

№1 Тема: Структура програми мовою Java. Типи даних, літерали, операції і оператори.

Мета: Ознайомлення з JDK платформи Java SE та середовищем розробки Eclipse IDE.

1 Індивідуальне завдання

1.1 Розробник

Студент Малохвій Едуард Едуардович, КІТ-26А, Варіант 8.

1.2 Вимоги

1. Вирішити три прикладні задачі на мові Java в середовищі Eclipse.
2. Продемонструвати покрокове виконання програми та результати роботи в режимі налагодження, не використовуючи виведення до консолі.
3. Виконати компіляцію і запуск програми в командному рядку за допомогою відповідних утиліт JDK.

1.3 Завдання

1. Обрати тип змінних та встановити за допомогою констант та літералів початкові значення:
 - число, що відповідає номеру залікової книжки за допомогою шістнадцяткового літералу;
 - число, що відповідає номеру мобільного телефона (починаючи з 380...) за допомогою десяткового літералу;
 - число, яке складається з останніх двох ненульових цифр номера мобільного телефону за допомогою двійкового літералу;
 - число, яке складається з останніх чотирьох ненульових цифр номера мобільного телефону за допомогою вісімкового літералу;
 - визначити збільшене на одиницю значення залишку від ділення на 26 зменшеного на одиницю номера студента в журналі групи;
 - символ англійського алфавіту в верхньому регістрі, номер якого відповідає знайденому раніше значенню.
2. Використовуючи десятковий запис цілочисельного значення кожної змінної знайти і підрахувати кількість парних і непарних цифр.
3. Використовуючи двійковий запис цілочисельного значення кожної змінної підрахувати кількість одиниць.

2 Розробка програми

2.1 Засоби ООП

Під час вирішення поставленої задачі було розділено вирішення кожного завдання на окремий клас, початкові дані винесено в окремий клас для зручнішого розділення між виконавцями завдання. Для створення об'єкту початкових даних було розроблено декілька будівників (детальніше див. Рис. 1), таким чином клас Builder забезпечує створення початкових даних із меншої кількості вхідних даних, оскільки інші можливо розрахувати із отриманих. Клас LiteralBuilder потребує встановити усі значення окремо, без попереднього розрахування під час створення.

