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 №4

по курсу «Графический интерфейс Android и Qt»

на тему «SharedPreferences для хранения данных»

**Цель работы:** изучить SharedPreferences для хранения данных.

**Задание**:

1. Переопределить методы жизненного цикла одной из activity для сохранения при её закрытии введенных пользовательских данных и загрузке сохраненных данных при восстановлении activity.
2. Переопределить методы жизненного цикла всех activity для сохранения при закрытии приложения общих настроек (например, языка интерфейса) и загрузке сохраненных данных при запуске приложения.

**Ход лабораторной работы:**

1. Переопределил методы жизненного цикла одной из activity для сохранения при её закрытии введенных пользовательских данных и загрузке сохраненных данных при восстановлении activity. Создал методы save и load для сохранения данных listView. В методе onCreate вызываем загрузку данных, а в onPause сохранение.

**Листинг MainActivity2:**

public class MainActivity2 extends AppCompatActivity {  
  
 private final static String *TAG* = "MainActivity2";  
 final String LIST = "list";  
  
 ArrayList<String> addString;  
 ArrayAdapter<String> adapter;  
 ArrayList<String> selectedString;  
 ListView listView;  
 SharedPreferences preferences;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main2*);  
  
 preferences = getSharedPreferences("listApp", *MODE\_PRIVATE*);  
 addString = new ArrayList<>();  
 adapter = new ArrayAdapter <>(this, android.R.layout.*simple\_list\_item\_multiple\_choice*, addString);  
 selectedString = new ArrayList<>();  
 listView = findViewById(R.id.*listView*);  
  
 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);  
 }  
 });  
  
 load();  
 }  
  
 public void save(){  
 SharedPreferences.Editor editor = preferences.edit();  
  
 Set<String> itemList = new HashSet<>();  
  
 for(int i=0; i<addString.size(); i++){  
 itemList.add(addString.get(i));  
 }  
 editor.putStringSet(LIST, itemList);  
 editor.apply();  
 }  
 public void load(){  
  
 Set<String> list = preferences.getStringSet(LIST, new HashSet<>());  
  
 for(String string : list){  
  
 addString.add(string);  
 adapter.notifyDataSetChanged();  
 listView.setAdapter(adapter);  
 }  
  
 }  
  
 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();  
 save();  
 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();  
 }  
  
