Лабораторна робота № 3

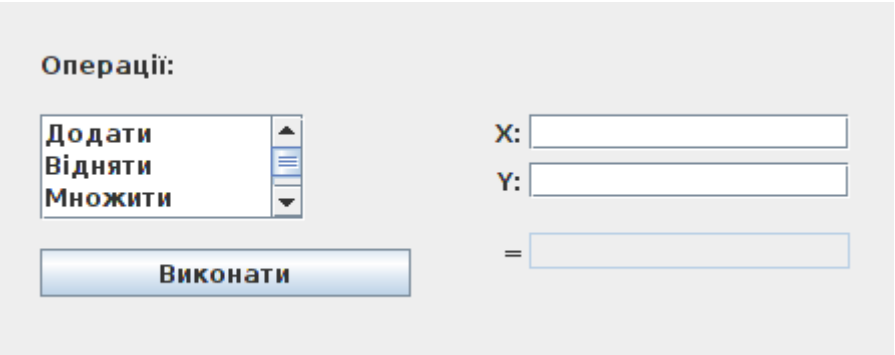
Робота зі списками

Основна мета роботи — навчитися використовувати елементи *Список* і *Поле зі списком* для формування певного списку текстових елементів та маніпулювання ними під час роботи програми. Для цього ви зробите 2 невеличких проекти. У першому ми знову повторимо простий калькулятор, аналогічний 2-му проекту у 2-й лабораторній роботі. Але тепер кожна операція буде виконуватися не окремою кнопкою, а за допомогою єдиної кнопки “Виконати”. Потрібну операцію будемо обирати за допомогою списку елементів (назв операцій). Другій проект — більш складний. За допомогою двох полів зі списком ми реалізуємо простий українсько-російський словник, який можна буде коригувати, використовуючи певні кнопки-операції.

I

1.1. Створіть новий проект (ім’я проекту визначте самостійно), не створюючи головний клас (вимкніть прапорець “Создать главный класс”). Створить нову порожню форму як основу проекту (визначте для неї зрозуміле ім’я) і відразу ж у властивостях проекту в категорії “Выполнить” вкажіть клас цієї форми як головний клас для подальшого запуску проекту.

1.2. Скомпонуйте усі елементи проекту приблизно наступним чином:



1.3. Перейменуйте елементи так, щоб їх імена були більш зрозумілими. Щоб у полі з результатом було неможливо виконувати будь-яке редагування, вимкніть його властивість *editable* (можливість редагування). Фоновий колір цього поля (*background*) можна зробити світло-сірим. Це буде нагадувати про неможливість редагування всередині поля.

1.4. Список операцій реалізується компонентом *Список*. Взагалі то цей компонент складається з двох компонентів: панелі прокрутки *JScrollPane* (для можливості прокрутки елементів, які не потрапляють у видиму область списку) і власне самого списку *JList,* який розташовується поверх панелі прокрутки (від панелі прокрутки ви бачите тільки смугу прокрутки, але і її можна не побачити, якщо висота/ширина списку дозволяє бачити в режимі Конструктора усі елементи). Властивості панелі прокрутки можна побачити, якщо клацнути по смузі прокрутки. Якщо слід налаштувати смуги прокрутки, а вони відсутні у режимі конструктора, необхідно зменшити висоту (ширину) прямокутника списку до їх появлення. Після створення список за замовченням буде заповнений елементами Item1, Item2, ... (Елемент1, Елемент2, ...). Для зміни списку елементів (вам потрібен список арифметичних операцій) скористуйтесь властивістю списку *model*. Для зручного редагування списку елементів скористайтеся кнопкою “...” поруч зі значенням цієї властивості. Ще одна важлива властивість списку — *selectionMode* — режим вибору, може мати наступні значення: SINGLE (одиночне значення), SINGLE_INTERVAL (одиночний інтервал — декілька елементів підряд обираються за допомогою клавіші Ctrl), MULTIPLE_INTERVAL (багатоінтервальний вибір — елементи можна обирати у різних інтервалах, також за допомогою клавіші Ctrl). Встановіть потрібне, на ваш погляд, значення.

1.5. Реалізуйте виконання вибраних в списку арифметичних дій при натисканні на кнопку Виконати. Основна робота з елементами списку здійснюється за допомогою їх порядкових номерів — індексів. Якщо отримати індекс за допомогою методу *getSelectedIndex()* списку, можна визначити, який елемент зараз вибраний користувачем (номер

