Custom Lint Rules

Игорь Таланкин

Tinkoff.ru

О чем доклад?

- Статический анализ
- Android Lint
- Custom lint rules

Tinkoff.ru

Статический анализ

- стилистика
  - отступы, форматирование
- семантика
  - ошибки в логике, недостижимые участки кода
- подсчет метрик
  - Source Lines Of Code, количество комментариев, сложность

Tinkoff.ru

Android Lint

Tinkoff.ru

Android Lint

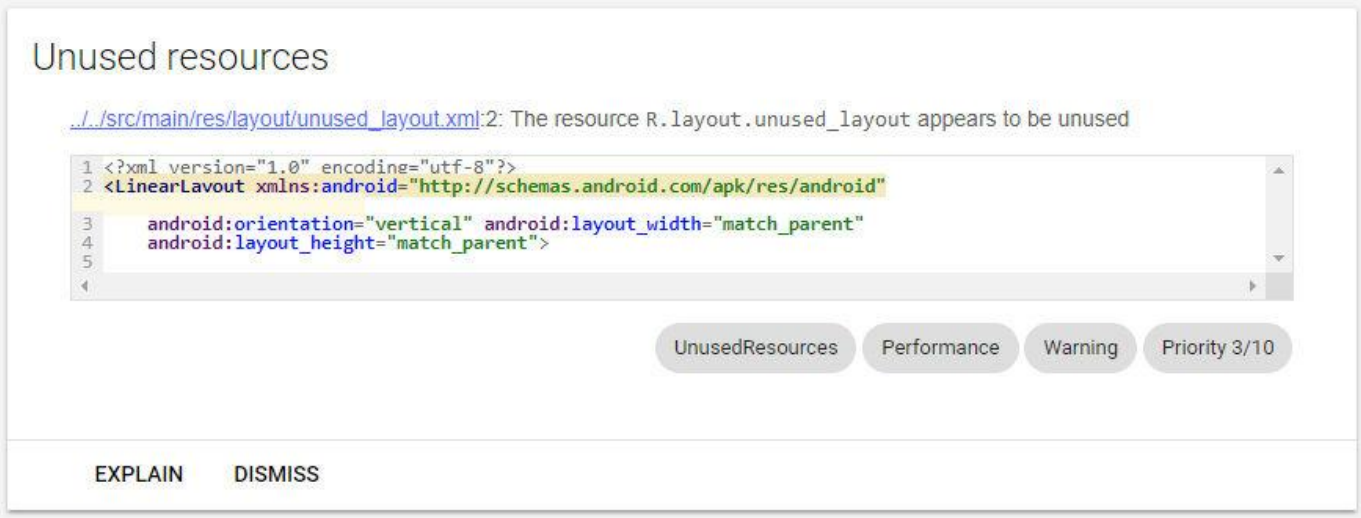
- статический анализатор кода
- работает в IDE из коробки
- мультидисциплинарный
  - XML, Java, Kotlin, JPEG, Properties, etc.
- можно поставлять вместе с библиотеками

Tinkoff.ru

```
$ ./gradlew lint

Ran lint on variant debug: 2 issues found
Ran lint on variant release: 1 issues found
Wrote HTML report to file:///.../build/reports/lint-results.html
Wrote XML report to file:///.../build/reports/lint-results.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<issues format="5" by="lint 3.3.1">
    <issue
        id="UnusedResources"
        severity="Warning"
        message="`R.layout.unused_layout` appears to be unused"
        category="Performance"
        priority="3"
        summary="Unused resources"
        explanation="..."
        errorLine1="&lt;LinearLayout"
        errorLine2="^">
        <location
            file="/.../unused_layout.xml"
            line="2"
            column="1"/>
    </issue>
</issues>
```

HTML



Tinkoff.ru

IDE

```
private void showErrorToast(CharSequence message) {
    Toast.makeText( context: this, message, Toast.LENGTH_SHOR
}
```

Toast created but not shown: did you forget to call show( ) ? more... (Ctrl+F1)

Tinkoff.ru

```
android {
    lintOptions {
        baseline file('baseline.xml')
        checkDependencies true
    }
}
```

```
android {
    lintOptions {
        baseline file('baseline.xml')
        checkDependencies true
    }
}
```

```
android {
    lintOptions {
        baseline file('baseline.xml')
        checkDependencies true
    }
}
```

Custom Lint Rules

Tinkoff.ru

Custom Lint Rules: зачем?

