

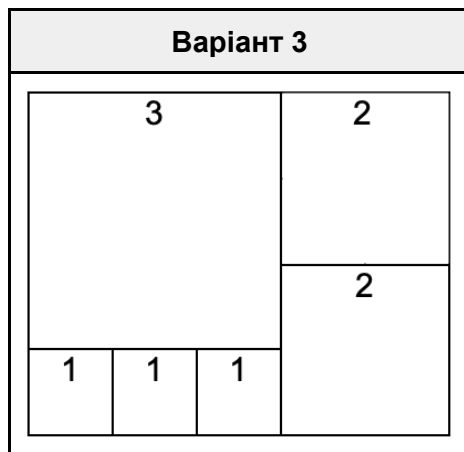
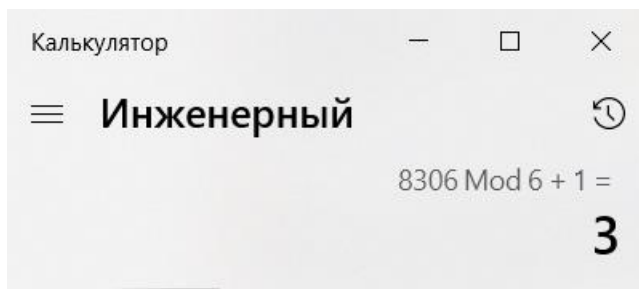
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

Лабораторна робота № 5  
з дисципліни  
“Програмування мобільних систем”

Виконав:  
студент групи ІВ-83  
ЗК ІВ-8306  
Грисюк Михайло

Київ 2021

## Вариант



## Лістинг коду

### MyCollectionViewCell.swift

```
//  
// MyCollectionViewCell.swift  
// Lab1-1  
//  
// Created by misha_hrysiuk on 30.04.2021.  
//  
import UIKit  
  
class MyCollectionViewCell: UICollectionViewCell {  
    @IBOutlet var myImageView: UIImageView!  
  
    override func awakeFromNib() {  
        super.awakeFromNib()  
    }  
  
    public func configure(image: UIImage?) {  
        myImageView.backgroundColor = .gray  
        myImageView.image = image  
    }  
}
```

### MyCollectionViewLayout.swift

```
//  
// MyCollectionViewLayout.swift  
// Lab1-1  
//
```

```

// Created by misha_hrysiuk on 30.04.2021.
//
import UIKit

protocol MyCollectionViewLayoutDelegate: AnyObject {
    func collectionView(_ collectionView: UICollectionView,
        heightForPhotoAtIndexPath indexPath: IndexPath) -> CGFloat
}

class MyCollectionViewLayout: UICollectionViewLayout {
    weak var delegate: MyCollectionViewLayoutDelegate?

    private let cellPadding: CGFloat = 1
    private var cache: [UICollectionViewLayoutAttributes] = []
    private var contentHeight: CGFloat = 0
    private var contentWidth: CGFloat {
        guard let collectionView = collectionView else {
            return 0
        }
        let insets = collectionView.contentInset
        return collectionView.bounds.width - (insets.left + insets.right)
    }

    override func invalidateLayout() {
        super.invalidateLayout()
        contentHeight = 0
        cache = []
    }

    override var collectionViewContentSize: CGSize {
        return CGSize(width: contentWidth, height: contentHeight)
    }

    override func prepare() {
        guard cache.isEmpty, let collectionView = collectionView else {
            return
        }

        let xOffset: [CGFloat] = [0, contentWidth/5*3, 0,
contentWidth/5,contentWidth/5*2]
        var yOffset:CGFloat = 0

        for item in 0..

```

```

        } else {
            frame = CGRect(x: xOffset[indexPath[1] % 6 - 1], y:
yOffset, width: height, height: height)
            yOffset += (indexPath[1] + 1) % 6 == 0 ? height : 0
        }
        let insetFrame = frame!.insetBy(dx: cellPadding, dy:
cellPadding)

        let attributes = UICollectionViewLayoutAttributes(forCellWith:
indexPath)
        attributes.frame = insetFrame
        cache.append(attributes)

        contentHeight = max(contentHeight, frame!.maxY)
    }

    }

    override func layoutAttributesForElements(in rect: CGRect) ->
[UICollectionViewLayoutAttributes]? {
        var visibleLayoutAttributes: [UICollectionViewLayoutAttributes] =
[]

        for attributes in cache {
            if attributes.frame.intersects(rect) {
                visibleLayoutAttributes.append(attributes)
            }
        }
        return visibleLayoutAttributes
    }

    override func layoutAttributesForItem(at indexPath: IndexPath) ->
UICollectionViewLayoutAttributes? {
        return cache[indexPath.item]
    }
}

```

### PhotosViewController.swift

```

//
//  PhotosViewController.swift
//  Lab1-1
//
//  Created by misha_hrysiuk on 30.04.2021.
//
import UIKit

class PhotosViewController: UIViewController {
    @IBOutlet var collectionView: UICollectionView!

    var arrayOfPictures:[UIImage] = []

    override func viewDidLoad() {
        super.viewDidLoad()

        if let layout = collectionView?.collectionViewLayout as?
MyCollectionViewLayout {
            layout.delegate = self
        }
    }
}

