

Задание 1 к занятиям в пятницу 28.02.2020	1
Задание 2 к занятиям в четверг 05.03.2020	1
Обязательное домашнее задание:	1
Дополнительное задание со звёздочкой:	3
Задание 3 к занятиям в пятницу 06.03.2020	4
Задание 4 к занятиям в четверг 12.03.2020	4
Задание 5 к занятиям в четверг 19.03.2020	8
Задание 6 к занятиям в четверг 02.04.2020	11
Задание 7 к занятиям в четверг 09.04.2020	11

Задание 1 к занятиям в пятницу 28.02.2020

Обязательно принести ноутбук

1. Установить JDK 11

<http://www.oracle.com/technetwork/java/javase/downloads/jdk9-downloads-3848520.html>

2. Установить Android Studio

<https://developer.android.com/studio/index.html>

3. Создать приложение из шаблона Basic Activity
4. Запустить приложение на эмуляторе (api 22, 28) и телефоне

<https://developer.android.com/studio/run/emulator.html>

<https://developer.android.com/studio/debug/dev-options.html>

5. Создать аккаунт на gitlab.com
6. Создать аккаунт в Google Firebase
7. Установить git
8. Посмотреть видео материал про Git & Workflow - <https://clck.ru/CbZQ8>
9. Посмотреть видео материал про CI/CD - <https://clck.ru/CbZQv>

Задание 2 к занятиям в четверг 05.03.2020

Ресурсы предыдущего семинара:

<https://gitlab.com/kondratyonok/kondratyonok>

Обязательное домашнее задание:

1. Просмотреть лекции:

- Git & Workflow - <https://clck.ru/CbZQ8>

- CI/CD - <https://clck.ru/CbZQv>

- Изучить Раздел Полезные ссылки из презентации 2019_03-04_CI_CD

2. Gitlab:

2.1 Убедиться, что у репозитория Вашего приложения существует минимум 3 ветки (master, dev, feature), при этом

feature – ветка для новой функциональности

dev - ветка по умолчанию, защищена (не включая администратора)

master - ветка для релизов, защищена (включая администратора):

Защита:

- от удаления и от force-push-a

- требует создание pull request-а для мержа в неё

- проверяет статус сборки на CI прежде, чем разрешить мержинг

2.2 Посылать сборки, собранные в ветке master, в Firebase

2.3. Версия первой релизной версии приложения из ветки master - 1.0 (проставить тег)

3. Приложение:

3.1. Назвать своё приложение так, чтобы по его названию и по package_name мы могли без труда определить, кто его автор.

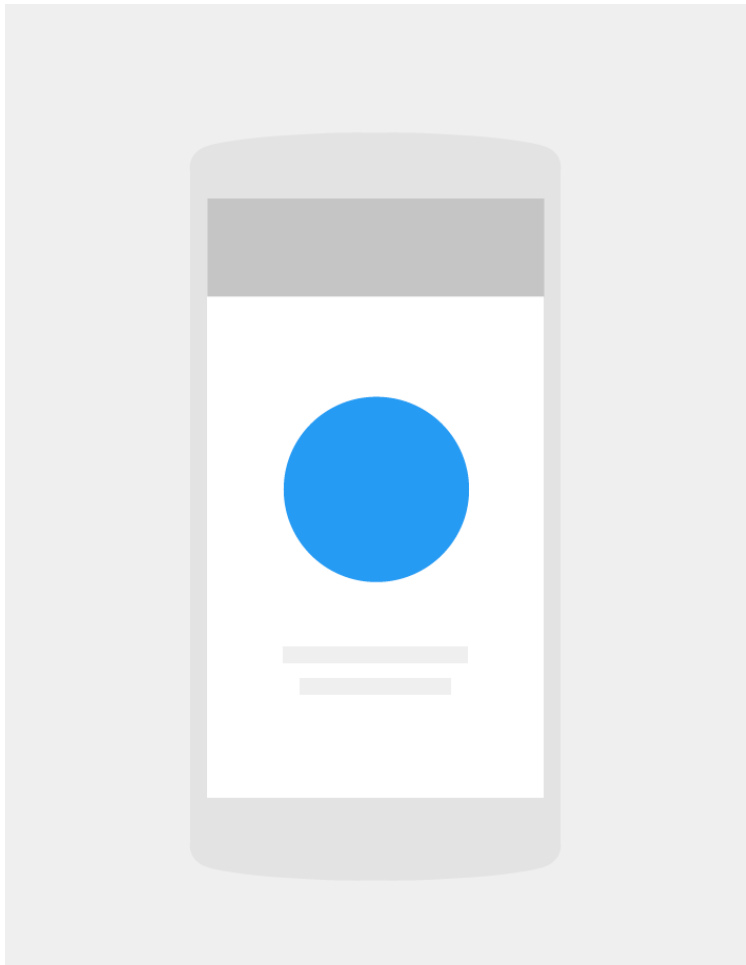
3.2. На главном экране Вашего приложения отобразить информацию о себе согласно макету, приложенному к заданию:

