

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

МЕТОДИЧНІ ВКАЗІВКИ ДО КУРСОВОЇ РОБОТИ З ДИСЦИПЛІНИ  
"ПРИКЛАДНЕ ПРОГРАМУВАННЯ JAVA"  
для студентів спеціальності 122 Комп'ютерні науки

Укладач  
проф. Льїн О.О.

Київ 2020

## 1. Вступ

В межах курсової роботи з дисципліни "Прикладне програмування Java" студентам пропонується створити прикладний додаток з графічним інтерфейсом користувача. Концептуально, додаток являє собою прототип геоінформаційної системи, дозволяє відобразити карту місцевості, позначки різних за обсягом подій, здійснювати моделювання появи подій, здійснювати статистичні обрахунки та відобразити результати у графічному виді. Вікно типового додатку зображене на рис.1.

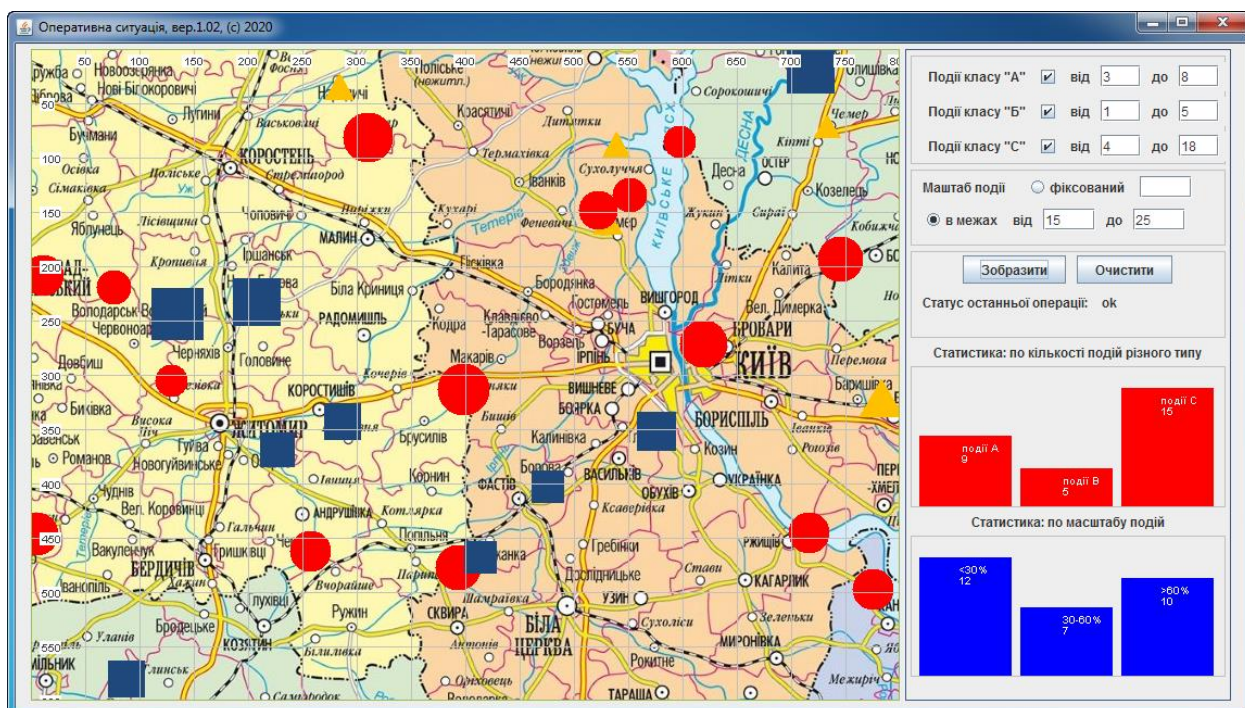


Рисунок 1. Вікно додатку з позначками подій та статистичною інформацією

Реалізація кожним студентом власного програмного додатку відповідно до сформульованого технічного завдання, дозволяє закріпити отримані у курсі "Прикладне програмування Java" загальні та професійні компетентності, а також компетентності, які сформульовані відповідно до вимог роботодавців, продемонструвати програмні результати навчання.

