

Лабораторная – Класс Activity

Изучение класса Activity

Цели:

Познакомиться с классом Activity, жизненным циклом класса Activity и процессом реконфигурации в операционной системе Android. Создайте и произведите мониторинг простого приложения: наблюдайте за несколькими объектами Activity, по мере того, как они проходят различные стадии своего жизненного цикла

По завершении этой Лабораторной вы изучите: класс Activity, жизненный цикл Activity, как запускать Activity программно и как управлять реконфигурацией Activity.

Часть 1: Класс Activity

Эта часть состоит из двух упражнений. Цель этих упражнений – познакомиться с жизненным циклом Activity в Android и лучше понимать как Android управляет реконфигурацией и как это связано с жизненным циклом Activity.

Упражнение А:

Приложение, которое вы будете использовать называется ActivityLab. Будучи запущенным, оно отображает пользовательский интерфейс как показано ниже. Вам предоставлены все ресурсы, включая макеты приложения. Не изменяйте, не модифицируйте их.



Это приложение включает два класса Activity. Первый Activity, названный “ActivityOne”, выводит сообщения в журнал(Log), используя метод Log.i(), каждый раз, когда вызываются методы жизненного цикла Activity: onCreate(), onRestart(), onStart(), onResume(), onPause(), onStop() и onDestroy(). Этот Activity должен также мониторить и показывать информацию о

вызовах методов жизненного цикла этого Activity: onCreate(), onStart() и onResume(). Для этого Activity содержит по одному счетчику для каждого из этих методов. Эти счетчики подсчитывают число раз, которые каждый из этих методов были вызваны с тех пор как ActivityOne был запущен в последний раз. Имена методов и их текущее количество вызовов должны всегда отображаться вне зависимости от видимости пользовательского интерфейса ActivityOne.

Замечание:

Не объявляйте переменные счетчиков статическими, т.к. в следующем упражнении мы будем практиковаться в сохранении состояния между реконфигурациями.

Когда пользователь нажимает на кнопку “Start ActivityTwo,” ActivityOne отвечает активацией второго Activity, названного “ActivityTwo.” По мере того как пользователь переходит между ActivityOne и ActivityTwo, вызываются различные методы жизненного цикла и все ассоциированные счетчики увеличиваются. ActivityTwo содержит кнопку с подписью “Close Activity” для закрытия этого экрана (пользователь также может нажать кнопку «Назад» в Android для перехода между Activity). Напоминаем еще раз, что вам уже даны все файлы макетов, поэтому вам не требуется их реализовывать и вы не должны их модифицировать. Как и ActivityOne, ActivityTwo будет отслеживать четыре специфичных вызова методов жизненного цикла Activity и подсчитывать количество вызовов. Он также выводит в журнал сообщение каждый раз, когда ActivityTwo вызывает какой-либо метод жизненного цикла.



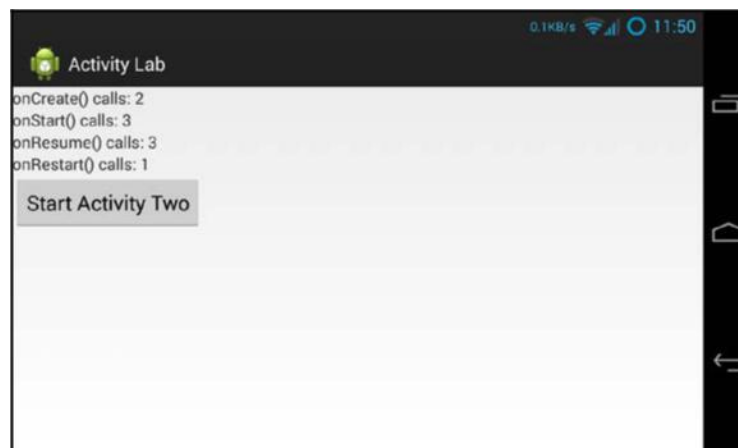
Убедитесь, что пользовательский интерфейс ActivityOne отображает корректное число вызовов метода по мере того как пользователь переходит из ActivityTwo обратно в ActivityOne.



См. видео ActivityNoReconfig.mp4, которое входит в пакет Лабораторной.

Упражнение В:

Когда пользователь меняет ориентацию своего устройства Android, изменяя, скажем из Портретного режима в Пейзажный режим или наоборот, Android обычно убивает текущее Activity и затем делает его рестарт. Вы можете переориентировать ваше устройство в эмуляторе нажатием Ctrl+F12 (Command+F12 на Mac). Когда это происходит ваше текущее Activity убивается и рестартует, при этом вызываются методы жизненного цикла Activity.



В этом упражнении вы будете модифицировать ваше приложение из Exercise A таким образом, чтобы показания счетчиков сохранялись даже после того как Activity было убито и воссоздано из-за реконфигурации. Обычно же, если Activity убивается (например, путем нажатия кнопки "Close Activity Two" или нажатием кнопки Back), а затем делается его рестарт, счетчики также рестартуют с нуля.