- Ваша фотография

- Имя Фамилия

- Account Name в gitlab или ссылка

При этом не прибегать к сторонним библиотекам. Можно использовать только библиотеки, представленные в <https://developer.android.com/topic/libraries/index.html>



Круглую фотографию можно сделать, сразу вставив готовую, или любым другим способом.

4. Firebase

Настроить оповещение себе и преподавателям курса ветки master, в которой все протестировано и работает корректно.

email преподавателей:

- ys.minsk.md@yandex.ru

При этом предыдущие пункты домашнего задания к этому моменту должны быть выполнены.

Дополнительное задание со звёздочкой:

1. Добавить статические анализаторы кода Checkstyle, PMD, FindBugs, Lint, которые будут исполняться при каждой сборке ветки master.
2. Оптимизировать сборку, настроив файл конфигурации, используя Job, dependencies, parallel, caching

Задание 3 к занятиям в пятницу 06.03.2020

1. Склонировать проект github.com/tewnn/Kotlin-Intro (<https://github.com/tewnn/Kotlin-Intro>)
2. Разобраться со следующими примерами:
 - Collections.kt
 - Dfs.kt
 - Expr.kt
 - Fold.kt
 - ForEach.kt
3. Поиграться с play.kotlinlang.org (<https://play.kotlinlang.org/>)
4. Посмотреть java bytecode для delegated property и понять в чём проблема с ней. См. как декомпилировать тут: <https://stackoverflow.com/questions/34957430/how-to-convert-a-kotlin-source-file-to-a-java-source-file>
5. Посмотреть Sequences.
6. Пройти Kotlin Koans (<https://kotlinlang.org/docs/tutorials/koans.html>) online или в Android Studio/Idea (самое важное).

Задание 4 к занятиям в четверг 12.03.2020

1) Создать стартовую Welcome Page для своего приложения, согласно макетам. Приложение всегда запускается с этой активити.

- Welcome Page представляет из себя 4 экрана с описанием приложения и его настройкой. Переход между экранами осуществляется по нажатию на кнопку "Далее". Последний экран Welcome Page должен перевести нас на LauncherActivity.
- Элементы, указанные на макетах, рассчитывались для устройства "Google Pixel": <https://market.yandex.ru/product/14283154/spec?track=tabs>, но отображаться должны одинаково на любом устройстве.
- Размеры элементов экрана Welcome Page:

отступы - 48 px

иконка приложения - 456 px

название приложения - 60 px

приветствие - 48 px

описание приложения - 48 px

выбор темы приложения - 72 px

выбор макета - 60 px

описание макета - 42 px

кнопка - Widget.AppCompat.Button.Colored стиль

- Элементы не стандартной формы (круглые, закругленные и проч) желательно делать с помощью XML shape drawable.