## 2. Технології для застосування

Базовою технологією, яка застосовується в роботі, є Java. Для організації роботи важливе значення має необхідне програмне забезпечення розробника. Його основу складає JDK(1.8), середовище розробника IDE Eclipse або Netbeans. Ці засоби необхідні для створення коду програми, її компіляції та запуску. Проектування графічного інтерфейсу користувача необхідно здійснювати без візуального редактора, за допомогою графічного фреймворку Swing.

### 3. Опис роботи додатку

Програмний додаток *логічно* складається з чотирьох блоків:

1. Блок обробки вхідних даних.
2. Блок моделювання.
3. Блок візуалізації даних про події в графічному виді.
4. Блок статистичної обробки даних та її візуалізації.
5. Блок інтерактивної взаємодії із користувачем.

**Блок обробки вхідних даних** призначений для інтерпретації вхідних даних та їх підготовки для зображення на карті місцевості у вигляді замкнених графічних примітивів (коло, квадрат, трикутник), які відповідають подіям трьох типів - умовна подія «А», умовна подія «Б», умовна подія «С». Координати центру фігур на карті відповідають координатам фіксування подій (використовується декартова система координат на площині). Площі фігур пропорційні масштабу подій. Внаслідок відсутності реального джерела даних про події, в роботі пропонується застосувати **блок моделювання**, який на основі вказаних користувачем налаштувань, генерує дані про положення подій та їх масштабів. Кількість подій кожного типу генеруються довільним чином, відповідно до налаштувань. Координати центру події (x та y) також генеруються довільним чином, але в межах зображення карти. Масштаб події характеризується її площею, задається радіусом, і призначається одним для всіх подій, або генерується довільним чином у вказаних користувачем межах.

Отриманий набір даних про події передається у **блок візуалізації даних у графічному виді**. Блок призначений для відображення карти місцевості, координатної сітки із позначками положень горизонтальних та вертикальних ліній сітки, а також відображення позначок подій, відповідно до наданих даних. Позначки подій відображаються поверх карти місцевості, але під координатною сіткою. Карта місцевості має відобразитись при запуску програми.

Вхідні дані, одночасно із відображенням на карті місцевості, передаються до **блоку статистичної обробки інформації та її візуалізації**, яка складається із двох діаграм. Перша стовпчикова діаграма відображає кількість подій за типами. Друга стовпчикова діаграма відображає масштаб подій, але відповідно до наступних правил: в перший стовпчик виводиться кількість подій (будь-яких типів), але які за масштабом становлять не перебільшують 30% розміру події максимального масштабу; в другий стовпчик виводяться події за масштабом від 30% до 60% від події максимального масштабу; і в третій стовпчик виводяться кількість подій від 60% до 100% від події максимального масштабу. Масштаб події обраховується як площа фігури, за відомим радіусом.

**Блок інтерактивної взаємодії із користувачем** призначений для управління роботою програми. Він складається із графічних компонентів для вводу даних, інформування про похибки, допущені при вводі даних, кнопок для управління роботою додатку. Дані про тип подій, їх бажану кількість, їх

масштаб, необхідно задавати у чисельному вигляді та за допомогою відповідних кнопок вибору варіантів. Для зображення даних про події на карті призначена кнопка «Зобразити», для видалення всіх подій з карти - кнопка «Очистити». При натискуванні на кнопку «Зобразити» перевіряються всі обрані пункти, введені дані, і у випадку відсутності необхідних налаштувань або похибки у даних, користувача надається попередження про тип похибки і місце її виникнення. При успішно пройденій перевірці, події відображаються на карті і зображуються відповідні статистичні діаграми. При повторному натисненні на кнопку «Зобразити» до вже існуючого набору даних додаються нові дані, згенеровані у блоці моделювання відповідно до поточного налаштування. Утворений масив даних відображається на карті і у блоці статистики. При натисненні на кнопку «Очистити» всі дані видаляються, позначки даних та статистичні діаграми видаляються з екрану також. При цьому зберігаються налаштування у блоці інтерактивної взаємодії.