2.2 Ієрархія та структура класів

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

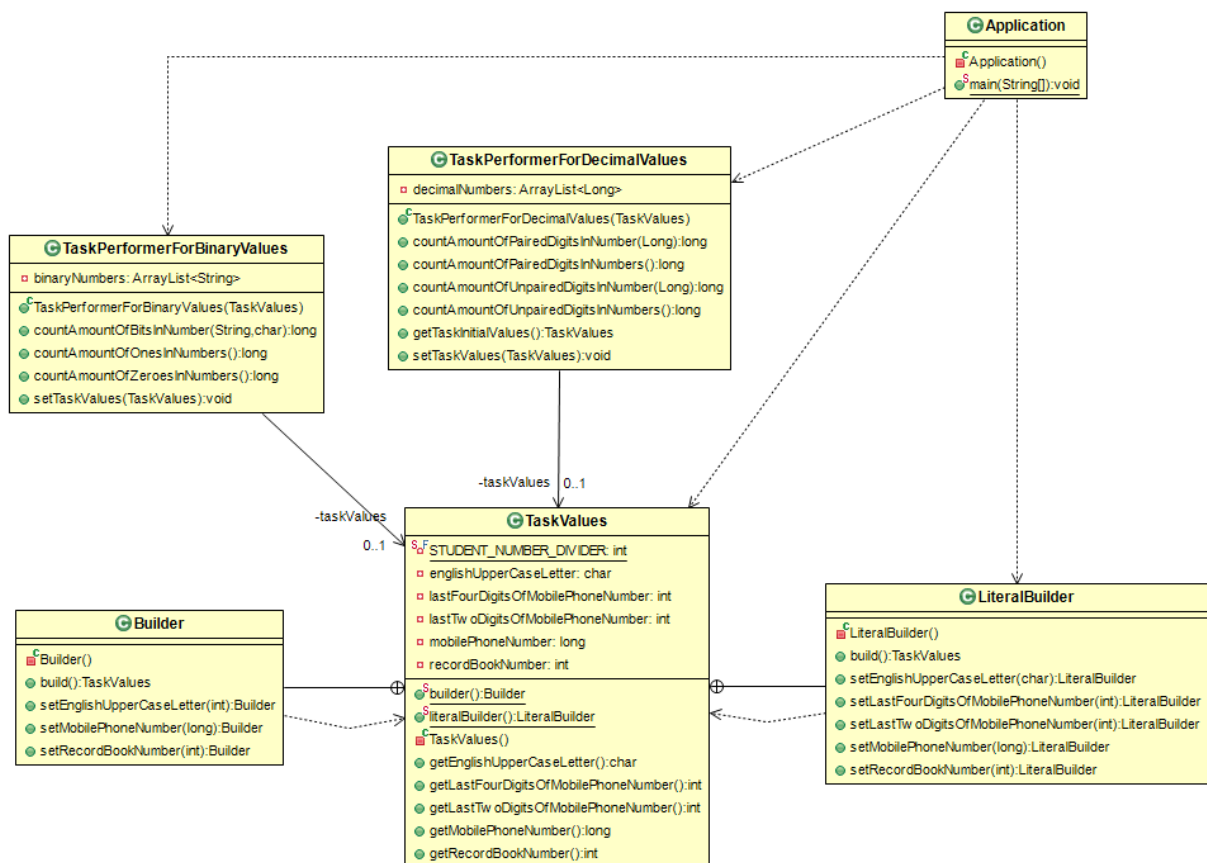


Рис. 1 - Діаграма класів

2.3 Опис програми

За умовою лабораторної роботи, виведення поточної інформації відсутнє. Усі перевірки та дослідження поведінки розробленої програми були виконані за допомогою перспективи Debug у середовищі Eclipse.

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/malokhvii01/).

```
package ua.khpi.malokhvii01;

import java.util.ArrayList;

public class TaskPerformerForDecimalValues {

    private ArrayList<Long> decimalNumbers;
    private TaskValues taskValues;

    public TaskPerformerForDecimalValues(final TaskValues taskValues) {
        this.decimalNumbers = new ArrayList<Long>();
        this.setTaskValues(taskValues);
    }

    public long countAmountOfPairedDigitsInNumber(final Long number) {
        long numberCopy = number;
        long amountOfPairedNumbers = 0;
        while (numberCopy != 0) {
            if (((numberCopy % 10) % 2) == 0) {
                ++amountOfPairedNumbers;
            }
            numberCopy /= 10;
        }

        return amountOfPairedNumbers;
    }

    public long countAmountOfPairedDigitsInNumbers() {
        long amountOfPairedDigits = 0;
        for (Long number : this.decimalNumbers) {
            amountOfPairedDigits += this
                .countAmountOfPairedDigitsInNumber(number);
        }

        return amountOfPairedDigits;
    }

    public long countAmountOfUnpairedDigitsInNumber(final Long number) {
        long numberCopy = number;
        long amountOfUnpairedNumbers = 0;
```

```

        while (numberCopy != 0) {
            if (((numberCopy % 10) % 2) != 0) {
                ++amountOfUnpairedNumbers;
            }
            numberCopy /= 10;
        }

        return amountOfUnpairedNumbers;
    }

    public long countAmountOfUnpairedDigitsInNumbers() {
        long amountOfUnpairedDigits = 0;
        for (Long number : this.decimalNumbers) {
            amountOfUnpairedDigits += this
                .countAmountOfUnpairedDigitsInNumber(number);
        }

        return amountOfUnpairedDigits;
    }

    public TaskValues getTaskInitialValues() {
        return this.taskValues;
    }

    public void setTaskValues(final TaskValues taskValues) {
        this.taskValues = taskValues;

        this.decimalNumbers.clear();
        this.decimalNumbers
            .add(new Long(this.taskValues.getMobilePhoneNumber()));
        this.decimalNumbers.add(new Long(
            this.taskValues.getLastTwoDigitsOfMobilePhoneNumber()));
        this.decimalNumbers.add(new Long(
            this.taskValues.getLastFourDigitsOfMobilePhoneNumber()));

        this.decimalNumbers
            .add(new Long(this.taskValues.getRecordBookNumber()));
        this.decimalNumbers
            .add(new Long(this.taskValues.getEnglishUpperCaseLetter()));
    }
}

```

3 Результати роботи

Нижче наведено покрокове виконання програми та результати роботи в режимі налагодження, не використовуючи виведення до консолі, згідно отриманого завдання.