Чтобы это сделать вы будете сохранять получать и обнулять различные счетчики по мере реконфигурации приложения. В частности вы будете сохранять счетчики в Bundle, когда Activity закрывается, и вы будете получать и восстанавливать счетчики из Bundle, когда Activity воссоздается.

См. "Recreating an Activity" по адресу <http://developer.android.com/training/basics/activity-lifecycle> для более подробной информации о сохранении и получении данных из Bundle.

См. Видео ActivityReconfig.mp4, которое включено в Лабораторную.

Заметки по реализации:

1. **Упражнение для разогрева:** Перед реализацией Упражнений сделайте следующее упражнение в качестве разогрева. Создайте текстовый файл Activity.txt и запишите в него свои ответы на вопросы, представленные ниже.
 - а) Откройте файл ActivityLifecycleWalkthrough.pdf. Эта диаграмма отображает две машины состояний, представляющих ActivityOne и ActivityTwo. Если вы хотите, вы можете вырезать маленькие кружочки и использовать их как маркеры по мере прохождения этого упражнения.
 - б) Предположим пользователь запустил приложение, что приводит к открытию ActivityOne. Затем пользователь нажимает кнопку для запуска ActivityTwo, и ActivityTwo затем появляется на экране.

- a. Перечислите методы жизненного цикла Activity, которые были вызваны в ActivityOne и в ActivityTwo, с момента запуска приложения в том порядке, в котором они происходили.
 - c) Затем, предположим пользователь возвращается обратно к ActivityOne, нажав на кнопку “Close Activity” ActivityTwo. ActivityTwo закрывается и затем ActivityOne появляется вновь. Начиная с того момента, где вы закончили предыдущий шаг:
 - a. Перечислите методы жизненного цикла Activity, которые были вызваны в ActivityOne и в ActivityTwo в том порядке, в котором они происходили.
 - d) Затем, пользователь снова нажимает на кнопку для запуска ActivityTwo. Как только ActivityTwo появляется, пользователь нажимает на кнопку Home устройства. Начиная с момента, где вы закончили предыдущий шаг:
 - a. Перечислите методы жизненного цикла Activity, которые были вызваны в ActivityOne и в ActivityTwo в том порядке, в котором они происходили.
 - e) Затем, пользователь стартует приложение снова, нажав на его иконку в списке приложений. С момента перезапуска приложения и начиная с того момента, где вы закончили предыдущий шаг:
 - a. Перечислите методы жизненного цикла Activity, которые были вызваны в ActivityOne и в ActivityTwo в том порядке, в котором они происходили.
2. **Exercise A:** Скачайте скелет проекта приложения и проимпортируйте его в вашу IDE.
- a) Прodelайте шаги от *a* до *c*, приведенные ниже для обоих классов ActivityOne (в ActivityOne.java), и для ActivityTwo (в ActivityTwo.java). Реализуйте шаг *d* для ActivityOne и шаг *e* для ActivityTwo.
 - a. Создайте не статические переменные счетчиков, каждый из которых соответствует различному методу жизненного цикла: onCreate(), onStart(), onResume(). Инкрементируйте эти переменные, когда вызываются соответствующие методы жизненного цикла.
 - b. Создайте четыре переменных TextView, каждый из которых будет отображать значение различных переменных счетчиков. Если вы откроете файл *layout.xml* в директории *res/layout* и исследуете элемент *<textview>* вы увидите его *id*. Переменные TextView должны быть доступны из любых методов, т.к. они должны быть изначально присвоены в методе onCreate().
 - c. Перекройте четыре вызова методов жизненного цикла, которые вы будете мониторить. В каждом из этих методов обновите соответствующий счетчик и вызовите метод *displayCounts()* для обновления пользовательского интерфейса.

- d. Реализуйте OnClickListener для launchActivityTwoButton. (только для ActivityOne.java)

```
launchActivityTwoButton.setOnClickListener(new OnClickListener() { public
    void onClick(View v) {
        // Эта функция запускает ActivityTwo
        // Используйте: в Context метод startActivity()
        ...
    }
})
```

- e. Реализуйте OnClickListener для closeButton. (только для ActivityTwo.java)

```
closeButton.setOnClickListener(new OnClickListener() { public
    void onClick(View v) {
        // Эта функция закрывает ActivityTwo
        // Подсказка: используйте в Context метод finish()
        ...
    }
})
```

3. **Упражнение В:** реализуйте следующие расширения к той работе, которую вы проделали в Упражнении А. См. “Recreating an Activity” по адресу <http://developer.android.com/training/basics/activity-lifecycle> для подробной информации по сохранению и получению данных с помощью Bundle.

