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. 3530901/80202 Ткаченко Д. Ю.

(подпись)

Преподаватель А. О. Алексюк

(подпись)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Санкт-Петербург

2020

1. Цели  
1. Ознакомиться с жизненным циклом Activity.

2. Изучить основные возможности и свойства Alternative resources.

Вариант 15.

Задание для задачи № 2: Night mode.

2. Программа работы

2.1. Задача 1. Activity

Продемонстрируйте жизненный цикл Activity на любом нетривиальном примере.

Пример: Выключение телефона

Когда пользователь выключает телефон вызываются методы onPause() и onStop(), после чего телефон выключается.



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

Пример: совершение вызова на телефон во время нахождения в приложении.

Когда пользователю звонят и он отвечает, происходит вызов метода onPause(). При возврате в приложения вызывается метод onResume().



2.2. Задача 2. Alternative Resources

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

Варинт : API Level

Для демонстрации работы альтернативного ресурса были созданы два layout-ресурса для значений квалификаторов v15 и v30.

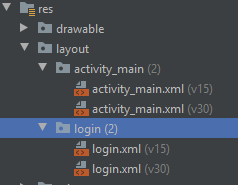


Рис. 5.1. Структура папок с альтернативными ресурсами

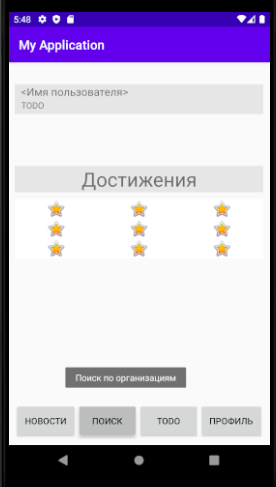


Рис.5.2. Запущенное приложение на устройствах с конфигурацией v30 позволяет получить информацию о кнопках в приложение в виде выскакивающих подсказок

В папке activity\_main находятся два xml файла с одинаковыми названиями и разными конфигурациями. При запуске приложения выбирается файл, соответствующий конфигурации устройства.

Удобство данного альтернативного ресурса заключается в том, что пользователь получает подсказку при долгом нажатии на кнопку.

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 итоге. Ответ докажите.

|  |
| --- |
| Вариант 15: |
|  | ==================================== |
|  | Конфигурация устройства: |
|  | LOCALE\_LANG: en |
|  | LOCALE\_REGION: rCA |
|  | SCREEN\_SIZE: small |
|  | SCREEN\_ASPECT: notlong |
|  | ROUND\_SCREEN: round |
|  | ORIENTATION: port |
|  | UI\_MODE: television |
|  | NIGHT\_MODE: night |
|  | PIXEL\_DENSITY: xxxhdpi |
|  | TOUCH: notouch |
|  | PRIMARY\_INPUT: qwerty |
|  | NAV\_KEYS: nonav |
|  | PLATFORM\_VER: v25 |
|  |  |
|  | Конфигурация ресурсов: |
|  | ~~(default)~~ |
|  | ~~en-xxxhdpi-dpad-v25~~ |
|  | ~~fr-rFR-long-port-car-xxxhdpi-v26~~ |
|  | ~~en-normal-qwerty-trackball~~ |
|  | ~~small-v25~~ |
|  | ~~port-tvdpi-nonav~~ |
|  | **rCA-round-qwerty** |
|  | ~~fr-rUS-xlarge-round-xxhdpi-dpad~~ |
|  | ~~12key-v25~~ |
|  | ~~normal-round-port-notouch-nokeys-wheel-v26~~ |
|  | ~~rFR-small-night-finger-nokeys-wheel~~ |

Сначала вычеркиваем все, что противоречит, а после из оставшихся выбираем те, где наиболее «важный» ресурс.

Остаётся один ресурс, который и будет выбран: rCA-round-qwerty

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 создана переменная, в которую записывалось значение false при выходе из приложения и true при входе (Для чтения). В методе onPause() записывается текущее значение секундомера, а в методе onResume() это значение «достается».

**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:**

package ru.spbstu.icc.kspt.lab2.continuewatch  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import kotlinx.android.synthetic.main.activity\_main.\*  
  
class MainActivity : AppCompatActivity() {  
 var secondsElapsed: Int = 0  
 var isStart = false  
  
 var backgroundThread = Thread **{** while (true) {  
 Thread.sleep(1000)  
 if (isStart) {  
 textSecondsElapsed.post **{** textSecondsElapsed.setText("Seconds elapsed: " + secondsElapsed++)  
 **}** }  
 }  
 **}** override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 backgroundThread.start()  
 }  
  
 override fun onResume() {  
 isStart = true  
 super.onResume()  
 }  
  
 override fun onPause() {  
 isStart = false  
 super.onPause()  
 }  
  
 override fun onSaveInstanceState(outState: Bundle) {  
 outState.putInt("secs", secondsElapsed)  
 super.onSaveInstanceState(outState)  
 }  
  
 override fun onRestoreInstanceState(savedInstanceState: Bundle) {  
 secondsElapsed = savedInstanceState.getInt("secs")  
 textSecondsElapsed.post **{** textSecondsElapsed.setText("Seconds elapsed: " + secondsElapsed++)  
 **}** super.onRestoreInstanceState(savedInstanceState)  
 }  
}