

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Інститут комп'ютерних наук та інформаційних технологій  
Кафедра автоматизованих систем управління



**Звіт**  
до лабораторної роботи № 1  
з дисципліни  
*Прикладне програмування*  
на тему:  
**“Основи Java”**

Виконав: студент КН-105  
**Руденко Руслан**

Прийняв:  
**Мельник Р.В.**

Львів – 2022

## ***Лабораторна робота № 1***

### **Тема роботи: “Основи Java”**

**Мета роботи:** ознайомитись з основними компонентами інтегрованого середовища програмування мовою Java; навчитись створювати, відлагоджувати і виконувати програмні проекти;

#### **Послідовність роботи:**

1. Згідно з індивідуальним завданням напишіть програму (з використанням об'єктно-орієнтованого підходу), що відповідає наступним вимогам:

- Програма визначає і опрацьовує задані числа Фібоначчі (1, 1, 2, 3, 5, ... ) або Люка (1, 3, 4, 7, 11, ...). Номер числа та інші вхідні дані користувач має можливість передати через командний рядок, або в іншому випадку вводить з клавіатури.

- Створіть клас, який містить дані про номер та значення числа Фібоначчі або Люка, методи доступу до даних та інші методи, відповідно до індивідуального завдання.

- У головній функції оголосіть об'єкти або масив об'єктів створеного класу, які зберігають та опрацьовують введені дані.

- Програма виводить на екран вхідні дані і результати обчислень та перевірок із відповідними підказками.

- Скомпілюйте та запустіть програму через командний рядок та за допомогою обраного середовища розробки.

2. Продокументуйте програму з п. 1 з використанням JavaDoc і згенеруйте на основі нього документацію.

#### **Завдання лабораторної роботи, 22 варіант:**

Визначити суму  $N$  перших чисел Люка (спробуйте вивести і перевірити відповідну формулу).

## Текст програми:

### Constants.java

```
package lab1;

/**
 * Contains Golden ratio and max number in luke sequence constants
 *
 * @author Ruslan
 * @version 1.0.0
 */
public class Constants {
    static int MAX_LUKE_NUMBER = 89;
    static double GOLDEN_RATIO = (1 + Math.sqrt(5)) / 2.0;
}
```

### Sequence.java

```
package lab1;

/**
 * Base sequence class
 *
 * @author Ruslan
 * @version 1.0.0
 */
public interface Sequence {
    void setAmountOfNumbers(int amountOfNumbers);

    int getAmountOfNumbers();

    long getSumOfFirstNNumbers();
}
```

## LukeSequence.java

```
package lab1;

/**
 * Implements Sequence, allows to set amount of numbers and
 * calculate sum of them
 *
 * @author Ruslan
 * @version 1.0.0
 */
public class LukeSequence implements Sequence {
    private int amountOfNumbers;

    /**
     * amountOfNumbers setter
     * Note: when number is bigger than MAX_LUKE_NUMBER
     * it sets amountOfNumbers to MAX_LUKE_NUMBER
     */
    @Override
    public void setAmountOfNumbers(int amountOfNumbers) {
        if (amountOfNumbers > Constants.MAX_LUKE_NUMBER) {
            System.out.printf("Maximum amount of numbers is %d\n",
                Constants.MAX_LUKE_NUMBER);
            amountOfNumbers = Constants.MAX_LUKE_NUMBER;
        }
        this.amountOfNumbers = amountOfNumbers;
    }

    /**
     * amountOfNumbers getter
     */
    @Override
    public int getAmountOfNumbers() {
        return amountOfNumbers;
    }

    /**
     * Returns sum of first N numbers of Luke sequence
     */
    @Override
    public long getSumOfFirstNNumbers() {
        if (amountOfNumbers < 1) return 0;
        return getNLukeNumber(amountOfNumbers + 1) - 1;
    }

    /**
     * if n is less than 68 uses fast calculation via Golden ratio
     * otherwise uses standard iterative approach
     *
     * @param n from 1 to 89
     */
    private long getNLukeNumber(int n) {
        if (n <= 67) return getNLukeNumberFast(n);
    }
}
```

```

        long prev = 2;
        long cur = 1;
        long temp;

        for (int i = 1; i < n; i++) {
            temp = cur;
            cur += prev;
            prev = temp;
        }

        return cur;
    }

    /**
     * Calculates Luke sequence number using golden ratio
     *
     * @param n when over 67 gives not precise results
     */
    private long getNLukeNumberFast(int n) {
        return Math.round(Math.pow(Constants.GOLDEN_RATIO, n));
    }
}

```

## Lab1.java

```

package lab1;

import java.util.Scanner;

/**
 * This class solves task for Lab1, 22 variant
 *
 * @author Ruslan
 * @version 1.0.0
 */
public class Lab1 {

    public static void main(String[] args) {
        new Lab1().run(args);
    }

    /**
     * Performs necessary calculations and outputs result for Lab1
     */
    void run(String[] args) {
        Sequence lukeSequence = new LukeSequence();

        lukeSequence.setAmountOfNumbers(readLukeNumber(args));

        long sumOfLukeNumbers =
lukeSequence.getSumOfFirstNNumbers();

        System.out.printf(

```

```

        "Sum of first %d Luke sequence numbers: %d\n",
        lukeSequence.getAmountOfNumbers(),
        sumOfLukeNumbers);
    }

    /**
line  * Reads number from arguments, if not able then from command
    */
    int readLukeNumber(String[] args) {
        try {
            return Integer.parseInt(args[0]);
        } catch (Exception e) {
            return readLukeNumberFromTerminal();
        }
    }

    /**
    * Reads number from terminal
    */
    int readLukeNumberFromTerminal() {
        System.out.println("Enter amount of Luke numbers to
calculate sum: ");
        Scanner input = new Scanner(System.in);
        return input.nextInt();
    }
}

