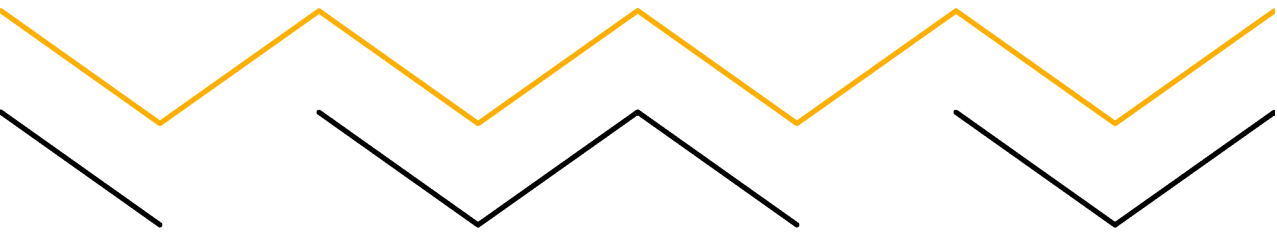


Разработка библиотек для мобильных приложений

Шкряба Ольга



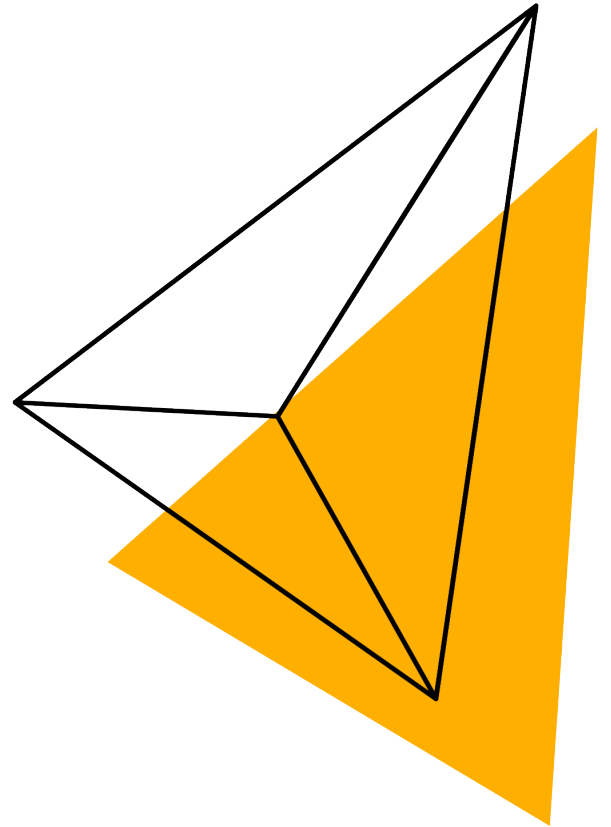
Ольга Шкряба

Занимаюсь разработкой SDK, любительскими исследованиями и мобильной безопасностью

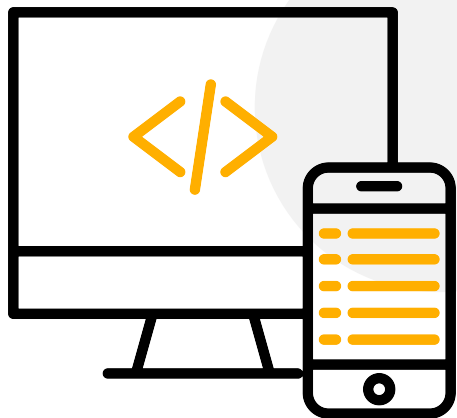
Для кого доклад...

