**Лабораторная работа №8**

# Сохранение настроек

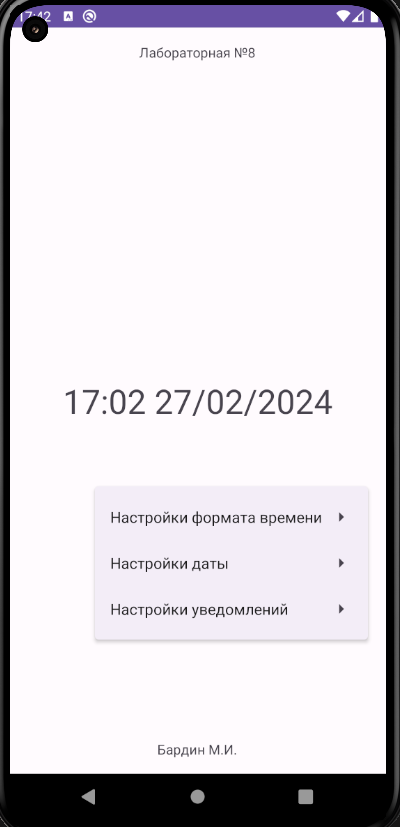
**Цель работы:** Научиться сохранять настройки приложения с применение SharedPreferences.

**Задание:**

1. Модифицируйте приложение, созданное в лабораторной работе №7.
2. Добавьте пункт меню, позволяющий полностью отключить toast-уведомление.
3. Обеспечьте сохранение настроек с помощью SharedPreferences.

Ход работы:

public class MainActivity extends AppCompatActivity {  
 final int MENU\_GROUP\_TIME = 1;  
 final int MENU\_ITEM\_24H = 0;  
 final int MENU\_ITEM\_12H = 1;  
  
 final int MENU\_GROUP\_DATE = 2;  
 final int MENU\_ITEM\_ON = 0;  
 final int MENU\_ITEM\_OFF = 1;  
  
 final int MENU\_GROUP\_TOAST = 3;  
  
 private int OPTION\_TIME\_FORMAT = MENU\_ITEM\_12H;  
 final String TIME\_FORMAT\_PREF = "OPTION\_TIME\_FORMAT";  
 private int OPTION\_DATE\_STATE = MENU\_ITEM\_ON;  
 final String DATE\_STATE\_PREF = "OPTION\_DATE\_STATE";  
 private int OPTION\_TOAST = MENU\_ITEM\_ON;  
 final String TOAST\_PREF = "OPTION\_TOAST";  
  
 SharedPreferences sPref;  
 final Handler handler=new Handler();  
 private boolean toasted = false;  
 private String datetime\_format;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 loadPreferences();  
  
 handler.postDelayed(updateTime,0);  
 handler.postDelayed(show\_toast,0);  
  
 ConstraintLayout parentLayout = findViewById(R.id.rootLayout);  
  
 registerForContextMenu(parentLayout);  
 updateSettings();  
 }  
  
 private void updateSettings() {  
 String TIME\_FORMAT="", DATE\_STATE="";  
 switch (OPTION\_TIME\_FORMAT){  
 case MENU\_ITEM\_24H: TIME\_FORMAT="HH:MM"; break;  
 case MENU\_ITEM\_12H: TIME\_FORMAT="hh:MM aa"; break;  
 }  
 switch (OPTION\_DATE\_STATE){  
 case MENU\_ITEM\_ON: DATE\_STATE=" dd/MM/YYYY"; break;  
 case MENU\_ITEM\_OFF: DATE\_STATE=""; break;  
 }  
 datetime\_format = TIME\_FORMAT+DATE\_STATE;  
 set\_current\_time();  
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v,  
 ContextMenu.ContextMenuInfo menuInfo) {  
 MenuItem saved\_item;  
 SubMenu subMenuTime = menu.addSubMenu(R.string.subMenuTime);  
  
 subMenuTime.add(MENU\_GROUP\_TIME, MENU\_ITEM\_24H, Menu.NONE, R.string.MENU\_ITEM\_24H);  
 subMenuTime.add(MENU\_GROUP\_TIME, MENU\_ITEM\_12H, Menu.NONE, R.string.MENU\_ITEM\_12H);  
 subMenuTime.setGroupCheckable(MENU\_GROUP\_TIME, true, true);  
  
 saved\_item = subMenuTime.getItem(OPTION\_TIME\_FORMAT);  
 saved\_item.setChecked(true);  
  