<https://developer.android.com/guide/topics/resources/drawable-resource?hl=ru#Shape>

Пример создания круглой картинки:

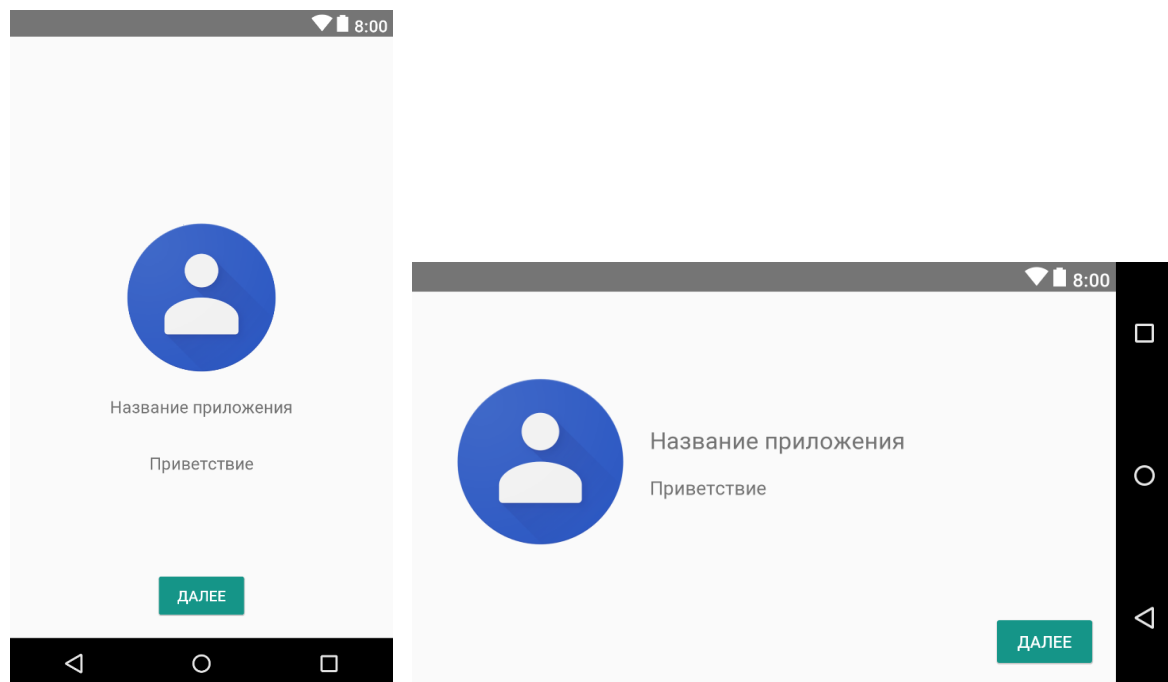
<https://android--code.blogspot.com/2016/03/android-circle-shape-in-xml-drawable.html>

