МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО - КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

**Курсова робота**

З дисципліни “Прикладне програмування Java” на тему:

розробка калькулятора для вирішення квадратних рівнянь

Роботу виконав здобувач вищої освіти,

Студент групи КНД-12

Черних А.О

Роботу перевірив\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата первірки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ, 2024р.

Зміст

**Опис..................................................................................................................... 3**

**Імпорт бібліотек і оголошення змінних.............................................................. 4**

**Створення метода для відображення вікна і його налаштування...................... 5**

**Створення і налаштування панелей для виводу текста і створення міток для відображення тексту............................................................................................** **6**

**Створення і налаштування кнопок, створення текстових полів для виводу результата............................................................................................................. 7**

**Метод для кнопки “Erase” і встановлення розмірів компонентів ...................... 8**

**Метод для кнопки “Exit” та початок підключення кнопки “Calculate”................ 9**

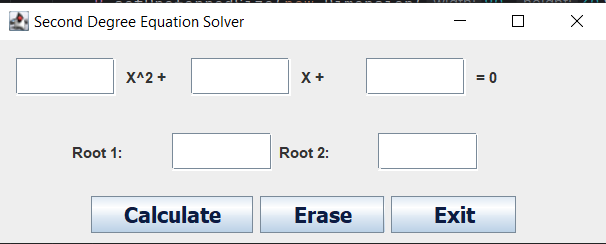
**Додавання компонентів до головної панелі....................................................... 11**

**Метод підрахунку чисел і код програми............................................................. 12**

Опис

Калькулятор квадратичного рівняння, який був запропонований у методичці для прикладу виконання курсової роботи. Відрізняє від прикладу у методичці те, що калькулятор у цьому варіанті розробки рахує будь яке квадратичне рівняння яке має два розв’язки, в не залежності від цілих чи не цілих чисел.

Після запуску програми у Вас відкриється вікно у якому ви зможете порахувати рівняння яке Вам потрібно. У поля А, В, С, Ви можете ввести числа для обрахування рівняння. Після того як Ви ввели числа для обрахування натисніть клавішу Calculate, програма автоматично порахує Ваше рівнняня і відповідь запишеться у поля Root1, Root2. При натисканні хрестика, або кнопки Exit, програма закриється. Якщо Вам треба очистити всі поля, Ви можете скористатися клавішою Erase, всі дані які були у полях зникнуть і можна буде ввести рівняння знову.

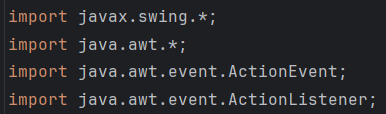


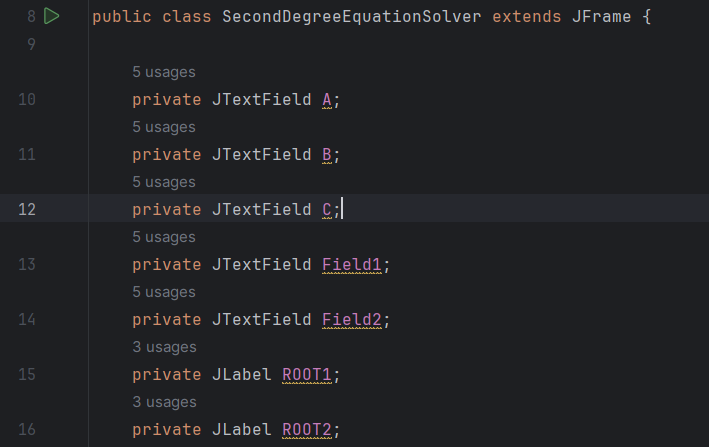
Сам код починаться з імпорту тих бібліотек які нам знадобляться, import javax.swing.\* імпортує всі класи із пакету java swing. Цей пакет містить багато класів для створення компонентів GUI, створення кнопок, поля для вводу текста, панелі та списки.

import java.awt.\* імпортує всі класи із пакету java.awt.\*, тут є самі основні класи для роботи з графікою та інтерфейсом користувача, такі як Color, Font, Graphics.

import java.awt.event.ActionEvent, цей рядок імпортує клас ActionEvent, він використовується для використання подій, натискання кнопки тощо.

import java.awt.event.ActionListener, цей рядок імпортує інтерфейс ActionListener. Цей рядок визиває метод, який викликається коли відбувається подія ActionEvent.



