

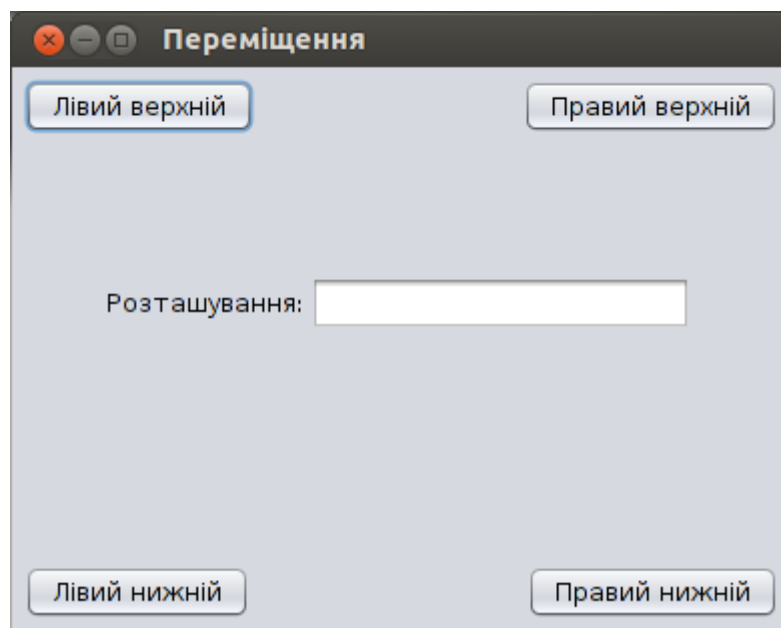
Робота з полями і кнопками

У цій роботі ви зробите три проекти, які допоможуть вам засвоїти використання простих візуальних елементів: полів та кнопок. Перший проект навчить вас використовувати координатну графічну систему екрану і за допомогою кнопок переміщати форму в різні кути екрану. У другому проекті ви зробите простий калькулятор, у якому треба навчитися перетворювати символічні дані, що вводяться в текстові поля, у числові дані і виконувати з ними необхідні арифметичні операції. Третій проект схожий на створення стандартного калькулятора операційної системи, в якому слід навчитися вводити дані в текстове поле натисканням різних кнопок.

I

1.1. Створіть новий проект з ім'ям **Lab2.1**, не створюючи головний клас (вимкніть прапорець “Создать главный класс”). Створіть нову порожню форму з ім'ям **MovableForm** як основу проекту і відразу ж у властивостях проекту в категорії “Выполнить” вкажіть клас цієї форми як головний клас для подальшого запуску проекту.

1.2. Розташуйте об'єкти Кнопка і Текстовое поле з написом (міткою) ліворуч, як показано на наступному рисунку:



Заголовок форми “Переміщення” визначте за допомогою властивості **title**.

1.3. Запрограмуйте кнопки таким чином, щоб при їх натисканні форма переміщувалась у відповідний кут екрану.

Нагадаємо, що центр координатної системи екрану (0, 0) розташований у лівому верхньому куті екрану. Вісь абсцис (X) спрямована направо, а вісь ординат (Y) — вниз. Тобто точка з координатами (100, 150) відносно до лівого верхнього кута знаходиться на 100 пікселів праворуч і 150 пікселів нижче. Координати будь-якого прямокутного об'єкту також прив'язуються до його лівого верхнього куту. Тобто, якщо говорять, що форма розташована в координатах (100, 150), це означає, що її лівий верхній кут знаходиться у цій точці.

Для програмування кнопок використовуйте подію (События) Action — `actionPerformed`, яка схожа на `Mouse — mouseClicked`, але остання реагує тільки на клацання миші, а перша — як на мишу, так і на клавішу пробілу.

Розташування форми задається функцією `setLocation(x,y)`; тому для програмування кнопки “Лівий верхній” до коду події достатньо вписати команду `setLocation(0, 0)`;

1.4. Для програмування інших кнопок вам знадобляться функції для отримання ширини/висоти екрану і форми. Наприклад, для форми, яка розташується у правому верхньому куті екрану, верхній лівий кут знаходиться у координатах (ширина_екрану – ширина_форми, 0), а форма у лівому нижньому куті — у координатах (0, висота_екрану – висота_форми).

Для отримання розмірів форми вам знадобиться функція **getSize()**, що повертає розмір форми у вигляді об'єкта класу **java.awt.Dimension**, який містить наступні властивості:

width – ширина;

height – висота.

Для розмірів форми використовується об'єкт класу **java.awt.Rectangle**, який має такі ж властивості, що і об'єкт класу **java.awt.Dimension**.

Щоб використовувати ці об'єкти, слід оголосити змінні цих класів. Рекомендуємо зробити це наприкінці класу форми, відразу ж після блоку змінних, що відповідають вашим кнопкам і текстовому полю. Це може виглядати, наприклад, так:

```
...

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JLabel jLabel1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

private java.awt.Dimension f; // змінна f для розмірів форми

private java.awt.Rectangle e; // змінна e для розмірів екрану

}
```

Тоді перед використанням функції **setLocation(x,y)** слід отримати розміри форми (змінній **f** присвоїти результат функції **getSize()**):

```
f = getSize();
```

Аналогічно можна отримати розміри екрану у змінну **e**, тільки замість функції **getSize()** слід використати більш складну конструкцію:

```
java.awt.GraphicsEnvironment.getLocalGraphicsEnvironment().getMaximumWindowBounds();
```

Тоді різницю ширин можна отримати як

```
e.width – f.width
```

а різницю висот — як

```
e.height – f.height
```

і підставити ці вирази у **setLocation(x,y)** замість відповідних координат.

