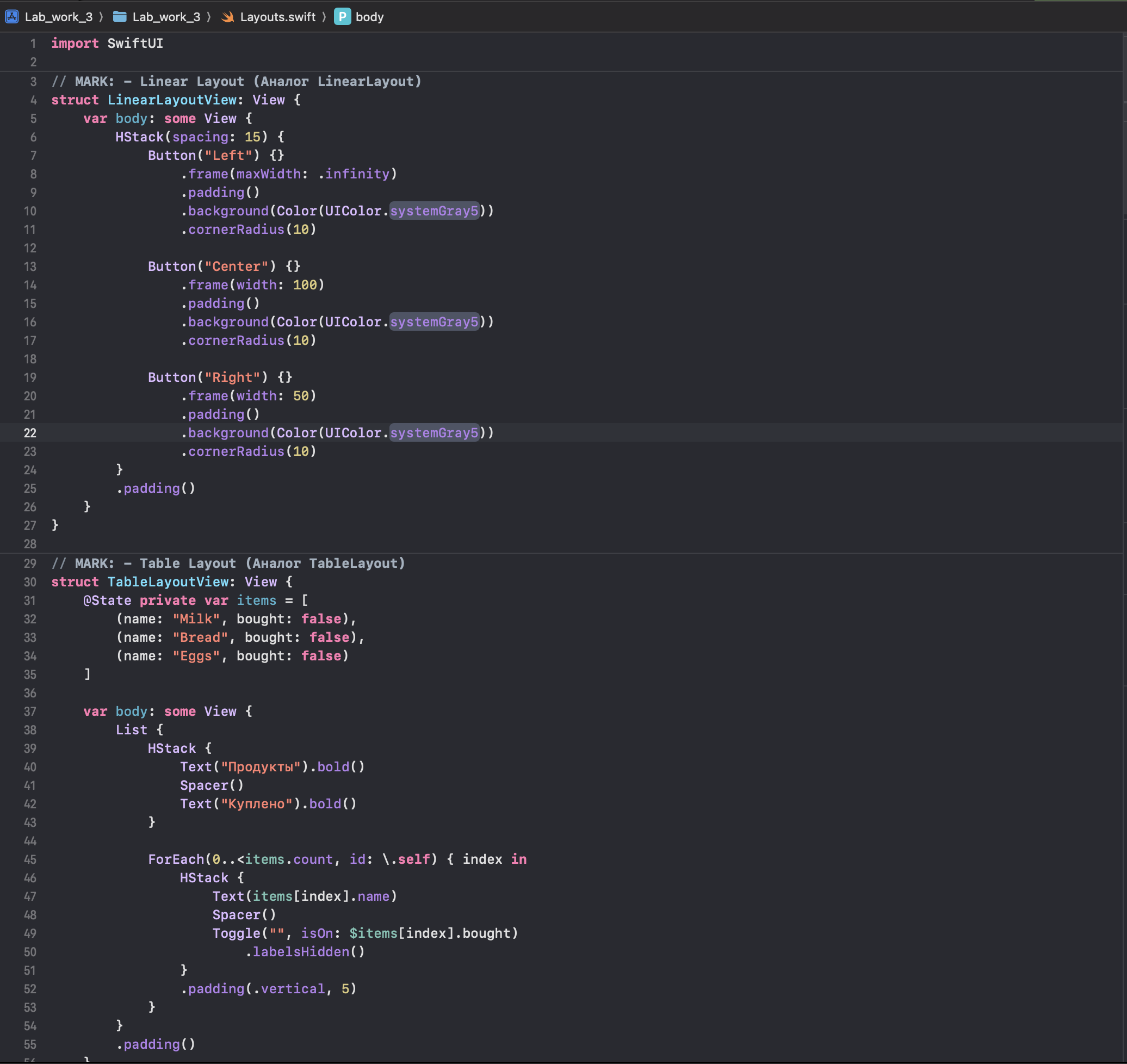
