

№2 Тема: Консольні програми для платформи Java SE. Прості алгоритми обробки даних.

Мета: Розробка простих консольних програм для платформи Java SE.

1 Індивідуальне завдання

1.1 Розробник

Студент Малохвій Едуард Едуардович, КІТ-26А, Варіант 8 (Завдання №6).

1.2 Вимоги

- Розробити та продемонструвати програму мовою Java в середовищі Eclipse для вирішення прикладної задачі за номером, що відповідає збільшеному на одиницю залишку від ділення на 10 зменшеного на одиницю номера студента в журналі групи.
- Для визначення вхідних даних використовувати генератор псевдовипадкових чисел (`java.util.Random`) та забезпечити циклічне (принаймні десять ітерацій) знаходження результату рішення прикладної задачі.
- Забезпечити виведення до консолі відповідних значень вхідних даних та результатів обчислень у вигляді таблиці. 4.Забороняється використання даних типу `String` та масивів при знаходженні рішення прикладної задачі. 5.Застосувати функціональну (процедурну) декомпозицію і забезпечити рішення прикладної задачі за допомогою відповідних статичних методів.

1.3 Завдання

Перевірити чи дорівнює сума перших трьох цифр сумі останніх трьох цифр в десятковому запису 6-значного цілого числа.

2 Розробка програми

2.1 Засоби ООП

Під час вирішення поставленої задачі було акцентовано увагу на інкапсуляції процесу обробки та отримання вхідних даних. Тобто об'єкт циклу

подій оброблює контейнер даних який забезпечує загальний інтерфейс для певного набору методів (отримання, обробку, відображення). Об'єкт циклу лише контролює послідовність подій під час виконання програми, за обчислення відповідає лише контейнер даних.

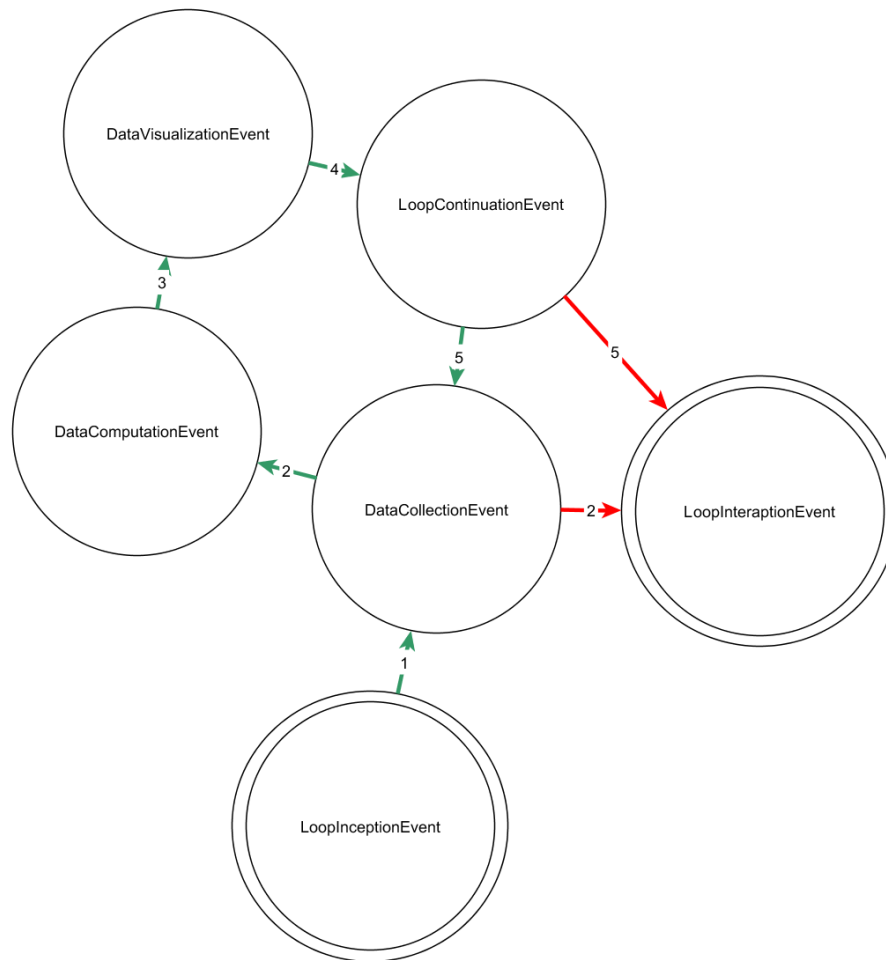


Рис. 1 - Діаграма взаємодії подій

2.2 Ієрархія та структура класів

Проект містить наступні пакети:

- data - містить у собі реалізацію логіки обробки даних
- event - містить у собі реалізацію логіки обробки подій, їх збереження то контролю.

