**MD5.1 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.
3. В основной активности создайте две конопки. При нажатии на первую интент, вызывающий созданный сервис (явный интент) рассылается методом startService(), при нажатии второй - методом bindService().
4. Запустите приложение и понаблюдайте за жизненным циклом сервиса.

Работа приложения:

Нажатие на кнопку запустить сервис

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Нажатие на кнопку остановить сервис

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

### Контрольные вопросы

1. Когда происходит создание и удаление связанного сервиса?

Когда мы используем метод onBind(), в момент первой привязки он вызывает также метод onCreate(), который создает сервис. Если сервис не в запущенном состоянии и клиентский компонент отвязывается от сервиса, то он уничтожается и вызывается onDestroy(). Чтобы удалить сервис его нужно отвязать методом onUnbind().

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

Основные методы:

protected void onCreate()

protected void onStart()

protected void onRestart()

protected void onResume()

protected void onPause()

protected void onStop()

protected void onDestroy()

Сначала вызывается метод onCreate()

После него идет метод onStart() (В некоторых случаях метод onCreate() не вызывается перед методом onStart() – например в случае возобновления работы приостановленного приложения)

Метод onResume() вызывается после метода onStart()

Метод onPause() вызывается, когда пользователь переходит к работе с новым окно

Метод onStop() вызывается, когда окно пропадает из видимости пользователя

Если окно возвращается в приоритетный режим после вызова onStop(), то в этом случае вызывается метод onRestart()

Метод onDestroy() вызывается по окончании работы активности

1. Как соотносится ЖЦ активности и сервиса?

И в ЖЦ активности и в ЖЦ сервиса при первом запуске мы создаем методом onCreate(), а уничтожаем методом onDestroy(). После создания в сервисе мы вызываем метод onStartCommand(), что означает что сервис переходит в активное состояние, а в активности вызывается метод onStart(), а далее при взаимодействии с пользователем onResume(), также ее можно поставить на паузу методом onPause(). В отличии от активности, сервисы могут быть связанные и несвязанные, тогда вместо onStartCommand(), вызывается метод onBind(), а впоследствии onUnbind(). Сервис может быть запущен как самостоятельная единица, а впоследствии может быть привязан к некоторым компонентам. В этом случае в сервисе должны быть обязательно реализованы оба метода onStartCommand() и onBind(). Сервисы существуют дольше активностей, до тех пор, пока ее не остановят или же она не остановится сама. Жизненный цикл сервиса намного проще жизненного цикла активности, однако он может быть осложнен тем, что сервис работает в фоновом режиме и пользователь может и не осознавать, что в некоторых случаях он имеет дело с работой сервисов.