Содержание

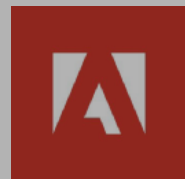
- Терминология
- Языки разработки
- Типы внедрения
- Минификация и обфускация
- Версионирование
- API
- Транзитивные зависимости
- Конфликты имен ресурсов
- Ошибки и работа с фичами



Библиотека vs SDK



Примеры SDK



Firebase

Когда нужно делать библиотеку

01

Вынести протестированный функционал для переиспользования

02

Решение задачи руками внешней команды без доступа к основному коду

03

Работа с посторонним API



Когда не нужно делать библиотеку

01

Не планируете использовать ее в нескольких приложениях

02

Уже есть стабильное решение в публичном доступе

03

Нет времени

Java ↔ Kotlin

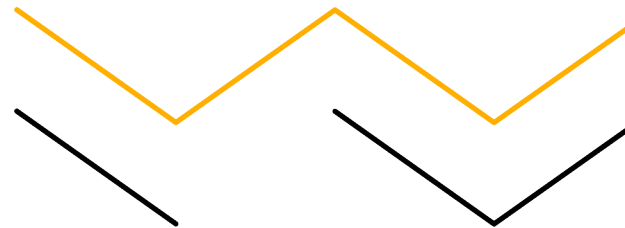
```
class Example {  
    private String tag;  
  
    Example(String tag) {  
        this.tag = tag;  
    }  
  
    public void setTag(String tag) {  
        this.tag = tag;  
    }  
  
    public String getTag() {  
        return tag;  
    }  
}
```



```
data class Example(var tag: String)
```

- Совместимость с Java
- Обратная совместимость
- Прямая совместимость
- std-lib, std-lib-jdk7, std-lib-jdk8

Прямая совместимость



```
tasks.withType(org.jetbrains.kotlin.gradle.tasks.KotlinCompile).all {  
    kotlinOptions {  
        apiVersion = "1.2"  
        languageVersion = "1.2"  
    }  
}
```



Obj-C ↔ Swift



```
@class  
MySwiftClass
```



```
#import  
ProductModuleName-Swift.h
```



- Обратная совместимость?
- libswiftCore версии(dylib)
- Совместимости с Obj-C

Виды зависимостей

01

Внешние
зависимости проекта

02

Проектные
зависимости

03

Файловые
зависимости

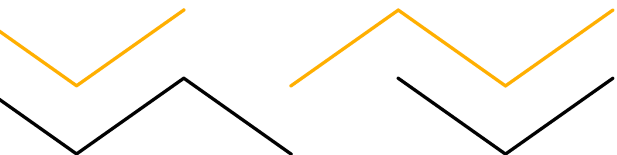
Тип внедрения



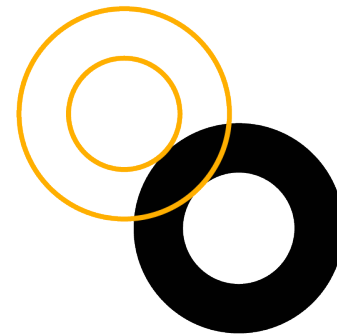
- implementation
- aar/jar



- pod install (and etc)
- framework



Расширения библиотек



Снаружи:

.jar/aar

.framework

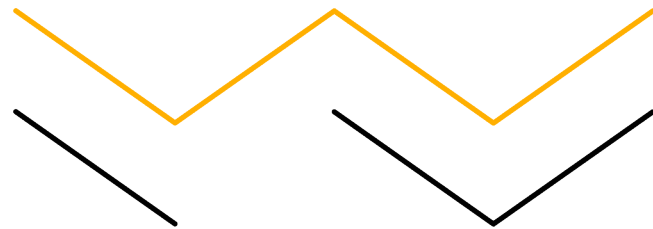
Расширения библиотек

Внутри:

