

Práctica #3

Crear un nuevo proyecto.

Añadir la extensión de AndroidAnnotations.

Añadir los siguientes nuevos paquetes:

```
apps
beans
extras
```

Dentro del paquete apps añadir la siguiente clase:

```
@EApplication
public class NewApp extends Application {
}
```

Dentro del paquete beans añadir las siguientes clases.

Primero:

```
@EBean
public class FirstBean {

    @Bean
    SecondBean secondBean;

    @App
    NewApp newApp;

    public FirstBean() {
        System.out.println("}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}: FirstBean::se
        System.out.println("}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}: FirstBean::se
    }

    // https://github.com/androidannotations/androidannotations/wiki
    @AfterInject
    public void initAfterInjection() {
        System.out.println("+++++++: FirstBean::in
        System.out.println("+++++++: FirstBean::se
    }

}
```

Poniendo un log en el constructor se visualizará que las dos variables no están inicializadas.

Posteriormente, cuando pase por el método "initAfterInjection" se verá que sendas

variables ya están inicializadas.

Segundo:

```
@EBean
public class SecondBean {
}
```

En el package de extras añadir la siguiente clase:

```
@EActivity(R.layout.activity_main)
public class SecondActivity extends Activity {

    @Extra
    String message;

    @Bean
    FirstBean firstBean;

    @App
    NewApp newApp;

    @ViewById(R.id.tv_hello_world)
    TextView tvHelloWorld;

    public SecondActivity() {
        System.out.println("*****: SecondActivit
        System.out.println("*****: SecondActivit
        System.out.println("*****: SecondActivit
        System.out.println("*****: SecondActivit
        System.out.println("*****: SecondActivit
    }

    // https://github.com/androidannotations/androidannotations/wiki
    @AfterExtras
    public void initAfterExtrasInjection() {
        System.out.println("*****: SecondActivit
        System.out.println("*****: SecondActivit
        System.out.println("*****: SecondActivit
        System.out.println("*****: SecondActivit
        System.out.println("*****: SecondActivit
    }

    // https://github.com/androidannotations/androidannotations/wiki
    @AfterInject
    public void initAfterInjection() {
        System.out.println("+++++: SecondActivit
        System.out.println("+++++: SecondActivit
        System.out.println("+++++: SecondActivit
```

```

        System.out.println("+++++: SecondActivit
        System.out.println("+++++: SecondActivit
    }

    // https://github.com/androidannotations/androidannotations/wiki
    @AfterViews
    void initAfterViewInjection() {
        System.out.println("-----: SecondActivit
        System.out.println("-----: SecondActivit
        System.out.println("-----: SecondActivit
        System.out.println("-----: SecondActivit
        System.out.println("-----: SecondActivit
    }
}

```

Con esta clase podremos ver que en el constructor no hay nada inicializado. Posteriormente, desde la anotación "AfterExtras" estarán todos los objetos ya inicializados salvo los de UI que quedará inicializado después "AfterViews".

Finalmente en la clase "MainActivity" poner las anotaciones @AfterExtras a un método, @AfterInject a un método y @AfterViews a otro método. Adicionalmente poner sobreescritos los siguientes métodos:

```

protected void onCreate (Bundle savedInstanceState)
protected void onStart()
protected void onResume()

```

En cada uno poner un log y verificar cuál se ha lanzado primero.

Lanza la aplicación

El orden será de ejecución será:

- (1) @AfterExtras
- (2) @AfterInject
- (3) protected void onCreate (Bundle savedInstanceState)
- (4) @AfterViews
- (5) protected void onStart()
- (6) protected void onResume()

El orden de inicialización:

- **Constructor:** nunca hay nada inicializado
 - **@AfterExtras y @AfterInject:** está todo inicializado, menos objetos de UI
 - **@AfterViews:** Inicializa los objetos de UI