

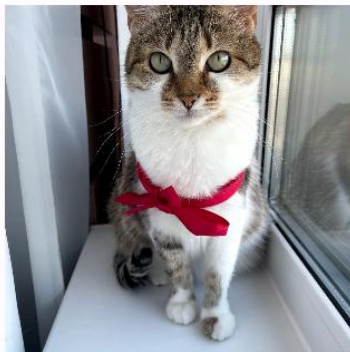
Практическая работа

Цели практической работы

- Научиться применять базовые компоненты пользовательского интерфейса.
- Создать экран профиля пользователя, используя такие компоненты, как TextView, ImageView, TextInputLayout/TextInputEditText, RadioButton/RadioGroup, MaterialCheckBox, SwitchMaterial, LinearProgressLayout, Button, Snackbar, Toast.

Что нужно сделать

Создайте экран профиля пользователя, к элементам которого применены стили и который имеет следующий вид:



Имя

Кошачий Пользователь



20/40

Телефон

89083 156678

Пол

☒ М

☐ Ж

Получать уведомления



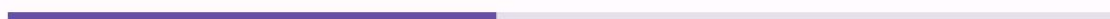
Об авторизации на устройстве



О новинках и предложениях

Баллы

44/100



Сохранить

Экран состоит из следующих компонентов:

1. **Фото пользователя.** Квадратное изображение размером 120 × 120 dp. Добавьте подходящее изображение. Для оптимального отображения картинки попробуйте воспользоваться разными scale type.
2. **Имя.** Поле для ввода текста. Текстовая клавиатура, ограничение на ввод — не более 40 символов. Доступна опция удаления текста.
3. **Телефон.** Поле для ввода текста. Тип ввода — номер телефона. Применён стиль FilledBox, цвет фона — фиолетовый (#BB86FC).
4. **Заголовок «Пол».** Стиль: размер текста — 16 sp, цвет текста — чёрный (#000000).
5. RadioGroup с заголовком для выбора пола.
6. Switch для включения/отключения уведомлений.
7. Два checkbox для выбора уведомлений, которые пользователь хочет получать. Стиль: цвет — бирюзовый (#03DAC5).
8. **Заголовок «Баллы».** Стиль: размер текста — 16 sp, цвет текста — чёрный (#000000).
9. Линейный прогресс отображает количество баллов пользователя от 0 до 100.
10. **Текстовая информация о количестве баллов.** Стиль: размер текста — 16 sp, цвет текста — серый (#444444).
11. Кнопка сохранения.

Требования к стилям

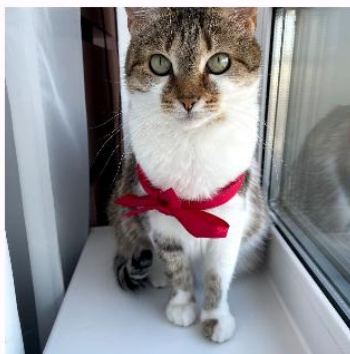
- Стили текстов применяются через textAppearance.
- Похожие стили наследуются друг от друга (заголовки и количество баллов).
- Стили checkbox применяются в теме (атрибут в теме checkboxStyle — по аналогии с materialButtonStyle из видео, базовый стиль — для Checkbox Widget.Material3.CompoundButton.CheckBox).

Настройки темы

Используя свои цвета, переопределите атрибуты темы colorPrimary, colorSecondary и другие. Изучите, каким образом будет меняться внешний вид элементов, к которым вы не применяли собственные стили.

Логика работы Switch — переключателя уведомлений

Если Switch выключен, список получаемых уведомлений блокируется. Иначе пользователь может выбирать из множественного списка.



Имя

Кошачий Пользователь

20/40

Телефон

89083 156678

Пол

☒ М

☐ Ж

Получать уведомления



☐ Об авторизации на устройстве

☐ О новинках и предложениях

Баллы

44/100



Сохранить

Логика работы линейного прогресса

Установите значение прогресса и текстового поля, описывающее прогресс случайным образом с помощью

```
Random.nextInt(101)
```

который вернёт число от 0 до 100 включительно.

