Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Розрахунково-графічна робота

з дисципліни
«Об’єктно орієнтоване програмування»

Виконав:

Студент групи ІП-04

Пащенко Дмитро Олексійович
номер у списку групи: 19

Перевірив:

Порєв Віктор Миколайович

Київ 2021

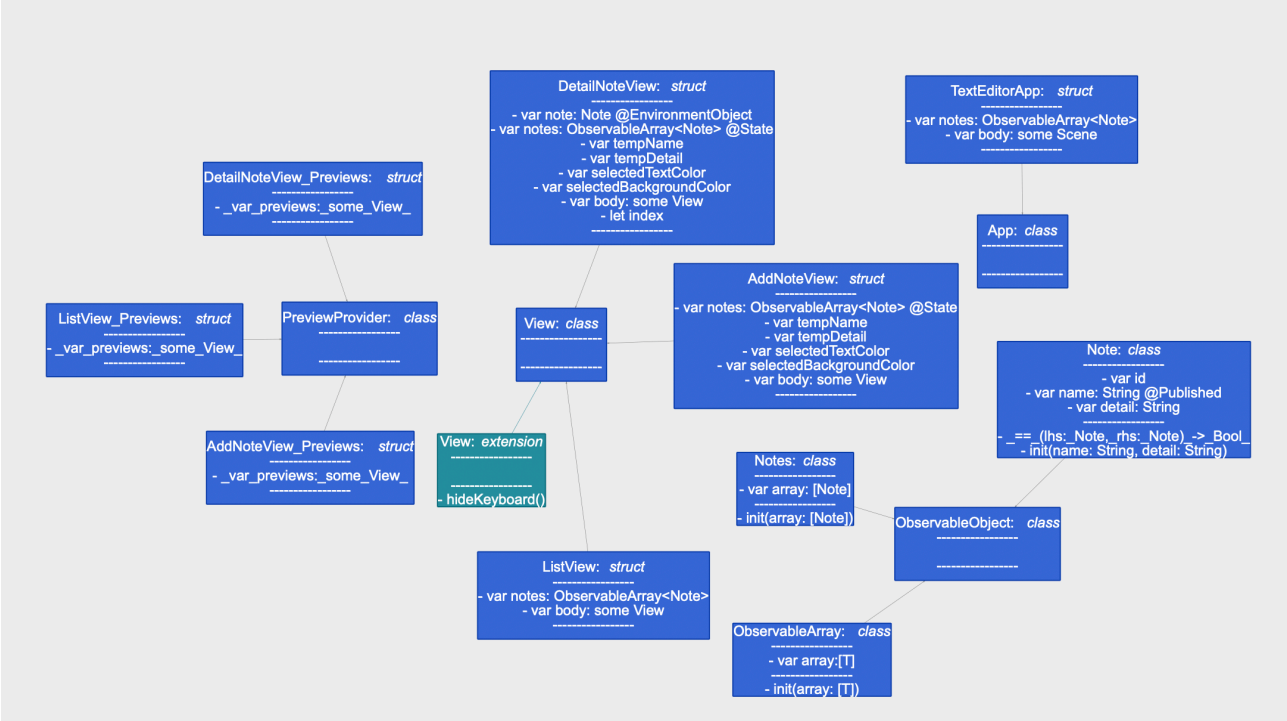
Варіант 19:

Створити програму для вводу та обробки текстів.

