

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 №8
з дисципліни «Алгоритми та структури даних-1.
Основи алгоритмізації»
«Дослідження рекурсивних алгоритмів»
Варіант 8

Виконав
студент

ІП-13, Гончаров Євген Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірів

Наталія Вечерковська Сергіївна

(прізвище, ім'я, по батькові)

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

Лабораторна робота 8

Дослідження рекурсивних алгоритмів

Мета - дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

Індивідуальне завдання

Варіант 8

Завдання

Обчислити суму елементів геометричної прогресії, що убиває: початкове значення – 50, кінцеве значення – 1, крок – 2.

1. П о с т а н о в к а з а д а ч і

Спочатку використаємо 3 змінні(start, finish, step) щоб зберегти в них початковий, кінцевий елементи та крок геометричної прогресії. Наступною дією обчислимо суму цієї прогресії за допомогою рекурсивної підпрограми й запишемо результат у змінну result. У кінці виведемо результат.

2. П о б у д о в а м а т е м а т и ч н о ї м о д е л і.

С к л а д е м о т а б л и ц ю і м е н з м і н н и х .

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Змінна start	Цілочисельний	start	Введення,проміжні данні
Змінна finish	Цілочисельний	finish	Введення,проміжні данні
Змінна step	Цілочисельний	step	Введення,проміжні данні
Змінна sum	Цілочисельний	sum	Проміжні данні
Функція findSum	Функція(цілочисельний)	findSum(start, finish, step)	Обчислення
Змінна result	Цілочисельний	result	Кінцеві данні

Реалізуємо знаходження суми прогресії всередині функції findSum шляхом поступового додавання змінної start до змінної sum, та зменшення змінної start ,на значення змінної step, поки вона не буде менша за змінну finish.

Р о з в ' я з а н н я

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначемо основні дії.

Крок 2. Введемо значення змінних start, finish. step.

Крок 3. Підрахуємо значення змінної result за допомогою функції findSum.

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 4. Виведемо result

П с е в д о к о д

Основна програма:

Крок 1

Початок

Визначемо основні дії

Введемо значення змінних start, finish. step.

Підрахуємо значення змінної result за допомогою функції findSum.

Виведемо result

кінець

Крок 2

Початок

Визначемо основні дії

Введемо значення змінних start, finish. step.

Підрахуємо значення змінної result за допомогою функції findSum.

Виведемо result

кінець

Крок 3

Початок

Визначемо основні дії

Start = 50, finish = 1, step = 2

Підрахуємо значення змінної result за допомогою функції findSum.

Виведемо result

Кінець

Крок 4

Початок

Визначемо основні дії

Start = 50, finish = 1, step = 2

Result = findSum(start, finish, step)

Виведемо result

Кінець

Алгоритми та структури даних.

Основи алгоритмізації

Крок 5

Початок

Визначемо основні дії

Введемо значення змінних `start`, `finish`, `step`.

`Result` = `findSum`(`start`, `finish`, `step`)

Виведемо `result`

кінець

Основи алгоритмізації

Підпрограми:

Крок 1

```
findSum(start, finish, step)
    sum = start
    start /= step;
    sum += findSum(start, finish, step)
    кінець рекурсії
    зворотній хід рекурсії
Кінець findSum
```

Крок 2

```
findSum(start, finish, step)
    sum = start
    start /= step;
    якщо start >= finish то:
        sum += findSum(start, finish, step)
    все якщо
    зворотній хід рекурсії
Кінець findSum
```

Крок 3

```
findSum(start, finish, step)
    sum = start
    start /= step;
    якщо start >= finish то:
        sum += findSum(start, finish, step)
    все якщо
    повернути sum
Кінець findSum
```

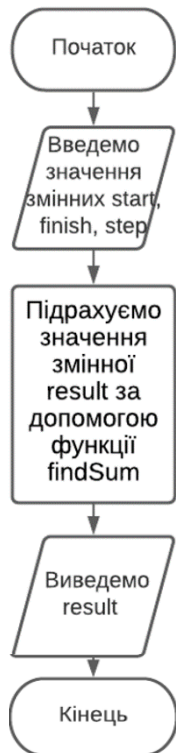
Алгоритми та структури даних.