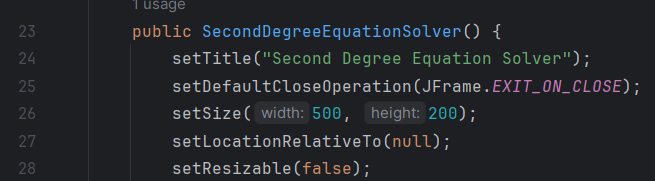


8. Створення класу SecondDegreeEquationSolver який є підкласом JFrame;

10, 11, 12: Оголошення трьох приватних змінних(A, B, C) які є текстовими полями для вводу значень користувачем;

13, 14: Оголошення двох текстових полів (Field1, Field2) для виводу результата;

15, 16: Оголошення двух приватних полів ROOT1 i ROOT2;



В цьому блоці коду ми створюємо метод для відображення вікна і задаємо йому налаштування які нам потрібні, назва сторінки, завершення програми на натиск хрестика, розміри та розміщення на екрані.

23: створення методу для налаштування вікна в якому буде калькулятор;

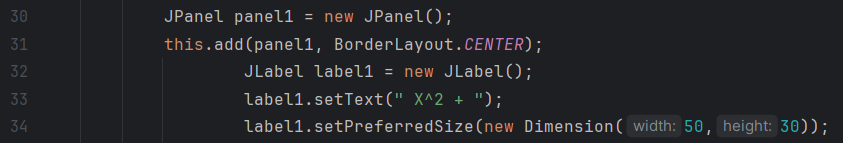
24: зміна назви вікна на “Second Degree Equation Solver”$

25: команда щоб програма закінчувалася коли натискаємо хрестик;

26: розмір вікна;

27: команда щоби вікно з’являлося не в лівому верхньому куті, а в центрі екрана;

28: зміна розміру вікна - неможлива;

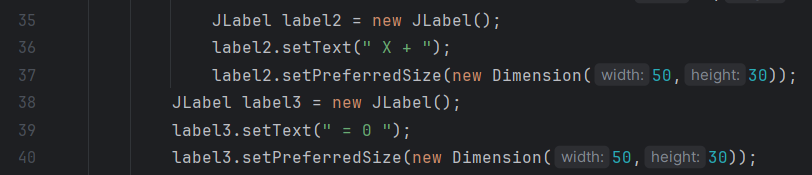
Після створення і налаштування панелі, я створюю label1, label2, label3 для візуального представлення квадратного рівняння. Я задаю кожному з них однакову характеристику, але з різним текстом для представлення. 30: цей рядок створює нову панель з назвою panel1;

31: цією командою ми додаємо панель у центр вікна;

32: цей рядок створює новий компонент(label1) для відображення тексту;

33: у label1 додаємо текст який хочемо відобразити у вікні;

34: задаємо розмір label1;

Аналогічно робимо ще такі ж дві панелі label2 i label3.

35: створення панелі label2;

36: додавання тексту у панель label2;

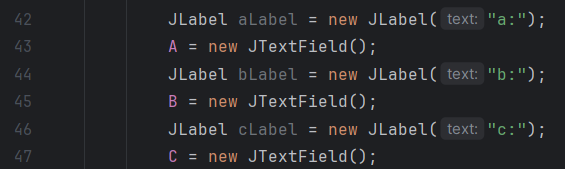
37: зміна розміру панелі label2 до такого же як на label1;

38: створення панелі label3;

39: додавання тексту до панелі label3;

40: зміна розміру панелі label3 до таких як у label1 i label2;

У цьому блоці коду я створив три текстових поля( A, B, C ) для того аби користувач вводив свої числа.



42: створює мітку яка відображає текст а;

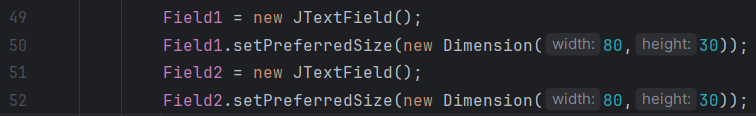
43: це текстове поле у якому користувач може ввести змінну А;

44: створює мітку яка відображає текст b;

45: текстове поле у якому користувач може ввести змінну В;

46: створює мітку яка відображає текст с;

47: текстове поле у якому користувач може ввести текст С;

49: створюю текстове поле Field1для виводу результата;

50: встановлюю розмір текстового поля такий як мені потрібен;

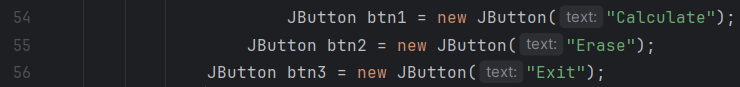
51: створюю текстове поле Field2 для виводу другого результата;

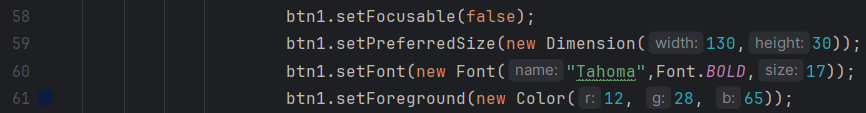
52: встановлюю розмір текстового поля як для Field1;

Btn1 - кнопка “Calculate”, при її натисканні починається команда підрахунку.