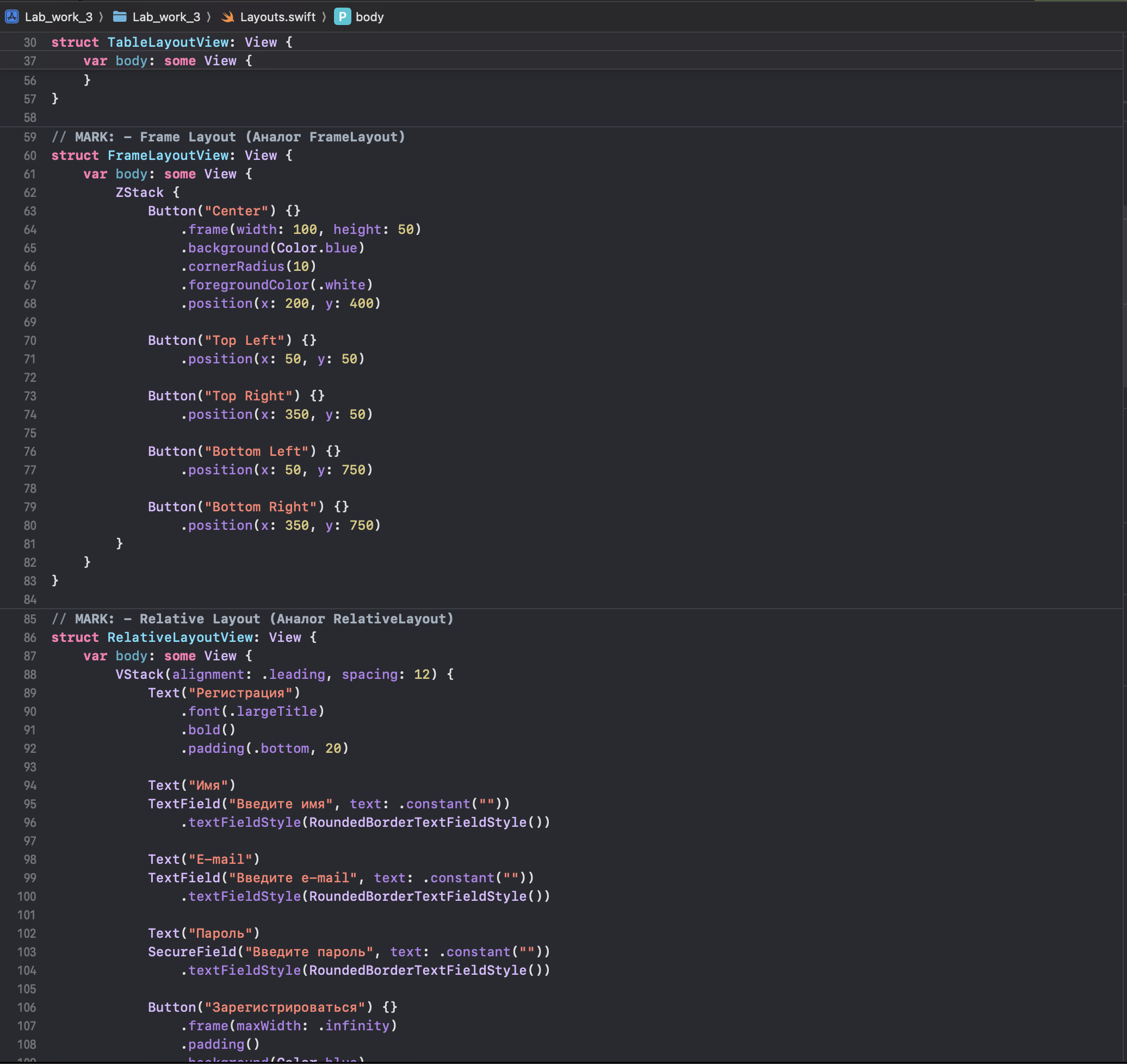
Лабораторная работа 3