Нижче наведено діаграму класів отриману за допомогою розширення для середовища розробки Eclipse - ObjectAid UML Diagram.

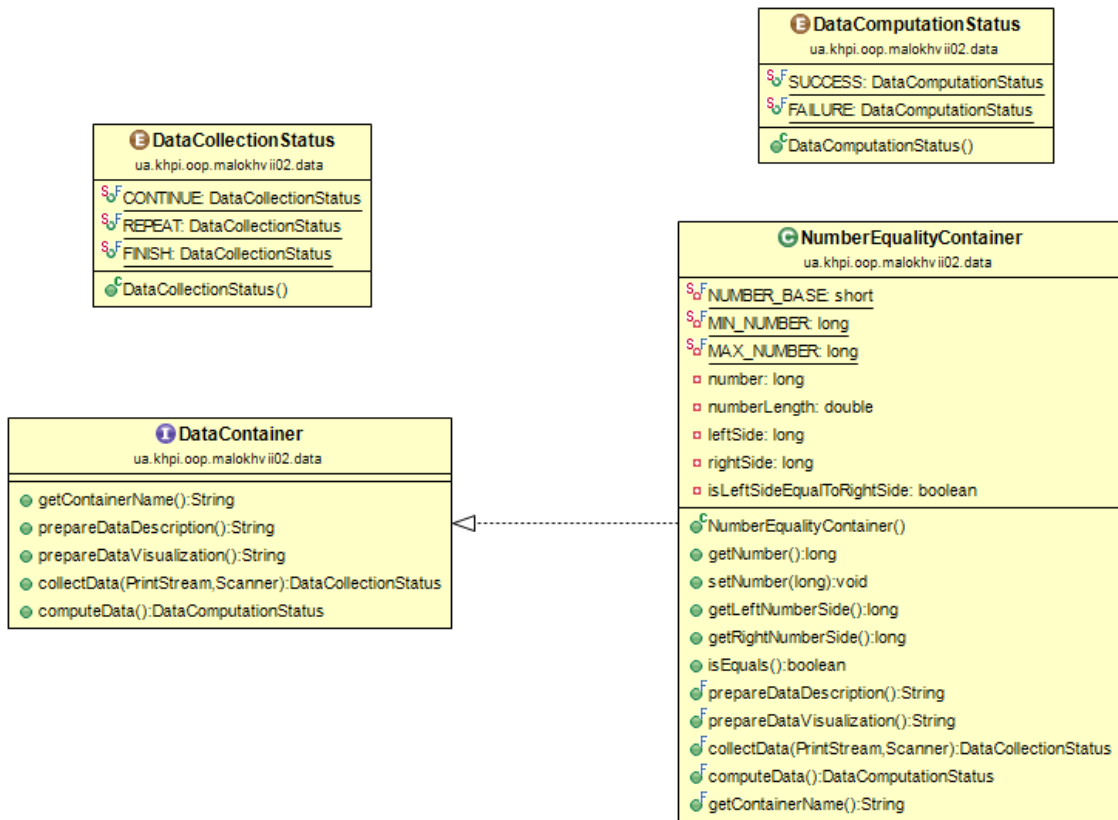


Рис. 2 - Діаграма класів із пакету data

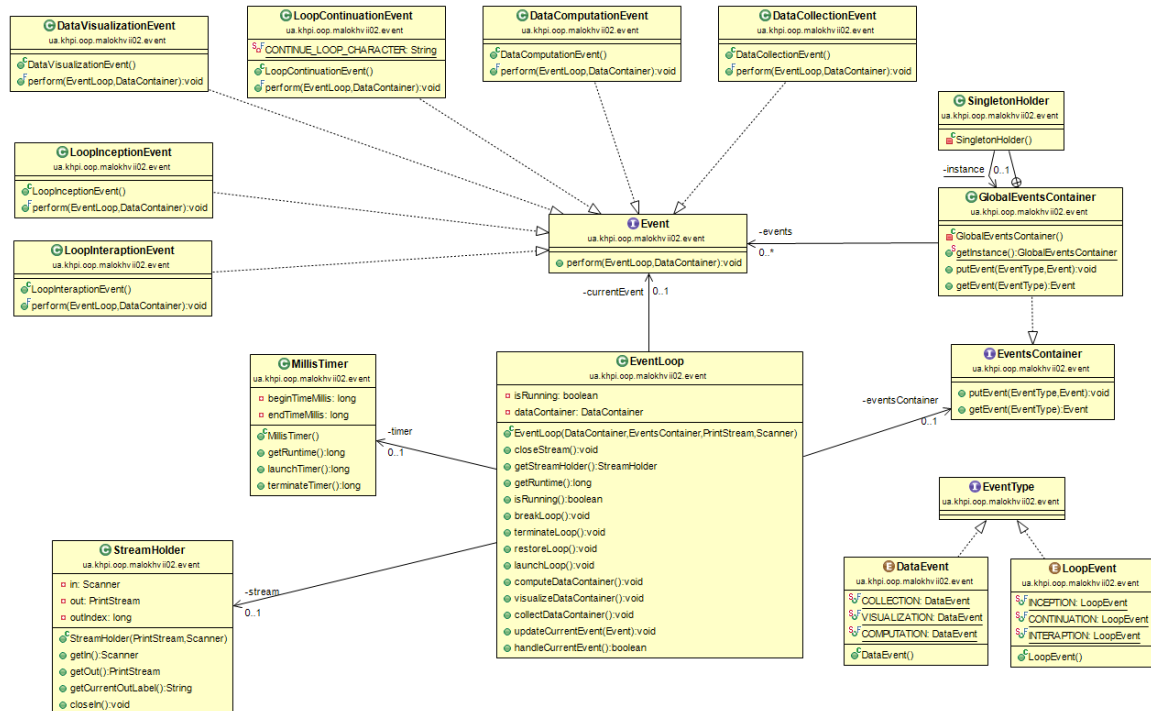


Рис. 3 - Діаграма класів із пакету event

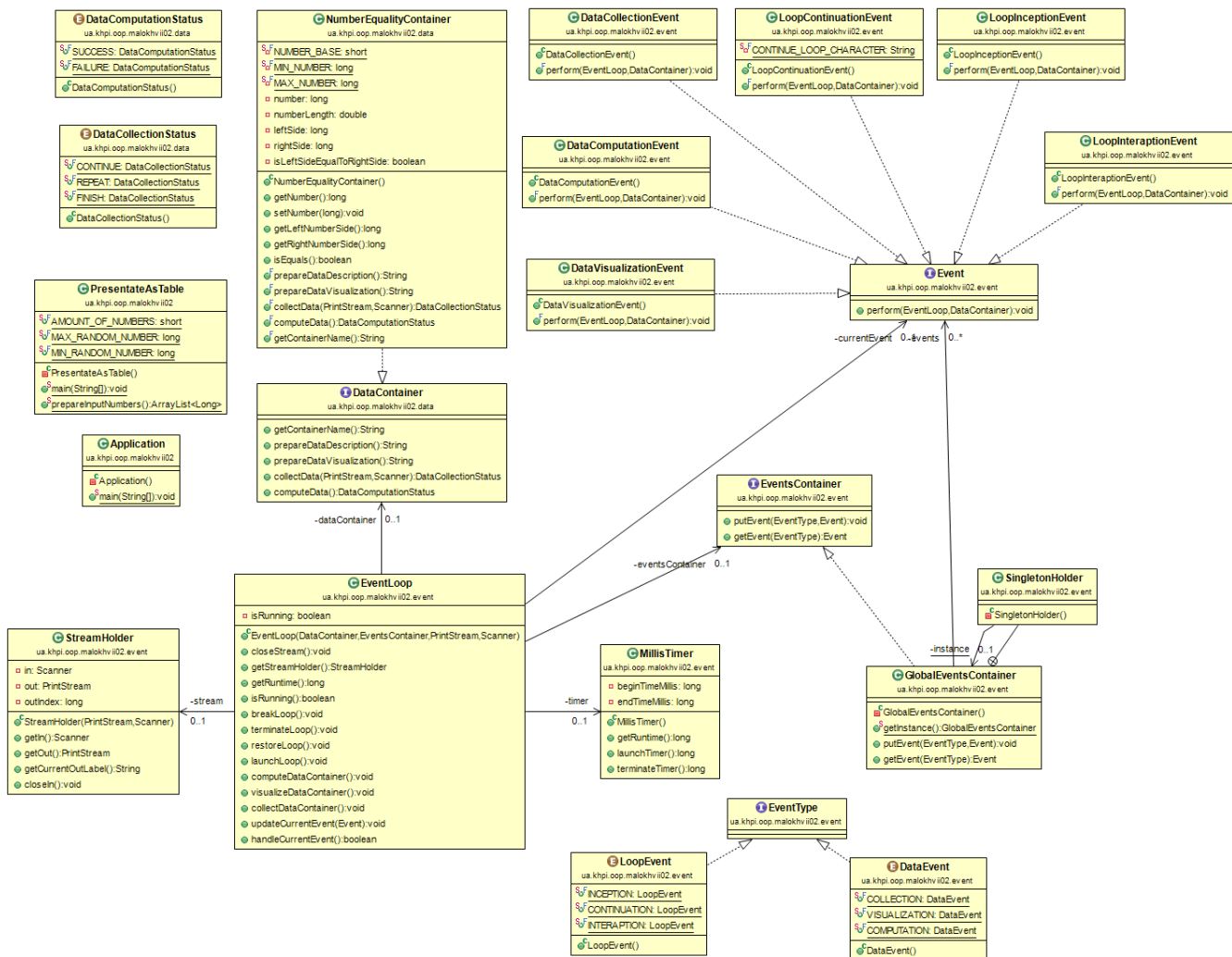


Рис. 4 - Загальна діаграма класів

2.3 Опис програми

За умовою лабораторної роботи було виведено отримані значення під час обчислення у вигляді таблиці, вхідні значення згенеровано за допомогою класу `java.util.Random`. Для форматування виведення було використано `format` та `printf`.

2.4 Важливі фрагменти програми

Нижче наведено фрагмент виконавця завдання для десятичних значень. Інші фрагменти детальніше див за посиланням (https://sourceforge.net/p/kit26a-cpp/code/HEAD/tree/malokhvii_eduard/src/ua/khpi/oop/malokhvii02/).

```

package ua.khpi.oop.malokhvii02;

import java.util.Scanner;

import ua.khpi.oop.malokhvii02.data.DataContainer;

```

```

import ua.khpi.oop.malokhvii02.data.NumberEqualityContainer;
import ua.khpi.oop.malokhvii02.event.EventLoop;
import ua.khpi.oop.malokhvii02.event.EventsContainer;