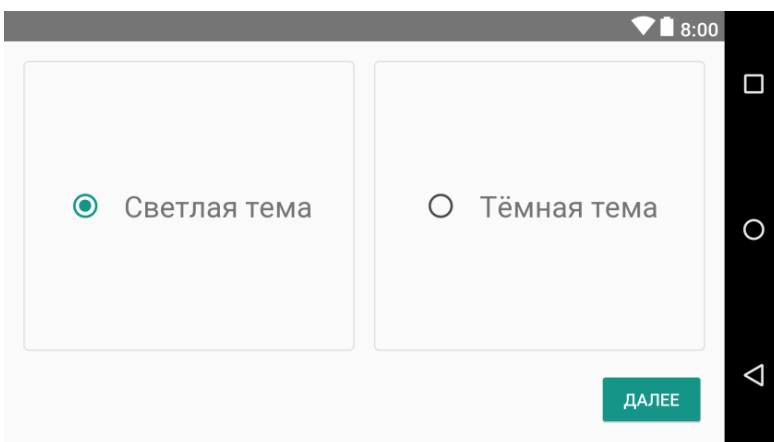
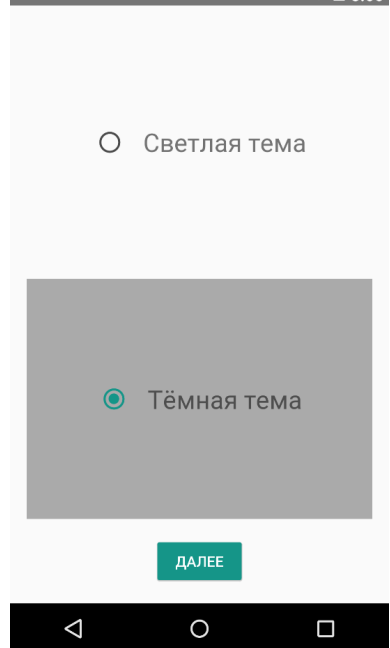
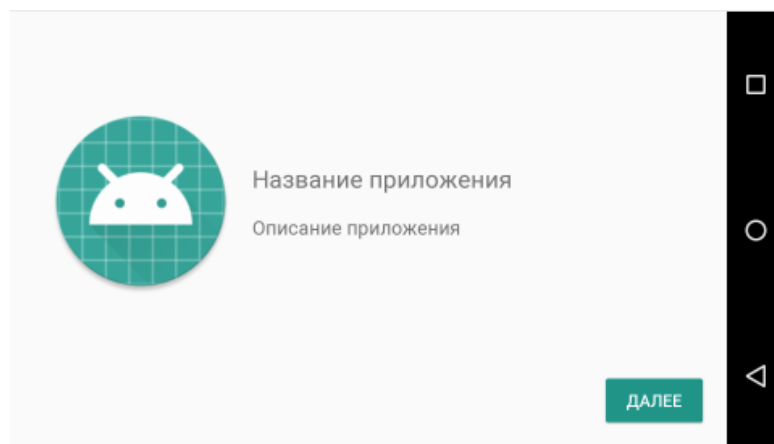
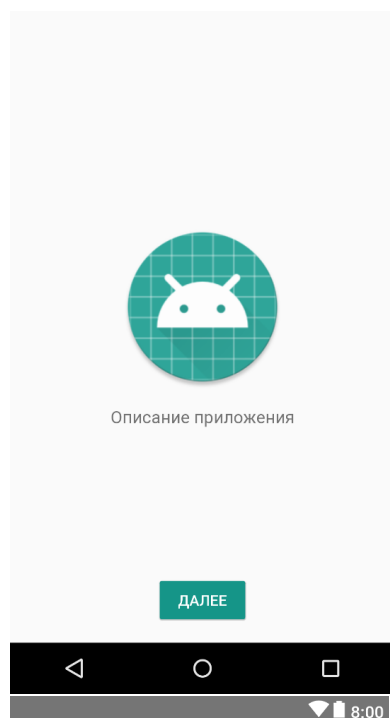
Пример работы с radio button/radio group:

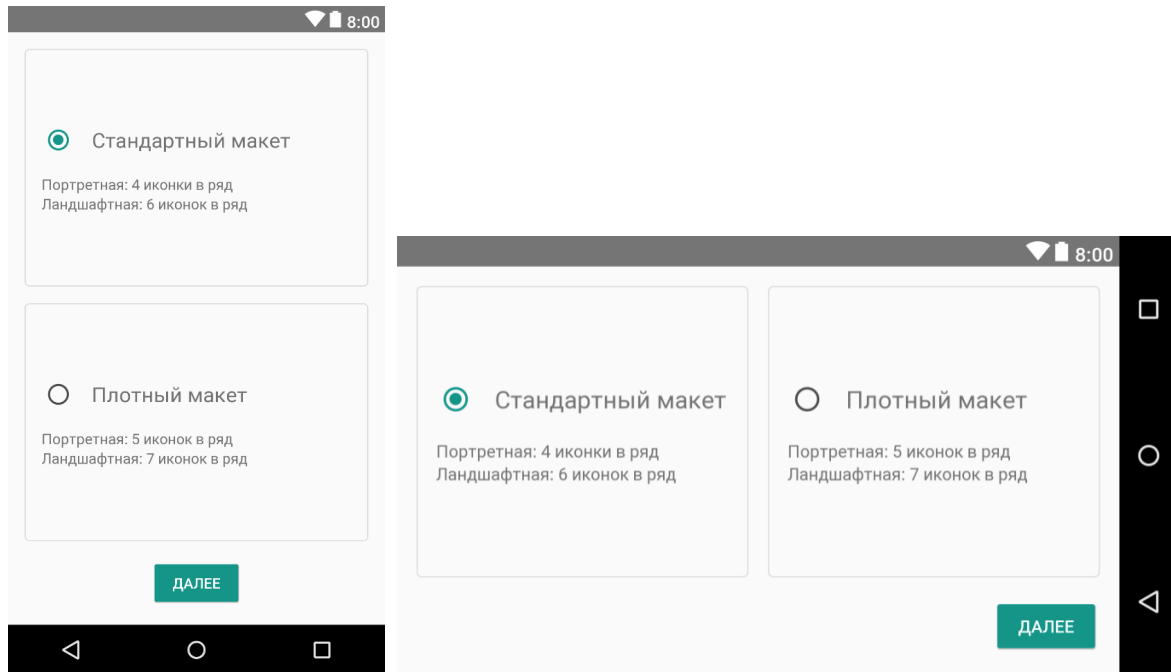
<https://android--code.blogspot.com/2015/08/android-radio-group-get-selected.html>

- Макеты представлены в портретном и ландшафтном режиме. Приложение должно уметь отображать Welcome Page в двух вариантах (использовать Fragment <https://developer.android.com/guide/components/fragments>).
- Приложение должно уметь учитывать пользовательские настройки для шрифта, определяемые в настройках телефона.
- элементов, которые не описаны на макетах - на экране Welcome Page быть не должно.
- Константы необходимо хранить в предназначенных для них ресурсах (strings/colors/arrays/dimens, ...)
- Старайтесь использовать стили в дублирующихся или похожих элементах

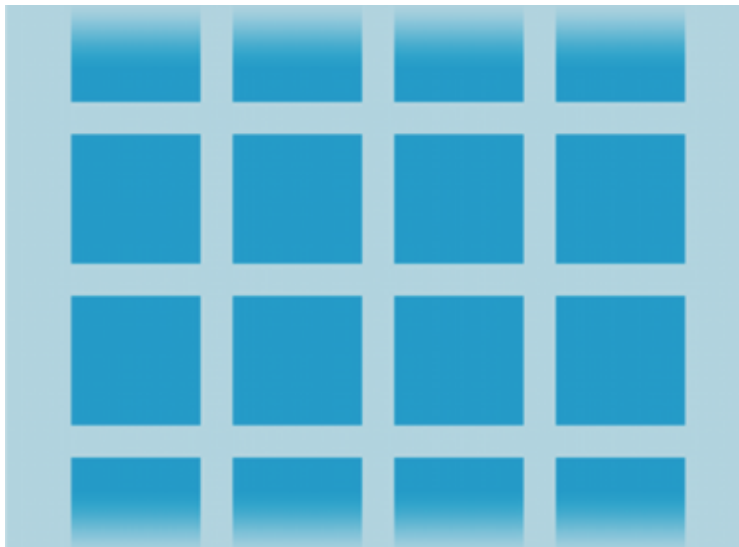
<https://developer.android.com/intl/ru/guide/topics/ui/themes.html>







2) Создать **LauncherActivity**, представляющую из себя экран с иконками, согласно макету.



