Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 1

З дисципліни «Алгоритми та методи обчислень»

На тему «Поняття алгоритму.

Задавання алгоритму у вигляді блок-схеми»

ВИКОНАВ:

студент 2 курсу ФІОТ

групи ІО-41

Логвинчук А. І.

Варіант 20

ПЕРЕВІРИВ:

ст. в. Порєв В. М.

Київ – 2016

**Тема:** Поняття алгоритму. Задання алгоритму у вигляді блок-схеми.

**Мета:** Навчитись створювати блок-схеми лінійного алгоритму; розгалуженого алгоритму та циклічного алгоритму за допомогою редактора MS Visio.

**Завдання**

Відповідно до варіанту завдання розробити блок-схеми очислення виразів для лінінйного алгоритму, алгоритму, що розгалужується та циклічного алгоритму. У відповідності до блок-схеми створити програму обчислення виразу на алгоритміній мові.

**Блок-схеми алгоритмів**

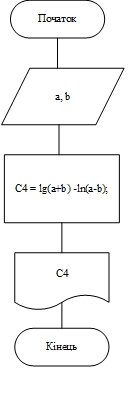


рис. 1. Блок-схема лінійного алгоритму

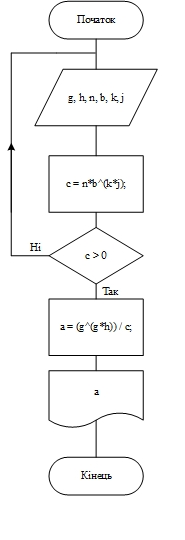
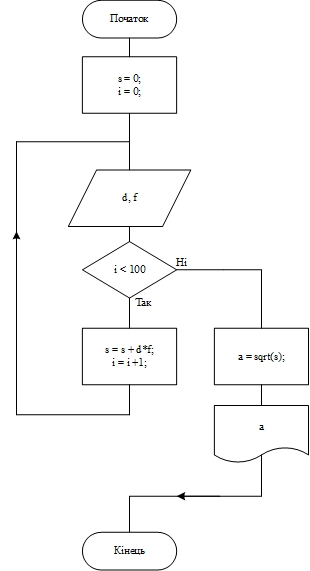


рис. 2. Блок-схема розгалуженого алгоритму

рис. 3. Блок-схема циклічного алгоритму

**Текст програми**

Лінійний алгоритм:

public void compute1(View view) {  
  
 EditText inputA = (EditText) findViewById(R.id.*input\_a*);  
 EditText inputB = (EditText) findViewById(R.id.*input\_b*);  
 TextView result = (TextView) findViewById(R.id.*result*);  
  
 double a = 0;  
 double b = 0;  
   
 try {  
 a = Double.*parseDouble*(inputA.getText().toString());  
 b = Double.*parseDouble*(inputB.getText().toString());  
 } catch (NumberFormatException e) {  
 return;  
 }  
  
 if (a < b) {  
 result.setText(«Невірні вхідні дані»);  
 return;  
 }  
  
 double res = Math.*log10*(a + b) – Math.*log*(a – b);  
  
 result.setText(«Результат: « + Double.*toString*(res));  
   
}

Розгалужений алгоритм:

public void compute2(View view) {  
 EditText inputG = (EditText) findViewById(R.id.*input\_g*);  
 EditText inputH = (EditText) findViewById(R.id.*input\_h*);  
 EditText inputN = (EditText) findViewById(R.id.*input\_n*);  
 EditText inputB = (EditText) findViewById(R.id.*input\_bb*);  
 EditText inputK = (EditText) findViewById(R.id.*input\_k*);  
 EditText inputJ = (EditText) findViewById(R.id.*input\_j*);  
  
 TextView condition = (TextView) findViewById(R.id.*condition*);  
 TextView result = (TextView) findViewById(R.id.*result2*);  
  
 double g = 0;  
 double h = 0;  
 double n = 0;  
 double b = 0;  
 double k = 0;  
 double j = 0;  
 try {  
 g = Double.*parseDouble*(inputG.getText().toString());  
 h = Double.*parseDouble*(inputH.getText().toString());  
 n = Double.*parseDouble*(inputN.getText().toString());  
 b = Double.*parseDouble*(inputB.getText().toString());  
 k = Double.*parseDouble*(inputK.getText().toString());  
 j = Double.*parseDouble*(inputJ.getText().toString());  
 } catch (NumberFormatException e) {  
 return;  
 }  
  
 double c = Math.*pow*(n\*b, k\*j);  
  
 if (c < 0) {  
 condition.setText(Double.*toString*(c) + " не більше 0");  
 result.setText("Введіть інші дані");  
  
 inputG.setText("");  
 inputH.setText("");  
 inputN.setText("");  
 inputB.setText("");  
 inputK.setText("");  
 inputJ.setText("");  
  
