1. Реалізовані зміни для кращого object-oriented design

Нова версія коду використовує перерахування з ім'ям Priority для надання пріоритетів завдання. Перелік Priority надає кілька властивостей та методів для отримання та керування пріоритетами, такими як ідентифікатор, заголовок та колір. З іншого боку, стара версія коду представляє пріоритети у вигляді рядків і використовує метод styleForPriority для повернення відповідного кольору для кожного пріоритету. Перерахування Priority у новій версії коду є більш природним та об'єктно-орієнтованим способом представлення пріоритетів, оскільки він інкапсулює різні властивості та поведінку, пов'язані з пріоритетами, в одному типі.

Нова версія коду надає структуру з ім'ям TaskView, яка визначає візуальне подання завдання. Структура TaskView приймає об'єкт Task та відображає його інформацію у рядку, включаючи пріоритет, заголовок та статус обраного. З іншого боку, стара версія коду не надає окремого уявлення для подання завдання. Натомість він вбудовує уявлення задачі всередині структури ContentView, що робить її менш модульною і менш придатною для повторного використання. Відділення подання уявлення від моделі даних є найкращим принципом об'єктно-орієнтованого проектування, оскільки забезпечує краще поділ завдань та сприяє модульності та повторному використанню.

Нова версія коду використовує властивості, що обчислюються, для отримання кольору пріоритету з перерахування Priority, а також з об'єкта Task. З іншого боку, стара версія коду використовує функцію, яка приймає рядок, що має пріоритет, і повертає відповідний колір. Обчислювані властивості - це найбільш природний та ідіоматичний спосіб надання властивості, що залежить від інших властивостей. Крім того, реалізація колірної логіки в новій версії коду коду більш лаконічна та зручна для читання.

Загалом нова версія коду забезпечує більш об'єктно-орієнтований та модульний дизайн, ніж стара версія коду. Він використовує перерахування та обчислювані властивості для інкапсуляції даних та поведінки, сприяє поділу завдань та забезпечує чітку та лаконічну реалізацію.

2. Порівняння попередньої та оновленої версії програми :

Оновленна програма використовує анімацію SwiftUI та жести під час дотику, що може зробити її трохи повільнішою, ніж попердня програма, яка не використовує жодної анімації.

Розмір коду: оновленна програма трохи довша за другу через включення спеціальної структури TaskView та функцій updateTask і deleteTask. Однак оновленна програма більш модульна, і вона розділяє код, пов’язаний зі створенням, оновленням і видаленням завдань, на окремі функції, що полегшує його читання та змінення.

Підсумовуючи: оновленна програма є більш модульною та використовує анімацію SwiftUI, але вона трохи довша та може працювати повільніше, ніж попередня програма.