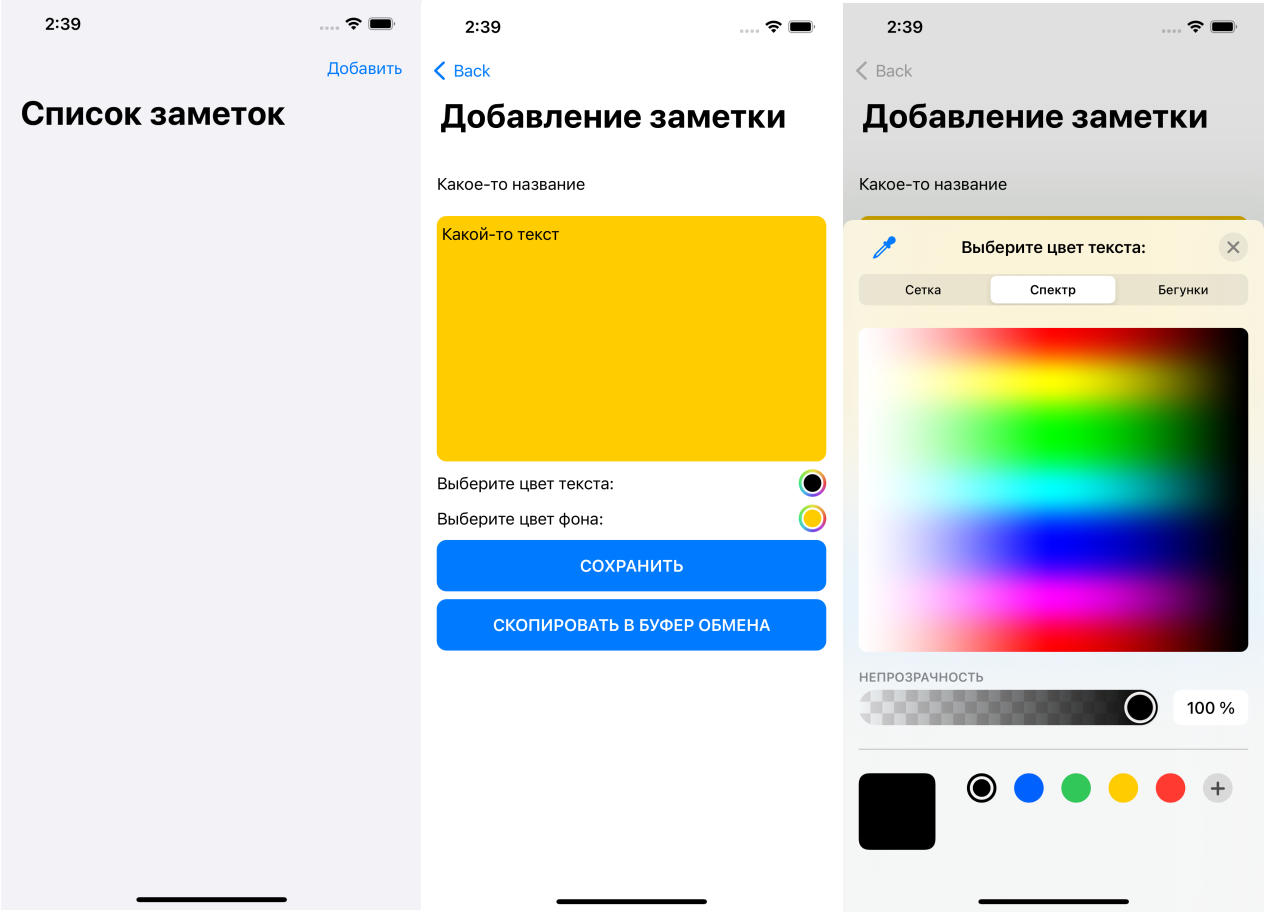
Використаний стек технологій:

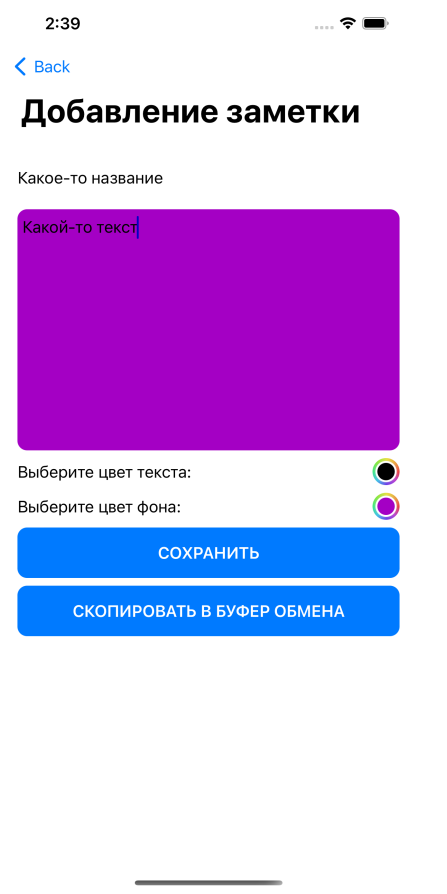
Мова програмування Swift, фреймворк для інтерфейсу SwiftUI.