```

# Результат генерування java doc:

All Classes

Constants  
Lab1  
LukeSequence  
Sequence

PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

Package lab1

Interface Summary

Interface	Description
Sequence	Base sequence class

Class Summary

Class	Description
Constants	Contains Golden ratio and max number in luke sequence constants
Lab1	This class solves task for Lab1, 22 variant
LukeSequence	Implements Sequence, allows to set amount of numbers and calculate sum of them

PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

All Classes

Constants  
Lab1  
LukeSequence  
Sequence

PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD    DETAIL: FIELD | CONSTR | METHOD

lab1

Class Constants

java.lang.Object  
lab1.Constants

```
public class Constants
extends java.lang.Object

Contains Golden ratio and max number in luke sequence constants

Version:
1.0.0

Author:
Ruslan
```

Constructor Summary

Constructors

Constructor and Description
Constants()

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

All Classes

Constants  
Lab1  
LukeSequence  
Sequence

PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD    DETAIL: FIELD | CONSTR | METHOD

lab1

Class Lab1

java.lang.Object  
lab1.Lab1

```
public class Lab1
extends java.lang.Object

This class solves task for Lab1, 22 variant

Version:
1.0.0

Author:
Ruslan
```

Constructor Summary

Constructors

Constructor and Description
Lab1()

Method Summary

All Methods    Static Methods    Concrete Methods

Modifier and Type	Method and Description
static void	main(java.lang.String[] args)

PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

All Classes

Constants

Lab1

LukeSequence

Sequence

PACKAGE

CLASS

TREE

DEPRECATED

INDEX

HELP

PREV CLASS

NEXT CLASS

FRAMES

NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHODDETAIL: FIELD | CONSTR | METHOD

lab1

Class LukeSequence

java.lang.Object

lab1.LukeSequence

All Implemented Interfaces:  
Sequence

public class LukeSequence

extends java.lang.Object

implements Sequence

Implements Sequence, allows to set amount of numbers and calculate sum of them

Version:  
1.0.0

Author:  
Ruslan

Constructor Summary

Constructors

Constructor and Description

LukeSequence()

All Classes

Constants

Lab1

LukeSequence

Sequence

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method and Description
int	getAmountOfNumbers() amountOfNumbers getter
long	getSumOfFirstNNumbers() Returns sum of first N numbers of Luke sequence
void	setAmountOfNumbers(int amountOfNumbers) amountOfNumbers setter Note: when number is bigger than MAX_LUKE_NUMBER it sets amountOfNumbers to MAX_LUKE_NUMBER

Methods inherited from class java.lang.Object  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

LukeSequence

public LukeSequence()

All Classes

Constants

Lab1

LukeSequence

Sequence

Method Detail

setAmountOfNumbers

public void setAmountOfNumbers(int amountOfNumbers)

amountOfNumbers setter Note: when number is bigger than MAX\_LUKE\_NUMBER it sets amountOfNumbers to MAX\_LUKE\_NUMBER

Specified by:  
setAmountOfNumbers in interface Sequence

getAmountOfNumbers

public int getAmountOfNumbers()

amountOfNumbers getter

Specified by:  
getAmountOfNumbers in interface Sequence

getSumOfFirstNNumbers

public long getSumOfFirstNNumbers()

Returns sum of first N numbers of Luke sequence

Specified by:  
getSumOfFirstNNumbers in interface Sequence

PACKAGE

CLASS

TREE

DEPRECATED

INDEX

HELP

PREV CLASS

NEXT CLASS

FRAMES

NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHODDETAIL: FIELD | CONSTR | METHOD



Interface Sequence

All Known Implementing Classes:  
LukaSequence

public interface **Sequence**

Base sequence class

Version:  
1.0.0

Author:  
Ruslan

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type		Method and Description
int		getAmountOfNumbers()
long		getSumOfFirstNNumbers()
void		setAmountOfNumbers(int amountOfNumbers)

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type		Method and Description
int		getAmountOfNumbers()
long		getSumOfFirstNNumbers()
void		setAmountOfNumbers(int amountOfNumbers)

Method Detail

<b>setAmountOfNumbers</b>
void setAmountOfNumbers(int amountOfNumbers)
<b>getAmountOfNumbers</b>
int getAmountOfNumbers()
<b>getSumOfFirstNNumbers</b>
long getSumOfFirstNNumbers()

## Результати роботи програми:

Enter amount of Luke numbers to calculate sum:

5

Sum of first 5 Luke sequence numbers: 17

Process finished with exit code 0

Enter amount of Luke numbers to calculate sum:

-10

Sum of first -10 Luke sequence numbers: 0

Process finished with exit code 0

Enter amount of Luke numbers to calculate sum:

100

Maximum amount of numbers is 89

Sum of first 89 Luke sequence numbers: 6440026026380244497

Process finished with exit code 0

## **Висновок**

В ході лабораторної роботи я ознайомився з основними компонентами інтегрованого середовища **IntelliJ Idea** для програмування мовою Java. Навчився створювати, відлагоджувати і виконувати програмні проекти, вивчив принципи оголошення класу та об'єктів, рівні захисту елементів класу та застосував ці знання на практиці виконавши індивідуальне завдання.