- a. Напишите исходный код, необходимый для сохранения значений счетчиков методов жизненного цикла. Когда Activity убивается, но может быть перезапущено Android вызывает метод onSaveInstanceState(). Это дает Activity шанс сохранить любые данные объекта, которые его могут понадобиться, если Activity в будущем восстановится. Заметим, что если Android не ожидает, что Activity будет перезапущено, тогда этот метод не будет вызван. Например, метод не будет вызван когда пользователь нажимает кнопку закрытия Activity в ActivityTwo. См: <http://developer.android.com/reference/android/app/Activity.html>, особенно метод onSaveInstanceState(android.os.Bundle) для более полной информации.

```
// Сохранение данных в Bundle (коллекция пар ключ-значение).
public void onSaveInstanceState(Bundle savedInstanceState) {
    ...
}
```

- b. Реализуйте исходный код, необходимый для восстановления значений счетчиков методов жизненного цикла. Есть несколько способов это сделать. В этой лабораторной, реализуйте логику в методе onCreate().

```
protected void onCreate (Bundle savedInstanceState)
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_one);

    // Было ли сохранено предыдущее состояние?
    if (savedInstanceState != null){
```

```
        // Восстановите значения счетчиков из сохраненного состояния
    }
}
```

Другой способ, каким это можно было бы сделать (но не в этой Лабораторной) это определить метод `onRestoreInstanceState()`. Убедитесь, что вы понимаете когда и почему вызывается этот метод. См: <http://developer.android.com/reference/android/app/Activity.html> для подробной информации.

```
protected void onRestoreInstanceState (Bundle savedInstanceState) {
    // Восстанавливаем значения счетчиков из сохраненного состояния
}
```

Тестирование и Отчет

Тестирование для этой Лабораторной будет включать несколько ручных шагов. Мы запускали тесты на эмуляторе, используя Galaxy Nexus AVD с API level 18. Чтобы ограничить проблемы конфигурации, вам следует тестировать ваше приложение на подобном AVD. В добавок, когда тестируете, помните, что нужно начинать тесты на вашем устройстве в портретном режиме и с разблокированным экраном. Также, если вы установили ваши Настройки разработчика убивать Activity, когда они уходят в фон, тогда эти тест-кейсы будут провалены.

Предостережение: Документированы некоторые проблемы с эмулятором и Robotium, о которых вы должны быть осведомлены. Первая проблема в том, что некоторые эмуляторы не поворачиваются. Похоже, что это, прежде всего эффект API уровней 19 и 20. См: <https://code.google.com/p/android/issues/detail?id=61671>. В добавок, мы идентифицировали случаи, в которых некоторые методы жизненного цикла некорректно не вызывались при Robotium тестировании.

Когда вы будете готовы отправить вашу работу на проверку, вы вручную выполните каждый из тестовых сценариев и подготовить текстовый файл отчета о результатах выполнения. В начале каждого тестового сценария вы должны стартовать приложение с домашнего экрана в портретной ориентации. Вы затем выполните набор специфических операций. После последнего шага вы запишите показания счетчиков, отображенных на экране. Вы затем завершаете тест, убивая приложение (не помещая его в фон нажатием кнопки Home). В итоге вы затем подготовите **четырехстрочный** текстовый файл (с расширением **.txt**), содержащий счетчики, отображенные в вашем тестовом сценарии и отправьте этот файл в качестве отчета. В пакете с Лабораторной содержатся шаблоны текстовых файлов, которые вам нужно отправить. Вам лишь нужно заполнить пропущенные значения счетчиков.

Счетчики должны быть в точности в следующем формате:

```
onCreate() calls: A
onStart() calls: B
onResume() calls: C
onRestart() calls: D
```

Замените A , B , C и D числами, отображающимися на вашем устройстве. Убедитесь, что записываете в правильном порядке и что текстовый файл содержит только четыре строки.

Тестовые сценарии, которые нужно выполнить:

Test1: StartActivityOneTest

1. Запустите ActivityLab
2. Запишите показания счетчиков жизненного цикла.

Test2 : DoubleRotateActivityOneTest

1. Запустите ActivityLab
2. Поверните экран в пейзажный режим
3. Поверните экран в портретный режим
4. Запишите показания счетчиков жизненного цикла.

Test3 : StartActivityTwoTest

1. Запустите ActivityLab
2. Нажмите на кнопку “Start Activity Two”
3. Запишите показания счетчиков жизненного цикла.

Test4 : DoubleRotateActivityTwoTest

1. Запустите ActivityLab
2. Нажмите на кнопку “Start Activity Two”
3. Поверните экран в пейзажный режим
4. Поверните экран в портретный режим
5. Запишите показания счетчиков жизненного цикла.

Test5 : CloseActivityTwoTest

1. Запустите ActivityLab
2. Нажмите на кнопку “Start Activity Two”
3. Нажмите на кнопку “Close Activity”
4. Запишите показания счетчиков жизненного цикла.

Test6 : ReopenActivityTwoTest

1. Запустите ActivityLab
2. Нажмите на кнопку “Start Activity Two”
3. Нажмите на кнопку “Close Activity”
4. Нажмите на кнопку “Start Activity Two” еще раз
5. Запишите показания счетчиков жизненного цикла.