- находить ошибки

Tinkoff.ru

Custom Lint Rules: зачем?

Gson (de)serialization issues caused by ProGuard

../../src/main/java/ru/tinkoff/sme/ContractRecord.kt:10: May be lost during Gson deserialization

```
 7 data class ContractRecord(
 8     @SerializedName("id")
 9     var id: String,
10     var contractNumber: String?,
11     @SerializedName("contractDate")
12     var contractDate: Date,
13     @SerializedName("contractRecordNumber")
```

../../src/main/java/ru/tinkoff/sme/ContractRecordApi.kt:26: Used here

```
23 interface ContractRecordApi {
24
25     @GET("getContractRecords")
26     fun getContractRecords(): Single<List<ContractRecord>>
27
28
```

SerializedNameDetector | SME | Warning | Priority 5/10

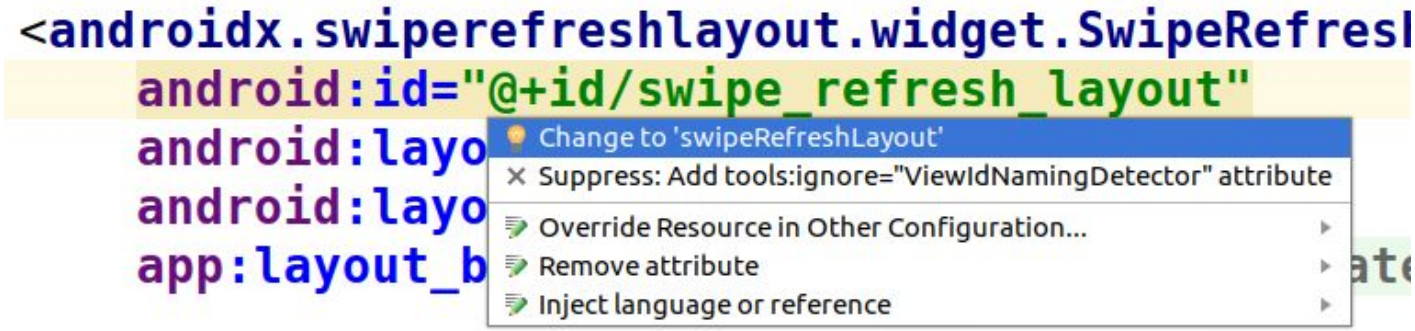
Tinkoff.ru

Custom Lint Rules: зачем?

- находить ошибки
- контролировать соблюдение соглашений

Tinkoff.ru

Custom Lint Rules: зачем?



Tinkoff.ru

Custom Lint Rules: зачем?

- находить ошибки
- контролировать соблюдение соглашений
- избежать рутинных и повторяющихся действий

Tinkoff.ru

Custom Lint Rules: зачем?

- находить ошибки
- контролировать соблюдение соглашений
- избежать рутинных и повторяющихся действий
- рефакторить