Принцип роботи додатку можна подивитись на відповідному відео, посилання на яке розміщено у електронному курсі.

#### **4. Опис курсової роботи у документації (опис роботи)**

Разом із розробкою програмного додатку, необхідно розробити опис курсової роботи. Він представляє собою документ у паперовому виді, із титульною сторінкою, як зразку у додатку. А. Опис курсової роботи містить формулювання завдання щодо розробки, порядок розробки, необхідні пояснення разом із UML-діаграмами, скріншоти вікон додатку, **коди всіх класів java.**

Опис складається з кількох розділів. У розділі №1 наводиться отримане завдання. В розділі №2 описується математична модель, використана для формування вхідних даних та опису масштабу подій. Доцільно описати, яким чином отримуються вхідні дані для події кожного типу, яким чином обраховується масштаб події, разом із формулами, потрібними ілюстраціями та окремими фрагментами коду на java. В розділі №3 наводиться зображення графічного інтерфейсу розробленого додатку, описуються функціональні можливості додатку, UML-діаграми. У висновку підводяться підсумки виконаної роботи, відповідно до основних етапів розробки. В розділі №4 наводяться коди всіх класів java, їх UML-діаграми.

Обсяг опису роботи, без кодів програм - не більше 10 сторінок.  
Порядок слідування частин роботи:

- 1) титульний лист (додаток А даних методичних рекомендацій);
- 2) зміст роботи;
- 3) розділ 1,2,3,...
- 4) бібліографічний перелік літератури або посилань Інтернет;

Робота оформлюється на листах А4, нумерується (перший лист без номера), скріплюється степлером з боку більшого відступу (ліве поле).

Для основного тексту роботи використовується шрифт Times New Roman 12 пт. Для програмного коду – моношириний шрифт Courier New 10пт.

Код програми має містити коментарі.

## 5. Порядок захисту роботи

Для захисту курсової роботи, її опис разом із кодом програмного додатку, має бути заздалегідь (за 3 дні до захисту) завантажений через систему електронного навчання. Код програми має бути в архіві zip, у вигляді папки src та img, в яких розташовані інші пакети та коди класів java, використані зображення карт (у папці img). Оцінка складається з кількох частин:

Що оцінюється	Макс. бал
1. Функціонування програмного додатку	40
2. Відповідність вимогам щодо дизайну та принципу роботи	35
3. Відповіді на питання по внутрішньому устрою та принципу роботи додатку (питання по коду java, наведеному у описі)	25

# ДОДАТКИ

*Додаток А*

## **Приклад оформлення титульного листа курсової роботи**

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

### **Курсова робота**

з дисципліни "Прикладне програмування Java" на тему: розробка програмного додатку моделювання та візуалізації подій

Роботу виконав здобувач вищої освіти Іванов В.О.

