Пенза 2022

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

Выполнил студент группы 19ВВ2:

Серов Д.В.

Принял:

Митрохин М.А.

Патунин Д.В.

**ОТЧЕТ**

по лабораторной работе №3

по курсу «Графический интерфейс Android и Qt»

на тему «Жизненный цикл Activity в Android-приложениях. Intent»

**Цель работы:** изучить жизненный цикл Android приложений.

**Задание**:

1. Переопределить методы жизненного цикла первой activity.
2. Добавить в переопределенные методы жизненного цикла код для вывода сообщений на экран или в системный журнал.
3. Создать intent для вызова второй activity.
4. Проанализировать порядок вызова методов жизненного цикла на основе выводимых сообщений при различных действиях пользователя.
5. Передать данные из одной activity в другую при помощи intent и продемонстрировать результат.

**Ход лабораторной работы:**

1. Переопределил методы жизненного цикла первой activity.
2. Добавил в переопределенные методы жизненного цикла код для вывода сообщений на экран или в системный журнал.

**Листинг MainActivity2:**

public class MainActivity2 extends AppCompatActivity {  
  
 private final static String *TAG* = "MainActivity2";  
  
 ArrayList<String> addString;  
 ArrayAdapter<String> adapter;  
 ArrayList<String> selectedString;  
 ListView listView;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main2*);  
  
 addString = new ArrayList<>();  
 adapter = new ArrayAdapter <>(this, android.R.layout.*simple\_list\_item\_multiple\_choice*, addString);  
 selectedString = new ArrayList<>();  
  
 Button buttonAdd = findViewById(R.id.*button1*);  
 listView = findViewById(R.id.*listView*);  
 ImageButton imgBtn = findViewById(R.id.*imgBut*);  
  
 Bundle arguments = getIntent().getExtras();  
 String name = arguments.get("login").toString();  
 addString.add(name);  
 adapter.notifyDataSetChanged();  
 listView.setAdapter(adapter);  
  
  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener(){  
 @Override  
 public void onItemClick(AdapterView<?> parent, View v, int position, long id)  
 {  
 *// получаем нажатый элемент* String res = adapter.getItem(position);  
 if(listView.isItemChecked(position))  
 selectedString.add(res);  
  
 else  
 selectedString.remove(res);  
 }  
 });  
  
  
  
 }  
  
 public void add(View view){  
  
 EditText editText = findViewById(R.id.*edit*);  
 String res = editText.getText().toString();  
 if(res.isEmpty()){  
 editText.setHintTextColor(Color.*rgb*(255,0,0));  
 }  
 else{  
 editText.setHintTextColor(Color.*GRAY*);  
 addString.add(res);  
 adapter.notifyDataSetChanged();  
 listView.setAdapter(adapter);  
 editText.setText("");  
 }  
 }  
 public void remove(View view){  
 *// получаем и удаляем выделенные элементы* for(int i=0; i< selectedString.size();i++){  
 adapter.remove(selectedString.get(i));  
 }  
 *// снимаем все ранее установленные отметки* listView.clearChoices();  
 *// очищаем массив выбраных объектов* selectedString.clear();  
  
 adapter.notifyDataSetChanged();  
 }  
  
 @Override  
 protected void onDestroy(){  
 super.onDestroy();  
 Log.*d*(*TAG*, "onDestroy");  
  
 Toast.*makeText*(MainActivity2.this, "APP onDestroy", Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 protected void onStop(){  
 super.onStop();  
 Log.*d*(*TAG*, "onStop");  
  
 Toast.*makeText*(MainActivity2.this, "APP onStop", Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 protected void onStart(){  
 super.onStart();  
 Log.*d*(*TAG*, "onStart");  
  
 Toast.*makeText*(MainActivity2.this, "APP onStart", Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 protected void onPause(){  
 super.onPause();  
 Log.*d*(*TAG*, "onPause");  
 Toast.*makeText*(MainActivity2.this, "APP onPause", Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 protected void onResume(){  
 super.onResume();  
 Log.*d*(*TAG*, "onResume");  
  
 Toast.*makeText*(MainActivity2.this, "APP onResume", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 protected void onRestart(){  
 super.onRestart();  
 Log.*d*(*TAG*, "onRestart");  
  
 Toast.*makeText*(MainActivity2.this, "APP onRestart", Toast.*LENGTH\_SHORT*).show();  
 }  
  
}

1. Проанализировал порядок вызова методов жизненного цикла на основе выводимых сообщений при различных действиях пользователя.

