Міністерство освіти і науки України

Національний технічний університет України „КПІ”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки

інформації та управління

**ЗВІТ**

до лабораторної роботи № 2

з предмету:

„Основи технологій програмування”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Виконав**  **студент** |  | *ІП-61 Кушка Михайло Олександрович, 2-й курс, ІП-6116* |  |  |
|  |  | (№ групи, прізвище, ім’я, по батькові, курс, номер залікової книжки) |  |  |
|  |  |  |  |  |
| **Прийняв** |  | *Подрубайло О.О.* |  |  |
|  |  | (посада, прізвище, ім’я, по батькові ) |  |  |

Київ 2018

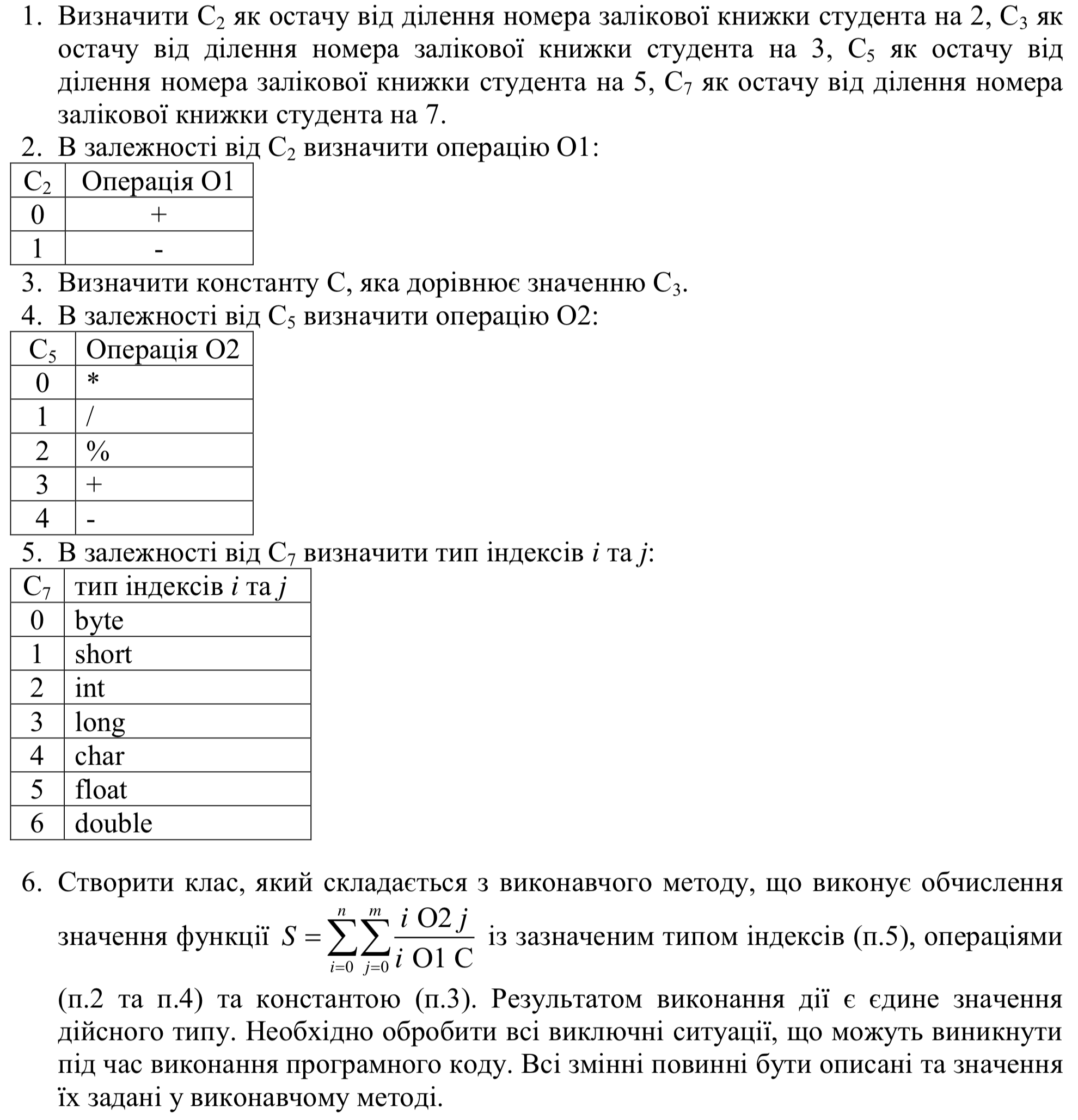
ЗМІСТ

[1. постановка задачі 3](#_Toc506126329)

[2. Висновок 4](#_Toc506126330)

[3. Код програми 5](#_Toc506126331)

# постановка задачі



6116 % 5 == 1 ()

6116 % 7 == 5 ()

6116 % 11 == 0 ( Обчислити суму найменших елементів кожного стовпця матриці)

# Висновок

Незвичним в даній лабораторній роботі було те, що в мові програмування Java відсутнє таке поняття як «покажчик», тому довелося реалізовувати клас з матрицею без можливості зміни її розмірності.

# Код програми

/\*\*

\* Java labs - Lab2

\* **@version** 1.0 2018-02-26

\* **@author** Misha Kushka

\*/

**class** Matrix {

**private** **final** **char**[][] matr;

**private** **final** **int** n;

**private** **final** **int** m;

Matrix(**char**[][] newMatr) {

matr = newMatr;

n = matr.length;

m = matr[0].length;

}

**void** printMatrix() {

System.***out***.print('\t');

**for** (**int** i = 0; i < **this**.n; ++i) {

**for** (**int** j = 0; j < **this**.m; ++j) {

System.***out***.print(**this**.matr[i][j] + " ");

}