Tinkoff.ru

Custom Lint Rules

1. Добавить модуль

Tinkoff.ru

```
// custom-lint-rules/build.gradle
apply plugin: 'kotlin'

dependencies {
    compileOnly "com.android.tools.lint:lint-api:$lintVersion"
    testCompile "com.android.tools.lint:lint-tests:$lintVersion"
}

jar {
  manifest {
    attributes("Lint-Registry-v2": "ru.tinkoff.CustomIssueRegistry")
  }
}
```

```
// custom-lint-rules/build.gradle
apply plugin: 'kotlin'

dependencies {
    compileOnly "com.android.tools.lint:lint-api:$lintVersion"
    testCompile "com.android.tools.lint:lint-tests:$lintVersion"
}

jar {
  manifest {
    attributes("Lint-Registry-v2": "ru.tinkoff.CustomIssueRegistry")
  }
}
```

```
lintVersion = androidGradlePluginVersion + 23.0.0

// например
androidGradlePluginVersion = "3.5.0"
lintVersion = "26.5.0"
```

```
// custom-lint-rules/build.gradle
apply plugin: 'kotlin'

dependencies {
    compileOnly "com.android.tools.lint:lint-api:$lintVersion"
    testCompile "com.android.tools.lint:lint-tests:$lintVersion"
}

jar {
  manifest {
    attributes("Lint-Registry-v2": "ru.tinkoff.CustomIssueRegistry")
  }
}
```

Custom Lint Rules

1. Добавить модуль
2. Реализовать IssueRegistry

Tinkoff.ru

```
class CustomIssueRegistry : IssueRegistry() {
    override val api = CURRENT_API

    override val issues: List<Issue> = listOf()
}
```

```
class CustomIssueRegistry : IssueRegistry() {
    override val api = CURRENT_API

    override val issues: List<Issue> = listOf()
}
```

```
import com.android.tools.lint.detector.api.CURRENT_API
```

```
class CustomIssueRegistry : IssueRegistry() {
    override val api = CURRENT_API

    override val issues: List<Issue> = listOf()
}
```

Custom Lint Rules

1. Добавить модуль
2. Реализовать IssueRegistry
3. Подключить модуль к проекту

Tinkoff.ru

```
// app/build.gradle
dependencies {
    lintChecks project(':custom-lint-rules')
}
```

```
// some-kotlin-module/build.gradle
apply plugin: 'kotlin'
apply plugin: 'com.android.lint'

dependencies {
    lintChecks project(':custom-lint-rules');
}
```

Custom Lint Rules

1. Добавить модуль
2. Реализовать IssueRegistry
3. Подключить модуль к проекту
4. Реализовать Detector

Tinkoff.ru

Пример

Tinkoff.ru

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE) {
        CropImage.ActivityResult result = CropImage.getActivityResult(data);
        if (requestCode == Activity.RESULT_OK) {
            Uri resultUri = result.getUri();
            try {
                Bitmap bitmap = getBitmap(getContentResolver(), resultUri);
                profile_image.setImageBitmap(bitmap);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE) {
        CropImage.ActivityResult result = CropImage.getActivityResult(data);
        if (requestCode == Activity.RESULT_OK) {
            Uri resultUri = result.getUri();
            try {
                Bitmap bitmap = getBitmap(getContentResolver(), resultUri);
                profile_image.setImageBitmap(bitmap);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Интерфейсы

- `XmlScanner`
- `SourceCodeScanner`
- `ClassScanner`
- `GradleScanner`
- `BinaryResourceScanner`
- `ResourceFolderScanner`, `OtherFileScanner`, ...

Tinkoff.ru

SourceCodeScanner

- вызовы методов с именем `methodName`
- обращение к символам с именем `symbolName`
- наследование от класса `ClassName`
- использование аннотаций `AnnotationName`
- любые другие конструкции