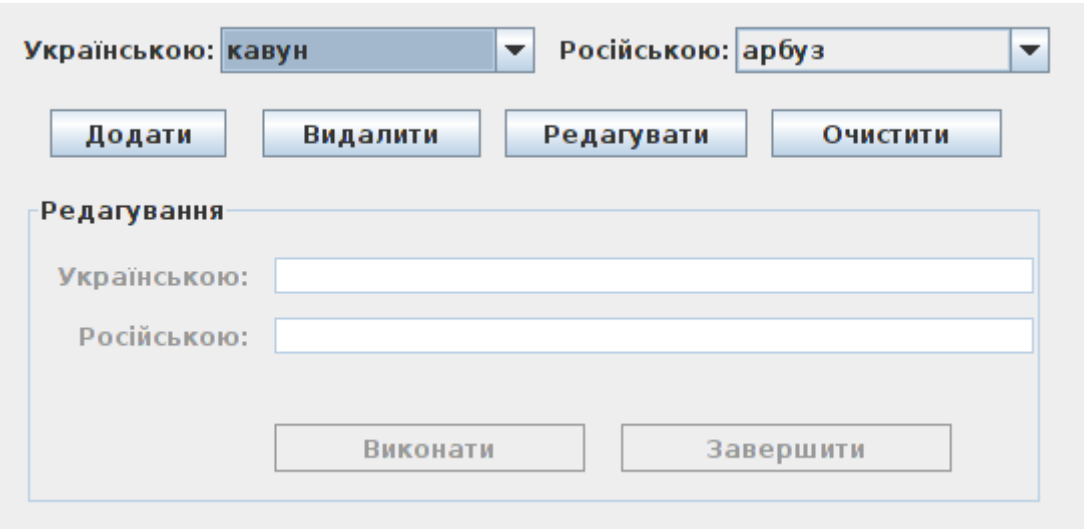
першого елементу починається з нуля, якщо немає вибраних елементів, індекс = -1). Якщо ж змусити список виконати метод *setSelectedIndex(номер),* відповідний елемент списку стане вибраним. Рекомендуємо запрограмувати стандартний метод кнопки *actionPerformed*, у якому спочатку отримати текстові значення полів *X* та *Y*, перетворити їх у відповідні числові (float) значення локальних змінних *x* і *y*, оголосити локальну змінну для результату, потім отримати індекс обраного елементу зі списку операцій і зберегти його у якійсь локальній змінній (int). Нарешті можна визначити результат операції за обраним індексом операції. Для цього доцільно скористатись оператором розгалуження *switch:*

```
switch (вираз) {
  case значення1: програмний код1; break;
  case значення2: програмний код2; break;
  …
  case значенняN: програмний кодN; break;
  default: програмний код, що виконується,
           коли не виконались попередні умови;
}
```

Виконання арифметичних дій здійснюється аналогічно до попередньої лабораторної роботи. Наприкінці не забудьте помістити текст результату у відповідне текстове поле. Робіть це тільки у тому випадку, коли користувач дійсно не забув вибрати операцію у списку, тобто індекс >= 0.

II

2.1. Створіть наступний проект з новою формою приблизно наступного виду (словник англійською — dictionary):



2.2. Як завжди, надайте елементам нові зрозумілі імена. Елементи списків українських і російських слів — це Поля со списком. Налаштуйте їх схожим способом, як звичайний Список. Властивості Поля со списком selectedIndex і selectedItem вказують на елемент, який буде обрано відразу ж після запуску програми. У нас це — перший елемент (індекс 0).

2.3. Основний принцип роботи словника — однакові індекси українських і відповідних російських слів. При виборі українського слова автоматично у другому списку повинне встановитися відповідне російське і навпаки. Програмування події вибору елемента зі списку повністю аналогічне програмуванню натискання на кнопку, тобто для обох полів зі списком слід запрограмувати подію *actionPerformed*. У цьому методі слід спочатку отримати індекс свого обраного елементу, а потім для іншого поля зі списком встановити елемент у цей же індекс (методи *getSelectedIndex()* і *setSelectedIndex(номер)*).

2.4. Розмістіть кнопки на формі поки що без програмування. Потім розташуйте рамку з написом “Редагування”. Рамка — це елемент *JPanel*, на якому можуть розміщатися інші елементи, тобто це — контейнер. Шукайте на Палітрі елемент Панель у секції “Контейнеры Swing”. Далі слід налаштувати властивість *border*

(границя) цієї Панелі. Виберіть тип рамки “Рамка с надписью” (TitledBorder) і встановіть властивість Заголовок — “Редагування”. Далі можна розміщати поля і кнопки на цієї Панелі.

2.5. Після запуску проекту за замовченням режим редагування словника вимкнутий. Тому усі елементи на панелі Редагування повинні бути деактивовані (пригашені). Взагалі то це можна зробити за допомогою властивості *enabled* (дозволений) кожного окремого елементу. Однак нам прийдеться це робити неодноразово та ще і прямо під час роботи програми. Тому зробимо це програмно. Це можна зробити за допомогою циклу for, перебираючи усі компоненти контейнера (Панелі — рамки з написом) і встановлюючи їх властивість *enabled* в значення *false* (хибність):

```
for (int i = 0; i < jPanel1.getComponentCount(); i++) {
  jPanel1.getComponent(i).setEnabled(false);
}
```

Тут *jPanel1* — наша рамка з написом; метод *getComponentCount()* повертає кількість компонентів на панелі; метод *getComponent(i)* повертає об’єкт, відповідний і-му компоненту; метод *setEnabled(false)* встановлює властивість *enabled* для цієї і-й компоненти. Такий цикл можна встановити тільки всередині того методу, який відповідає за початкову ініціалізацію усіх об’єктів форми. Такий метод є у кожного класу. Він називається *конструктором класу* і його ім’я повинне співпадати з назвою самого класу. Наприклад, якщо клас вашої форми називається DicForm, слід шукати всередині нього метод DicForm():

```
public DicForm() {
  initComponents();
}
```

Функція *initComponents()* саме і виконує таку ініціалізацію всіх об’єктів форми. Безпосередньо після цієї функції слід додати у конструктор класу форми наведений вище цикл деактивації елементів рамки з написом. Потім у потрібному місці можна навести такий же цикл і навпаки активувати усі елементи значенням *true* (істина).