 SubMenu subMenuDate = menu.addSubMenu(R.string.subMenuDate);  
 subMenuDate.add(MENU\_GROUP\_DATE, MENU\_ITEM\_ON, Menu.NONE, R.string.MENU\_ITEM\_ON);  
 subMenuDate.add(MENU\_GROUP\_DATE, MENU\_ITEM\_OFF, Menu.NONE, R.string.MENU\_ITEM\_OFF);  
 subMenuDate.setGroupCheckable(MENU\_GROUP\_DATE, true, true);  
  
 saved\_item = subMenuDate.getItem(OPTION\_DATE\_STATE);  
 saved\_item.setChecked(true);  
  
 SubMenu subMenuToast = menu.addSubMenu(R.string.subMenuToast);  
 subMenuToast.add(MENU\_GROUP\_TOAST, MENU\_ITEM\_ON, Menu.NONE, R.string.MENU\_ITEM\_ON);  
 subMenuToast.add(MENU\_GROUP\_TOAST, MENU\_ITEM\_OFF, Menu.NONE, R.string.MENU\_ITEM\_OFF);  
 subMenuToast.setGroupCheckable(MENU\_GROUP\_TOAST, true, true);  
  
 saved\_item = subMenuToast.getItem(OPTION\_TOAST);  
 saved\_item.setChecked(true);  
 }  
  
  
 @Override  
 public boolean onContextItemSelected(MenuItem item)  
 {  
 if(item.getGroupId()!=Menu.NONE)  
 switch (item.getGroupId()){  
 case MENU\_GROUP\_TIME:  
 OPTION\_TIME\_FORMAT= item.getItemId();  
 break;  
 case MENU\_GROUP\_DATE:  
 OPTION\_DATE\_STATE = item.getItemId();  
 break;  
 case MENU\_GROUP\_TOAST:  
 OPTION\_TOAST = item.getItemId();  
 break;  
 }  
 item.setChecked(true);  
 updateSettings();  
 return true;  
 }  
 final Runnable show\_toast=new Runnable() {  
 @Override  
 public void run() {  
 if(OPTION\_TOAST==MENU\_ITEM\_OFF){  
 toasted=true;  
 return;  
 }  
 if(toasted)  
 return;  
  
 toasted = true;  
 Toast.makeText(MainActivity.this, R.string.Toast,  
 Toast.LENGTH\_LONG).show();  
 }  
 };  
  
 private void set\_current\_time() {  
 DateFormat df = new SimpleDateFormat(datetime\_format);  
 TextView tv = findViewById(R.id.DateTime);  
 tv.setText(df.format(Calendar.getInstance().getTime()));  
 }  
 final Runnable updateTime=new Runnable() {  
 @Override  
 public void run() {  
 set\_current\_time();  
 handler.postDelayed(this,1000);  
 }  
 };  
  
 //Remembering Prefs  
 @Override  
 public void onSaveInstanceState(Bundle savedInstanceState) {  
  
 savedInstanceState.putBoolean("toasted", toasted);  
 super.onSaveInstanceState(savedInstanceState);  
  
 }  
 @Override  
 public void onRestoreInstanceState(Bundle savedInstanceState) {  
 super.onRestoreInstanceState(savedInstanceState);  
 toasted = savedInstanceState.getBoolean("toasted");  
 updateSettings();  
 }  
  
 //Saving Prefs  
 void savePreferences() {  
 sPref = getPreferences(MODE\_PRIVATE);  
 SharedPreferences.Editor sPref\_editor = sPref.edit();  
 sPref\_editor.putInt(TIME\_FORMAT\_PREF, OPTION\_TIME\_FORMAT);  
 sPref\_editor.putInt(DATE\_STATE\_PREF, OPTION\_DATE\_STATE);  
 sPref\_editor.putInt(TOAST\_PREF, OPTION\_TOAST);  
 sPref\_editor.apply();  
 }  
  
 void loadPreferences() {  
 sPref = getPreferences(MODE\_PRIVATE);  
 OPTION\_TIME\_FORMAT = sPref.getInt(TIME\_FORMAT\_PREF, MENU\_ITEM\_12H);  
 OPTION\_DATE\_STATE= sPref.getInt(DATE\_STATE\_PREF, MENU\_ITEM\_ON);  
 OPTION\_TOAST = sPref.getInt(TOAST\_PREF, MENU\_ITEM\_ON);  
 updateSettings();  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 savePreferences();  
 }  
 @Override  
 protected void onPause() {  
 super.onPause();  
 savePreferences();  
 }  
  
  
}



**Вывод:** в ходе работы мы научились сохранять настройки приложения с применение SharedPreferences