}

**Результат:**

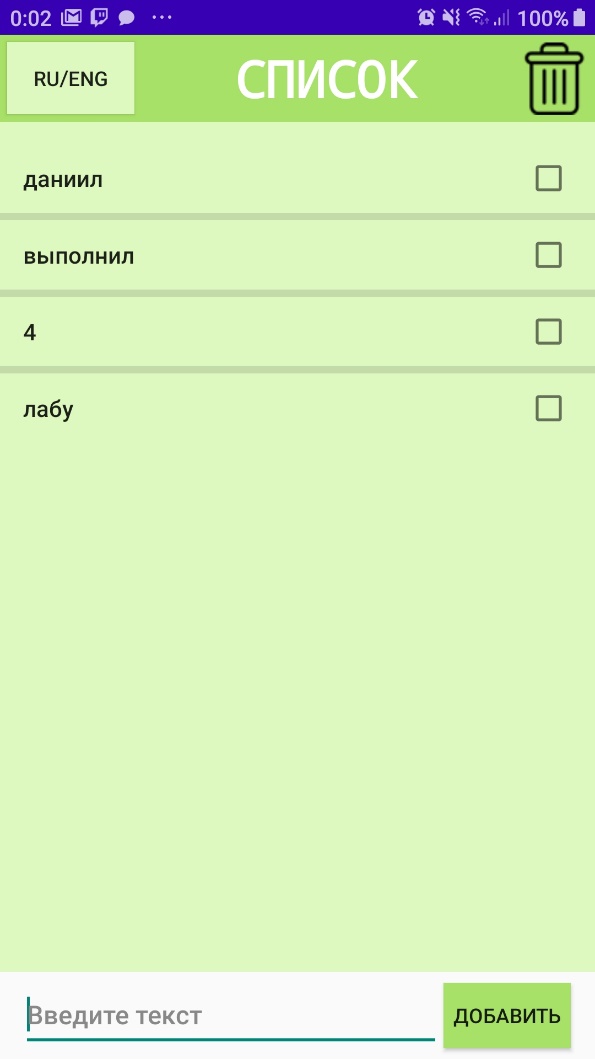


Рис.1 Добавил записи в список

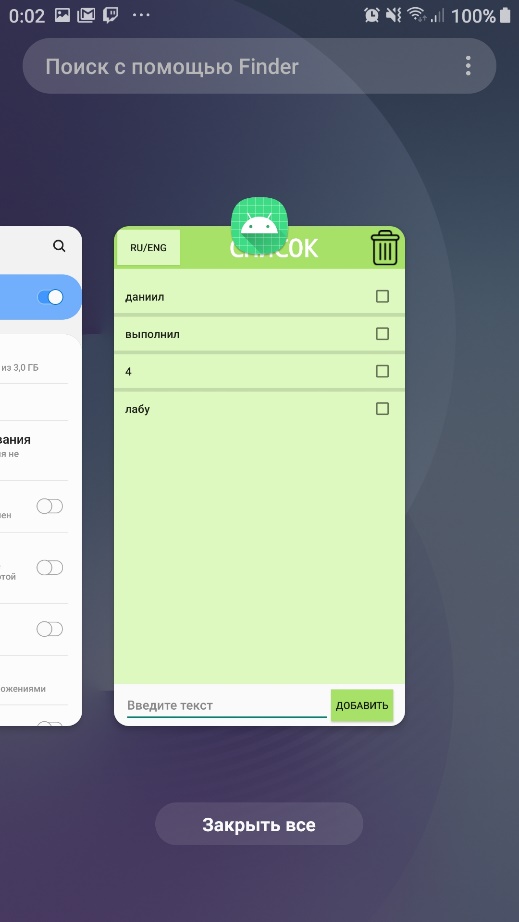


Рис.2 Закрыл приложение (список сохранился)

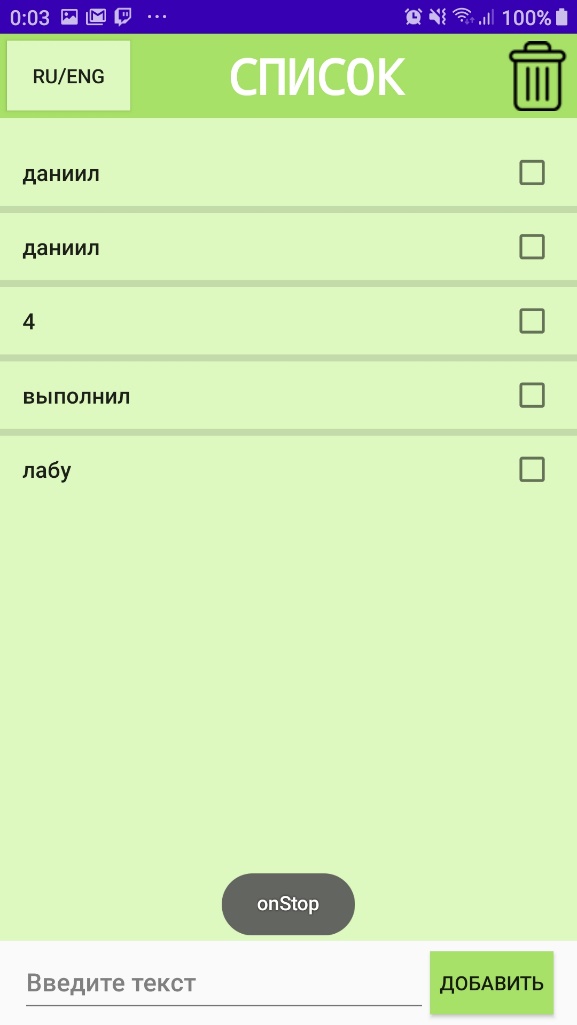


Рис.3 Повторный вход в приложение (список загрузился)

1. Переопределил методы жизненного цикла activity для сохранения при закрытии приложения общих настроек (например, языка интерфейса) и загрузке сохраненных данных при запуске приложения. В две активити добавлены методы сохранение выбранного языка интерфейса и передача состояния между активити.