- количество иконок в ряду определяется в Welcome Page. Достаточно реализовать 3 ряда.
- в портретном и ландшафтном режиме кол-во иконок в ряду различно.
- иконкой, на данном этапе, будем называть обычную **View**, представленную в виде квадрата рандомного цвета.
<https://developer.android.com/reference/android/view/View>
- при длительном нажатии на иконку - должен появиться Toast с описанием цвета иконки.
<https://developer.android.com/reference/android/view/View.OnLongClickListener>
<https://developer.android.com/guide/topics/ui/notifiers/toasts>

3) В качестве родительской activity для LauncherActivity, определяем активити, отображающую профиль студента из Задания 1.

Подробнее про родительские активити почитать тут:

<https://developer.android.com/training/appbar/up-action>

4) При длительном нажатии на иконку профиля студента - мы возвращаемся на первый экран Welcome Page.

5) Приложение должно быть полностью локализовано на 5 языков (качество перевода не имеет принципиального значения).

6) В качестве иконки приложения используйте иконку из вашего профиля. Подготовьте её для альтернативных ресурсов:

<https://developer.android.com/studio/write/image-asset-studio.html>

7) Обновить версию приложения до 2.0.

Полезные ссылки:

- Supporting Multiple Screens - https://developer.android.com/guide/practices/screens_support.html
- Единицы измерения - <http://developer.alexanderklimov.ru/android/theory/scales.php>
- Units & measurements - <https://material.io/guidelines/layout/units-measurements.html>
- Layouts - <https://developer.android.com/guide/topics/ui/declaring-layout.html>
<https://developer.android.com/guide/topics/ui/declaring-layout.html?hl=ru>
- Linear Layout - <https://developer.android.com/guide/topics/ui/layout/linear.html>
- Relative Layout - <https://developer.android.com/guide/topics/ui/layout/relative.html>
- ConstraintLayout - <https://developer.android.com/reference/android/support/constraint/ConstraintLayout.html>
- ConstraintLayout - <https://habrahabr.ru/company/touchinstinct/blog/326814/>
- Create App Icons with Image Asset Studio - <https://developer.android.com/studio/write/image-asset-studio.html>
- Styles and Themes - <https://developer.android.com/guide/topics/ui/look-and-feel/themes.html>
- Design App Themes - <https://developer.android.com/studio/write/theme-editor.html>
- XML shape drawable - <https://developer.android.com/guide/topics/resources/drawable-resource?hl=ru#Shape>
- Localizing with Resources - <https://developer.android.com/guide/topics/resources/localization.html>

Задание 5 к занятиям в четверг 19.03.2020

1) Реализовать launcher activity, содержащий множество иконок, с использованием RecyclerView и GridLayoutManager

- в качестве иконки выступает квадратная view, рандомного цвета.
- иконка имеет подпись цвета в формате #RRGGBB.
- offset = 8dp;

2) Реализовать list activity, содержащую список из множества элементов с помощью LinearLayoutManager, согласно макету "Three-line list with avatar":

https://storage.googleapis.com/material-design/publish/material_v_12/assets/0B6Okdz75tqQsSHRDUIlnV1kxSFE/components-lists-keylines-three8.png

* **subheader** (на картинке это текст "Today") **можно не добавлять**

- в качестве аватарки выступает круглая view, рандомного цвета.
- в качестве title используется подпись цвета в формате #RRGGBB.
- в качестве вторичного текста можно использовать заготовленный в string-array текст.

Полезные ссылки:

<https://material.io/guidelines/components/lists.html>

<https://developer.android.com/guide/topics/ui/layout/recyclerview.html>

<https://habr.com/en/post/237101/>

<https://developer.android.com/guide/topics/ui/layout/recyclerview>

3) Множество данных для пунктов 1 и 2 состоит из 1000 элементов.

