

## Тестовое задание

Тестовое задание состоит в написании приложения для вывода прогноза погоды по г.

Новосибирск с использованием открытых данных сервера gismeteo.ru.

На выполнение тестового задания дается одна неделя (от даты отправки).

При выполнении задания разрешается (и рекомендуется) использовать любые доступные источники: книги, форумы, документацию и пр. Естественно задание необходимо выполнить самостоятельно (будет «защита» выполненного задания).

В результате выполнения задания необходимо получить:

Код приложения (папка проекта)

Подписанный арк-файл, для установки на устройство.

### Важно:

1. Написанное приложение должно быть снабжено инструкциями по сборке и развертыванию

арк файла на физическом устройстве и эмуляторе

2. Крайне желательно при разработки использовать gradle

Проверка задания включает в себя:

1. Установку приложения на устройство и проверку сценария выполнения (см. ниже)

2. Code review.

Тестовое задание необходимо реализовать на эмуляторе или на реальном устройстве начиная с версии android 4.0.x

### Ресурсы

1. XML-информер с сервера Гис метео (описание формата данных):

Данные по Новосибирску: [http://informer.gismeteo.ru/xml/29634\\_1.xml](http://informer.gismeteo.ru/xml/29634_1.xml)

По другим городам: <http://servisvk.ru/status/pogoda/russia.php>

2. <http://developer.android.com> - описание API и платформы.

3. <http://developer.android.com/sdk/installing.html> - описание процесса установки и развертывания среды разработки.

### Описание приложения

Приложение предназначено для получения прогноза погоды по г. Новосибирску с сервера Гисметео.

## Сценарий использования

1. Пользователь запускает приложение. Он видит пустой экран с надписью «Нет данных» и кнопку «Обновить»
2. Пользователь нажимает кнопку «Обновить». Приложение загружает данные с сервера и списком отображает 4 текущих прогноза
3. Пользователь кликает по прогнозу в списке. Открывается новый экран с подробной информацией по выбранному прогнозу
4. При помощи системной кнопки «Назад» пользователь возвращается к экрану со списком прогнозов.
5. Пользователь кликает по другому прогнозу в списке. Открывается новый экран с подробной информацией по выбранному прогнозу. И т. д.
6. Находясь на экране со списком, пользователь может нажать кнопку «Обновить» повторно. При этом текущий список очищается и с сервера загружаются новые данные.

## Исключительные ситуации

1. Приложение должно корректно обрабатывать ситуацию, когда сервер недоступен.
2. Приложение должно корректно обрабатывать поворот экрана

## Замечания

1. Дизайн приложения желательно выполнить в соответствии material design guidelines <https://www.google.com/design/spec/material-design/introduction.html>
2. Персистентность и возможность настройки не нужны. Допускается указать адрес сервера (ссылку на XML-данные) «хардкодом».

3. При загрузке данных (пункт 2 и 6 сценария использования) приложение не должно зависать. Дополнительная индикация того, что данные загружаются, не требуется и остается на Ваше усмотрение.

## Подсказки.

### Основные подсказки для начинающих разработчиков

1. Для экранов 1, 2 используйте одну и ту же Activity для фрагментов.
2. Экран на 3 является отдельной Activity.
3. Для загрузки данных в фоне удобно использовать класс, наследуемый от AsyncTask<Parameter, Progress, Result>.

### Для продвинутых:

Было бы круто использовать injection dependency, reactive java, retrofit и архитектурные ухищрения

Желаем Вам удачи.