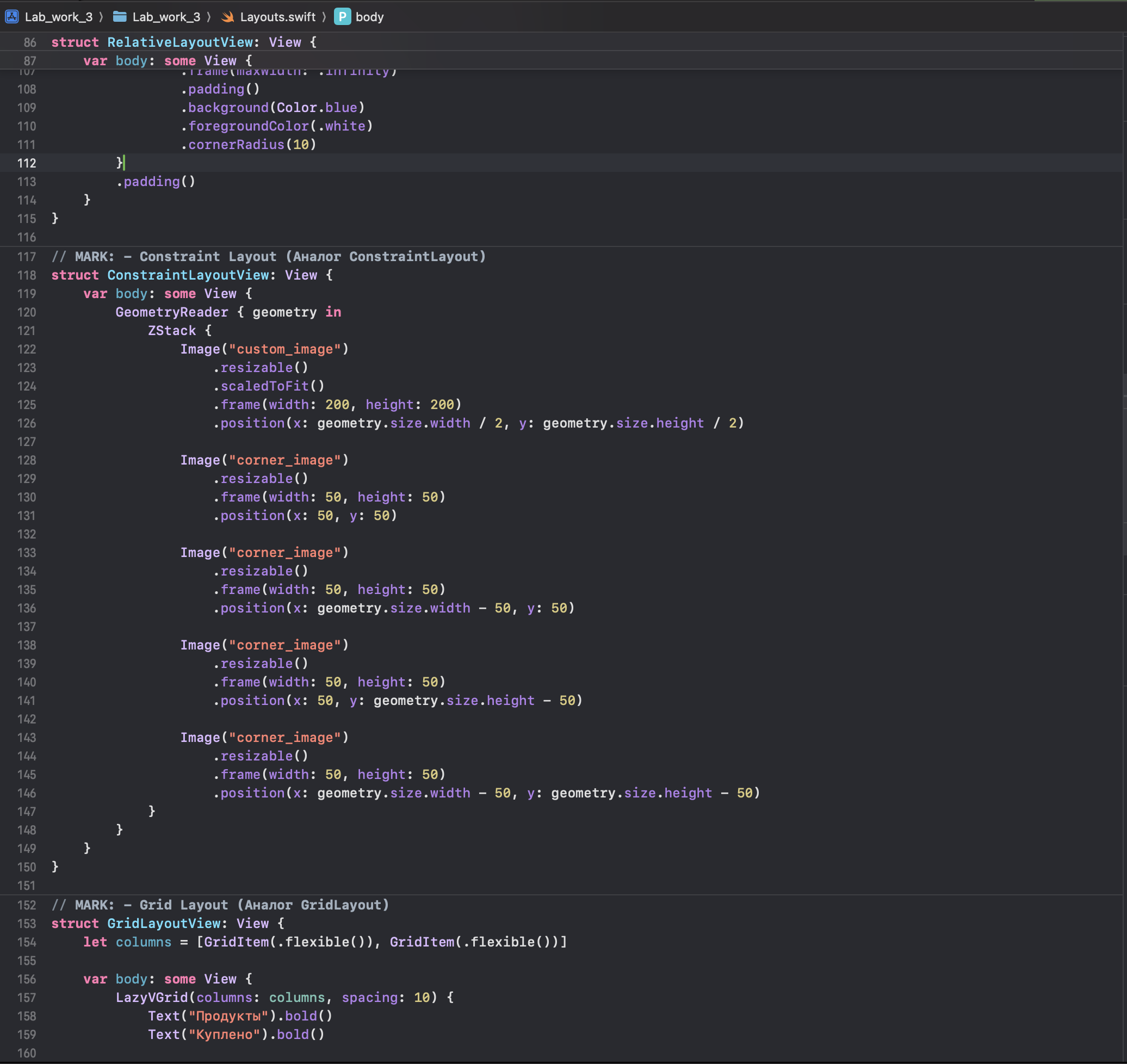
Менеджеры компоновки

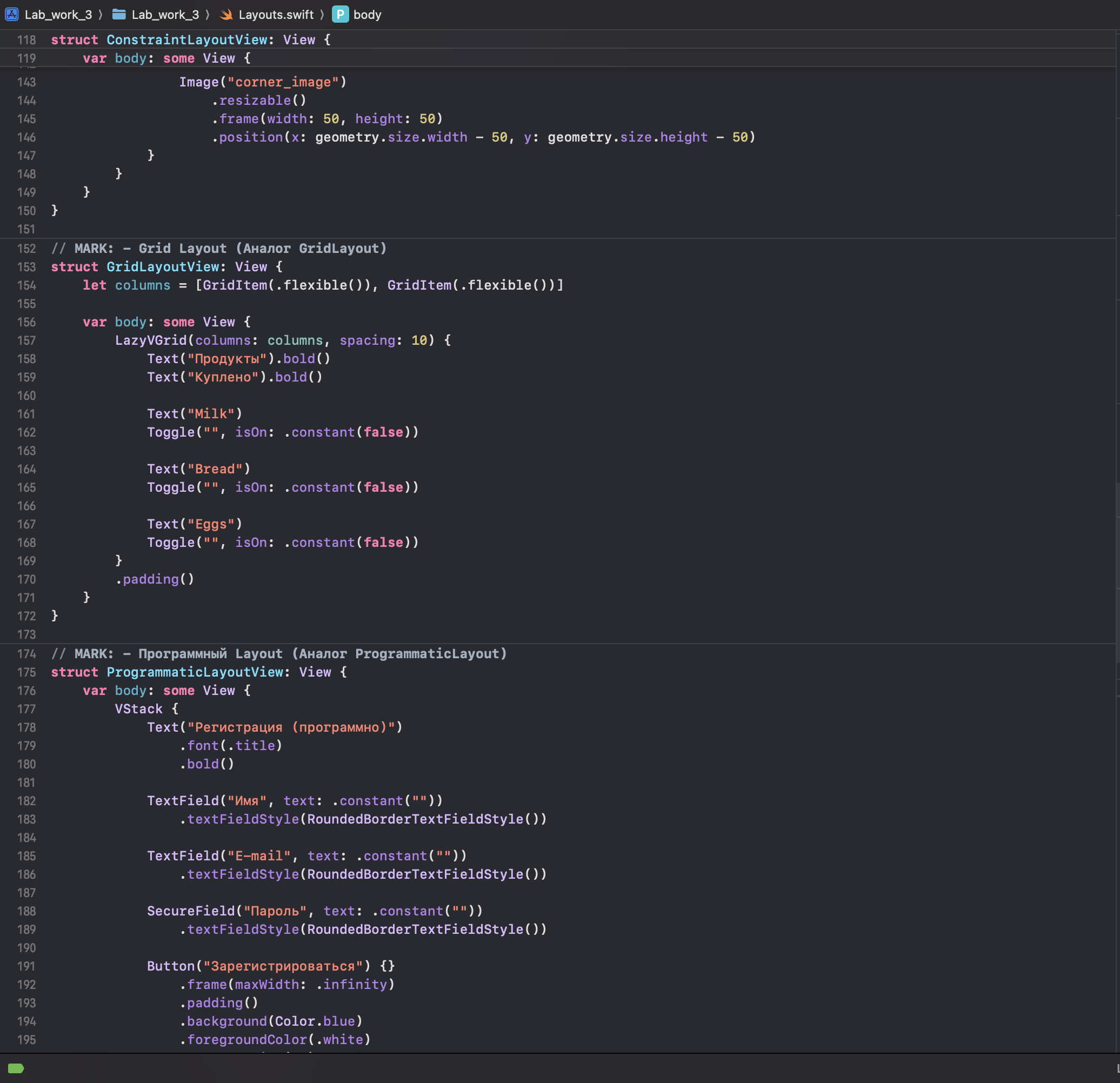