Tinkoff.ru

UAST

- Unified Abstract Syntax Tree
- универсальный анализ Java и Kotlin

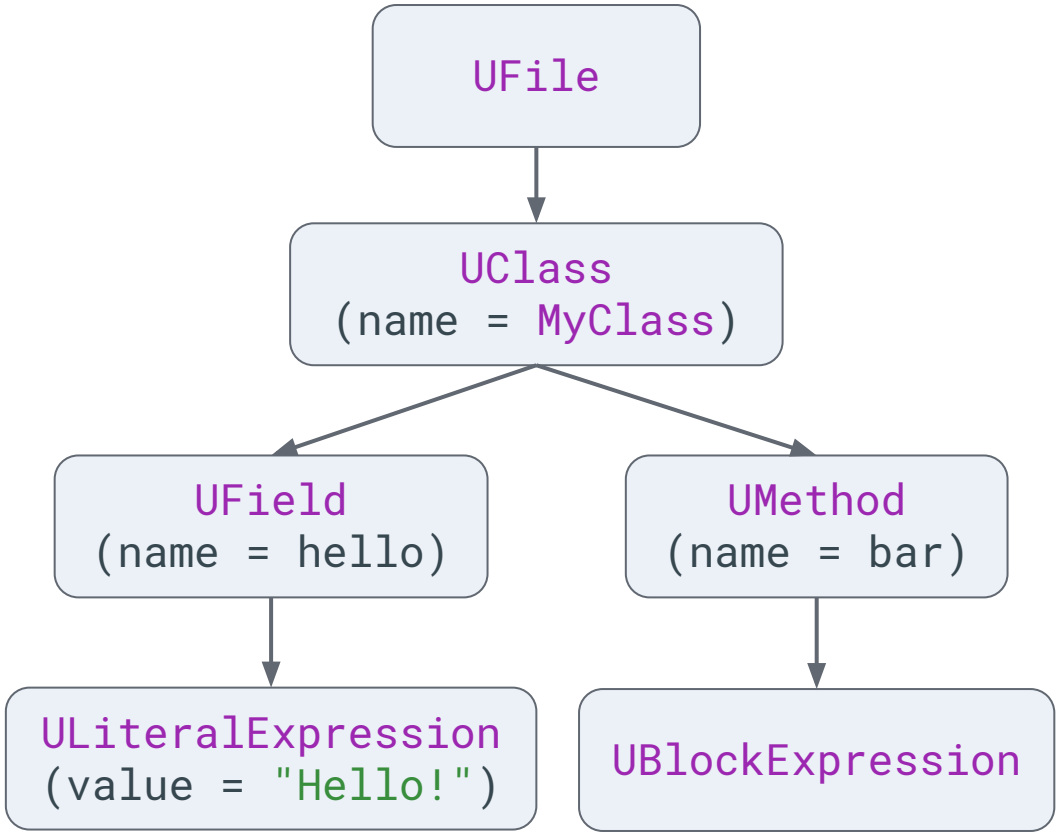
Tinkoff.ru

```java
// MyClass.java
public class MyClass {
    String hello = "Hello!";
    void bar() { }
}
```

```kotlin
// MyClass.kt
class MyClass {
    val hello = "Hello!"
    fun bar() { }
}
```

```java
// MyClass.java
public class MyClass {
    String hello = "Hello!";
    void bar() { }
}
```

```kotlin
// MyClass.kt
class MyClass {
    val hello = "Hello!"
    fun bar() { }
}
```



Алгоритм

1. Найти использование констант `RESULT_OK`, `RESULT_CANCELLED`
2. Проверить второй операнд ==
3. Если это `requestCode`, то сообщить об ошибке

