

# ALG1 - Semestrální práce

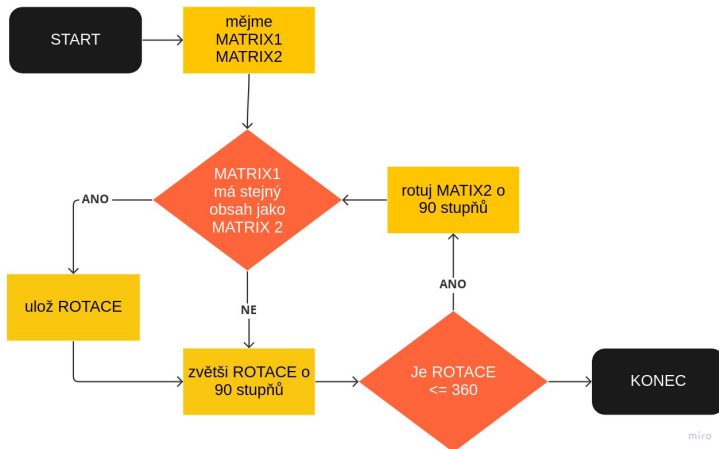
Daniel Adámek

19. prosince 2022

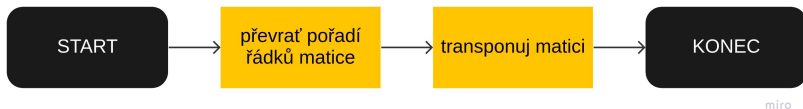
## **Zadání č. 21:**

Vytvořte program, který po zadání rozměru čtvercové matice načte 2 data matice a vyhodnotí, zda-li není jedna transformací - rotace druhé matice, případně o kolik stupňů.

# Algoritmus validace rotace matic



# Rotace matice o 90 stupňů po směru hodinových ručiček



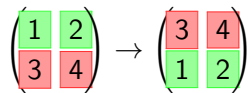
miro

# Rotace matice o 90 stupňů postup

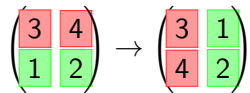
Mějme matici  $2 \times 2$

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Převrátíme pořadí řádků


$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix}$$

Transponujeme matici


$$\begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & 1 \\ 4 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 1 \\ 4 & 2 \end{pmatrix}$$

## Nejzajímavější část kódu (rotace)

```
int[] rotationsMem = new int[4];

// rotation of matrix
Matrix m2Rotated = m2.deepCopy();
int rotation = 0;
while (rotation < 4) {
    if (Matrix.hasSameContent(m1,m2Rotated))
        rotationsMem[rotation] = rotation*90;

    m2Rotated.rotate(90);
    rotation++;
}
```

# JUnit testování

```
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class MatrixTest {
    /**
     * Manual rotation.
     */
    @Test
    @DisplayName("Matrix turn method - manual rotation")
    void manualRotation() {
        // Testing cases
        Matrix a = new Matrix(2);
        a.data = new int[][]{
            {1, 2},
            {3, 4}
        };
        a.rotate(90);
        a.print();

        Matrix ninetyDegrees = new Matrix(2);
        ninetyDegrees.data = new int[][]{
            {3, 1},
            {4, 2}
        };
        ninetyDegrees.print();
        assertTrue(Matrix.hasSameContent(a,ninetyDegrees));
    }
}
```

✓ MatrixTest (cz.tul.alg.daniela.matrix)	35 ms
✓ Matrix turn method - manual rotation	23 ms
✓ Matrix turn method - Even sizes	4 ms
✓ Matrix hasSameContent	1 ms
✓ Matrix turn method - edge cases	1 ms
✓ Matrix data - deep copy of data	1 ms
✓ Matrix turn method - Odd sizes	5 ms

# Testování programu

Vstup		Výstup	Reference
3	3	Rozměr matic	Rozměr matic
1 2 3	1 2 3	První matice	První matice
4 5 6	4 5 6	Druhá matice	Druhá matice
7 8 9	7 8 9	Rotace 270	Rotace 270
7 4 1	1 4 7	Rozměr matic	Rozměr matic
8 5 2	2 5 8	První matice	První matice
9 6 3	3 6 9	Druhá matice	Druhá matice
3	-1	Rotace 90	Rotace 90
1 2 3		Rozměr matic	Rozměr matic
4 5 6		První matice	První matice
7 8 9		Druhá matice	Druhá matice
3 6 9		Není rotací	Není rotací
2 5 8		Rozměr matic	Rozměr matic
1 4 7			

Záznam o testu v složce */test*