Цель: Разработка простого приложения, работа с Layouts.

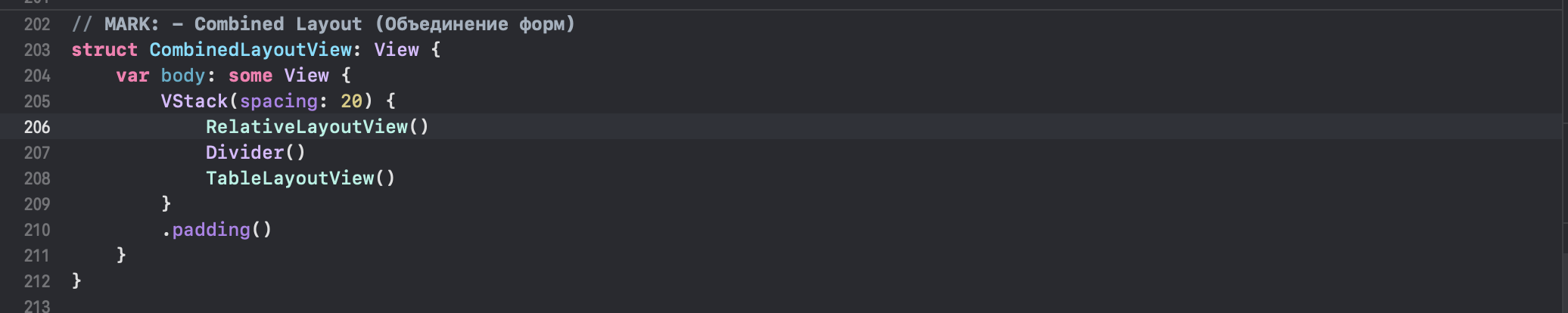
Пояснение реализации:

Рисунок 1. Скриншот 1 из файла Layouts

Рисунок 2. Скриншот 2 из файла Layouts

Рисунок 3. Скриншот 3 из файла Layouts

Рисунок 4. Скриншот 4 из файла Layouts

Рисунок 5. Скриншот 5 из файла Layouts

Этот файл содержит реализацию различных типов макетов (Layout) в SwiftUI, аналогичных стандартным макетам в Android (LinearLayout, TableLayout, FrameLayout и т. д.).

Структура файла:

• LinearLayoutView – горизонтальное расположение элементов.

• TableLayoutView – таблица с динамическим списком.

• FrameLayoutView – абсолютное позиционирование элементов.

• RelativeLayoutView – форма с полями ввода.

• ConstraintLayoutView – элементы с привязками (GeometryReader).

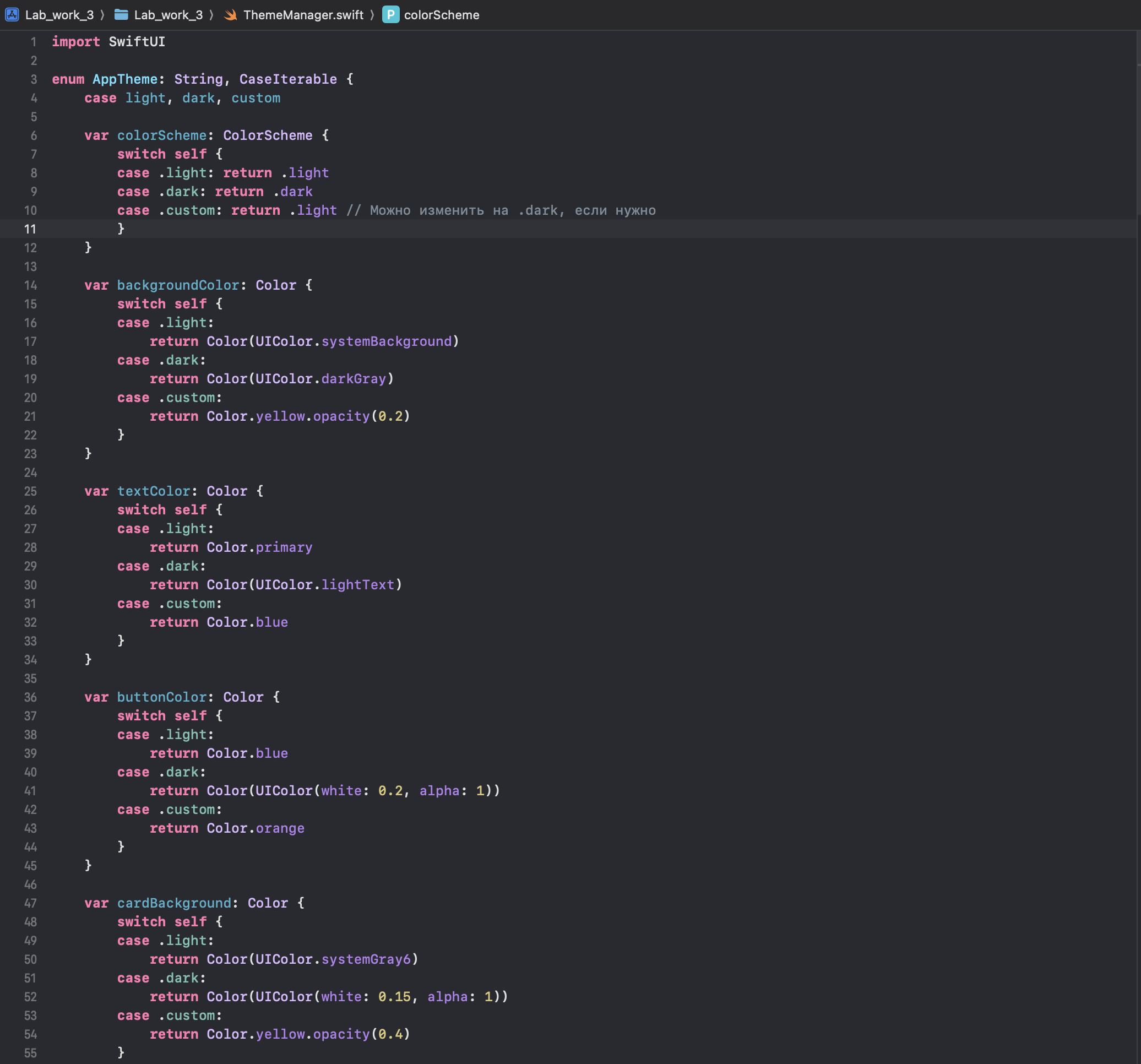
• GridLayoutView – сетка с двумя колонками.