**Листинг MainActivity:**

public class MainActivity extends AppCompatActivity {  
  
 private final static String *TAG* = "MainActivity";  
 EditText editText1, editText2;  
 Button singIn, singUp, lang;  
 SharedPreferences preferences\_main;  
  
 final String LOGIN = "login";  
 final String PASS = "pass";  
 final String STATE = "state";  
 int state = 0;  
 int count = 0;  
 @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*);  
 lang = findViewById(R.id.*lang*);  
 preferences\_main = getSharedPreferences("authentication", *MODE\_PRIVATE*);  
  
 singIn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
  
 String login = preferences\_main.getString(LOGIN,"нет данных");  
 String pass = preferences\_main.getString(PASS, "нет данных");  
 Log.*i*("login", login);  
 Log.*i*("password", pass);  
 int condition = 0;  
  
 if (!(pass.isEmpty() && login.isEmpty())){  
 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\_main.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();  
 }  
 });  
  
 lang.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
  
 count += 1;  
 if(count % 2 == 0){  
 state = 0;  
  
 }else  
 state = 1;  
 setLang(state);  
  
 Log.*i*("state", String.*valueOf*(state));  
 Log.*i*("count", String.*valueOf*(count));  
  
 }  
 });  
 }  
 public void save(){  
 SharedPreferences.Editor editor = preferences\_main.edit();  
 editor.putInt(STATE, state);  
 editor.apply();  
 Log.*i*("stateSave", String.*valueOf*(state));  
 int loadState = preferences\_main.getInt(STATE,3);  
  
 Log.*i*("loooad", String.*valueOf*(loadState));  
  
 Toast.*makeText*(MainActivity.this, "SaveState", Toast.*LENGTH\_SHORT*).show();  
 }  
 public void load(){  
  
 int loadState = preferences\_main.getInt(STATE,3);  
 state = loadState;  
 Log.*i*("stateLoad", String.*valueOf*(state));  
 }  
 public void setLang(int stateLang){  
 if(stateLang == 1){  
 editText1.setHint(R.string.*nameEn*);  
 editText2.setHint(R.string.*passEn*);  
 singIn.setText(R.string.*singInEn*);  
 singUp.setText(R.string.*singUpEn*);  
 }  
 if(stateLang == 0){  
 editText1.setHint(R.string.*name*);  
 editText2.setHint(R.string.*pass*);  
 singIn.setText(R.string.*singIn*);  
 singUp.setText(R.string.*singUp*);  
 }  
 }  
 @Override  
 protected void onDestroy(){  
 super.onDestroy();  
 Log.*d*(*TAG*, "onDestroy");  
 Toast.*makeText*(MainActivity.this, "onDestroy", Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 protected void onStop(){  
 super.onStop();  
  
 Log.*d*(*TAG*, "onStop");  
 Toast.*makeText*(MainActivity.this, "onStop", Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 protected void onStart(){  
 super.onStart();  
 load();  
 setLang(state);  
 Log.*d*(*TAG*, "onStart");  
 Toast.*makeText*(MainActivity.this, "onStart", Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 protected void onPause(){  
 super.onPause();  
 save();  
 Log.*d*(*TAG*, "onPause");  
 Toast.*makeText*(MainActivity.this, "onPause", Toast.*LENGTH\_SHORT*).show();  
 }  
 @Override  
 protected void onResume(){  
 super.onResume();  
 Log.*d*(*TAG*, "onResume");  
 *//load();* Toast.*makeText*(MainActivity.this, "onResume", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 protected void onRestart(){  
 super.onRestart();  
  
 Log.*d*(*TAG*, "onRestart");  
 Toast.*makeText*(MainActivity.this, "onRestart", Toast.*LENGTH\_SHORT*).show();  
 }  
}

