

**Лабораторна робота №4****Застосування класів стандартної бібліотеки мови Java**

1. Використовуючи програму з лабораторної роботи №3 як допоміжний клас, розробити програму з графічним інтерфейсом користувача, яка у головному вікні дозволяє вводити дані та виводити результати відповідно до варіанту. Головне вікно програми повинно створюватись, як таке, що наслідується від стандартного класу JFrame, що належить до стандартної бібліотеки інтерфейсних елементів мови Java (Swing). Елементи, що розміщуються у цьому вікні, також повинні створюватись на основі бібліотеки Swing.
2. Результати роботи також вивести у вікні програми за допомогою стандартної бібліотеки класів мови Java.
3. Якщо в процесі написання програми знадобиться переробити структуру методів головного класу з другої лабораторної роботи, то для з'ясування коректності цих змін виконати тестування цього класу за допомогою тестового класу, що був створений у другій роботі.
4. По закінченню виконання роботи, виконати її тестування.

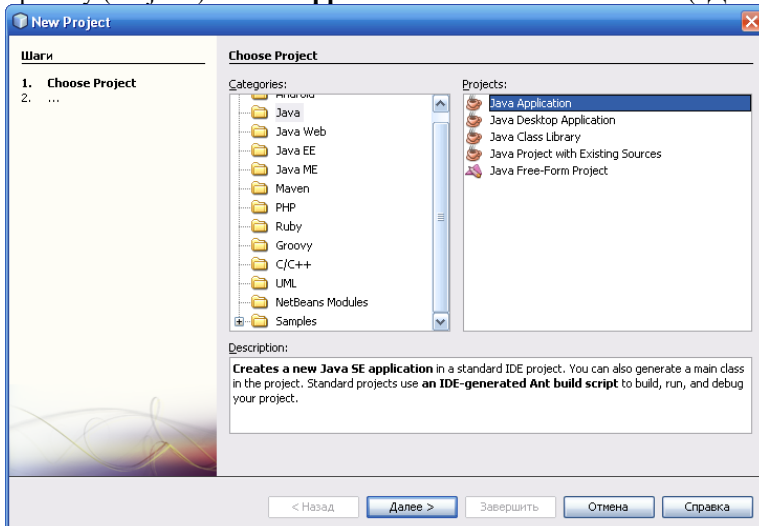
Варіант	Завдання
1	<ol style="list-style-type: none"> <li>1. Ввести елементи масиву.</li> <li>2. Обчислити суму найбільшого та найменшого.</li> <li>3. Обчислити значення функції при аргументі, що дорівнює найбільшому елементу масиву, найменшому та середньому значенні елементів масиву.</li> <li>4. Всі результати вивести у цьому ж вікні.</li> </ol>
2	<ol style="list-style-type: none"> <li>1. Ввести елементи масиву.</li> <li>2. Обчислити суму номерів найбільшого та найменшого елементів масиву.</li> <li>3. Обчислити суму всіх від'ємних елементів масиву.</li> <li>4. Обчислити середнє значення всіх додатних елементів масиву.</li> </ol>
3	<ol style="list-style-type: none"> <li>1. Ввести елементи масиву.</li> <li>2. Визначити "відстань" (кількість елементів) між найбільшим та найменшим елементами масиву.</li> <li>3. Визначити кількість від'ємних елементів масиву.</li> <li>4. Обчислити суму та середнє арифметичне всіх додатних елементів масиву.</li> </ol>
4	<ol style="list-style-type: none"> <li>1. Ввести елементи масиву.</li> <li>2. Визначити кількість елементів, ціла частина яких є непарним числом.</li> <li>3. Визначити значення елемента, що йде після найбільшого елемента. Якщо такого немає, вивести NaN</li> <li>4. Обчислити суму та середнє арифметичне елементів масиву, які більше вказаного числа.</li> </ol>
5	<ol style="list-style-type: none"> <li>1. Ввести елементи масиву.</li> <li>2. Визначити, скільки разів у масиві зустрічається число, що дорівнює найбільшому елементу.</li> <li>3. Обчислити суму всіх від'ємних елементів.</li> <li>4. Знайти середнє арифметичне всіх елементів, що менше найбільшого елемента.</li> </ol>
6	<ol style="list-style-type: none"> <li>1. Ввести елементи масиву.</li> <li>2. Знайти розв'язання лінійного рівняння <math>ax + b = 0</math>, де <math>a</math> - найбільший, <math>b</math> - найменший елементи масиву.</li> <li>3. Знайти суму та середнє арифметичне всіх елементів масиву.</li> <li>4. Обчислити кількість додатних елементів, які мають номери, більші, ніж найменший елемент масиву.</li> </ol>
7	<ol style="list-style-type: none"> <li>1. Ввести елементи масиву.</li> <li>2. Знайти найбільший від'ємний та найменший додатний елементи масиву.</li> <li>3. Обчислити середнє арифметичне всіх елементів масиву із парними номерами.</li> <li>4. Обчислити суму натуральних логарифмів від модулів найбільшого та найменшого елементів.</li> </ol>
8	<ol style="list-style-type: none"> <li>1. Ввести елементи масиву.</li> <li>2. Знайти найбільший елемент серед тих, що мають непарні номери.</li> <li>3. Знайти суму і середнє арифметичне значення всіх елементів масиву, без урахування того, що був знайдений у п.2.</li> <li>4. Обчислити суму усіх від'ємних елементів масиву, що мають номери, менші за найбільший елемент.</li> </ol>
9	<ol style="list-style-type: none"> <li>1. Ввести елементи масиву.</li> <li>2. Знайти середнє арифметичне від'ємних елементів масиву.</li> <li>3. Знайти суму елементів, більших значення, знайденого у п.2</li> <li>4. Визначити кількість додатних елементів, що розташовані між найбільшим та найменшим елементами.</li> </ol>
10	<ol style="list-style-type: none"> <li>1. Ввести елементи масиву.</li> <li>2. Знайти суму всіх елементів масиву.</li> <li>3. Знайти суму тих елементів, що більше ніж різниця найбільшого та найменшого елементів.</li> <li>4. Визначити кількість елементів, ціла частина яких є непарною та більше середнього арифметичного всіх елементів.</li> </ol>

