

# Лабораторная – Классы пользовательского интерфейса

---

Создайте сложный пользовательский интерфейс с классами пользовательского интерфейса Android

## Цели:

Познакомиться с классами пользовательского интерфейса (UI) Android. Создать простое приложение, которое использует разнообразные элементы UI, включая: Button, TextView и Checkbox. Вы также объедините те знания, которые получили в предыдущих уроках для реализации большей части приложения с нуля.

## Обзор:

В этой лабораторной вы создадите Приложение ToDo-менеджер. Приложение создает и управляет списком дел (ToDo, вещи, которые нужно сделать, “to do”). Вы спроектируете пользовательский интерфейс приложения, включая его макеты и файлы ресурсов. Вы также реализуете функциональность приложения, как вы делали в предыдущих лабораторных. Не модифицируйте идентификаторы ресурсов, содержащихся в файлах макетов скелета приложения, так как это приведет к повреждению тест-кейсов.

## Упражнение А: Базовый ToDo Менеджер

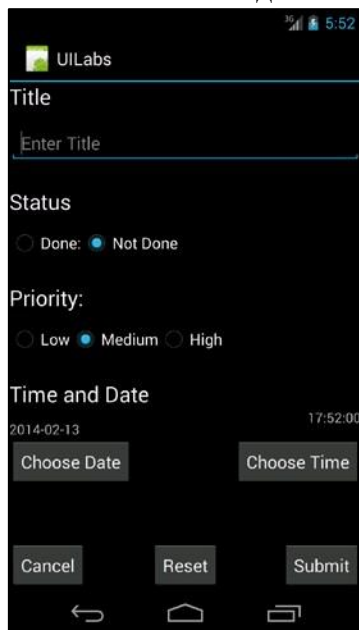
Главный Activity для данного приложения называется Lab4\_UI\_Labs. Когда оно запускается, но существуют ранее сохраненные элементы ToDo, ее изначальный UI будет выглядеть как-то так:



*Рис. 1. Изначальный Вид*

Этот пользовательский интерфейс содержит отдельный ListView, который отображает все существующие элементы ToDo. Как показано выше, последняя строка списка всегда содержит специальный View со словами “Add New ToDo Item.” Последняя позиция в списке ListView известна как футер. Вы можете добавить View в футер, используя метод `addFooterView()` класса ListView.

Когда пользователь нажимает на футер ListView, Приложение стартует новый экран, названный `AddToDoActivity` который позволяет пользователю создавать и изменять новый элемент ToDo.



*Рис. 2. Добавление нового элемента*

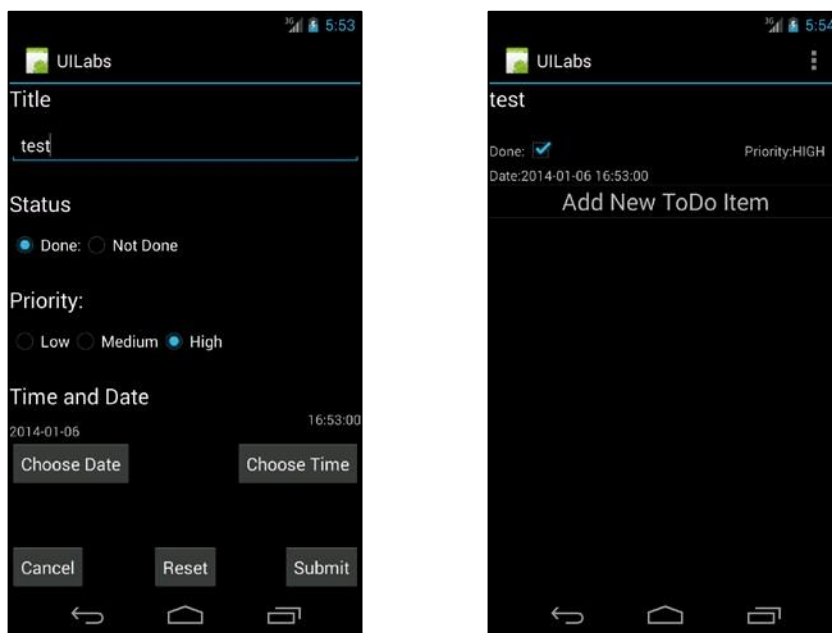
Элементы ToDo имеют как минимум следующий набор полей. Значения по умолчанию выделены жирным шрифтом:

- Title: заголовок, строка, введенная пользователем. Значение по умолчанию – пустая строка ("").
- Status: статус, один из {Done, **Not Done**}
- Priority: приоритет, один из {Low, **Med**, High}
- Date & Time: Время и Дата, дедлайн для выполнения этого элемента ToDo. Дедлайн по умолчанию **7 дней от текущей даты и времени**.