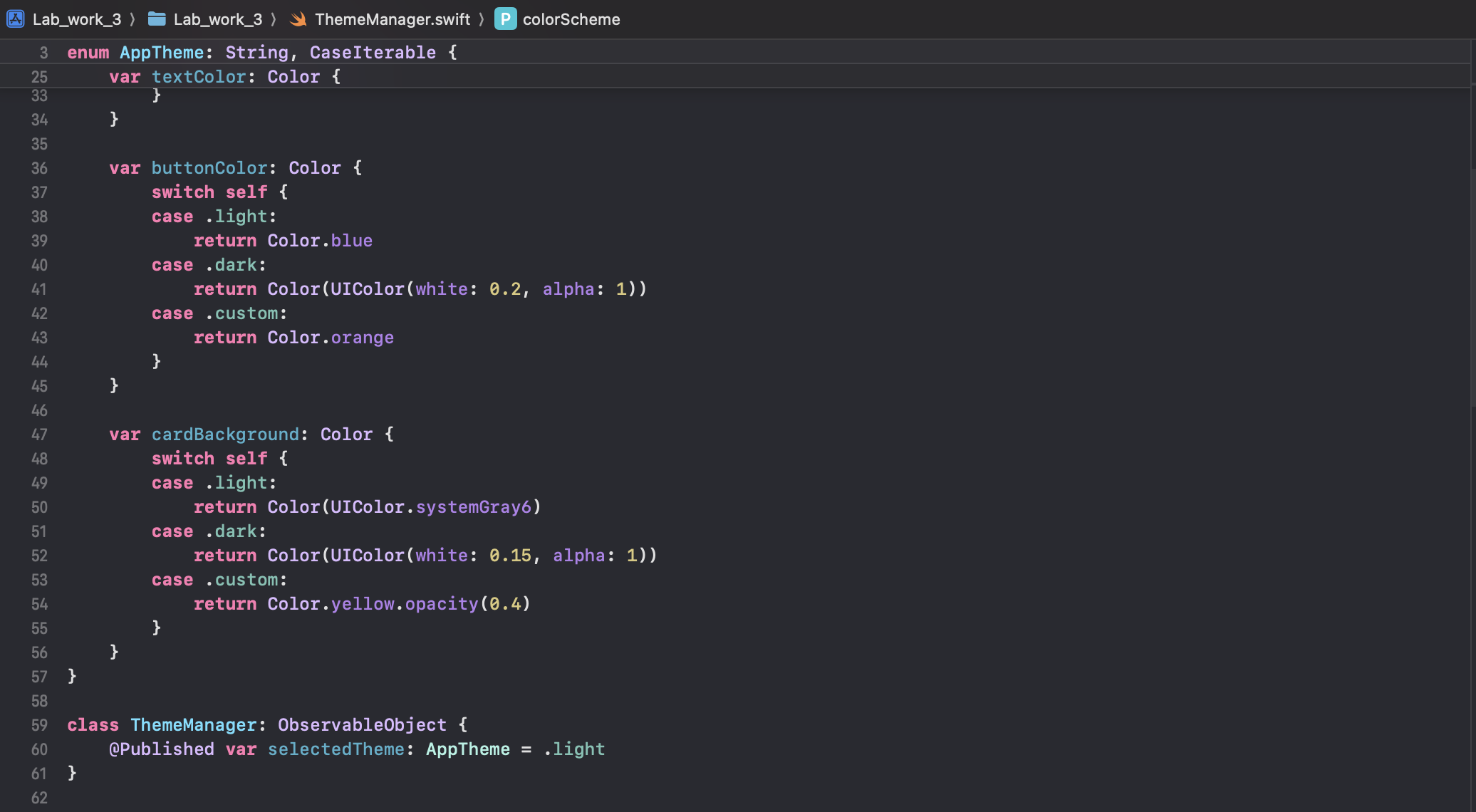
• ProgrammaticLayoutView – форма регистрации, построенная п программно.