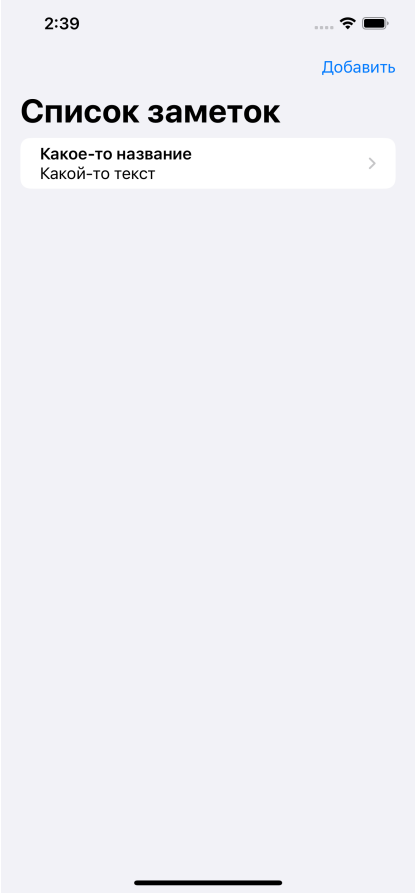
Діаграма класів:

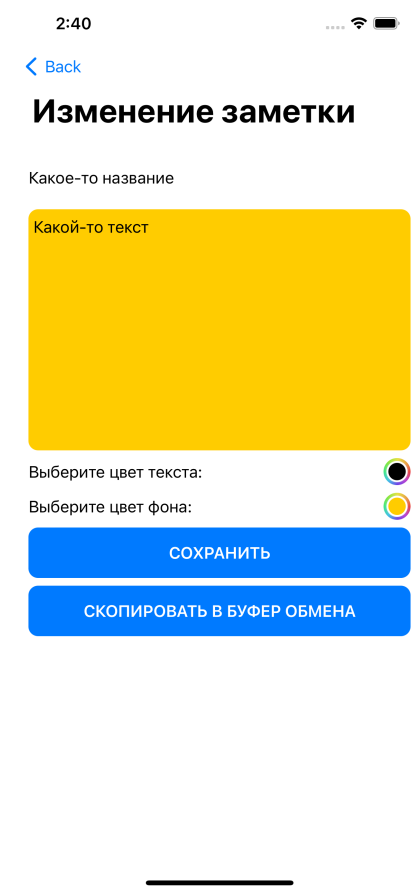


Скріншоти програми:









Код програми:

TextEditorApp.swift

```
import SwiftUI

@main
struct TextEditorApp: App {
    @State var notes: ObservableArray<Note> = ObservableArray(array: [])

    var body: some Scene {
        WindowGroup {
            ListView()
                .environmentObject(notes)
        }
    }
}
```

Model.swift

```
import Foundation

class ObservableArray<T>: ObservableObject {
    @Published var array:[T] = []

    init(array: [T]) {
        self.array = array
    }
}

class Note: ObservableObject, Identifiable, Equatable {
    static func == (lhs: Note, rhs: Note) -> Bool {
        return lhs.name == rhs.name && lhs.detail == rhs.detail
    }

    var id = UUID()
    @Published var name: String
    @Published var detail: String

    init(name: String, detail: String) {
        self.name = name
```

```
        self.detail = detail
    }
}

class Notes: ObservableObject {
    @Published var array: [Note]

    init(array: [Note]) {
        self.array = array
    }
}
```

ListView.swift

```
import SwiftUI

struct ListView: View {
    @EnvironmentObject var notes: ObservableArray<Note>

    var body: some View {
        NavigationView {
            List {
                ForEach(notes.array) { note in
                    NavigationLink(destination: DetailNoteView()
                                   .environmentObject(note)
                                   .environmentObject(notes)) {
                        VStack(alignment: .leading) {
                            Text(note.name)
                                .font(.headline)
                            Text(note.detail)
                                .font(.body)
                        }
                    }
                }
            }
            .navigationTitle("Список заметок")
            .toolbar {
                NavigationLink(destination:
AddNoteView().environmentObject(notes))
                {
                    Text("Добавить")
                }
            }
        }
    }
}

struct ListView_Previews: PreviewProvider {
    static var previews: some View {
        ListView()
    }
}
```