Порядок виконання роботи.

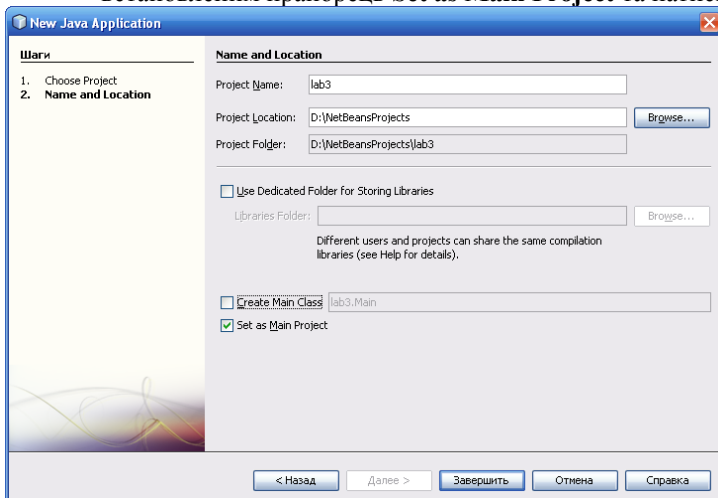
- A. У середовищі NetBeans створити новий проект. Для цього у головному меню програми оберіть: *File -> New Project*, у діалоговому вікні, що відкриється оберіть категорію проекту "Java" та в ній проект – "Java Application".
- B. Додайте до цього проекту головний клас – вікно (JFrame). Для цього у головному меню програми оберіть: *File -> New File*, у діалоговому вікні, що відкриється оберіть категорію файлу "Swing GUI Forms" (Форми графічного інтерфейсу користувача), в цій категорії оберіть "JFrame Form" (форма на основі класу JFrame).
- C. За допомогою маніпулятора "Миша" додайте з палітри "Palette" до головного вікна проекту елементи інтерфейсу:
  - JPanel (для розділення вікна на окремі області – панелі),
  - JLabel (для виведення повідомлень та написів у вікні),
  - JTextField (для введення даних у програму),
  - JButton (Для надання користувачеві можливості давати команди програмі)
  - JTextArea, JList чи JTable (для відображення великої кількості даних)
  - Будь-які інші, які вважаєте за потрібні.
- D. Скопіюйте у каталог **src** проекту файл з текстом класу, який був розроблений для Лабораторної роботи №2 та використайте його для виконання наступного пункту.
- E. У контекстному меню елементу управління JButton (що ви розмістили у вікні у пункті C) оберіть **Events->Action->actionPerformed** та у тексті метода, що відкриється для редагування опишіть реакцію на натискання цієї кнопки.