Пользовательский интерфейс этого Activity включает следующие кнопки:

- Cancel – завершить Activity без создания нового элемента ToDo.
- Reset – обнулить поля ToDo элемента к их изначальным значениям и обновить пользовательский интерфейс для отображения изменений.
- Submit – создать новый элемент ToDo, содержащий данные полей, введенные пользователем и вернуть их в ToDoManagerActivity. Когда приложение возвращается к ToDoManagerActivity, новый элемент ToDo должен появиться в списке ListView.

Например, если пользователь создает и отправляет новый элемент в пустой список, тогда при возврате приложения в ToDoManagerActivity, его ListView будет содержать новый элемент ToDo как показано ниже.



*Рис. 3. (слева): Пользователь создает новый элемент ToDo. (справа): После отправки нового элемента, приложение возвращается в главный Activity, отображая новый элемент*

Вернувшись на главный экран, пользователь сможет переключить чекбокс Done для обозначения того, что статус элемента изменился, скажем с Not Done на Done.

См. ролик UILabs.mp4 для того чтобы увидеть приложение в действии.

## Заметки по реализации:

1. Скачайте файлы скелета приложения для проекта Lab4\_UILab и импортируйте их в свою IDE. В случае с IntelliJ Idea Import Project -> Import project from existing sources.
2. Реализуйте проект в соответствии со спецификациями, описанными выше. Для реализации Лабораторной, см. на комментарии в файлах скелета приложения, содержащие строку "//TODO." Как в предыдущих лабораторных, эти комментарии содержат подсказки, что нужно сделать, чтобы завершить проект. Однако, будьте осторожны, т.к. начиная с этого момента данные подсказки не будут столь подробными, требуя от вас все больше принимать решения по поводу реализации всего проекта.
3. Не забудьте, что элементы View могут быть повторно использованы в ListAdapter в его методе getView(). Для повторного использования вам следует очистить ВСЕ старые состояния View.

## Тестирование

Тест-кейсы для этой лабораторной находятся в проекте Lab4\_UILabTest. Вы можете запускать их все разом или по одному с помощью контекстного меню. Тестовые классы используют Robotium.

По мере реализации и осуществления различных шагов Лабораторной, запускайте тесты для того чтобы увидеть, как далеко вы продвинулись к цели.

### Предостережения:

1. Эти тестовые кейсы были протестированы на Galaxy Nexus AVD с API level 18. Для ограничения проблем конфигурации вам следует тестировать ваше приложение на подобном AVD. Также, когда тестируете, убедитесь, что ваше устройство находится в портретном режиме, когда запускаются тест-кейсы.
2. При запуске этих тест кейсов удалите все существующие элементы ToDo.

## Отправка

Чтобы отправить отчет о работе вам нужно загрузить архив проекта, при этом очистив его от скомпилированных ресурсов. Это можно сделать с помощью gradlew clean или с помощью

Build->Clean или с помощью ручного удаления папок со сгенерированными .class и .dex файлами.

Если вы закончили Упражнение А и прошли все тесты, и чувствуете, что вы бы могли сделать больше, здесь несколько предложений по добавлению. Это необязательно и не на оценку, но если вы делаете что-то круто, это всегда потом проявляется. Делитесь своими продвинутыми результатами с помощью скриншотов в системе Буратино.

### Необязательное Упражнение В: Более приятное приложение ToDo Менеджер

Сейчас ToDo менеджер достаточно страшный. Попробуйте модифицировать файлы макетов для создания более привлекательного и легкого в использовании приложения. Например, поиграйтесь с размером шрифта текста, количеством отступов вокруг элементов, цветами фона и др.

### Необязательное Упражнение С: Улучшенное приложение ToDo Менеджер

Модифицируйте ваше приложение, чтобы ToDoItems чей статус Not Done отображался в Главном Activity с другим цветом фона, чем те, у кого статус Done. Если вы это сделаете, тогда когда пользователь переключает чекбокс Done цвет фона тоже должен изменить на соответствующий. Вы также возможно должны рассмотреть изменение цвета фона или добавления предупреждающей иконки, если дедлайн записи слишком близко. В итоге, прямо сейчас пользователь не может модифицировать приоритет элементов списка. Модифицируйте этот пользовательский интерфейс так чтобы он позволял перетаскивать элементы списка, позволяя пользователям задавать различный Приоритет.

### Необязательное Упражнение D: ToDo Менеджер, который вы возможно будете использовать

Модифицируйте ваше приложение так, чтобы при долгом нажатии на ListView выскакивал диалог, позволяющий удалить выбранный элемент списка. Если вы действительно чувствуете, что в приложении должны быть закладки, то сделайте это. Пусть одна из них отображает элементы упорядоченные по дедлайну. А другая закладка отсортирует по приоритету, а внутри элементов с одинаковым приоритетом по дедлайну.