

Зачетное задание 3

Формальные требования к зачету:

1. Задание должно быть сдано в виде коммита в репозиторий с исходными кодами согласно **дате в anytask**. Тем, кто сдал работу во время, добавляется 1 балл.
2. В качестве решения в anytask прикладывается ссылка на репозиторий.
3. Репозиторий должен содержать исходные коды.
4. Коммит для зачета должен быть помечен тегом **{номер задания}.{номер итерации}**, например, 3.0.
5. Приложение должно уметь собираться из консоли командой **./gradlew clean assembleRelease** из папки проекта.
6. Приложение должно быть опубликовано в Firebase с версией, которая соответствует гит тегу, и доступно для скачивания под аккаунтом *ys.minsk.md@yandex.ru*.
7. Набрать минимально необходимое количество баллов по каждому требованию или паразитировать приемную комиссию качеством исполнения.

Баллы:

За полную реализацию каждого пункта Требований дается 1 балл (если не указано иное), в противном случае дается 0 баллов.

Дизайн:

Если в Требованиях не указан дизайн элементов или представлено неполное описание функциональности, то ожидается соответствие тому, что описывалось в домашних заданиях.

Требования

1) Модифицировать исходный код приложения (максимум 5 баллов, минимум 2)

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

2) Реализовать безопасный ContentProvider (максимум 7 баллов, минимум 5):

Обязательно к зачету:

1. Добавить возможность расшарить историю использования приложений (с числом кликов больше 0), сделав отдельный ContentProvider с возможностью получать

информацию о последнем запущенном приложении и список всех используемых (имя приложения, его пакет и количество запусков).

2. Сделать отдельное приложение, которое будет обращаться к ContentProvider-у лончера. Приложение должно иметь возможность: запросить последнее запущенное приложение, запросить все хоть раз запущенные приложения. Данные должны выводиться текстом в простом формате. UI приложения стоит сделать предельно простым, но понятным.
3. Защитить доступ к ContentProvider-у для всех приложений. Предоставить доступ только приложению из п.3.
4. Защитить доступ к получению всего списка записей из ContentProvider-а.
5. Сделать несколько версий тестового приложения из п.3, давая каждому из них различные права доступа. Если приложение не сможет получить доступ к каким-то данным, то должно выводиться соответствующее сообщение.

Дополнительно:

6. Добавить возможность изменять через ContentProvider число запусков любого из приложений с отдельной защитой для операции изменения записи. Интегрировать в приложения из п5.
7. Добавить возможность получить всю доступную информацию из Профиля пользователя с отдельной защитой доступа. Интегрировать в приложения из п5.

3) AppMetrica SDK (максимум 2 баллов, минимум 2)

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

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

Обязательно для зачета:

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

4) Работа с приложениями (максимум 14 баллов, минимум 5).

Изменяем поведение приложения, заменив случайные значения в таблицах Списка приложений на реальные значения.

Обязательно для зачета:

1. Каждая ячейка служит для отображения эскиза приложения. Эскиз — это настоящая иконка приложения и подпись к ней в виде названия приложения.
2. По тапу на иконку (или всю ячейку) запускается приложение.
3. Реализуем слежение за изменением списка приложений в системе. При установке или удалении приложения с устройства должен обновляться список приложений.

4. При долгом нажатии на иконку (ячейку) появляется контекстное меню с пунктами "Удалить", "Частота", "Информация". При нажатии на меню "Удалить" происходит удаление приложения из системы. При нажатии на меню "Информация" открываются системные настройки приложения. В меню "Частота" отображается количество запусков приложения.
5. Добавить в Настройках возможность сортировать список приложений (по дате установки, в алфавитном порядке АЯ и ЯА, по частоте использования, без сортировки)

Дополнительно (2 балла за каждый пункт):

6. Реализовать функциональность Популярные приложения (см Задание 7 п. 11).
7. Реализовать функциональность "Избранное" с приложениями (см Задание 7 п. 13).
8. Реализовать функциональность "Избранное" с контактами (см Задание 7 п. 14).
9. Реализовать улучшения области "Избранное" (см Задание 7 п. 15).
10. В разделе Настройки добавить возможность скрывать разделы "Популярные" и "Избранное" - 1 балл.

5) Настройки (максимум 1 балл, минимум 0) - дополнительное задание

Дополнительно:

1. Добавить в настройки опцию "Показать Welcome-Page при следующем запуске". Welcome-page должна показаться только один раз.

6) Профиль студента (максимум 1 баллов, минимум 0) - дополнительное задание

Дополнительно:

1. Реализовать возможность редактирования данных профиля и сохранения измененных данных.