4) Добавить на экраны для пунктов 1 и 2 Floating Action Button, по нажатию на который в начало RecyclerView добавляется новый элемент.

5) Реализовать удаление иконки/элемента списка, через реализацию action в **SnackBar**.

SnackBar, включающий action для подтверждения удаления иконки, появляется при длительном нажатии на иконку. Длительность отображения SnackBar, в случае бездействия - 5 секунд.

<https://developer.android.com/reference/android/support/design/widget/Snackbar.html>

<https://material.io/components/snackbars/>

<http://developer.alexanderklimov.ru/android/android-support-design.php>

6) Добавление/Удаление пунктов RecyclerView, а также события (Snackbar.Callback, OnClickListener для action) SnackBar логируем с помощью android.util.Log класса.

Логи должны быть доступны в релизных версиях приложений.

<https://developer.android.com/reference/android/util/Log.html>

7) Добавить Navigation Drawer, включающий хидер и навигационное меню:

- хидер содержит Avatar и Имя Фамилию

- навигационное меню содержит 3 пункта: launcher activity, list activity, настройки

<https://material.io/components/navigation-drawer/>

8) По нажатию на Avatar из Navigation Header реализовать переход на экран профиля студента.

* поведение из предыдущего Домашнего Задания, согласно которому экран профиля студента являлся парентом для launcher activity считать устаревшим.

9) По нажатию на пункт навигационного меню переходить на экраны launcher_activity/list_activity

По нажатию на пункт настройки - переходить на экран настроек

10) Реализовать экран с настройками приложения, в который должны быть добавлены настройки из Welcome Page.

<https://developer.android.com/guide/topics/ui/settings.html>

<https://material.io/design/platform-guidance/android-settings.html>

11) Варианты выбора темы оформления приложения.

Приложение имеет возможность выбора двух тем оформления.

Цветовая гамма выбирается со страницы

<https://www.google.com/design/spec/style/color.html#color-color-palette>

согласно номеру пользователя в Курсе "Разработка мобильных приложений" AnyTask. Например, если ваш номер 10, то необходимо использовать цветовую схему (primary color) Green для светлой и темой темы. Акценты (secondary color) выбираются произвольные.

<https://material.io/guidelines/style/color.html>

<https://material.io/color/#!/view.left=0&view.right=0>

<https://www.materialpalette.com/>

Задание 6 к занятиям в четверг 02.04.2020

1) Сдать зачет

2) Модифицировать исходный код приложения

Взломать приложение с помощью apktool (или других), модифицировать согласно требованиям ниже и пересобрать так, чтобы можно было поставить на устройство. Пересобранное приложение залить в репозиторий в папку **./modified**.

Задания:

1. Изменить название приложения и сделать сборку DEBUGGABLE.
2. Изменить иконку приложения.
3. Открыть любой из закрытых провайдеров (выставить exported в true).
4. Получить любые данные из взломанного провайдера (с помощью adb или другого приложения).
5. Записать любые данные во взломанный провайдер (с помощью adb или другого приложения).

Взять приложение можно из любого источника, например, со своего телефона или с сайта 4pda.ru.

3) Интегрировать внешнюю аналитику AppMetrica SDK

Сервис: <https://appmetrica.yandex.ru>

Документация: <https://tech.yandex.ru/appmetrica/doc/quick-start/concepts/quick-start-docpage/>

Задания:

1. Подключить к приложению AppMetrica и настроить отслеживание активных пользователей и пользовательских сессий. Убедиться, что AppMetrica отслеживает краши приложения.
2. Добавить отправку минимум 20 событий с разными именами, которые позволяют отслеживать использование различных функциональных возможностей приложения.