AddNoteView.swift

```
import SwiftUI
extension View {
    func hideKeyboard() {

UIApplication.shared.sendAction(#selector(UIResponder.resignFirstResponder), to:
nil, from: nil, for: nil)
    }
}

struct AddNoteView: View {
    @EnvironmentObject var notes: ObservableArray<Note>

    @State var tempName = ""
```

```
    @State var tempDetail = ""
    @State var selectedTextColor = Color.black
    @State var selectedBackgroundColor = Color.yellow

    var body: some View {
        VStack {
            TextField("Название", text: $tempName)
                .frame(height: 50)

            TextEditor(text: $tempDetail)
                .frame(height: 250)
                .foregroundColor(selectedTextColor)
                .colorMultiply(selectedBackgroundColor)
                .cornerRadius(10)

            ColorPicker("Выберите цвет текста:", selection: $selectedTextColor)
            ColorPicker("Выберите цвет фона:", selection:
$selectedBackgroundColor)

            Button {
                hideKeyboard()
                notes.array.append(Note(name: tempName,
                                        detail: tempDetail))
            } label: {
                Text("Сохранить".uppercased())
                    .font(.headline)
                    .foregroundColor(.white)
                    .padding()
                    .frame(maxWidth: .infinity)
                    .background(.blue)
                    .cornerRadius(10)
            }

            Button {
                UIPasteboard.general.string = tempDetail
            } label: {
                Text("Скопировать в буфер обмена".uppercased())
                    .font(.headline)
                    .foregroundColor(.white)
                    .padding()
                    .frame(maxWidth: .infinity)
                    .background(.blue)
                    .cornerRadius(10)
            }

            Spacer()
        }
        .padding()
        .navigationTitle("Добавление заметки")
    }
}

struct AddNoteView_Previews: PreviewProvider {
    static var previews: some View {
        AddNoteView()
    }
}
```

DetailNoteView.swift

```
import SwiftUI

struct DetailNoteView: View {
    @EnvironmentObject var note: Note
    @EnvironmentObject var notes: ObservableArray<Note>
```

```
    @State var tempName = ""
    @State var tempDetail = ""
    @State var selectedTextColor = Color.black
    @State var selectedBackgroundColor = Color.yellow

    var body: some View {
        VStack {
            TextField("Название", text: $tempName)
                .frame(height: 50)

            TextEditor(text: $tempDetail)
                .frame(height: 250)
                .foregroundColor(selectedTextColor)
                .colorMultiply(selectedBackgroundColor)
                .cornerRadius(10)

            ColorPicker("Выберите цвет текста:", selection: $selectedTextColor)
            ColorPicker("Выберите цвет фона:", selection:
$selectedBackgroundColor)

            Button {
                let index = notes.array.firstIndex(of: note)

                hideKeyboard()
                notes.objectWillChange.send()
                notes.array[index!].name = tempName
                notes.array[index!].detail = tempDetail
            } label: {
                Text("Сохранить".uppercased())
                    .font(.headline)
                    .foregroundColor(.white)
                    .padding()
                    .frame(maxWidth: .infinity)
                    .background(.blue)
                    .cornerRadius(10)
            }

            Button {
                UIPasteboard.general.string = tempDetail
            } label: {
                Text("Скопировать в буфер обмена".uppercased())
                    .font(.headline)
                    .foregroundColor(.white)
                    .padding()
                    .frame(maxWidth: .infinity)
                    .background(.blue)
                    .cornerRadius(10)
            }

            Spacer()
        }
        .onAppear(perform: {
            tempName = note.name
            tempDetail = note.detail
        })
        .padding()
        .navigationTitle("Изменение заметки")
    }
}

struct DetailNoteView_Previews: PreviewProvider {
    static var previews: some View {
        DetailNoteView()
    }
}
```