Btn2 - кнопка “Erase” , при її натисканні всі текстові поля вмить очистяться і можна буде ввести дані заново.

Btn3 - кнопка “Exit”, при натисканні її програма закінчиться.

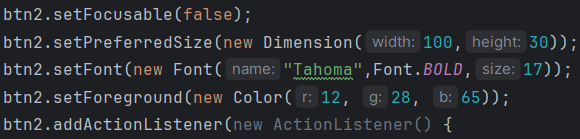
54-56: створюю три кнопки, btn1, btn2, btn3;

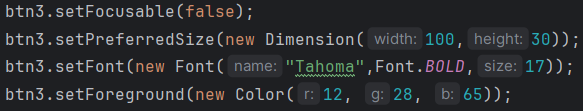
58: цей рядок робить кнопку неактивною, не можна нажати її якоюсь іншою клавішою чи ще чимось;

59: цей рядок встановлює бажаний розмір кнопки;

60: встановлюємо шрифт і розмір тексту в кнопці;

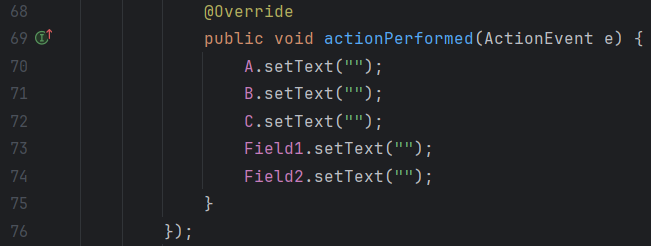
61: зміна кольору тексту в кнопці;





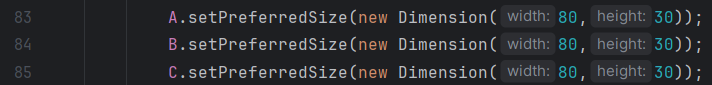
Аналогічні дії проводжу з другою та третюю кнопкою.

Для кожної з кнопок прописаний свій метод actionPerformed, який реагує на події, саме він робить так аби кнопки працювали і виконували свої функції. Тут я прописав такий метод для кнопки btn2. Друга кнопка відповідає за функцію “Clear”, тому я поставив у всіх полях для вводу чисел пусті значеня у вигляді подвійних лапок, щоб при натисканні на кнопку поля очищалися.

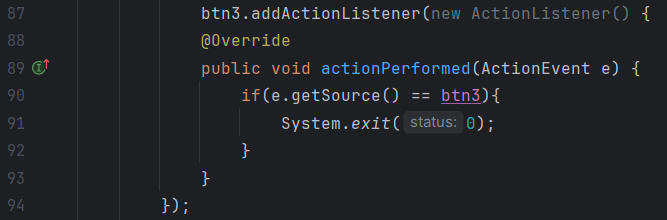
68: анотація @Override, що вказує на те що метод перевизначений;

69: сам метод actionPerformed;

70-74: ці рядки використовують метод setText для встановлення тексту кожного з поля на порожній рядок(“”);

В рядках 83-85 я встановлюю бажаний розмір для компонентів. В цьому випадку я встановив ширину і висоту яка мені була потрібна для текстових полів вводу чисел A, B, C.

У цій частині кода я прописав дію для третьої кнопки(btn3), вона відповідає за вихід, отже програма має завершитися при її натисканні.

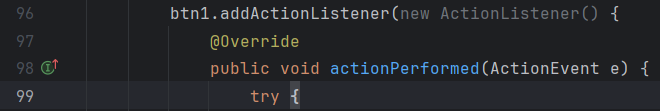
87: цей вираз додає actionListener до btn3;

88: анотація @Override, що вказує на те що метод перевизначений;

89: сам метод;

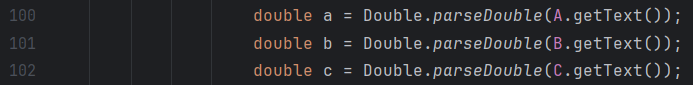
90: ця умова перевіряє чи є джерелом події кнопка btn3;

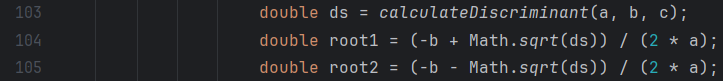
91: цей рядок завершує програму з кодом виходу 0, це значить що програма успішно завершена;