## Приклад створення програми, що розв'язує квадратні рівняння, з використанням стандартних класів для створення графічного інтерфейсу користувача мови Java.

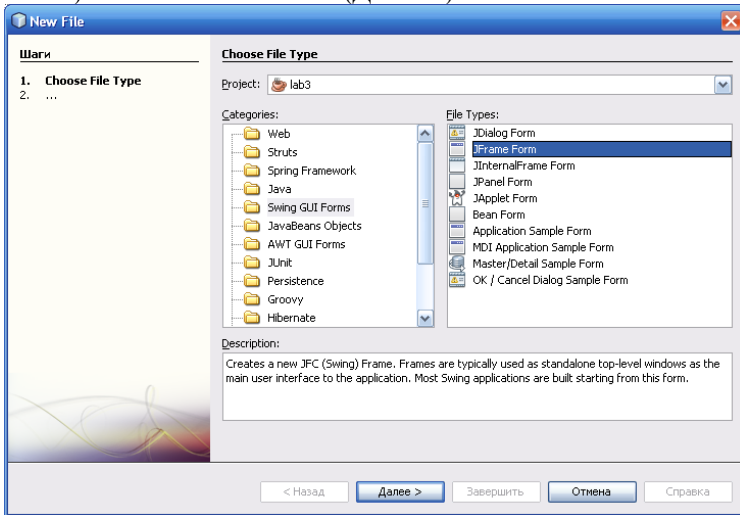
1. Створюємо новий проект: **File(Файл) -> New Project**, обираємо категорію (Categories) **Java**, а в ній тип проекту (Projects) – **Java Application**. Натискаємо "Next >" ("Далее >")...



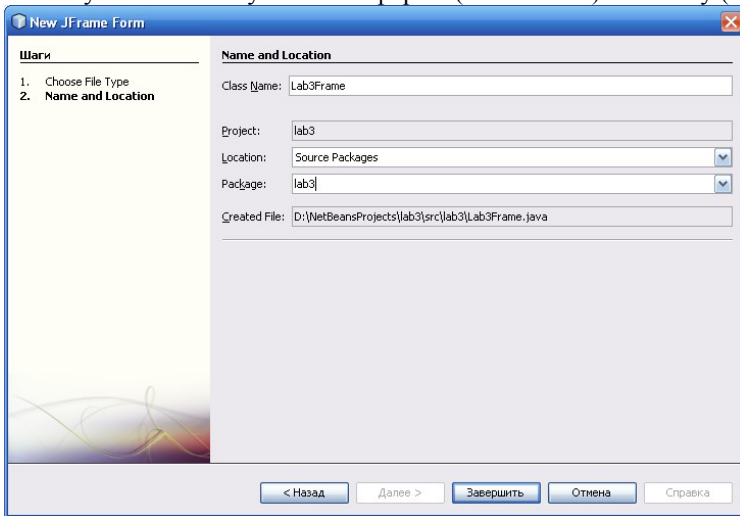
2. Вказуємо його ім'я, та розміщення. Знімаємо прапорець навпроти **Create Main Class**, залишаємо встановленим прапорець **Set as Main Project** та натискаємо **Finish** (Завершити):