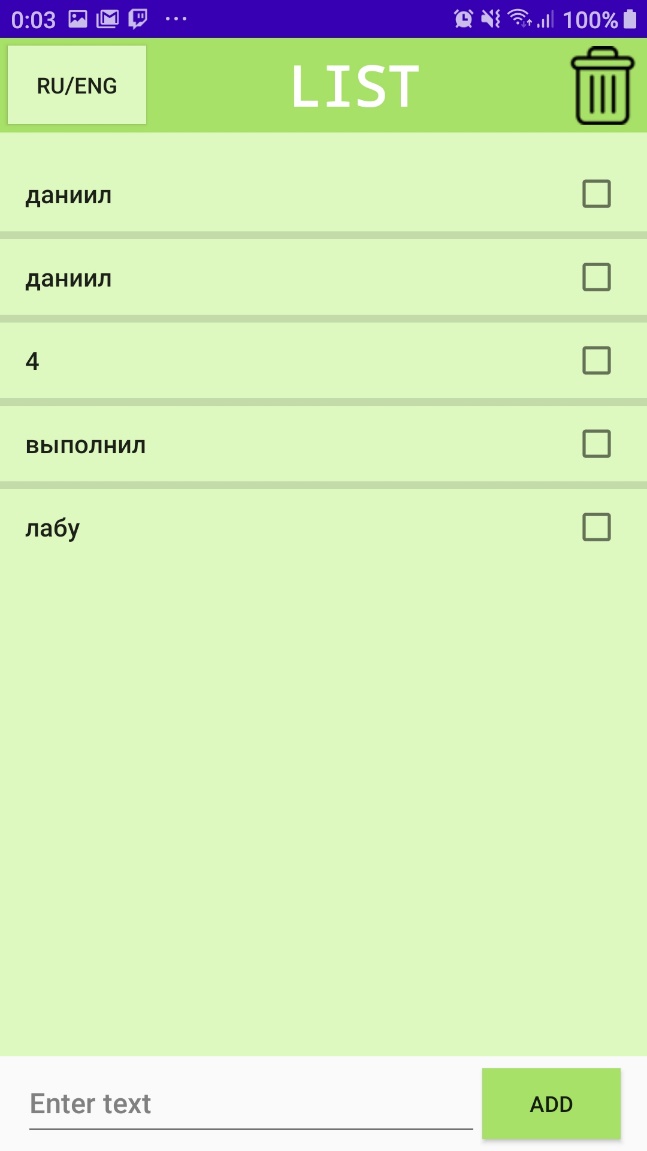
****

Рис.4 Смена языка

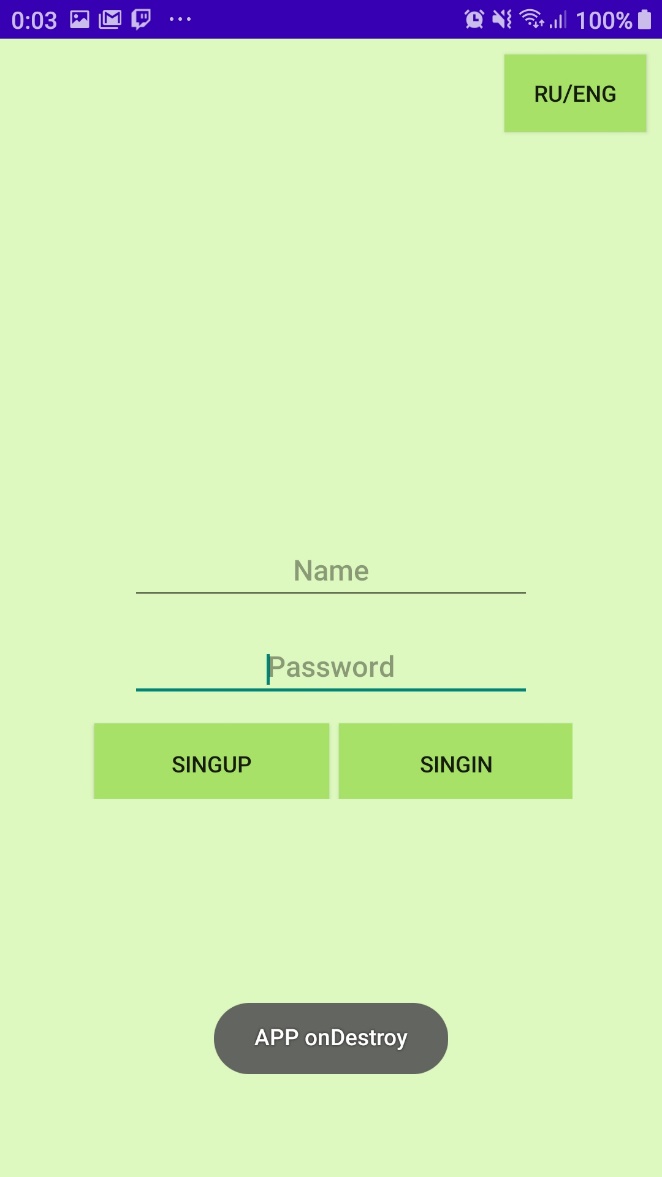


Рис.5 Переход в другую активити с сохранением выбранного языка

**Вывод:** изучили SharedPreferences для хранения данных и научились сохранять и получать данные из файла.