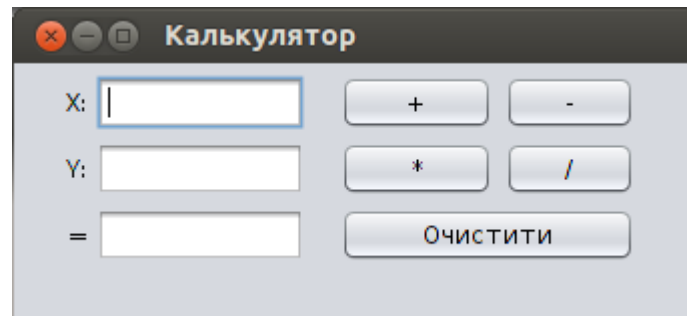
1.5. В центрі форми розташоване текстове поле **jTextField1** класу **JTextField**. Необхідно, щоб при переході в різні кути в цьому текстовому полі відображалось її положення (наприклад, “правий-нижній кут”). Це

можна зробити аналогічно тому, як ви виводили привітання у першій лабораторній роботі. В початковий момент поле повинно бути пустим.

Перевірте роботу форми при натисканні усіх кнопок.

II

2.1. Створіть другий проект з ім'ям **Lab2.2** повністю аналогічно проекту **Lab2.1**. Далі створіть форму з ім'ям **CalcForm** (не забувайте встановити “Головний клас проекту”) і розташуйте поля, мітки і кнопки приблизно наступним чином:



Необхідно так запрограмувати кнопки з арифметичними діями, щоб при введенні чисел в текстові поля X, Y результат був записаний у полі з міткою =. Тобто треба створити простий калькулятор.

2.2. Для зручності змініть назви змінних для текстових об'єктів. Для цього виділіть відповідний об'єкт і в контекстному меню виберіть пункт **Изменить имя переменной**. Вкажіть змістовні імена, наприклад: **xText**. Враховуйте, що прийнято назви змінних починати з маленької літери, а складові слова з великої. Також змініть назви кнопок так, щоб потім в тексті програми було легко визначити за їх назвою, яку дію треба виконати. Нагадаємо, що англійською додавання - **addition**, віднімання - **subtraction**, множення - **multiplication**, ділення - **division**, очистити — **clear**, тобто назви кнопок **addButton**, **subButton** і т.д.

Оскільки текстові поля зберігають свої значення у вигляді тексту в об'єктах **String** (назви класів, навпаки, прийнято починати з великої літери), а арифметичні дії допустимі для числових типів, зокрема **float** (це простий тип, не клас, тому з маленької літери), то перед виконання обчислень, необхідно перетворити текстове значення на числове. В Java для всіх простих типів (які не можуть динамічно розміщуватись в пам'яті) створені відповідні класи, які мають ту саму назву, але, як класи, з великої літери. Такі класи зазвичай мають всі необхідні методи для обробки відповідних типів, тому при необхідності, дивіться документацію саме по класу, а не типу. В нашому випадку для перетворення тексту в число з плаваючою комою потрібна наступна конструкція **Float.parseFloat(String s)**. В якості аргументу s має бути текстове значення.

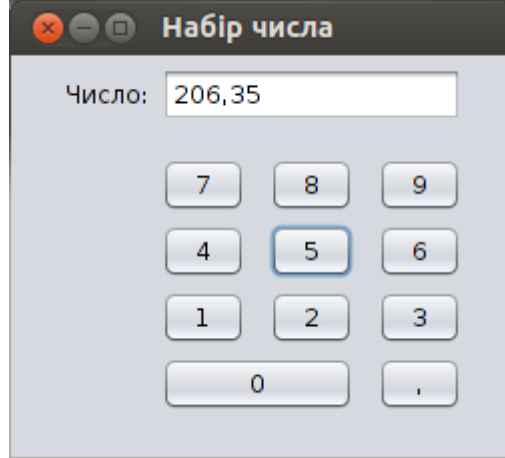
Значення тексту в текстових полях типу **JTextField** можна одержати за допомогою методу **getText()**, наприклад, **xText.getText()**.

Після того, як ви одержите числове значення результату, його необхідно перетворити в зворотну сторону: з числа на текст за допомогою конструкції **String.valueOf(Float f)**.

2.3. Додайте обробку кнопки «Очистити», так, щоб в усіх текстових полях зникав текст (встановити у поля порожній "" текст).

III

3.1. Створіть третій проект з ім'ям **Lab2.3** аналогічно попереднім проектам. Далі створіть форму з ім'ям **ButtonsForm** (не забувайте встановити “Головний клас проекту”) і розташуйте поле, мітку і кнопки приблизно наступним чином:



Запрограмуйте кнопки таким чином, щоб з їх допомогою можна було набирати число так, як це відбувається в звичайних калькуляторах.

Звіт по роботі

У текстовому файлі (формату .odt) після заголовку роботи по пунктах вказати послідовність дій для виконання поставлених завдань у кожному проекті. При цьому слід привести ті команди, які ви додавали до кожного потрібного методу, коментуючи кожен з команд. Рисунки з екрану вставляти у звіт не потрібно.

В кінці звіту запишіть висновки: які основні принципи роботи з координатною системою, перетворення текстових значень у числові, а також введення символів у поле, не стираючи попередніх, ви засвоїли з цієї роботи.

Папки з робочими проектами слід також заздалегідь надіслати до університету і розташувати її у потрібному місці.

Звіт надіслати викладачеві для перевірки до дня наступної роботи.