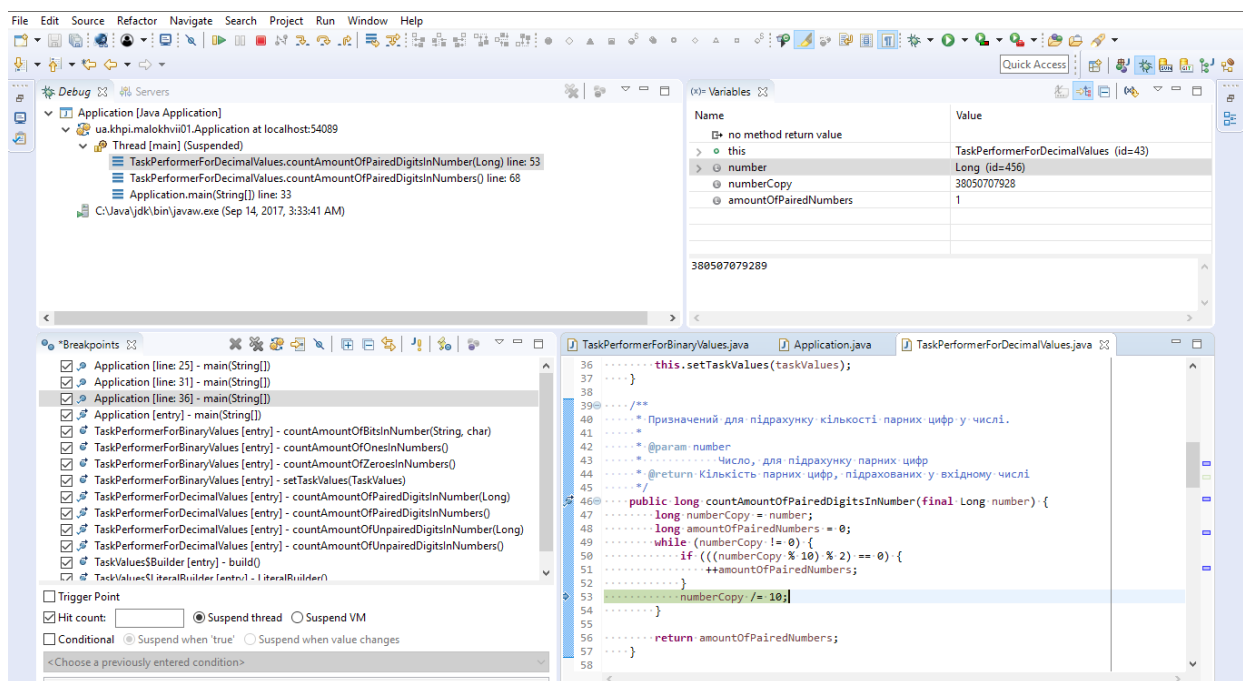


Рис. 2 - Демонстрація перспективи Debug

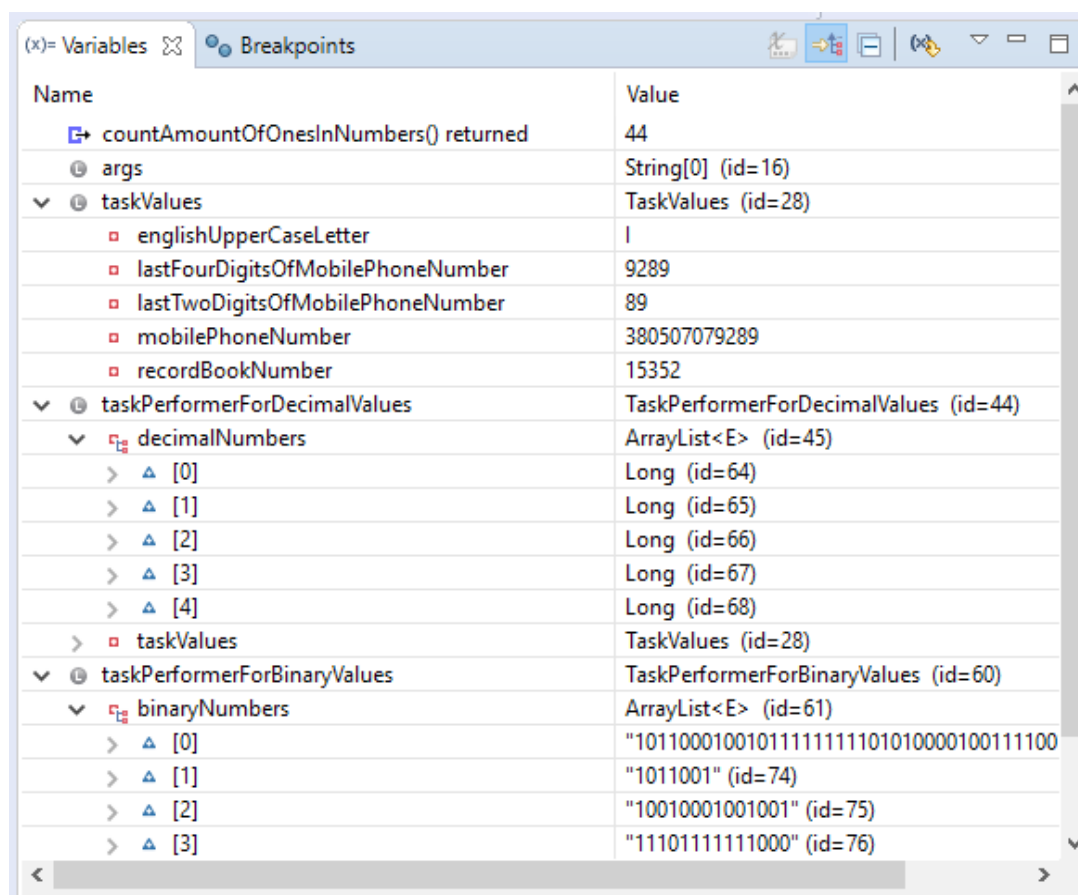


Рис. 3 - Демонстрація отриманих значень

Висновки

У ході виконання лабораторної роботи були опановані базові навички, та принципи використання середовища розробки Eclipse. Ознайомлено з платформою Java SE, опановано типи даних, операції.