2.6. Кнопка “Виконати” буде працювати як для режиму “Додати” (умовний код 1), так і для “Редагувати” (умовний код 2). Тому при натисканні кнопок “Додати” і “Редагувати” слід запам’ятати у глобальній змінній умовний код режиму. Таку змінну слід оголосити наприкінці класу форми (після блоку оголошення змінних для полів, кнопок і т.ін.), наприклад: *private int opMode = 0;*

Додавання нових слів у словник реалізуйте наступним чином: при натисканні на кнопку “Додати” панель “Редагування” повинна бути розблокована (активована). Після того, як користувач введе обидва варіанта слів, він повинен натиснути кнопку “Виконати” (тут як раз і можна за допомогою оператору *switch* аналізувати значення змінної *opMode*), після чого слова додаються в списки, що випадають, а текстові поля очищаються. Для додавання елементу в список, що випадає, скористайтеся його методом *insertItemAt(текст, позиція),* який додає елемент з вказаним текстом у вказану позицію. Зазвичай нові елементи додаються у кінець списку, тобто під номером, на 1 більшим номера останнього елементу. Для цього скористайтеся методом *getItemCount()* списку, який повертає кількість елементів списку. До речі, це і буде потрібний нам номер, оскільки нумерація елементів починається з нуля і номер останнього елементу буде на 1 меншим за кількість елементів. При необхідності зберігання отриманого значення для подальшої операції використовуйте відповідні локальні змінні. Таким чином, послідовність дій у режимі додавання буде такою:

- отримати індекс для вставки нового елементу (тобто кількість елементів);

- вставити українське слово в перший список під цим індексом;

- вставити російське слово в другий список під цим же індексом;

- очистити обидва поля;

- для обох полів зі списком встановити активний індекс на індекс з доданим словом (після додавання списки відразу ж показуватимуть додані слова).

2.7. Для режиму редагування кнопка “Редагувати” повинна розблокувати елементи панелі “Редагування” і помістити поточні слова списків у відповідні поля. Спочатку отримайте індекс вибраних елементів, наприклад, у змінну *i*. Отримати текст елементу списку (наприклад, ім’я списку - *ukrList*) за його індексом можна наступним чином: *ukrList.getItemAt(i).toString()*, тобто ви отримуєте сам елемент за його індексом і перетворюєте його у

текстове (*toString*) значення. Після редагування і натискання кнопки “Виконати” (відповідний *case* оператора *switch*) слід виконати таку послідовність дій:

- отримати індекс (номер) поточного активного слова (з будь-якого списку);

- видалити елементи з цим номером з обох списків за допомогою методу *removeItemAt(номер)* списку;

- додати текстові значення обох слів з відповідних полів до обох списків у ту ж саму позицію, під якою ви тільки що видаляли старі елементи (аналогічно як додавати у кінець списку);

- встановити активний поточний індекс обох списків у цей же самий номер.

2.8. Кнопка “Завершити” повинна очистити поля з українським і російським текстами і деактивувати усі елементи панелі “Редагування”.

2.9. Кнопка “Видалити” повинна просто видалити поточну пару слів з обох списків (метод *removeItemAt(номер)* за поточним номером).

2.10. Кнопка “Очистити” повинна повністю видалити усі слова з кожного поля зі списком. Для цього можна використати метод *removeAllItems()* для кожного поля зі списком.

2.11. Зробіть своєрідний захист операцій “Додати” і “Редагувати” від порожніх значень. Тобто, якщо хоча б одне з полів (українське або російське) є порожнім, операція додавання або редагування не повинна виконуватися. Для того, щоб перевірити, чи є пустим значення текстового поля, можна використати наступний метод текстового поля: *getText().equals("")*. Він повертає *true* (істину), якщо текст у полі дійсно є порожнім значенням.

Звіт по роботі

Звіт підготувати у формі, аналогічній звіту попередньої роботи. Тобто у текстовому файлі (формату .odt) після заголовку роботи по пунктах вказати послідовність дій для виконання поставленого завдання. При цьому слід привести ті команди, які ви додавали до кожного потрібного методу, коментуючи кожну з команд. Рисунки з екрану вставляти у звіт не потрібно.

В кінці звіту запишіть висновки: які основні принципи роботи зі списками ви засвоїли з цієї роботи.

Папку з робочим проектом слід також заздалегідь надіслати до університету і розташувати її у потрібному місці.

Звіт надіслати викладачеві для перевірки до дня наступної роботи.