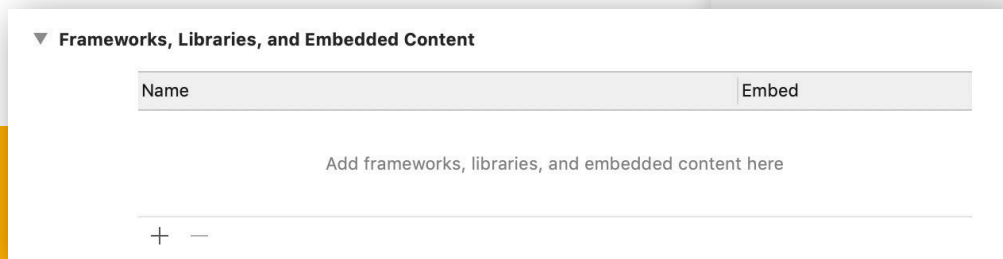
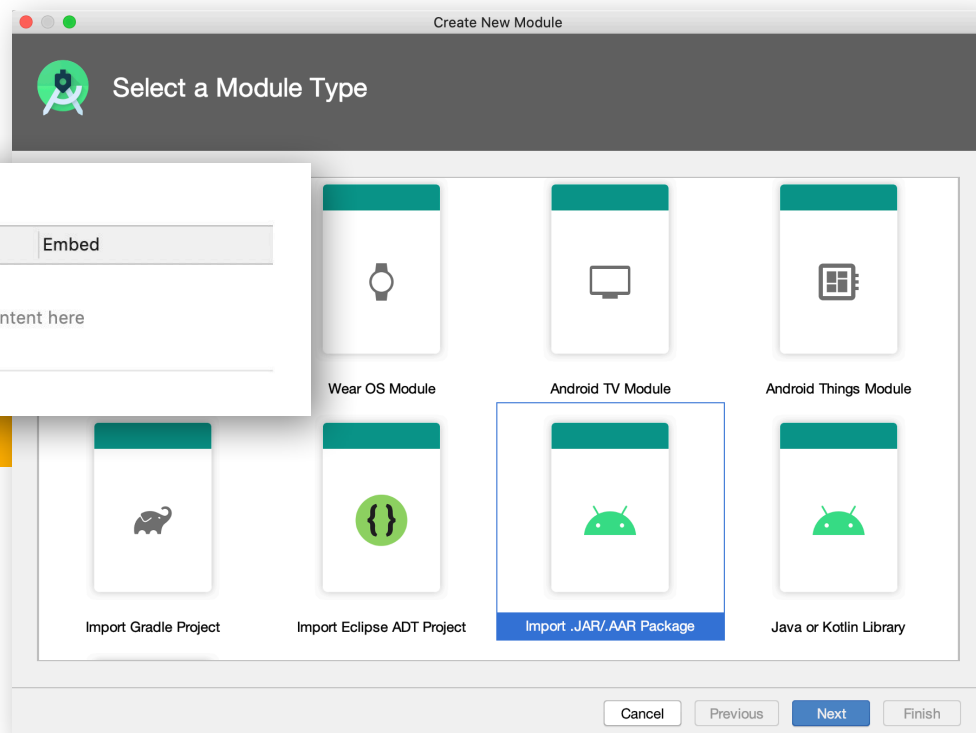
.dex

.so

.dylib



Добавление библиотек



MavenCentral vs jCenter

```
allprojects {  
    repositories {  
        mavenCentral()  
    }  
}
```

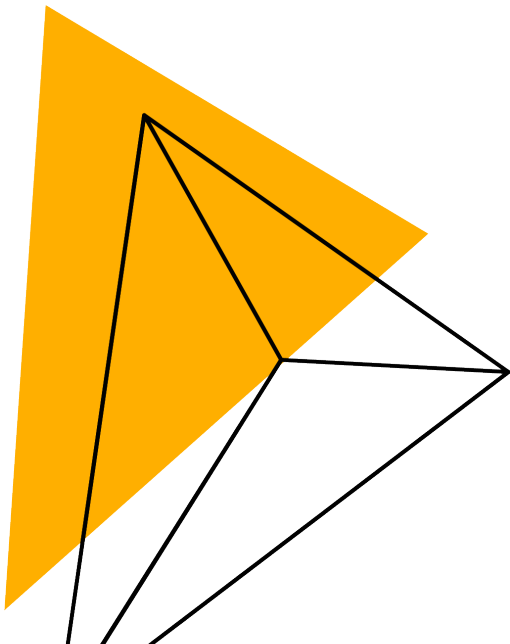
```
allprojects {  
    repositories {  
        jcenter()  
    }  
}
```

