

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №2
з дисципліни
“Програмування мобільних систем / Розроблення клієнтських додатків
для мобільних платформ”

Виконав:
студент групи ІВ-83
ЗК ІВ-8328
Чорноморець Влад

Київ 2021

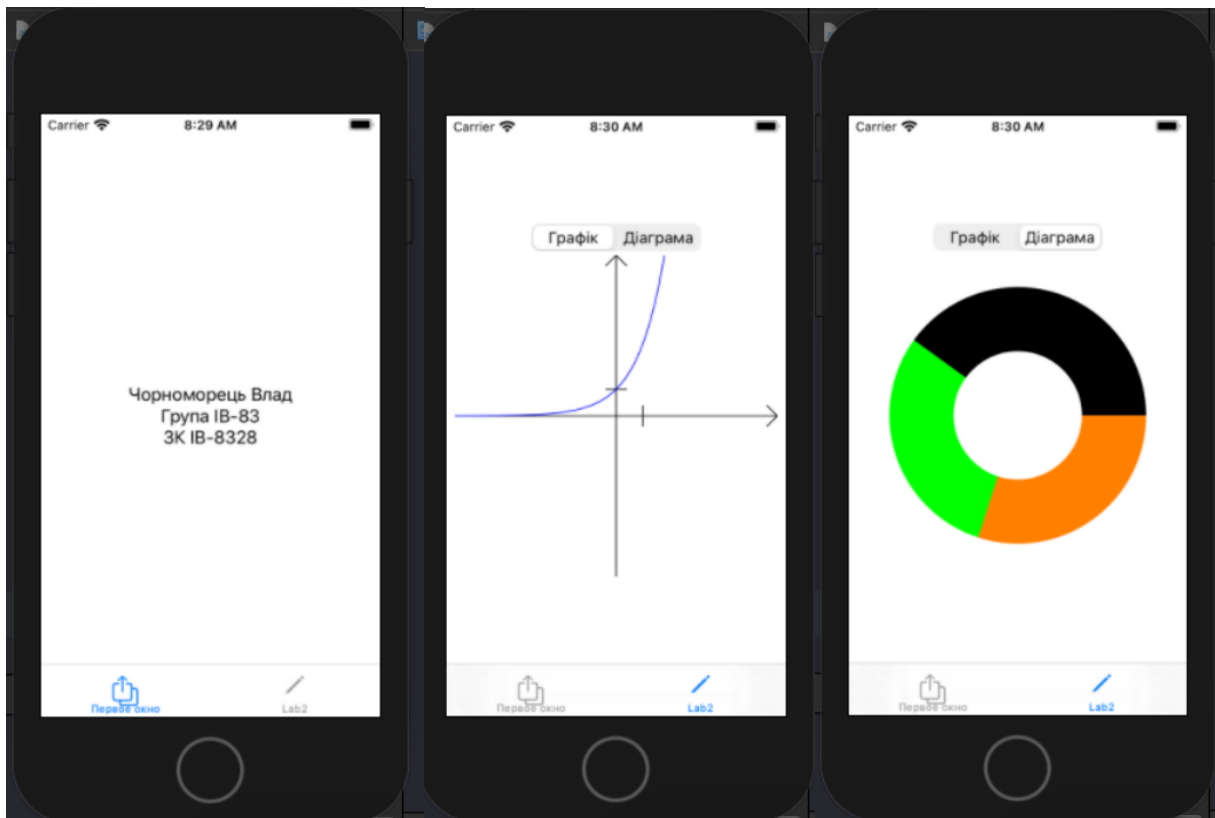
Варіант: 1

```
>>> 8328%6 + 1  
1
```

Варіант 1

- a. Функція $y = e^x$ на проміжку $x \in [-6; 6]$;
- b. Кільце кругової діаграми із секторами, що займають відповідний відсоток кола та мають відповідний колір: 30% (помаранчевий), 30% (зелений), 40% (чорний).