Tinkoff.ru

```
class ActivityResultDetector: Detector(), SourceCodeScanner {
    companion object {
        val ISSUE = Issue.create(...)
    }

    override fun getApplicableReferenceNames(): List<String>?

    override fun visitReference(
        context: JavaContext,
        reference: UReferenceExpression,
        referenced: PsiElement)
}
```

```
class ActivityResultDetector: Detector(), SourceCodeScanner {
    companion object {
        val ISSUE = Issue.create(...)
    }

    override fun getApplicableReferenceNames(): List<String>?

    override fun visitReference(
        context: JavaContext,
        reference: UReferenceExpression,
        referenced: PsiElement)
}
```

```
val ISSUE = Issue.create(
    id = "ActivityResultDetector",
    briefDescription = "Comparing result code to request code",
    explanation = "<...>",
    category = Category.CORRECTNESS,
    priority = 5,
    severity = Severity.ERROR,
    implementation = Implementation(
            ActivityResultDetector::class.java,
            Scope.JAVA_FILE_SCOPE)
)
```

```
val ISSUE = Issue.create(
        id = "ActivityResultDetector",
        briefDescription = "Comparing result code to request code",
        explanation = "<...>",
        category = Category.CORRECTNESS,
        priority = 5,
        severity = Severity.ERROR,
        implementation = Implementation(
                ActivityResultDetector::class.java,
                Scope.JAVA_FILE_SCOPE)
    )
```

```
val ISSUE = Issue.create(
        id = "ActivityResultDetector",
        briefDescription = "Comparing result code to request code",
        explanation = "<...>",
        category = Category.CORRECTNESS,
        priority = 5,
        severity = Severity.ERROR,
        implementation = Implementation(
                ActivityResultDetector::class.java,
                Scope.JAVA_FILE_SCOPE)
    )
```

```
val ISSUE = Issue.create(
        id = "ActivityResultDetector",
        briefDescription = "Comparing result code to request code",
        explanation = "<...>",
        category = Category.CORRECTNESS,
        priority = 5,
        severity = Severity.ERROR,
        implementation = Implementation(
                ActivityResultDetector::class.java,
                Scope.JAVA_FILE_SCOPE)
    )
```

```
val ISSUE = Issue.create(
        id = "ActivityResultDetector",
        briefDescription = "Comparing result code to request code",
        explanation = "<...>",
        category = Category.CORRECTNESS,
        priority = 5,
        severity = Severity.ERROR,
        implementation = Implementation(
                ActivityResultDetector::class.java,
                Scope.JAVA_FILE_SCOPE)
    )
```

```
enum class Severity {

    FATAL("Fatal"),

    ERROR("Error"),

    WARNING("Warning"),

    INFORMATIONAL("Information"),

    IGNORE("Ignore")
}
```

```
val ISSUE = Issue.create(
        id = "ActivityResultDetector",
        briefDescription = "Comparing result code to request code",
        explanation = "<...>",
        category = Category.CORRECTNESS,
        priority = 5,
        severity = Severity.ERROR,
        implementation = Implementation(
                ActivityResultDetector::class.java,
                Scope.JAVA_FILE_SCOPE)
    )
```

```
enum class Scope {
    RESOURCE_FILE,
    JAVA_FILE,
    CLASS_FILE,
    MANIFEST,
    PROGUARD_FILE,
    GRADLE_FILE,
    ...
}
```

```
class ActivityResultDetector : Detector(), SourceCodeScanner {

    override fun getApplicableReferenceNames(): List<String>?

    override fun visitReference(
        context: JavaContext,
        reference: UReferenceExpression,
        referenced: PsiElement)
}
```

```
class ActivityResultDetector : Detector(), SourceCodeScanner {

    override fun getApplicableReferenceNames(): List<String>?

    override fun visitReference(
        context: JavaContext,
        reference: UReferenceExpression,
        referenced: PsiElement)
}
```

```
override fun getApplicableReferenceNames(): List<String>? {
    return listOf("RESULT_OK", "RESULT_CANCELED")
}
```

```
override fun visitReference(
        context: JavaContext,
        reference: UReferenceExpression,
        referenced: PsiElement) {
    // TODO
}
```