Другие репозитории

```
dependencies {  
    implementation "your.library"  
}
```

```
repositories {  
    maven { url 'https://your.library.com/public' }  
}
```

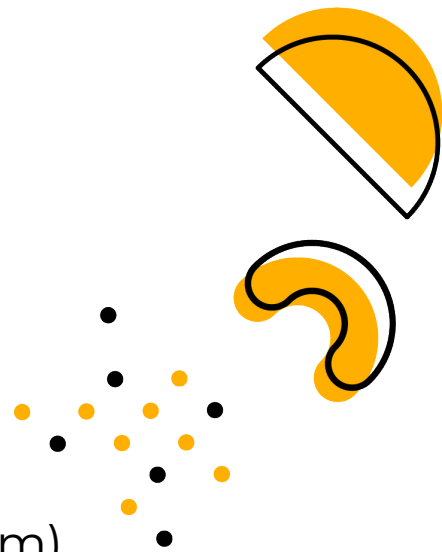
Доступы в библиотеке



- public/protected/private
- default packages
- internal
- proGuard

R8, ProGuard

- Аннотации
- Публичные классы, поля и методы
- Статические методы и перечисляемые типы (enum)
- Сериализованные классы
- Динамически создаваемые классы (Class.forName)



Версионирование библиотек

A – major release number

B – minor release number

C – bug fix number

D – build number



Планирование Library API

```
public interface LibraryController {  
    void setStartTime(long startTime);  
    void setTimeout(long timeout);  
    void setListener(LibraryListener listener);  
    void setUserId(String userId);  
    void setMessage(String message);  
}
```

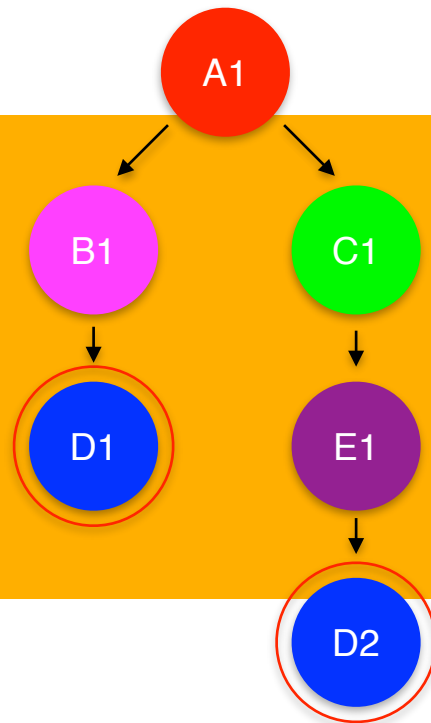
Asserts

```
fun startFunc(activity: Activity?) {  
    if (activity == null) {  
        throw Exception("activity is null")  
    }  
  
    // ...  
}
```

- Входящие аргументы
- Проверка данных в работе

Зависимости внутри библиотеки

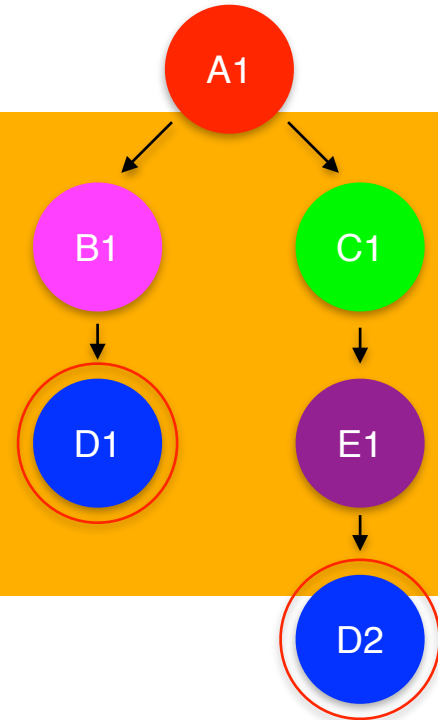
- Прямые
- Транзитивные



Разрешение конфликтов

★ build.gradle

```
dependencies {  
    implementation('commons-beanutils:commons-beanutils:1.9.4') {  
        exclude group: 'commons-collections', module: 'commons-collections'  
    }  
}
```