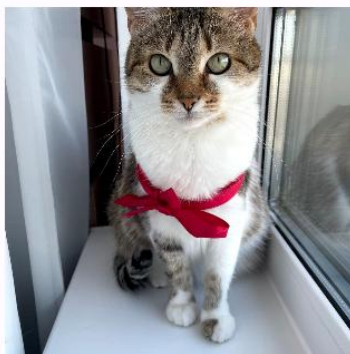
Логика работы кнопки «Сохранить»

Кнопка активна, если выполняются все следующие условия:

1. значение в поле «Имя» введено корректно, то есть поле не пустое и количество символов не превышает 40;
2. значение в поле «Телефон» введено корректно, то есть поле не пустое;
3. выбран пол;
4. если активен переключатель, должен быть выбран хотя бы один вид уведомлений.

При нажатии на активную кнопку «Сохранить» показывается уведомление о том, что информация сохранена.

8:03



Имя

Кошачий Пользователь

20/40

Телефон

89083 156678

Пол

☒ М

☐ Ж

Получать уведомления

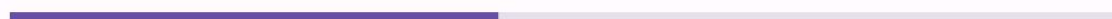


☐ Об авторизации на устройстве

☐ О новинках и предложениях

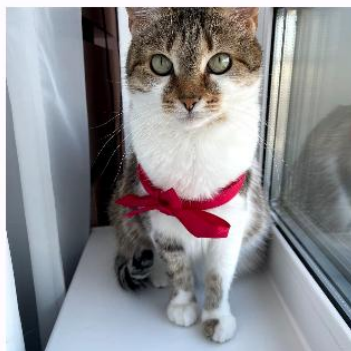
Баллы

44/100



Изменения сохранены

Для показа уведомления о сохранении можно использовать Snackbar или Toast.
Если нарушено хотя бы одно из условий a–d, кнопка «Сохранить» неактивна.



Имя

0/40

Телефон

Пол

☒ М

☐ Ж

Получать уведомления



☐ Об авторизации на устройстве

☐ О новинках и предложениях

Баллы

44/100

Сохранить

Порядок действий

1. Создайте пустой Android-проект. Убедитесь, что подключена библиотека Material Design.
2. Подготовьте стили и темы, необходимые для проекта, добавьте ресурсы. Найдите подходящее изображение для профиля.
3. Сверстайте layout по предложенному макету, примените все необходимые свойства для компонентов, чтобы их внешний вид и поведение соответствовали заявленным требованиям.
4. Последний шаг — это код. Опишите логику программы в главном Activity. Подпишитесь на уведомления об изменениях компонентов и на основании получаемых данных принимайте решения.

Советы и рекомендации

- Чётко следуйте плану. Начать лучше с ресурсов проекта и вёрстки.
- За блокировку view отвечает свойство isEnabled.
- Выносите значения строк, размерностей, цветов в ресурсы.
- При проверке условий используйте логическое И (&&), когда требуется выполнение всех частей. А && В истинно тогда и только тогда, когда и А и В истинны.
Когда достаточно выполнения хотя бы одного из условий, используйте логическое ИЛИ (||). А || В истинно, когда истинно только А, только В, и А и В.

Что оценивается

- UI содержит все необходимые компоненты, применены необходимые стили, тема настроена.
- Логика программы не нарушена.
- Нет критичных багов и вылетов.
- Код чистый, у переменных и компонентов понятные названия, соблюдаются принципы ООП-.

Как отправить работу на проверку

1. Используйте репозиторий android_dev_1_2022.
2. Скачайте изменения в репозитории на ваш компьютер.
3. Выполните практическую работу в папке m4_components. Создайте проект внутри папки, сделайте коммит или несколько коммитов. Отправьте коммиты в удалённый репозиторий.
- 4.