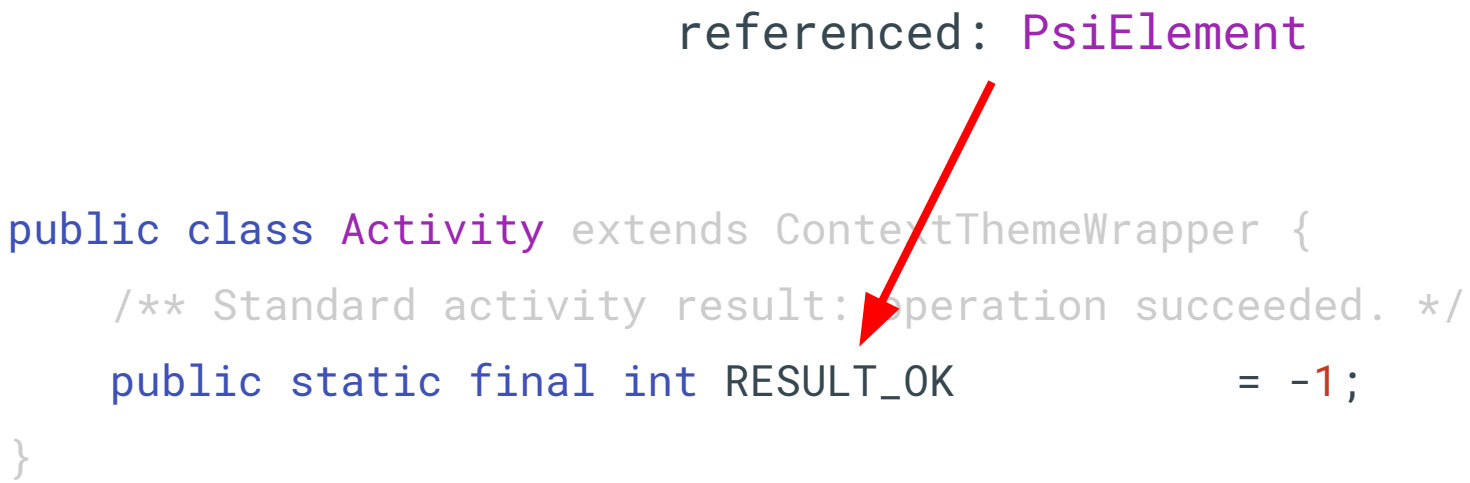
```
override fun visitReference(
        context: JavaContext,
        reference: UReferenceExpression,
        referenced: PsiElement) {
    // TODO
}
```

```
override fun visitReference(
        context: JavaContext,
        reference: UReferenceExpression,
        referenced: PsiElement) {
    // TODO
}
```

reference: UReferenceExpression

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    // ...
    if (requestCode == Activity.RESULT_OK) {
        Uri resultUri = result.getUri();
        // ...
    }
}
```