 } else {  
 double a = Math.*pow*(g, g\*h) / c;  
 condition.setText(Double.*toString*(c) + " > 0");  
 result.setText("Резуьтат: " + Double.*toString*(a));  
 }  
}

Циклічний алгоритм:

public void compute3(View view) {

TextView result = (TextView) findViewById(R.id.*result3*);  
  
 if (i < 10) {  
 EditText inputD = (EditText) findViewById(R.id.*input\_d*);  
 EditText inputF = (EditText) findViewById(R.id.*input\_f*);  
  
 TextView iter = (TextView) findViewById(R.id.*iter*);  
  
  
 double d = 0.;  
 double f = 0.;

try {  
 d = Double.*parseDouble*(inputD.getText().toString());  
 f = Double.*parseDouble*(inputF.getText().toString());  
 } catch (NumberFormatException e) {  
 return;  
 }  
  
  
 i++;  
 iter.setText("Ітерація: " + Integer.*toString*(i));  
 s += d\*f;  
 return;  
 }  
  
 result.setText("Кінцевий результат: " + Math.*sqrt*(s));

}

**Приклад результату роботи програми**

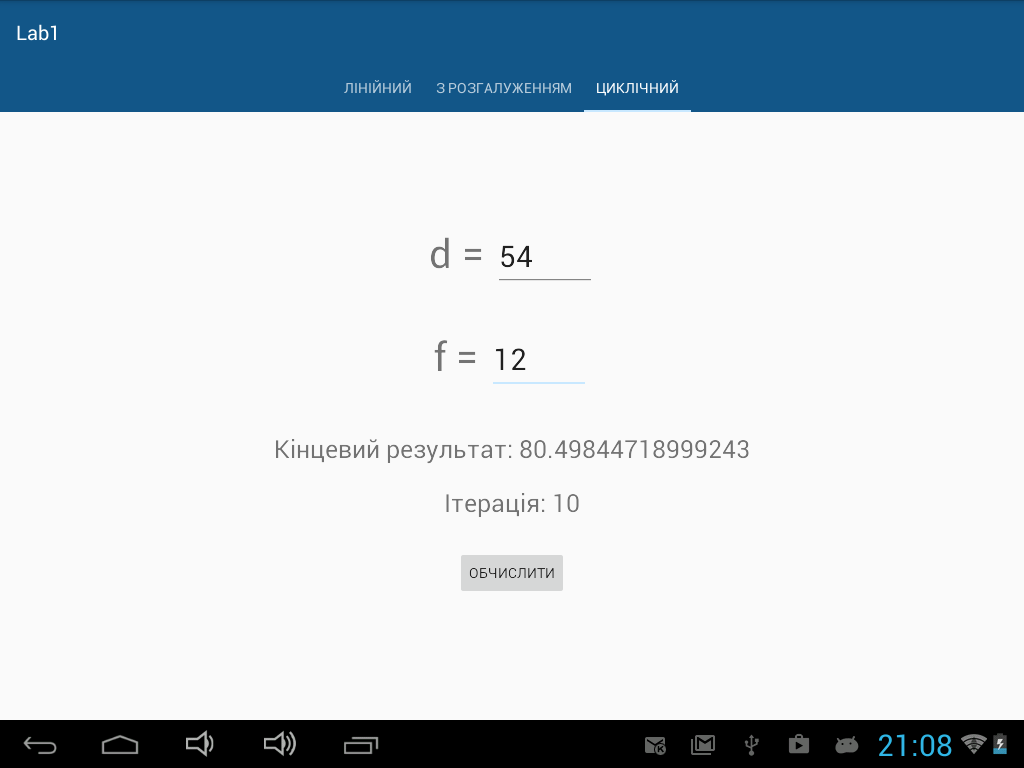


рис. 4. Знімок екрану під час роботи програми

**Висновки**

Алгоритм – набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій. Для візуалізації алгоритмів використовую блок-схеми.

Основні етапи розв’язання задачі:

* Введення вхідних даних.
* Перетворення їх за певними правилами.
* Виведення результатів.

Властивості алгоритму:

* скінченність
* детермінованість
* результативність
* масовість