Решения

01

Не
использовать

02

Регулярное
обновление

03

Учитывать более
старых версий

Предотвращение конфликтов имен ресурсов

```
<string name="example_name">Application</string>  
<string name="example_name">Library</string>
```

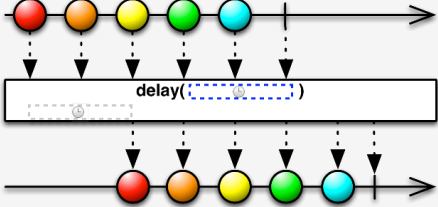
```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Library"/>
```

Предотвращение конфликтов имен ресурсов

```
<string name="example_name">Library</string>  
<string name="library_example_name">Library</string>
```

Документация

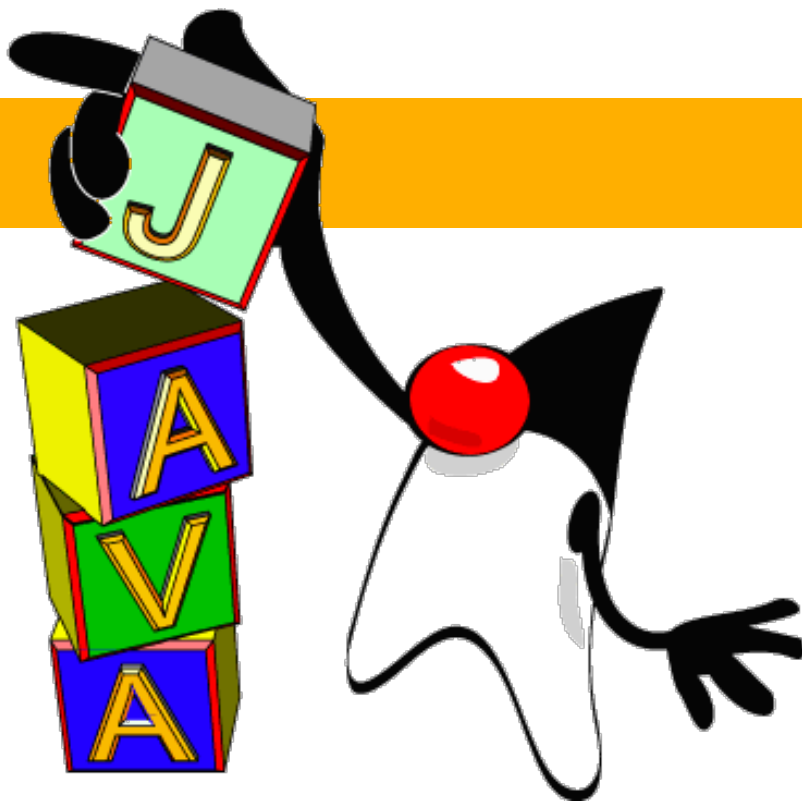
```
setStatusAllClients(CALL_STATE.CONNECTION_LOSS)
}
}.delay( delay: 1, TimeUnit.SECONDS)
}
io.reactivex.Observable<T>
.doOnE
if @CheckReturnValue
@SchedulerSupport(value="io.reactivex:computation")
public final Observable<T> delay(long delay,
TimeUnit unit)
}
Returns an Observable that emits the items emitted by the source ObservableSource shifted forward in time by a
specified delay. Error notifications from the source ObservableSource are not delayed.
Ti
}
.obser
.subsc
.flatM
co
se
```



- JavaDoc
- Dokka
- markdown

Теги JavaDoc

- @author
- @deprecated
- @inheritDoc
- @param
- @return
- @throws



Момент доверия

01

Кто
разработчик

02

Есть
ли тесты

03

Дата
последнего обновления

Выводы

- Архитектура
- Кто будет использовать библиотеку
- Для каких целей она будет написана
- Будет ли поддержка (как долго?)
- Время и ресурсы
- Тесты