

# Домашнее задание №3

## “Верстка”

### Постановка задачи

Из домашнего задания по интерфейсам, требуется реализовать минимум 2 экрана. Для наполнения экрана требуется использовать собственные объекты с данными, которые потом мапятся на поля верстки.

Если нет ДЗ по интерфейсам, то необходимо придумать экраны

- 1й со списком ячеек, например еды с названием, описанием и картинкой)
- 2й отдельная страница с подробной информацией
- добавлена навигация для переходов
- Возможно релизована навигация дополнительно через `bottomNavigation`, табы

Данные можно мапить из `api` или создать отдельный файл моковых данных, который симулирует ответы сервера (обязательно асинхронно со `sleep` в 500мс)

### Ограничения и требования

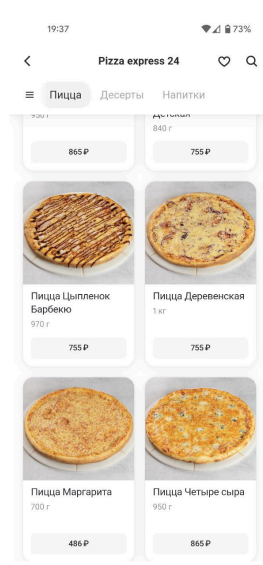
1. Требуется использовать `Fragment` или `Compose View`.
2. Приложение не должно содержать хардкод.
3. Приложение должно использовать ресурсы(`resources`) для работы
4. В коде можно оставлять комментарии, но в конечной версии нельзя оставлять `Log`
5. Между реализованными экранами должны быть переходы, или любой другой способ попасть на эти экраны после запуска приложения.
6. Данные экранов нельзя зашивать в верстке, они должны мапиться в верстку при помощи объектов с данными
7. Данные поставляться на экраны должны через не-синхронные вызовы, и задержка для получения данных должна быть в 2 секунды и иметь возможность ее увеличить (в коде).
8. Экраны или блоки с данными, должны поддерживать состояние загрузки.

### Что будет плюсом

1. Поддержка состояния ошибки всеми экранами и блоками
2. Поддержка пустого состояния блоками с данными и экранами
3. Реализация больше трех экранов
4. Использование `API` для получения данных для экранов.

## Основные сценарии проверки

1. Переход на каждый экран, ожидаем увидеть состояние загрузки, и состояние, когда данные загружены.
2. Меняем количество объектов в данных (если это списочные данные), и блок должен уметь обработать отображение как 1-го элемента, так и большого количества данных (за исключением тех моментов, когда блок реально рассчитан под конкретное количество данных).
  - а. Пример. У нас есть экран с меню. Где горизонтальная лента - это категории блюд, вертикальная лента - это список блюд в категории. Категории блюд - это блок, где количество элементов скорее будет иметь какие-то ограничения (количество категорий блюд - сильно меньше всех блюд). А вот уже список блюд - это уже блок, где количество элементов может очень сильно расти.



3. (Если есть обработка пустых состояний или ошибок) Переход на необходимый экран, видим загрузку. После загрузки видим ошибку или состояние блока, при отсутствии данных.