96: цей рядок додає actionListener до кнопки btn1;

97-98: метод і анотація @Override;

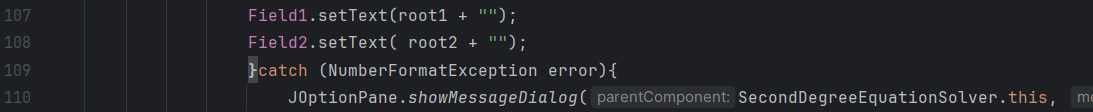
99: try - ключове слово перед місцем де може бути помилка

У рядках 100-102 ми отримуємо коефіцієнти рівняння з текстових полів A, B, C. Вираз getText бере текст який ввів користувач з текстових полів.

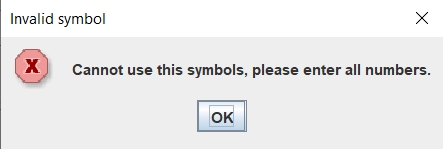
103: цей рядок зберігає значення дискримінанту у змінній ds, також він викликає метод calculateDiscriminant в який передає значення трьох аргументів і який буде показаний далі.

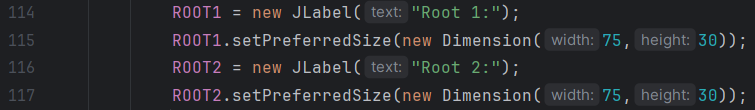
104: цей рядок розраховує першший корінь квадратного рівняння:

105: цей рядок рахує другий корінь рівняння;

107 і 108 строка встановлюють текст текстових полів Field1 i Field2;

109 і 110 строки виконуюються якщо відбулася помилка вводу чисел у текстові поля, JOptionPane.showMessageDialog ця команда виведе користувачеві діалогове вікно. "Cannot use this symbols, please enter all numbers.", це повідомлення яке отримає користувач при неправильному вводі. Ось таке повідомлення Ви отримаєте.

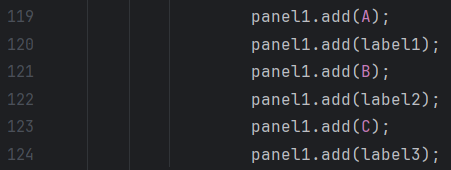


114: тут ми робимо текстовий рядок для виводу текста у лапках;

115: налаштовуємо розмір компонента ROOT1;

116: створюємо текстовий рядок для виводу текста у лапках;

117: налаштовуємо розмір компонента ROOT2;



119: додаю текстове поле А до головної панелі;

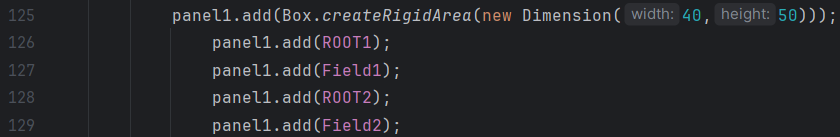
120: додаю рядок label1 до головної панелі;

121: додаю текстове поле В;

122: додаю рядок label2 до головної панелі;

123: додаю текстове поле С;

124: додаю рядок label3 до головної панелі;

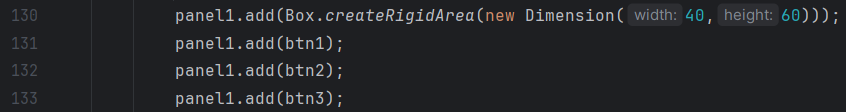
125: тут я додаю у панель невидимий жорсткий компонент і задаю йому одразу розміри, це для того аби розділити елементи між собою.

126: додаю у цій строці рядок з виводом тексту “Root1:” ;

127: додаю текстове поле у якому буде здійснюватися вивід результата;

128: додаю тут строку яка буде виводити текст “Root2”;

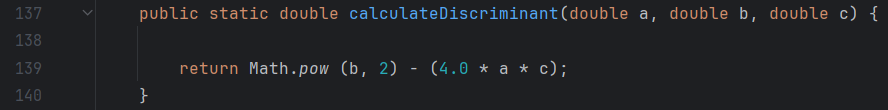
129: додаю друге текстове поле для вивода результата;

130: також додаю у панель нивидимий елемент для того аби розділити елементи;

131: додаю у панель першу кнопку “Calculate”;

132-133: додаю у панель кнопки два і три (Erase, Exit);

На останок представляю Вам метод про який йшлося раніше і який виконує функцію підрахунку. Метод приймає в себе три змінні які вводить користувач і по формулі дискримінанту шукає розв’язок.



**Код програми**



