**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 6**

по дисциплине «Разработка мобильных приложений»

**Тема: Unit 1 Начало работы “Жизненный цикл и состояние активности”**

Студент группы РИЗ-300028у: А.С. Севостьянов

Преподаватель: С.И.Тимошенко,

доц., к.т.н.

**Екатеринбург 2022**

содержание

[1 Постановка задачи 3](#_Toc118041527)

[2 Результаты проделанной работы 4](#_Toc118041528)

[3 Анализ полученных результатов 5](#_Toc118041529)

## 1 Постановка задачи

Добавить код в приложение TwoActivities из предыдущего практического руководства для реализации различных Activity обратных вызовов жизненного цикла для включения операторов ведения журнала.

Наблюдать за изменениями состояния по мере запуска приложения и при взаимодействии с каждым Activity из них в вашем приложении.

Изменить свое приложение, чтобы сохранить состояние экземпляра Activity, которое неожиданно воссоздается в ответ на поведение пользователя или изменение конфигурации на устройстве.

.

## Результаты проделанной работы

Программный код проекта. Изменили метод onCreat.

@Override  
protected void onCreate(Bundle savedInstanceState) {  
   super.onCreate(savedInstanceState);  
   setContentView(R.layout.activity\_main);  
  
   Log.d(LOG\_TAG, "-------");  
   Log.d(LOG\_TAG, "onCreate");  
  
   // Инициализируйте все переменные представления..  
   mMessageEditText = findViewById(R.id.editText\_main);  
   mReplyHeadTextView = findViewById(R.id.text\_header\_reply);  
   mReplyTextView = findViewById(R.id.text\_message\_reply);  
  
   // Восстановите сохраненное состояние.  
   if (savedInstanceState != null) {  
       boolean isVisible =   
                     savedInstanceState.getBoolean("reply\_visible");  
 // Показывать как заголовок, так и представления сообщений. Если isVisible является

// false или отсутствует в пакете, используйте макет по умолчанию.

if (isVisible) {  
           mReplyHeadTextView.setVisibility(View.VISIBLE);  
           mReplyTextView.setText(savedInstanceState  
                                  .getString("reply\_text"));  
           mReplyTextView.setVisibility(View.VISIBLE);  
       }  
   }  
}

Перечислены методы жизненного цикла из конструктора.

@Override  
protected void onStart() {  
        super.onStart();  
        Log.d(LOG\_TAG, "onStart");  
}  
  
@Override  
protected void onPause() {  
        super.onPause();  
        Log.d(LOG\_TAG, "onPause");  
}  
  
@Override  
protected void onRestart() {  
        super.onRestart();  
        Log.d(LOG\_TAG, "onRestart");  
}  
  
@Override  
protected void onResume() {  
        super.onResume();  
        Log.d(LOG\_TAG, "onResume");  
}  
  
@Override  
protected void onStop() {  
        super.onStop();  
        Log.d(LOG\_TAG, "onStop");  
}

@Override  
protected void onDestroy() {  
        super.onDestroy();  
        Log.d(LOG\_TAG, "onDestroy");  
} @Override  
public void onSaveInstanceState(Bundle outState) {  
   super.onSaveInstanceState(outState);

// Если заголовок виден, сообщение необходимо сохранить.

// В противном случае мы по-прежнему используем макет по умолчанию.

if (mReplyHeadTextView.getVisibility() == View.VISIBLE) {  
       outState.putBoolean("reply\_visible", true);  
       outState.putString("reply\_text",   
                      mReplyTextView.getText().toString());  
   }  
}

## **Анализ полученных результатов**

В результате выполнения данной лабораторной работы мной был разработан проект, в котором я понял, как Activity работает жизненный цикл.

Когда Activity запускается, приостанавливается, останавливается и уничтожается.

О методах обратного вызова жизненного цикла, связанных с Activity изменениями.

Влияние действий (таких как изменения конфигурации), которые могут привести к Activity событиям жизненного цикла.

Как сохранить Activity состояние во всех событиях жизненного цикла.