```

```

        collectionView.delegate = self
        collectionView.dataSource = self

        collectionView.register(UINib(nibName: "MyCollectionViewCell",
bundle: nil), forCellWithReuseIdentifier: "MyCollectionViewCell")

    }

    override func viewDidLoadSubviews() {
        super.viewDidLoadSubviews()
        collectionView.collectionViewLayout.invalidateLayout()
    }

}

extension PhotosViewController: UICollectionViewDelegate,
UICollectionViewDataSource{
    func collectionView(_ collectionView: UICollectionView,
numberOfItemsInSection section: Int) -> Int {
        return arrayOfPictures.count
    }

    func collectionView(_ collectionView: UICollectionView, cellForItemAt
indexPath: IndexPath) -> UICollectionViewCell {
        let cell = collectionView.dequeueReusableCell(withReuseIdentifier:
"MyCollectionViewCell", for: indexPath) as! MyCollectionViewCell
        cell.configure(image: arrayOfPictures[indexPath.item])

        return cell
    }

}

extension PhotosViewController: MyCollectionViewLayoutDelegate {
    func collectionView(_ collectionView:
UICollectionView,heightForPhotoAtIndex indexPath:IndexPath) -> CGFloat
{
        if((indexPath.item + 1) % 6 == 1) {
            return collectionView.frame.width/5*3
        } else if (((indexPath.item + 1) % 6 == 2 ) || (indexPath.item + 1)
% 6 == 3) {
            return collectionView.frame.width/5*2
        } else {
            return collectionView.frame.width/5
        }
    }

}

