverageElementsFromMasses(char[] mass1, char[] mass2){
18     char[] average1 = getElementsFromMassAbove55(mass1);
19     char[] average2 = getElementsFromMassAbove55(mass2);
20
21     int bigSize = 0;
22     if(average2.length > average1.length){ bigSize = average2.length; }
23     else { bigSize = average1.length; }
24     char[] result = new char[bigSize];
25     for(int i = 0; i < bigSize; i++){
26         if(i >= average1.length){
27             result[i] = (char)(average2[i]/2);
28         }
29         else if(i >= average2.length){
30             result[i] = (char)(average1[i]/2);
31         }
32         else{
33             int avg = ((average1[i] + average2[i])/2);
34             result[i] = (char) avg;
35         }
36     }
37     return result;
38 }
```

```
40 @ public static char[] getElementsFromMassAbove55(char[] mass){
41     int count = 0;
42     for(int i = 0; i < mass.length; i++){
43         if(mass[i] > 55){ count++; }
44         else{ mass[i] = 0; }
45     }
46     char[] result = new char[count];
47     for(int i = 0; i < mass.length; i++){
48         if (mass[i] != 0){
49             result[result.length - count] = mass[i];
50             count--;
51         }
52     }
53     return result;
54 }
```

Алгоритми та структури даних.

Основи алгоритмізації

```
55  @   public static char[] getMass1(int seed){
56      char[] mass1 = new char[10];
57      for(int i = 0; i < 10; i++){
58          mass1[i] = (char)(2 * (seed + i) + 45);
59      }
60      return mass1;
61  }
62  @   public static char[] getMass2(int seed){
63      char[] mass2 = new char[10];
64      for(int i = 0; i < 10; i++){
65          mass2[i] = (char)(61 - (2 * (seed + i)));
66      }
67      return mass2;
68  }
```

```
69      public static void printTests(char[] mass1, char[] mass2, char[] solution){
70          System.out.println("Перший масив");
71          for(int i = 0; i < 10; i++){
72              System.out.print((int)mass1[i] + " ");
73          }
74          System.out.println();
75          System.out.println("Другий масив");
76          for(int i = 0; i < 10; i++){
77              System.out.print((int)mass2[i] + " ");
78          }
79          System.out.println();
80          System.out.println("Масив середніх значень елементів чий код більше за 55");
81          for(int i = 0; i < solution.length; i++){
82              System.out.print((int)solution[i] + " ");
83          }
84      }
85  }
```

А л г о р и т м и т а с т р у к т у р и д а н и х .

О с н о в и а л г о р и т м і з а ц і ї

Висновки

Ми дослідили методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження середнього арифметичного елементів із двох списків які більші по коду за 55, декомпозиували задачу на 6 кроків: визначили основні дії, ввели намір для генерації масивів, згенерували масиви, знайшли всі елементи масивів і обчислили їх середнє арифметичне та вивели результат.