**Лабораторная работа № 4**

**SharedPreferences для хранения данных**

**Общие сведения.**

SharedPreferences  - механизм постоянного хранения данных приложения, который позволяет сохранять и восстанавливать данные и настройки приложения и activity на стадиях жизненного цикла и между запусками приложения.

Данные хранятся в виде пар «ключ - значение».

Физически SharedPreferences   реализовано в виде файла в котором данные хранятся в виде пар «ключ - значение». Данный файл доступен только конкретному приложению, которым он создавался.

Для доступа к хранилищу используется класс, экземпляр которого можно получить через методы:

**getPreferences(**int mode**)** - вызывается внутри activity и содержит данные, относящиеся к ней;

**getSharedPreferences(**String name, int mode**)** - содержит определенные для всего приложения данные, вызывается из любого контекста.

Для записи данных в SharedPreferences используется объект класса **SharedPreferences**.**Edit**. Его возвращает метод **edit**() объекта SharedPreferences.

В зависимости от типа сохраняемых данных используются методы:

**putBoolean(**String key, boolean value**)**;

**putFloat(**String key, float value**);**

**putInt(**String key, int value**);**

**putLong(**String key, long value**);**

**putString(**String key, String value**);**

**putStringSet(**String key, Set values**).**

После добавления данных необходимо вызвать метод **apply()** или **commit()**. Метод **apply()** изменяет данные в памяти и асинхронно записывает обновления в файл. Метод **commit()** осуществляет синхронную запись данных в файл, его вызова из основного потока следует избегать при записи большого объема данных, т.к. это может замедлить работу пользовательского интерфейса.

Для получения данных из SharedPreferences используются методы:

**getBoolean(**String key, boolean defValue**)**;

**getFloat(**String key, float defValue**)**;

**getInt(**String key, int defValue**)**;

**getLong(**String key, long defValue**)**;

**getString(**String key, String defValue**)**;

**getStringSet(**String key, Set<String> defValues**)**;

**getAll**().

В указанные методы передается значение ключа и, при необходимости, возвращаемое значение по умолчанию, если ключ отсутствует.

Например, если нам необходимо сохранить текст из текстового поля и затем его загрузить, можно использовать следующий код:

@Override  
**protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_helloact***);  
  
 TextView text1 = findViewById(R.id.***text1***);  
 Button button1 = findViewById(R.id.***button1***);  
  
 SharedPreferences sharedPref = **this**.getPreferences(Context.***MODE\_PRIVATE***);  
  
 SharedPreferences.Editor editor = sharedPref.edit();  
 editor.putString(**"text"**, **"Shared "**+text1.getText().toString());  
 editor.apply();  
  
 button1.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 text1.setText(sharedPref.getString(**"text"**,**""**));  
 }  
 });  
}

**Ход выполнения проекта.**

1 Откройте проект, созданный в лабораторной работе № 3.

2 Измените код в соответствии с заданием.

**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.