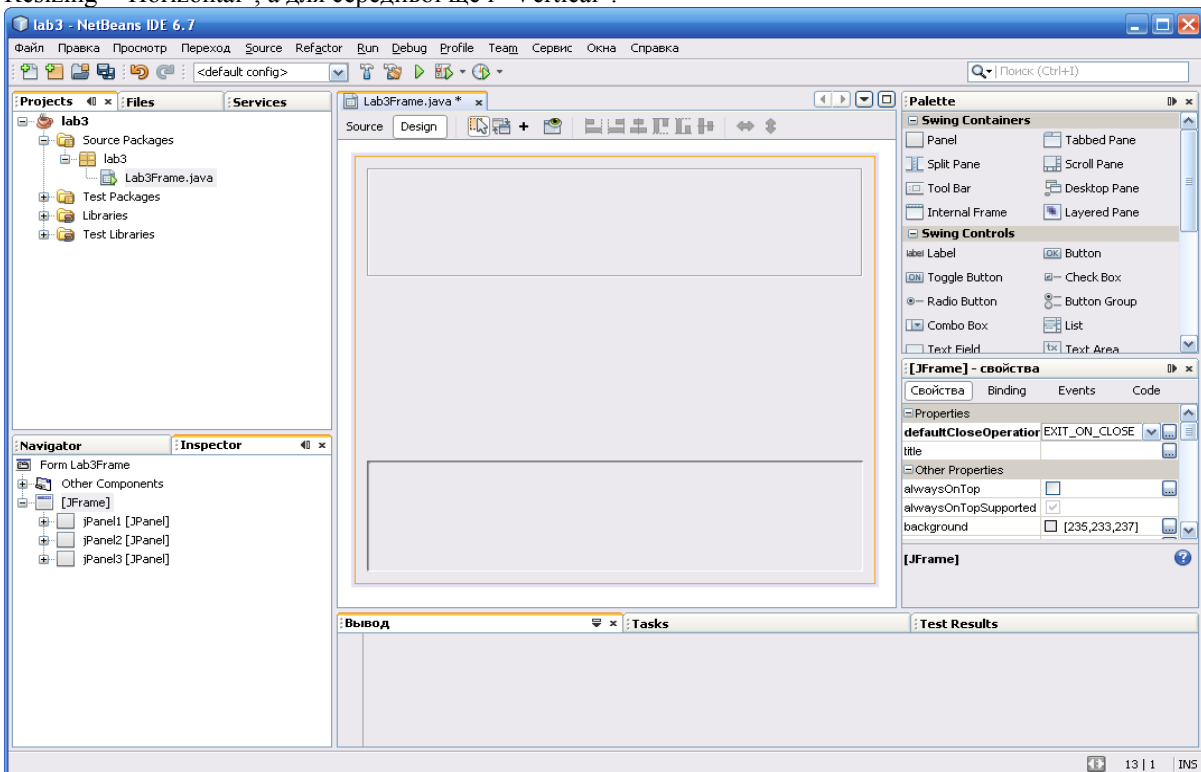
3. Додаємо до цього проекту новий файл типу JFrame Form з категорії Swing GUI Forms: (**File(Файл) -> New File...**). Натискаємо "Next >" (Далее >)...



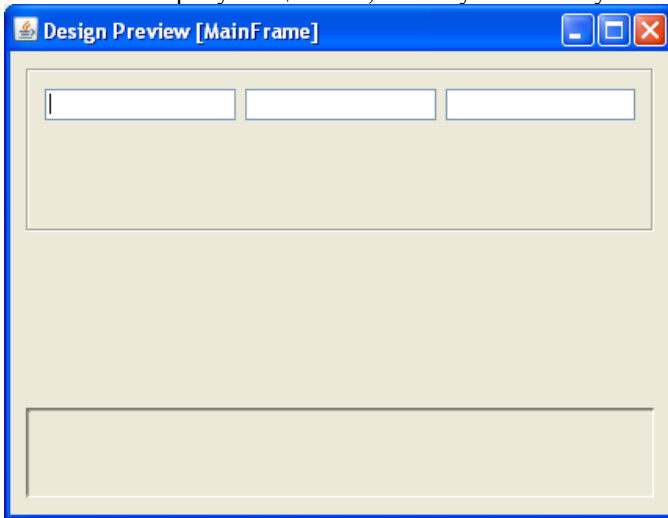
4. Вказуємо ім'я класу головної форми (Class Name) та пакету (Package), до якого ми її віднесемо:



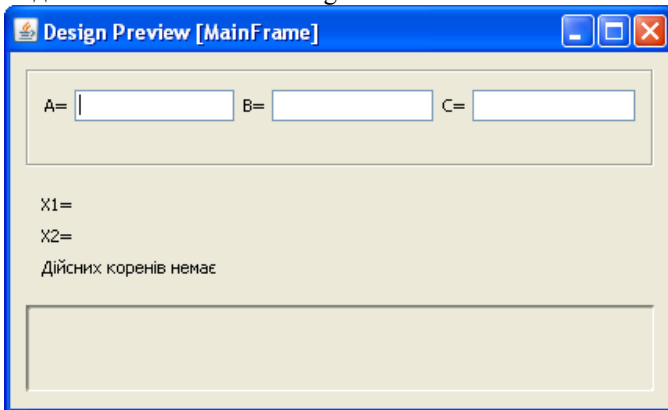
5. За допомогою компонентів JPanel розділяємо вікно на три горизонтальні області – панелі. При розміщенні панелей користуємося направляючими. Для верхньої панелі встановлюємо властивість Border у значення EtchedBorder, для середньої – залишаємо як є, а для нижньої обираємо SoftBevelBorder та у додатковому вікні тип границі – Lowered. Для всіх панелей у меню, що з'являється при натисканні правої кнопки миші обираємо "Auto Resizing ->Horizontal", а для середньої ще і "Vertical".