```
override fun visitReference(
        context: JavaContext,
        reference: UReferenceExpression,
        referenced: PsiElement) {
    // TODO
}
```


referenced: PsiElement
public class Activity extends ContextThemeWrapper {
/** Standard activity result: operation succeeded. */
public static final int RESULT_OK = -1;
}

PSI

- Program Structure Interface
- используются для представления внутренней структуры исходного кода

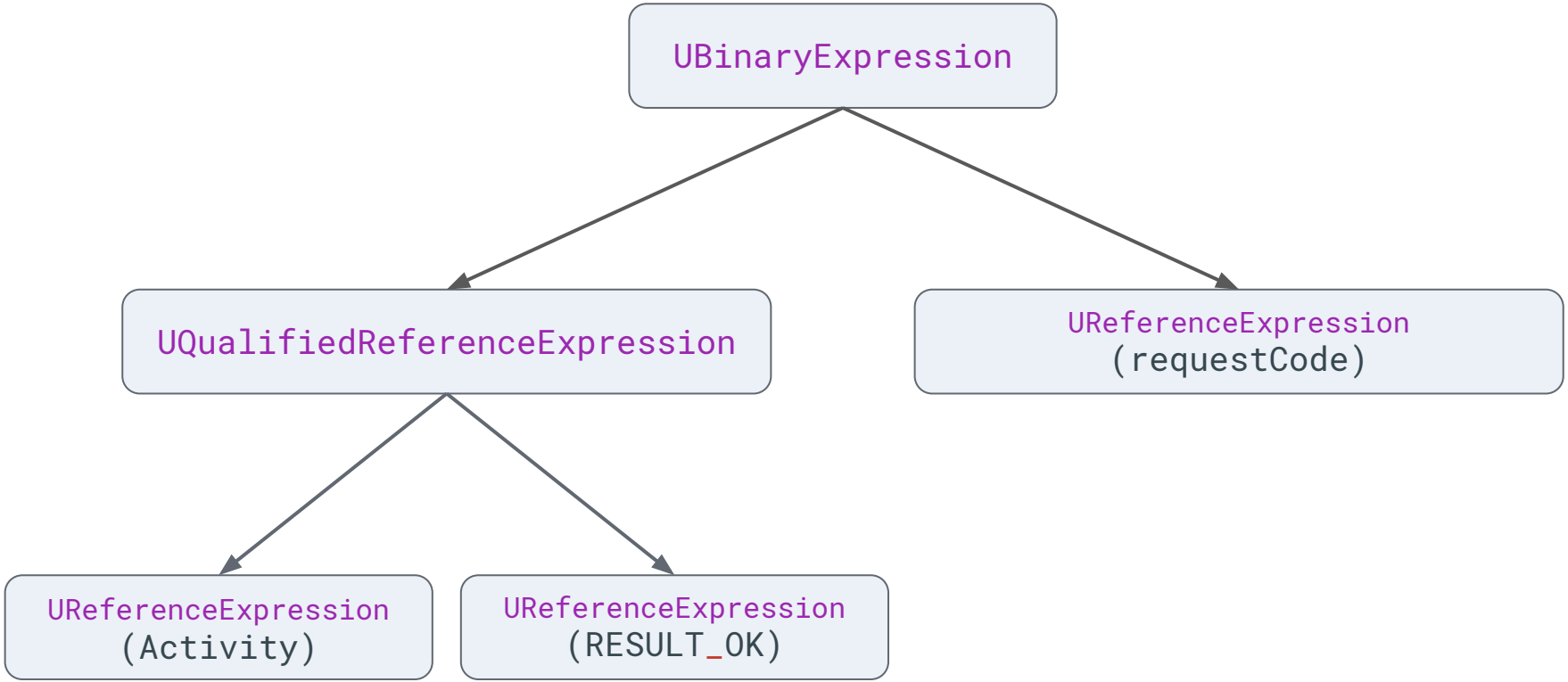
Tinkoff.ru

UAST vs PSI

- UAST строится на основе PSI
- UAST используется внутри тела методов

Tinkoff.ru

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    // ...
    if (requestCode == Activity.RESULT_OK) {
        Uri resultUri = result.getUri();
        // ...
    }
}
```


UBinaryExpression
UQualifiedReferenceExpression
UReferenceExpression
(requestCode)
UReferenceExpression
(Activity)
UReferenceExpression
(RESULT_OK)