```

```

extension PhotosViewController: UIImagePickerControllerDelegate,
 UINavigationControllerDelegate {
    @IBAction func openPhotos() {
        let imagePicker = UIImagePickerController()
        imagePicker.delegate = self
        imagePicker.sourceType = .photoLibrary
        present(imagePicker, animated: true)
    }

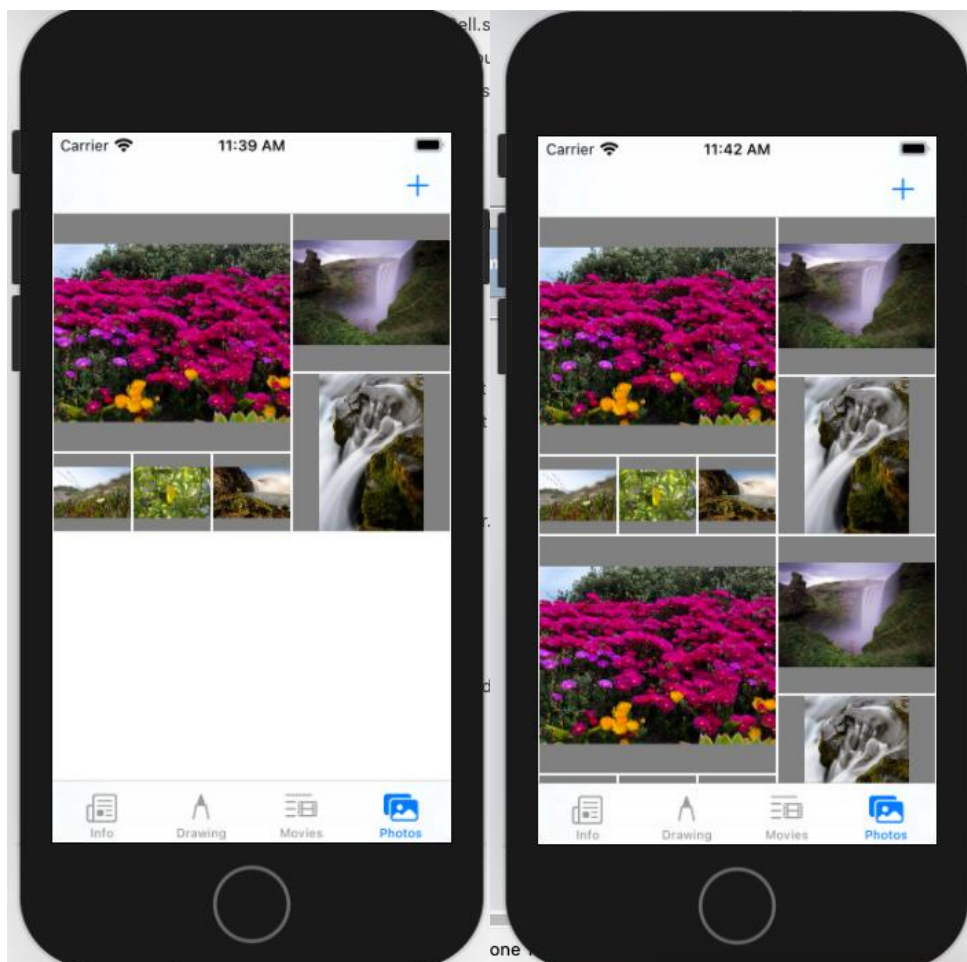
    func imagePickerController(_ picker: UIImagePickerController,
    didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey :
    Any]) {
        picker.dismiss(animated: true, completion: nil)
        guard let newImage = info[.originalImage] as? UIImage else { return
    }

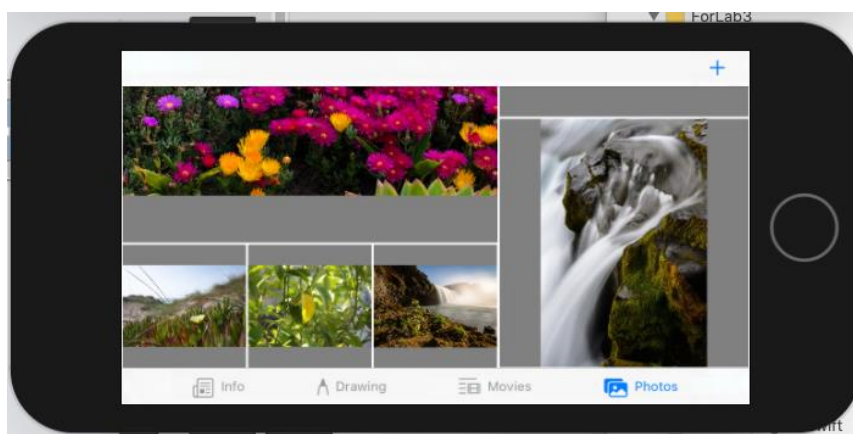
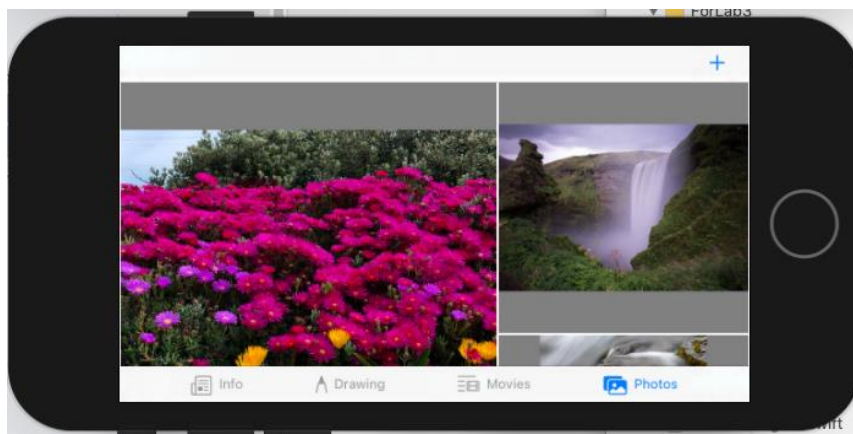
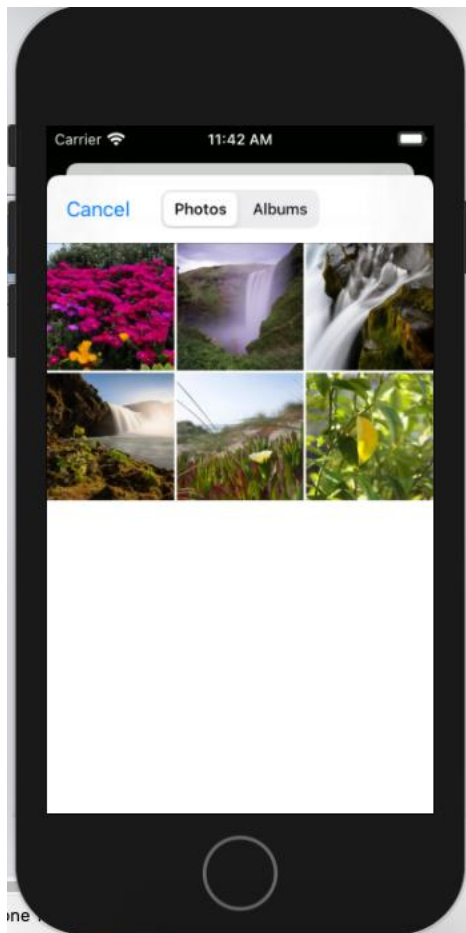
        arrayOfPictures.append(newImage)
        collectionView.reloadData()
    }

    func imagePickerControllerDidCancel(_ picker: UIImagePickerController)
    {
        picker.dismiss(animated: true, completion: nil)
    }
}

```

### Скріншот роботи програми





## **Висновок**

В цій лабораторній роботі була дороблена програма з лабораторної роботи №5. Було створено нову вкладку "Photos", в якому буде відображатись фотографії за шаблоном який був даний за варіантом. Як видно на скріншотах програма працює коректно при вертикальній і горизонтальній положеннях.