Санкт-Петербургский политехнический университет Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе №2**

**Дисциплина**: Проектирование мобильных приложений

**Тема**: Activity Lifecycle. Alternative resources.

Выполнил студент гр. 3530901/70203 И.Д. Иванов

(подпись)

Преподаватель И.С. Егорова

(подпись)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.

Санкт-Петербург

2019

1. Цели  
1. Ознакомиться с жизненным циклом Activity.

2. Изучить основные возможности и свойства Alternative resources.

Вариант 6.

Задание для задачи № 2: Screen aspect.

2. Программа работы

2.1. Задача 1. Activity

Продемонстрируйте жизненный цикл Activity на любом нетривиальном примере.

Пример: Выключение телефона

При открытии приложения происходят вызовы методов onCreate(), onStart() и onResume(). Когда пользователь выключает телефон вызываются методы onPause() и onStop(), после чего телефон выключается.

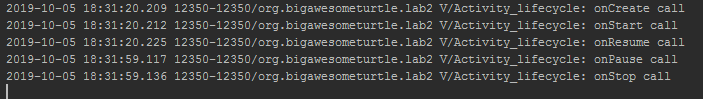


Рис.1. Демонстрация жизненного цикла Activity на примере выключения телефона

Пример: вызов Google Assistant во время нахождения в приложении

При открытии приложения происходят вызовы методов onCreate(), onStart() и onResume(). Когда пользователь вызывает Google Assistant, происходит вызов метода onPause(), но не происходит вызова метода onStop() до того момента, пока пользователь не закончит запрос:

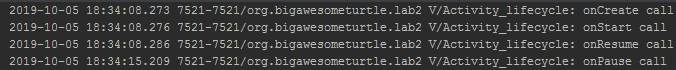


Рис.2. Вызовы методов Activity с момента открытия приложения до момента запроса к Google Assistant

Когда пользователь отправляет запрос, происходит вызов метода onStop():

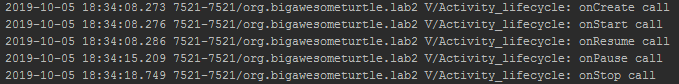


Рис.3. Запрос окончен

При возврате в приложения вызываются методы onRestart(), onStart() и onResume():

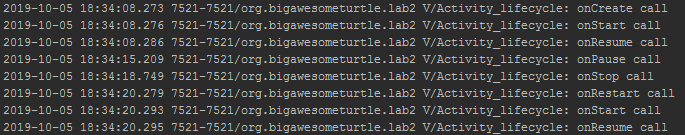


Рис.4. Возврат в приложение

2.2. Задача 2. Alternative Resources

Продемонстрируйте работу альтернативного ресурса (тип ресурса согласно варианту) на каком-либо примере.

Варинт 6: Screen aspect.

Для демонстрации работы альтернативного ресурса были созданы два layout-ресурса для значений квалификаторов long/notlong.

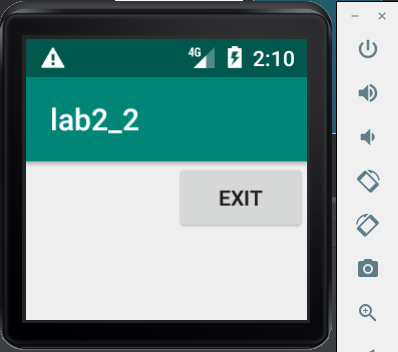
 

Рис.5. Запущенное приложение на устройствах с конфигурацией screen\_aspect: long (слева) и screen\_aspect: notlong (справа).

Layout-ресурсы расположены в папках layout-long и layout-notlong. При запуске приложения выбирается файл, соответствующий конфигурации устройства.

Удобство данного альтернативного ресурса заключается в том, что компоненты можно расположить более комфортно для пользователя. Например, для экрана типа long удобно расположить кнопку снизу, чтобы не «тянуться» в верхнюю часть экрана для совершения какого-либо действия, а для экрана типа notlong можно расположить кнопку в верхней части экрана, т.к. это не доставит неудобств пользователю.

2.3. Задача 3. Best-matching resource

Для заданного набора альтернативных ресурсов, предоставляемых приложением, и заданной конфигурации устройства (оба параметра согласно варианту) объясните, какой ресурс будет выбран в конечном итоге. Ответ докажите.