### **onStart()**

За onCreate() всегда следует вызов onStart(), но перед onStart() не обязательно должен идти onCreate(), так как onStart() может вызываться и для возобновления работы приостановленного приложения. При вызове onStart() окно ещё не видно пользователю, вызывается непосредственно перед тем, как активность становится видимой пользователю.

**onResume()**

Метод onResume() вызывается после onStart(), даже когда окно работает в приоритетном режиме и пользователь может его наблюдать. В этот момент пользователь взаимодействует с созданным вами окном. Приложение получает монопольные ресурсы. Запускает воспроизведение анимации, аудио и видео. Также может вызываться после onPause().

### **onPause()**

Когда пользователь решает перейти к работе с новым окном, система вызовет для прерываемого окна метод onPause(). По сути происходит свёртывание активности. Сохраняет незафиксированные данные. Деактивирует и выпускает монопольные ресурсы. Останавливает воспроизведение видео, аудио и анимацию. От onPause() можно перейти к вызову либо onResume(), либо onStop().

### **onStop()**

Метод onStop() вызывается, когда окно становится невидимым для пользователя. Это может произойти при её уничтожении, или если была запущена другая активность (существующая или новая), перекрывшая окно текущей активности. Всегда сопровождает любой вызов метода onRestart(), если активность возвращается, чтобы взаимодействовать с пользователем, или метода onDestroy(), если эта активность уничтожается.

### **onRestart()**

Если окно возвращается в приоритетный режим после вызова onStop(), то в этом случае вызывается метод onRestart(). Т.е. вызывается после того, как активность была остановлена и снова была запущена пользователем. Всегда сопровождается вызовом метода onStart().

### **onDestroy()**

Метод вызывается по окончании работы активности, так как все необходимые операции по освобождению ресурсов вы сделали в методе onStop(), то в этом методе вы можете подстраховаться и проверить ещё раз все неосвобождённые ресурсы.

**Результат вывода сообщений:**



Рис.1 Запуск второй activity(системный журнал)



Рис.2 Запуск второй activity(сообщение в приложении)

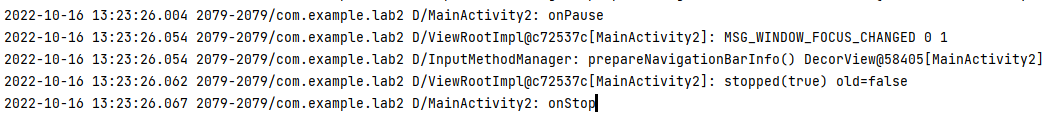


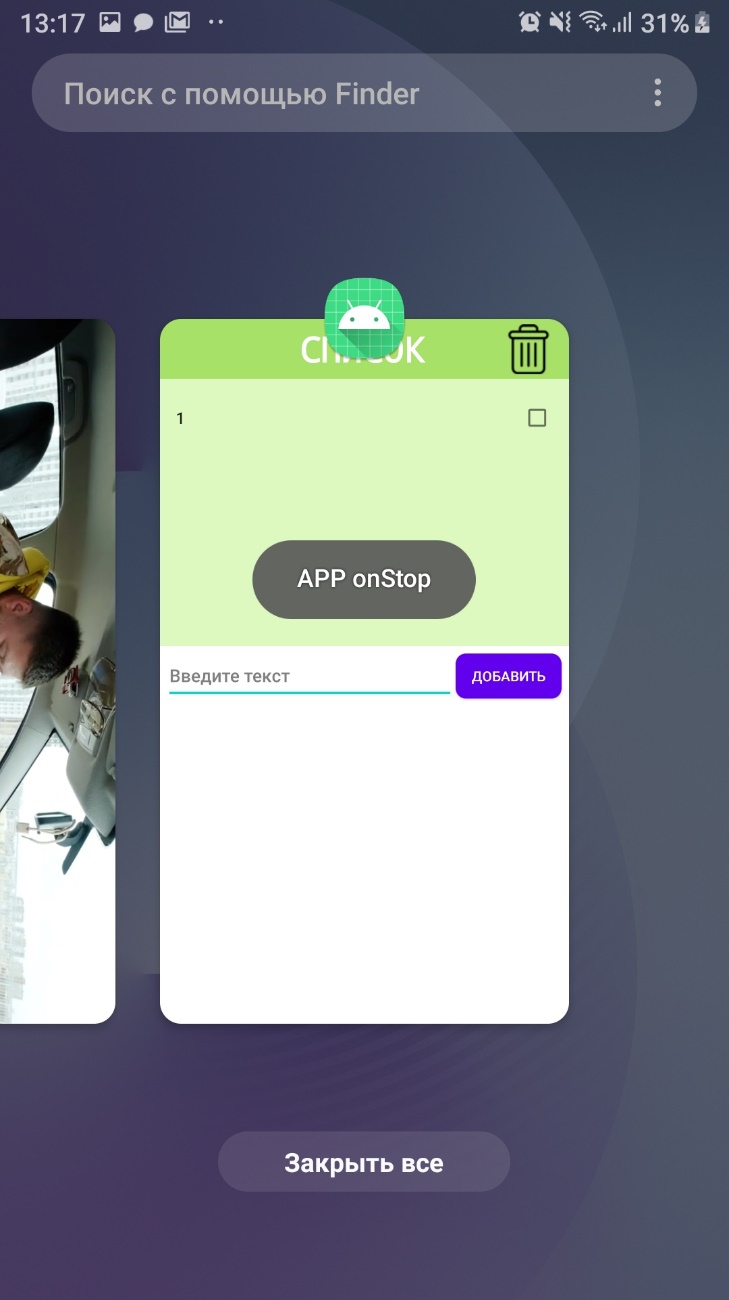
Рис.3 Перевод в невидимый режим второй activity(системный журнал)