import ua.khpi.oop.malokhvii02.event.GlobalEventsContainer;

public final class Application {

    private Application() {

    }

    public static void main(final String[] args) {
        DataContainer dataContainer = new NumberEqualityContainer();
        EventsContainer loopEvents = GlobalEventsContainer.getInstance();
        EventLoop eventLoop = new EventLoop(dataContainer, loopEvents,
            System.out, new Scanner(System.in));

        eventLoop.launchLoop();
        while (eventLoop.isRunning()) {
            eventLoop.handleCurrentEvent();
        }
        eventLoop.closeStream();
    }
}

```

3 Результати роботи

Нижче наведено виведення обчислень у вигляді таблиці та інтерактивної консолі.

```

Ed@malokhvii-ee MINGW32 /d/home/projects/ua.khpi.oop.malokhvii
$ java -jar malokhvii02.jar
[1]: Set up data container: NumberEqualityContainer
[2]: Divides the whole number by parts and compares their equality
[3]: Starting data collection...
[-]: Please, enter your number: 89998999
[4]: Finishing data collection...
[5]: Starting data computation...
[6]: Finishing data computation...
[7]: Starting data visualization...
[-]: Equality is true!
[-]: Left number side: 8999
[-]: Right number side: 8999
[-]: Number length: 8.00
[8]: Finishing data visualization...
[9]: Are you want to repeat data container computation (Y/N): N
[10]: Computation time 22726 ms
[11]: Tear down data container: NumberEqualityContainer

```

Рис. 5 - Результат виконання

```
Ed@malokhvii-ee MINGW32 /d/home/projects/ua.khpi.oop.malokhvii
$ java -jar malokhvii02.jar
+-----+-----+-----+-----+
| Number | Left Part | Right Part | Equality |
+-----+-----+-----+-----+
| 222223 | 222       | 223       | False    |
| 222229 | 222       | 229       | False    |
| 222228 | 222       | 228       | False    |
| 222228 | 222       | 228       | False    |
| 222224 | 222       | 224       | False    |
| 222223 | 222       | 223       | False    |
| 222223 | 222       | 223       | False    |
| 222229 | 222 data  | 229       | False    |
| 222227 | 222 event | 227       | False    |
| 222226 | 222       | 226       | False    |
+-----+-----+-----+-----+
```

Рис. 6 - Результат виконання

Висновки

У ході виконання лабораторної роботи були покращені навички взаємодії з потоковим виведення та введення, колекціями. Досліджено склад пакету java.util.