System.***out***.print("\n\t");

}

System.***out***.println();

}

**char**[][] transpose() {

**char**[][] outputMatrix = **new** **char**[**this**.m][**this**.n];

**for** (**int** i = 0; i < **this**.n; ++i) {

**for** (**int** j = 0; j < **this**.m; ++j) {

outputMatrix[j][i] = **this**.matr[i][j];

}

}

**return** outputMatrix;

}

**char** minElementsColumnSum() {

**int** minSum = 0;

**char** columnMinElement;

**for** (**int** i = 0; i < **this**.n; ++i) {

columnMinElement = Character.***MAX\_VALUE***;

**for** (**int** j = 0; j < **this**.m; ++j) {

**if** (matr[i][j] < columnMinElement) {

columnMinElement = matr[i][j];

}

}

// Overflow check

**if** (i == **this**.n-1 && minSum + (**int**)columnMinElement > Character.***MAX\_VALUE***) {

System.***err***.println("Overflow char type!");

}

minSum += (**int**)columnMinElement;

}

**return** (**char**)minSum;

}

}

**public** **class** Main {

**public** **static** **void** main(String[] args) {

**char**[][] matr = {

{'a', 'b', 'c', 'd'}, // {Character.MAX\_VALUE-3, Character.MAX\_VALUE-2, Character.MAX\_VALUE-1, Character.MAX\_VALUE},

{'e', 'f', 'g', 'h'},

{'i', 'j', 'k', 'l'}

};

Matrix matrix = **new** Matrix(matr);

// Input matrix

System.***out***.println("Input matrix:");

matrix.printMatrix();

// Transposed matrix

System.***out***.println("Transposed matrix:");

Matrix transpMatr = **new** Matrix(matrix.transpose());

transpMatr.printMatrix();

// Sum of the smallest elements from matrix columns

System.***out***.print("Sum of the smallest elements from matrix columns -> ");

**char** result = matrix.minElementsColumnSum();

System.***out***.println(result);

}

}