• CombinedLayoutView – комбинированный макет (форма + список)..

Рисунок 6. Файл TextStyles

Этот файл предоставляет удобный способ стилизации текста, позволяя легко изменять размер шрифта, автоматически применять жирность и моноширинный стиль. Также используется расширение (extension), что делает код более чистым и удобным для использования.

Рисунок 7. Скриншот 1 из файла ThemeManager

Рисунок 8. Скриншот 2 из файла ThemeManager

Этот файл создает систему, позволяя пользователю переключаться между тремя вариантами тем:

• Светлая (light) → стандартные цвета.

• Тёмная (dark) → глубокий чёрный фон, светло-серый текст.

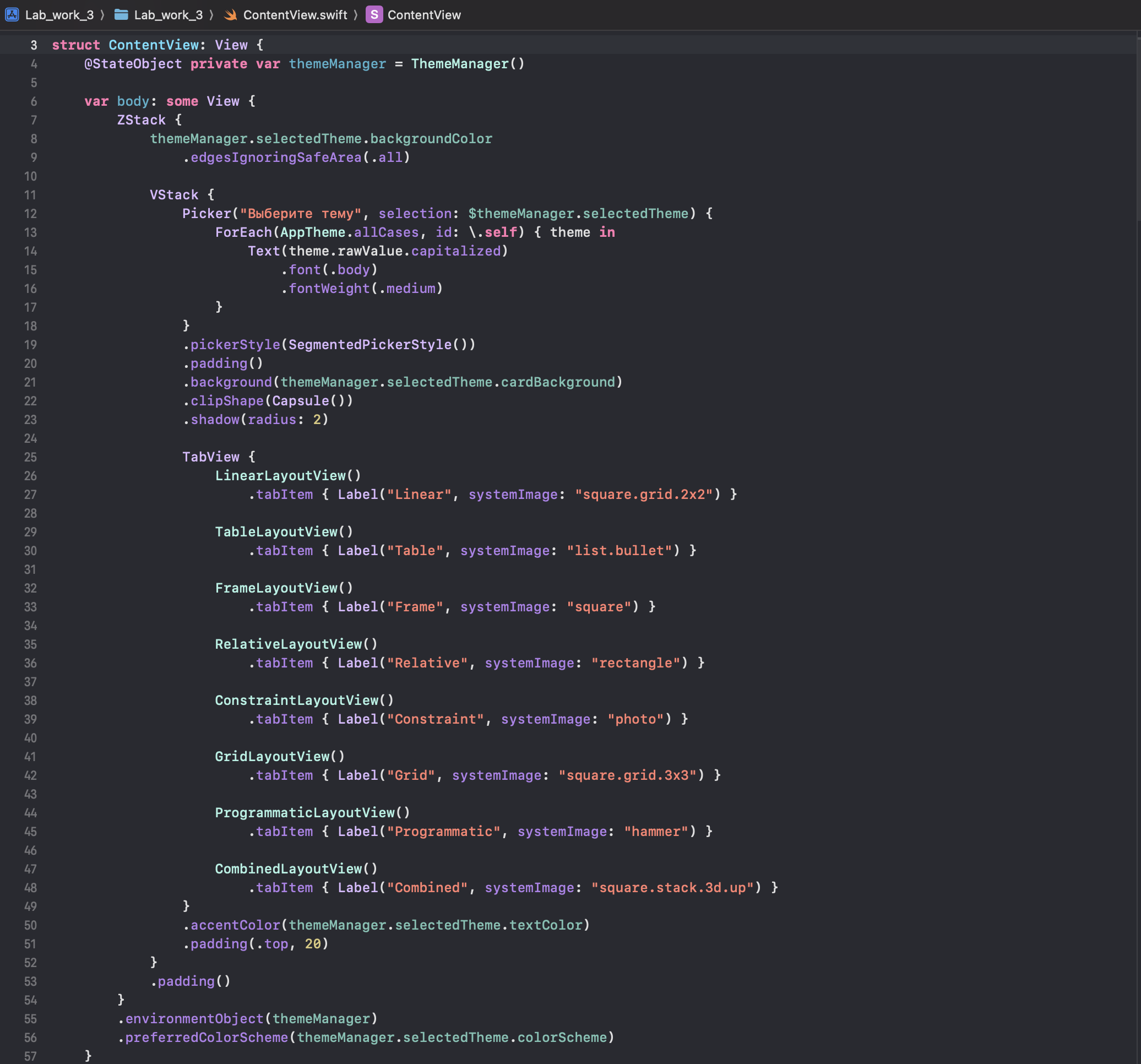
• Кастомная (custom) → жёлтые фоны, синий текст.