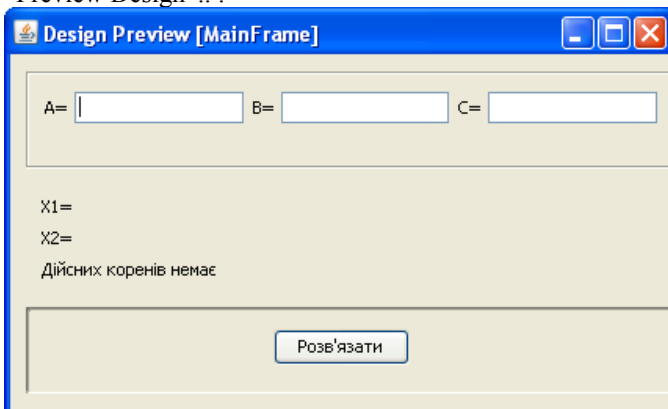
6. Розміщуємо на верхній панелі компоненти для введення коефіцієнтів А, В, С квадратного рівняння (JTextField), для них у вікні "Properties" очистити значення властивості "Text", та потім, за допомогою "Миші" відкоригувати їхні розміри у вікні так, щоб отримати результат, подібний до зображеного нижче. Крім того, оберіть ці компоненти та за допомогою правої кнопки миші у меню, що з'являється оберіть Auto Resizing -> Horizontal. Випробуйте це вікно, натиснувши кнопку "Preview Design".



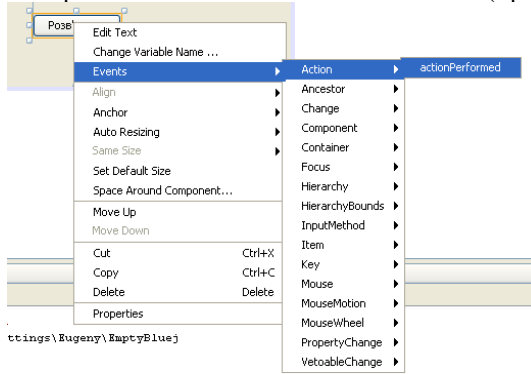
7. Розміщуємо на верхній панелі компоненти JLabel, міняємо їм властивість "Text" так, щоб вони підказували де вводити відповідні коефіцієнти, та на середній панелі такі ж компоненти для відображення результатів. Коригуємо розміри панелей так, щоб в них не залишалось "зайвого порожнього місця". Переглядаємо результат за допомогою "Preview Design":



8. Додаємо до нижньої панелі вікна кнопку (JButton) та змінюємо напис на ній. Далі, за допомогою правої кнопки миші у меню, що з'являється оберіть Auto Resizing -> Horizontal. Випробуйте це вікно, натиснувши кнопку "Preview Design".. :



9. Через контекстне меню кнопки JButton (права кнопка "миші") обираємо Events -> Action -> actionPerformed:



10. У редакторі коду описуємо поведінку програми при натисканні цієї кнопки – розв’язання задачі:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    /* отримуємо значення коефіцієнтів рівняння, введені користувачем */
    double a = Double.parseDouble(jTextField1.getText());
    double b = Double.parseDouble(jTextField2.getText());
    double c = Double.parseDouble(jTextField3.getText());

    /* обчислюємо дискримінант */
    double d = b * b - 4 * a * c;

    /* перевіряємо дискримінант на від'ємність */
    if (d < 0) {
        /* дійсних коренів немає - мітки для їхнього відображення ховаємо */
        jLabel4.setVisible(false);
        jLabel5.setVisible(false);
        /* робимо видимим напис "Дійсних коренів немає" */
        jLabel6.setVisible(true);
    } else {
        /* є дійсні корені - обчислюємо їх і показуємо */
        double x1 = (-b - Math.sqrt(d)) / (2 * a);
        double x2 = (-b + Math.sqrt(d)) / (2 * a);

        jLabel4.setText(String.format("X1 = %6.2f", x1));
        jLabel5.setText(String.format("X2 = %6.2f", x2));
        jLabel4.setVisible(true);
        jLabel5.setVisible(true);
        /* ховаємо напис "Дійсних коренів немає", оскільки вони є :) */
        jLabel6.setVisible(false);
    }
}
```

11. Запускаємо програму на виконання, та виконуємо її тестування.

**Спрощена ієрархія наслідування класів пакету Swing мови Java**