Вариант 6:

Конфигурация устройства:

LOCALE: fr-rFR

SCREEN\_SIZE: normal

SCREEN\_ASPECT: notlong

ROUND\_SCREEN: notround

ORIENTATION: land

UI\_MODE: desk

NIGHT\_MODE: notnight

PIXEL\_DENSITY: mdpi

TOUCH: finger

PRIMARY\_INPUT: 12key

NAV\_KEYS: wheel

PLATFORM\_VER: v25

Конфигурация ресурсов:

(default)

watch-ldpi-finger

land-notouch-qwerty

normal-round-notnight-v25

land-xxxhdpi-v25

fr-rCA-night-v27

port

en-rUS-finger

long-port-xxxhdpi-notouch

desk-finger-qwerty-dpad

long-land-car-tvdpi

Следуя алгоритму для best-matching resource:

1) У устройства Locale: **fr-rFR**, значит из ресурсов исключаются: **fr-rCA**-night-v27и **en-rUS**-finger.

2) У устройства Screen\_size: **normal**. По документации: использование квалификатора размера не подразумевает, что ресурсы предназначены только для экранов этого размера. Система не будет использовать ресурсы, которые использует квалификатор размера, превышающего текущий экран. На данном этапе нет противоречащих ресурсов.

3) У устройства Screen\_aspect: **notlong**, значит из ресурсов исключаются: **long**-port-xxxhdpi-notouch и **long**-land-car-tvdpi.

4) У устройства Round\_screen: **notround**, значит из ресурсов исключаетсяnormal-**round**-notnight-v25.

5) У устройства Orientation: **land**, значит из ресурсов исключается **port**.

6) У устройства UI\_mode: **desk**, значит из ресурсов исключается **watch**-ldpi-finger

7) У устройства Night\_mode: **notnight**, противоречащих ресурсов нет.

8) Pixel\_density: mdpi. По документации: Использование квалификатора плотности не подразумевает, что ресурсы предназначены только для экранов этой плотности. Если не предусмотрены альтернативные ресурсы с квалификаторами, лучше подходящими к текущей конфигурации устройства, система может использовать любые наиболее подходящие ресурсы. Значит, на этом этапе ресурсы не исключаются.

9) У устройства Touch: **finger**, значит из ресурсов исключаетсяland-**notouch**-qwerty.

10) У устройства Primary\_input: **12key**, значит из ресурсов исключается

desk-finger-**qwerty**-dpad.

11) У устройства Nav\_keys: **wheel**,противоречащих ресурсов нет.

12)У устройства Platform\_ver: **v25**,противоречащих ресурсов нет.

Остался один ресурс, который и будет выбран: **land-xxxhdpi-v25**.

2.4. Задача 4. Сохранение состояние Activity

Студент написал приложение continuewatch. Это приложение по заданию должно считать, сколько секунд пользователь провел в этом приложении. Найдите ошибки в этом приложении и исправьте их.

Найденные ошибки: приложение не сохраняет значение секундомера после закрытия приложения и поворота экрана (при каждом вызове метода onCreate() значение секундомера обнуляется). Также, приложение продолжает считать, когда оно не активно (после вызова методов onPause() и onStop()).

Решение (см. приложение 1): чтобы значение секундомера сохранялось при выходе из приложения (а также при повороте экрана) добавлен экземпляр класса SharedPreferences. В методе onPause() в данный экземпляр записывается текущее значение секундомера, а в методе onResume() это значение «достается». Чтобы поток, создаваемый в onResume(), мог завершиться, и, соответственно, секундомер переставал считать, добавлен флаг, значение которого задается равным true в методе onResume() и false в методе onPause().

**3. Выводы**

В ходе выполнения лабораторной работы были изучены: жизненный цикл Activity, основные возможности и свойства Alternative resources, алгоритм для выбора best-matching resource. Также, была проведена работа по нахождению ошибок, связанных с жизненным циклом activity, и исправлению их в заданном приложении.

**Приложение 1**

**MainActivity.kt:**

class MainActivity : AppCompatActivity() {  
 var secondsElapsed: Int = 0  
 private lateinit var sharedPref:SharedPreferences  
 var flag=true  
 private var backgroundThread = Thread **{** while (flag) {  
 Thread.sleep(1000)  
 textSecondsElapsed.post **{** textSecondsElapsed.*text* = "Seconds elapsed: " + secondsElapsed++  
 **}** }  
 **}** override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 }  
  
 override fun onResume() {  
 super.onResume()  
 sharedPref=*applicationContext*.getSharedPreferences("myPrefs", *MODE\_PRIVATE*)  
 secondsElapsed=sharedPref.getInt("secondsElapsed",0)  
 flag=true  
 backgroundThread.start()  
 }  
  
 override fun onPause() {  
 super.onPause()  
 flag=false  
 val editor=sharedPref.edit()  
 editor.putInt("secondsElapsed",secondsElapsed)  
 editor.apply()  
 }  
  
  
}