Основи алгоритмізації

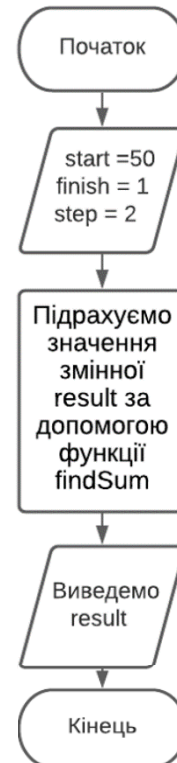
Блок-схема

Основна програма:

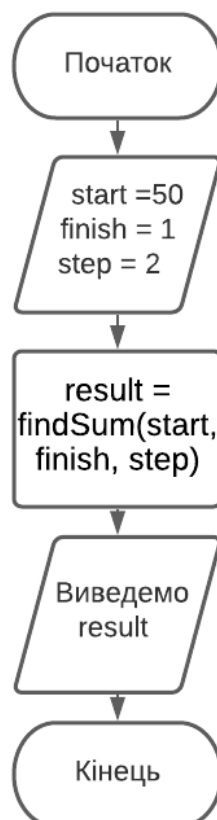
Крок 1, 2



Крок 3

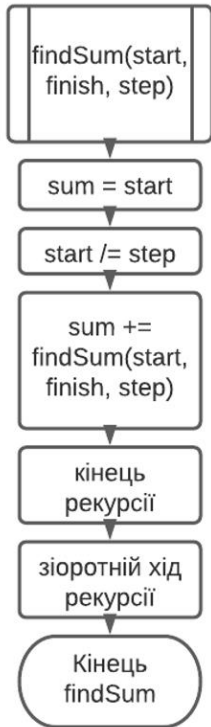


Крок 4

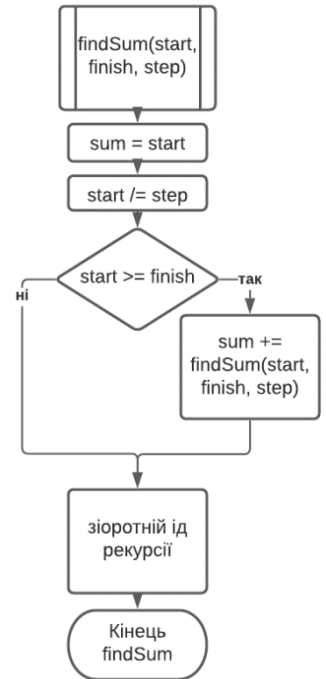


Підпрограми:

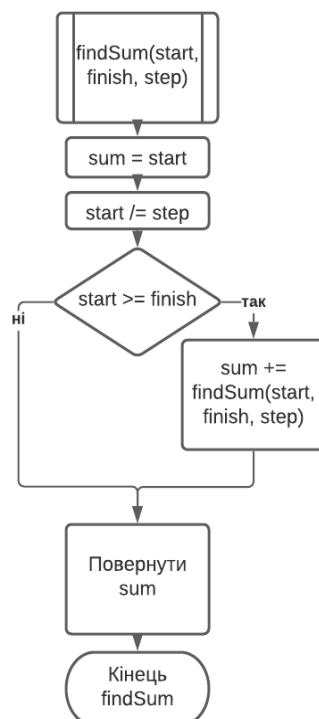
К р о к 1



К р о к 2



К р о к 3



Алгоритми та структури даних.

Основи алгоритмізації

Тестування

Б л о к	Д і я
	П о ч а т о к
1	Start = 4
2	Finish = 1
3	Step = 2
4	Sum = 4
5	Start = 2
6	Sum = 2
7	Start = 1
8	Sum = 1
9	Start = 0.5
10	Sum = 3
11	Sum = 7
12	Result = 7
	К і н е ц ь

К о д

```
1  ► public class tests {
2  ►  ► public static void main(String[] args) {
3      int start = 2;
4      int step = 2;
5      int finish = 1;
6      System.out.println(findSum(start, finish, step));
7  }
8  ► public static int findSum(int start, int finish, int step){
9      int sum = start;
10     start /= step;
11     if(start >= finish){
12     ►     sum += findSum(start, finish, step);
13     }
14     return sum;
15 }
16 }
```

Висновки

Ми дослідили особливості роботи рекурсивних алгоритмів та набули практичних навичок їх використання під час складання програмних специфікацій підпрограм. В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження суми всіх елементів геометричної прогресії від початкового елементу до фінального, декомпозували задачу на 4 кроки: визначили основні дії, ввели значення змінних start, finish, step, обчислили result та вивели його.