Основные элементы:

• AppTheme – управляет цветами для каждой темы.

• ThemeManager – хранит текущую выбранную тему.

• SwiftUI автоматически обновляет интерфейс при смене темы.

Рисунок 6. листинг файла ContentView

В этом файле описана главная страница приложения, на которой реализована возможность переключения тем с помощью themeManager, который управляет текущей темой. Также предусмотрена навигация между экранами с использованием панели вкладок (TabView), и автоматически изменяется фон и цвета элементов в зависимости от выбранной темы.

Ответы на контрольные вопросы:

1. Виды разметок (менеджеров компоновки):

LinearLayout

RelativeLayout

ConstraintLayout

FrameLayout

TableLayout

GridLayout

AbsoluteLayout (устаревший)

ScrollView

GridView

1. Ориентации экрана:

Вертикальная (portrait)

Горизонтальная (landscape)

1. Layout-файл:

Это XML-файл, который описывает интерфейс приложения и его компоненты (кнопки, текстовые поля и т.д.).

1. Папка для файлов разметки:

В папке res/layout внутри структуры проекта.

1. Спецификатор для горизонтальной ориентации экрана:

layout-land (создаётся папка res/layout-land для горизонтальной ориентации).

1. Разница между RelativeLayout и AbsoluteLayout:

RelativeLayout: Позволяет позиционировать элементы относительно других элементов.

AbsoluteLayout: Позиционирует элементы с точными координатами на экране (устаревший).

1. Атрибуты для одинакового размера и расположения вверху:

Для всех элементов в LinearLayout используйте layout\_width="100dp", layout\_height="100dp", и android:gravity="top".

1. Центрирование элементов (красный квадрат по центру внизу, зеленый слева, синий справа):

Используйте RelativeLayout и атрибуты:

Для красного: android:layout\_centerHorizontal="true" android:layout\_alignParentBottom="true".

Для зеленого: android:layout\_alignParentLeft="true" android:layout\_centerVertical="true".

Для синего: android:layout\_alignParentRight="true" android:layout\_centerVertical="true".

1. Формат записи ID в коде:

ID записывается как @+id/название\_элемента, например, @+id/button1.

1. Роль файла R.java:

Это автоматически сгенерированный файл, который содержит идентификаторы всех ресурсов (например, строки, изображения, элементы интерфейса) в проекте.

1. Работа метода findViewById:

Метод используется для получения ссылки на элемент интерфейса (например, кнопку или текстовое поле) по его ID.