Задание 7 к занятиям в четверг 09.04.2020

Обязательное домашнее задание:

- 1) Реализовать сохранение факта показа Welcome-page, то есть Welcome-page должна показываться только при первом запуске или после очистки данных.
- 2) Добавляем в настройки опцию "Показать welcome-page при следующем запуске". Welcome-page должна показываться только один раз.
- 3) На отдельном экране реализуем вывод списка приложений, которые можно запускать: иконка + название. Используем заготовку списка из предыдущего задания. Просто дублируем существующий экран и делаем на его основе. Стиль элементов не меняем.

<https://stackoverflow.com/questions/2695746/how-to-get-a-list-of-installed-android-applications-and-pick-one-to-run>

<https://developer.android.com/training/basics/intents?hl=en>

- 4) Реализуем возможность запуска приложения при тапе на иконку.
- 5) Реализуем слежение за изменением списка приложений в системе. При установке или удалении приложения с устройства должен обновляться список приложений.

<https://stackoverflow.com/questions/11246326/how-to-make-my-app-receive-broadcast-when-other-applications-are-installed-or-re>

- 6) Добавляем контекстное меню при долгом нажатии на приложение в списке:

<https://developer.android.com/guide/topics/ui/menus.html?hl=ru#context-menu>.

- 7) Добавляем в контекстное меню приложения возможность удалить приложение.
- 8) Добавляем возможность запоминать кол-во запусков каждого приложения по иконке. Рекомендуется использовать базу данных для этих целей. Если вы используете альтернативное хранилище, то вам нужно быть готовым обосновать его использование вместо базы данных. Необходимо предусмотреть возможность отображения кол-ва запусков любого приложения, например в контекстном меню.

- 9) Добавляем сортировку приложений в списке по дате установки и в алфавитном порядке. В настройках можно выбрать: сортировка по дате установки/сортировка в алфавитном порядке/отсутствие сортировки. Для сортировки в алфавитном порядке дополнительно можно выбрать: сортировать A-Z или Z-A.

- 10) Добавляем сортировку приложений в списке по кол-ву запусков на основании сведений из БД. По аналогии с предыдущим пунктом, определяется в настройках.

Дополнительное задание:

- 11) Добавляем в начало списка приложений визуально выделенную область, в которую помещается несколько приложений (популярные приложения). Выделенная область должна прокручиваться вместе со всем списком приложений. Приложение, попавшее в список популярных должно встречаться дважды: первый раз в выделенной области популярных

приложений, второй раз - в основном списке приложений на основании выбранного вида сортировки. В эту область можно выводить либо последние запущенные приложения, либо приложения, которые пользователь чаще всего запускал за последний час/6 часов/12 часов/сутки/неделю и т.д. (задается из настроек).

12) Реализовать возможность управления размером области популярных приложений (из предыдущего пункта задания) из настроек.

13) Добавить область "Избранное" в нижнюю часть списка приложений. Область "Избранное" должна визуально отделяться от списка приложений. Она не должна прокручиваться вместе со списком приложений. Представляет собой одну строку таблицы с ячейками, похожими на те, что используются в списке приложений. Изначально, избранная область заполнена пустыми ячейками. Окончательное решение по стилю и оформлению области "избранное" и отдельных ее ячеек, включая их кол-во, остается за вами. Необходимо реализовать возможность добавления любого приложения в любую ячейку избранной области и удаления приложения из нее.

14) Реализовать возможность добавления в область "Избранное" (из предыдущего пункта задания) не только приложения, но и контакты. Основное действие при нажатии на контакт - вызов абонента. Вместо иконки приложения использовать фотографию звонящего (можете обыграть на свое усмотрение ситуацию, когда фотографии нет или, если у вас не получится ее получить). Также в контекстное меню при долгом нажатии на контакт можно добавить переход на карточку контакта в приложении "Контакты".

15) Реализуем другие улучшения отображения области "Избранное" и ее элементов на собственное усмотрение. Также расширяем возможности по настройке этой области и ее отдельных элементов (это может быть как визуальное оформление, так и наполнение).