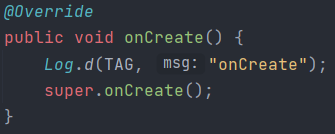
## Жизненный цикл сервиса

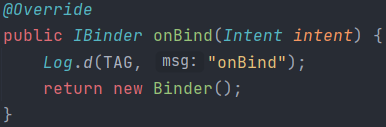
### Цель работы

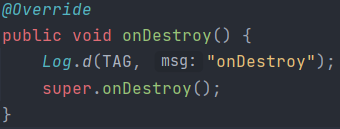
Познакомиться с жизненным циклом связанного и свободного сервиса в ОС Android, увидеть на практике порядок вызова методов жизненного цикла.

### Задания для выполнения

1. Создайте в приложении сервис, переопределив все основные методы жизненного цикла. Каждый метод должен выводить в консоль свое название при вызове.
2. Не забудьте вызов методов суперкласса и объявление сервиса в манифесте приложения.

Изображение выглядит как текст

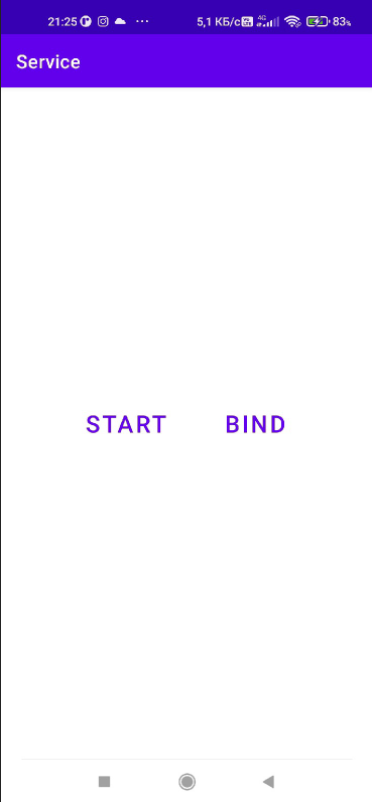
Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

1. В основной активности создайте две конопки. При нажатии на первую интент, вызывающий созданный сервис (явный интент) рассылается методом startService(), при нажатии второй - методом bindService().



1. Запустите приложение и понаблюдайте за жизненным циклом сервиса.



### Контрольные вопросы

1. Когда происходит создание и удаление связанного сервиса?

Связанный сервис создается, когда первый клиент присоединяется к сервису вызовом [bindService()](https://developer.android.com/reference/android/content/Context.html" \l "bindService(android.content.Intent,%20android.content.ServiceConnection,%20int)" \t "_blank). Связанный сервис удаляется, когда все клиенты отсоединились от сервиса вызовом метода [unbindService()](https://developer.android.com/reference/android/content/Context.html" \l "unbindService(android.content.ServiceConnection)" \t "_blank).

1. Какие методы жизненного цикла вызываются у сервиса и в какой последовательности?

У запущенного сервиса:

onCreate() -> onStartCommand() -> OnDestroy()

У связанного сервиса:

onCreate() -> onBind() -> onUnbind() -> OnDestroy()

1. Как соотносится ЖЦ активности и сервиса?

Так же как и активность, у сервиса есть методы ЖЦ создания и удаления onCreate() и OnDestroy(), однако в силу своего функционала у сервиса нет остальных методов ЖЦ, похожих на методы активности. Вместо этого он имеет свои: onStartCommand() для запущенного и onBind(), onUnbind() и onRebind() у связанного.

### Дополнительные задания

1. Добавьте в сервис полезное фоновое действие, которое выполняется продолжительное время. Как изменилось поведение сервиса?
2. Добавьте возможность из клиента (активности) убить сервис.
3. Добавьте логирование методов жизненного цикла активности. Пронаблюдайте за ней.

MainActivity.java:

package com.example.service;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.annotation.SuppressLint;  
import android.content.ComponentName;  
import android.content.Intent;  
import android.content.ServiceConnection;  
import android.os.Bundle;  
import android.os.IBinder;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
 final String TAG = "Service";  
 boolean bound = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 findViewById(R.id.start).setOnClickListener(this);  
 findViewById(R.id.bind).setOnClickListener(this);  
 }  
  
 @SuppressLint("NonConstantResourceId")  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(this, MyService.class);  
 switch (view.getId()) {  
 case R.id.start:  
 startService(intent);  
 stopService(intent);  
 break;  
 case R.id.bind:  
 ServiceConnection serviceConnection = new ServiceConnection() {  
 @Override  
 public void onServiceConnected(ComponentName componentName, IBinder iBinder) {  
 Log.d(TAG, "onServiceConnected");  
 }  
  
 @Override  
 public void onServiceDisconnected(ComponentName componentName) {  
 Log.d(TAG, "onServiceDisconnected");  
 }  
 };  
 if (!bound) {  
 bindService(intent, serviceConnection, BIND\_AUTO\_CREATE);  
 bound = true;  
 } else {  
 unbindService(serviceConnection);  
 bound = false;  
 }  
 break;  
 default:  
 break;  
 }  
 }  
}

activity\_main.xml:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="horizontal"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/start"  
 style="?android:attr/buttonBarButtonStyle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="20dp"  
 android:text="@string/start"  
 android:textSize="25sp" />  
  
 <Button  
 android:id="@+id/bind"  
 style="?android:attr/buttonBarButtonStyle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="20dp"  
 android:text="@string/bind"  
 android:textSize="25sp" />  
  
</LinearLayout>

MyService.java:

package com.example.service;  
  
import android.app.Service;  
import android.content.Intent;  
import android.os.Binder;  
import android.os.IBinder;  
import android.util.Log;  
  
public class MyService extends Service {  
 final String TAG = "Service";  
  
  
 public MyService() {  
 }  
  
 @Override  
 public void onCreate() {  
 *Log*.*d*(TAG, "onCreate");  
 super.onCreate();  
 }  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 Log.d(TAG, "onStartCommand");  
 return super.onStartCommand(intent, flags, startId);  
 }  
  
 @Override  
 public IBinder onBind(Intent intent) {  
 Log.d(TAG, "onBind");  
 return new Binder();  
 }  
  
 @Override  
 public boolean onUnbind(Intent intent) {  
 Log.d(TAG, "onUnBind");  
 return super.onUnbind(intent);  
 }  
  
 @Override  
 public void onDestroy() {  
 Log.d(TAG, "onDestroy");  
 super.onDestroy();  
 }  
}

AndroidManifest.xml:

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.service">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Service">  
 <service  
 android:name=".MyService"  
 android:enabled="true"  
 android:exported="true" />  
  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>