Рис.4 Перевод в невидимый режим второй activity(сообщение в приложении)

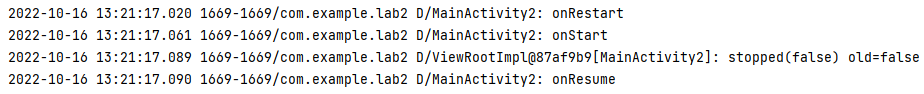


Рис.5 Повторный вход во вторую activity из невидимого режима (системный журнал)



Рис.6 Закрытие второй activity (системный журнал)

1. Создать intent для вызова второй activity. Передать данные из одной activity в другую при помощи intent и продемонстрировать результат.

**Листинг MainActivity:**

public class MainActivity extends AppCompatActivity {  
  
 EditText editText1, editText2;  
 Button singIn, singUp;  
 SharedPreferences preferences;  
  
 final String LOGIN = "login";  
 final String PASS = "pass";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 editText1 = findViewById(R.id.*login*);  
 editText2 = findViewById(R.id.*pass*);  
 singIn = findViewById(R.id.*singIn*);  
 singUp = findViewById(R.id.*singUp*);  
 preferences = getSharedPreferences("authentication", *MODE\_PRIVATE*);  
  
 singIn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
  
 String login = preferences.getString(LOGIN,"");  
 String pass = preferences.getString(PASS, "");  
 int condition = 0;  
  
 String cmpLogin = editText1.getText().toString();  
 String cmpPass = editText2.getText().toString();  
  
 if(cmpLogin.equals(login)){  
 condition += 1;  
 editText1.setText("");  
 }else{  
 Toast.*makeText*(MainActivity.this, "Неверный логин", Toast.*LENGTH\_SHORT*).show();  
 editText1.setText("");  
 condition = 0;  
 }  
  
 if (cmpPass.equals(pass)){  
 condition += 1;  
 editText2.setText("");  
  
 }else{  
 Toast.*makeText*(MainActivity.this, "Неверный пароль", Toast.*LENGTH\_SHORT*).show();  
 editText2.setText("");  
 condition = 0;  
 }  
  
 if(condition == 2){  
 Intent intent = new Intent(MainActivity.this, MainActivity2.class);  
 intent.putExtra(LOGIN, cmpLogin);  
 startActivity(intent);  
 }  
  
 }  
 });  
  
 singUp.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
  
 SharedPreferences.Editor editor = preferences.edit();  
 editor.putString(LOGIN, editText1.getText().toString());  
 editor.putString(PASS, editText2.getText().toString());  
 editor.apply();  
 Toast.*makeText*(MainActivity.this, "Пользователь зарегистрирован", Toast.*LENGTH\_SHORT*).show();  
  
 }  
 });  
 }  
}

****

Рис.7 Запуск приложения



Рис.8 Регистрация пользователя



Рис.9 Переход в другую activity и передача имени пользователя

**Вывод:** изучили жизненный цикл Android приложения и научились переходить и предавать данные между activity.