Перевірив: \_\_\_\_\_

Київ 2020

### Приклад схеми екранної форми додатку

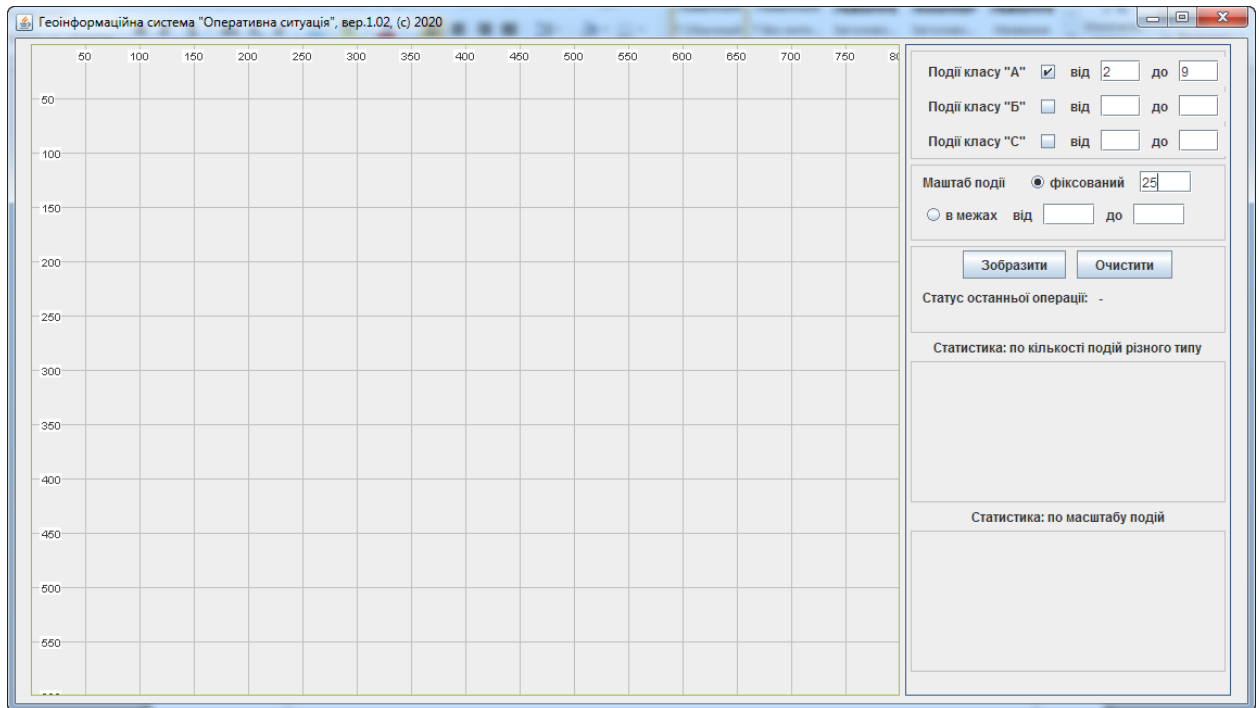


Рисунок 5. Екранна форма додатку із введеними даним та результатом їх обробки

## Приклад оформлення коду програми

```
260private Stuff getStuff() {
261     Stuff e = new Stuff();
262
263     // check if all fields are filled and correct
264     boolean isAllFieldsCorrect = true;
265     if (lastNameRowTextField.getText().equals(""))
266         isAllFieldsCorrect = false;
267     if (firstNameRowTextField.getText().equals(""))
268         isAllFieldsCorrect = false;
269     if (ageRowTextField.getText().equals(""))
270         isAllFieldsCorrect = false;
271     if (adressRowTextField.getText().equals(""))
272         isAllFieldsCorrect = false;
273     if (cityRowTextField.getText().equals(""))
274         isAllFieldsCorrect = false;
275     if (contactsRowTextField.getText().equals(""))
276         isAllFieldsCorrect = false;
277
278     if (isAllFieldsCorrect) {
279
280         e.setId(idRowTextField.getText().equals("") ?
0 : Integer.parseInt(idRowTextField.getText()));
281         e.setLastName(lastNameRowTextField.getText());
282
283         e.setFirstName(firstNameRowTextField.getText());
284
285         e.setAge(Integer.parseInt(ageRowTextField.getText()));
286         e.setAddress(adressRowTextField.getText());
287         e.setCity(cityRowTextField.getText());
288         e.setPhone(contactsRowTextField.getText());
289         e.setGender(rbMale.isSelected() ? 1 : 2);
290     } else
291         e = null;
292
293     return e;
294 }
```