Скріншот роботи додатка



Лістинг коду

GraphView.swift

```
import
Foundation

import UIKit

class GraphView: UIView {
    override func draw(_ rect: CGRect) {
        let centerOfAxis = (x: bounds.width/2, y: bounds.height/2)
        let cursor = UIBezierPath()

        cursor.move(to: CGPoint(x: 0, y: centerOfAxis.y))
        cursor.addLine(to: CGPoint(x: bounds.width, y: centerOfAxis.y))
        cursor.close()

        cursor.move(to: CGPoint(x: centerOfAxis.x, y: 0))
        cursor.addLine(to: CGPoint(x: centerOfAxis.x, y: bounds.height))
        cursor.close()

        cursor.move(to: CGPoint(x: centerOfAxis.x - 10, y: 10))
        cursor.addLine(to: CGPoint(x: centerOfAxis.x, y: 0))
        cursor.move(to: CGPoint(x: centerOfAxis.x, y: 0))
        cursor.addLine(to: CGPoint(x: centerOfAxis.x + 10, y: 10))
        cursor.close()

        cursor.move(to: CGPoint(x: bounds.width - 10, y: centerOfAxis.y - 10))
        cursor.addLine(to: CGPoint(x: bounds.width, y: centerOfAxis.y))
        cursor.move(to: CGPoint(x: bounds.width, y: centerOfAxis.y))
        cursor.addLine(to: CGPoint(x: bounds.width - 10, y: centerOfAxis.y + 10))
        cursor.close()

        cursor.move(to: CGPoint(x: centerOfAxis.x + centerOfAxis.x/6, y:
centerOfAxis.y - 10))
        cursor.addLine(to: CGPoint(x: centerOfAxis.x + centerOfAxis.x/6, y:
centerOfAxis.y + 10))
        cursor.close()

        cursor.move(to: CGPoint(x: centerOfAxis.x - 10, y: centerOfAxis.y -
centerOfAxis.y/6))
        cursor.addLine(to: CGPoint(x: centerOfAxis.x + 10, y: centerOfAxis.y -
centerOfAxis.y/6))
        cursor.close()

        var color = UIColor.black
        color.setStroke()
        cursor.stroke()
        cursor.lineWidth = 1
    }
}
```

```

let cursorOfGraph = UIBezierPath()
let part = centerOfAxis.x/6;

cursorOfGraph.move(to: CGPoint(x: 0, y: centerOfAxis.y))

for i in stride (from: -6, to: 6, by: 0.01){
    let funcValue = pow(M_E,i)
    cursorOfGraph.addLine(to: CGPoint(x: centerOfAxis.x + part *
CGFloat(i), y: centerOfAxis.y - part * CGFloat(funcValue)))
    cursorOfGraph.move(to: CGPoint(x: centerOfAxis.x + part * CGFloat(i),
y: centerOfAxis.y - part * CGFloat(funcValue)))
}

cursorOfGraph.close()
color = UIColor.blue
color.setStroke()
cursorOfGraph.stroke()
cursorOfGraph.lineWidth = 1
}
}

```

DiagramView.swift

```

import
UIKit

class DiagramView: UIView {

    override func draw(_ rect: CGRect) {
        let values = [(value: 0.3, color: UIColor.orange),
            (value: 0.3, color: UIColor.green),
            (value: 0.4, color: UIColor.black)]

        let center = CGPoint(x: bounds.width/2, y: bounds.height/2)
        var currAngle:CGFloat = 0

        for i in values {
            let cursor = UIBezierPath()
            cursor.move(to: center)

```

```

        cursor.addArc(withCenter: center, radius: bounds.width/2.5, startAngle:
currAngle, endAngle: currAngle + (CGFloat.pi * 2 * CGFloat(i.value)), clockwise:
true)

        currAngle = currAngle + (CGFloat.pi * 2 * CGFloat(i.value))
        cursor.close()
        let color = i.color
        color.setFill()
        cursor.fill()
    }
    let cursor2 = UIBezierPath()
    cursor2.addArc(withCenter: center, radius: bounds.width/5, startAngle: 0,
endAngle: CGFloat.pi*2, clockwise: true)
    let color = UIColor.white
    color.setFill()
    cursor2.fill()

}

}

```

Висновок:

В цій лабораторній роботі була доповнена робота 1.1, а саме створена нова вкладка, в якій малюються діаграма та графік. Малювання діаграми та графіку реалізовано в класах `DiagramView.swift` та `GraphView.swift`.