reference: UReferenceExpression

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    // ...
    if (requestCode == Activity.RESULT_OK) {
        Uri resultUri = result.getUri();
        // ...
    }
}
```

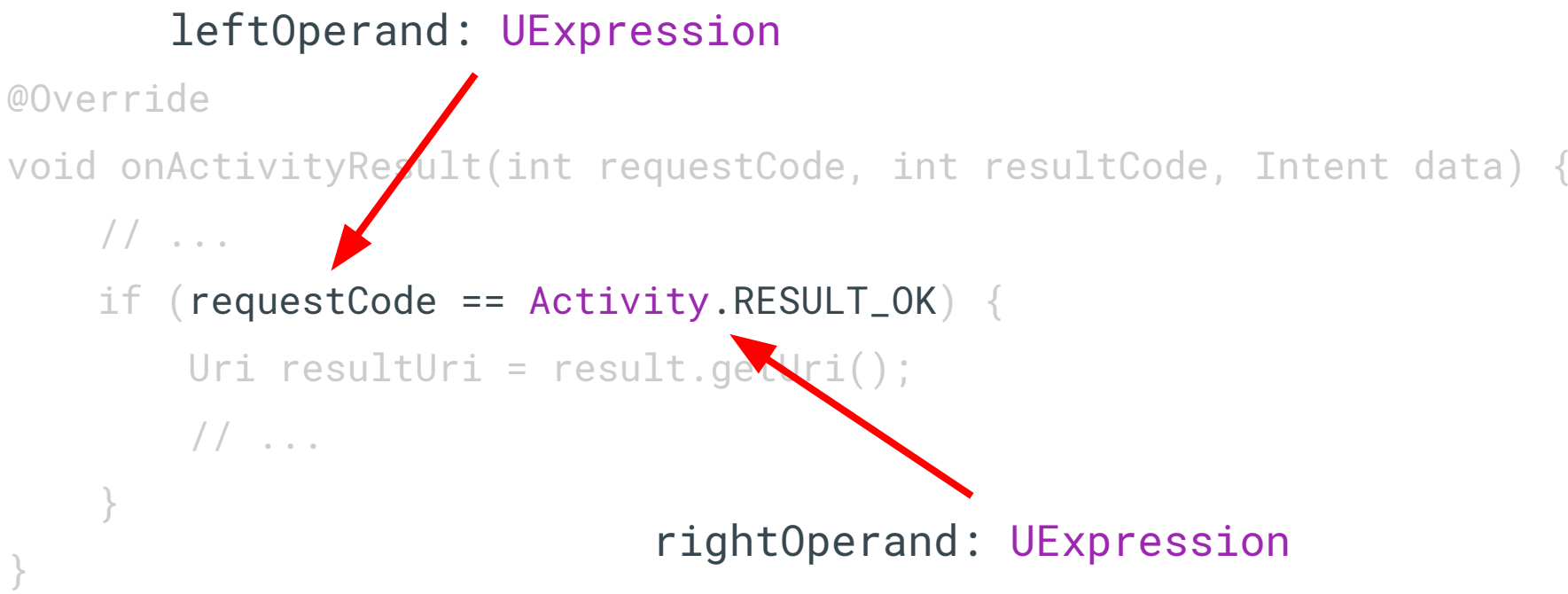
UQualifiedReferenceExpression

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    // ...
    if (requestCode == Activity.RESULT_OK) {
        Uri resultUri = result.getUri();
        // ...
    }
}
```

UBinaryExpression

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    // ...
    if (requestCode == Activity.RESULT_OK) {
        Uri resultUri = result.getUri();
        // ...
    }
}
```

```
val comparison: UBinaryExpression =
    reference.getParentOfType(UBinaryExpression::class.java)
```


leftOperand: UExpression
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
// ...
if (requestCode == Activity.RESULT_OK) {
Uri resultUri = result.getUri();
// ...
}
}
rightOperand: UExpression

```
val operand =
        comparison.leftOperand as? UReferenceExpression
```

method: UMethod

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    // ...
    if (requestCode == Activity.RESULT_OK) {
        Uri resultUri = result.getUri();
        // ...
    }
}
```

```
method.parameterList.parameters[0]
```

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    // ...
    if (requestCode == Activity.RESULT_OK) {
        Uri resultUri = result.getUri();
        // ...
    }
}
```

```
val method: UMethod? =
        reference.getParentOfType(UMethod::class.java)

val paramRequestCode = method.parameterList.parameters[0]
```

```
if (operand.resolve() == paramRequestCode) {
    context.report(
        issue = ActivityResultDetector.ISSUE,
        location = context.getLocation(comparison),
        message = "Error!"
    )
}
```

```
if (operand.resolve() == paramRequestCode) {
    context.report(
        issue = ActivityResultDetector.ISSUE,
        location = context.getLocation(comparison),
        message = "Error!"
    )
}
```

operand: UReferenceExpression

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == Activity.RESULT_OK) {
        Uri uri = data.getData();
        // ...
    }
}
```

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == Activity.RESULT_OK) {
        Uri uri = data.getData();
        // ...
    }
}
```

resolve()

```
if (operand.resolve() == paramRequestCode) {
    context.report(
        issue = ActivityResultDetector.ISSUE,
        location = context.getLocation(comparison),
        message = "Error!"
    )
}
```

```
if (operand.resolve() == paramRequestCode) {
    context.report(
        issue = ActivityResultDetector.ISSUE,
        location = context.getLocation(comparison),
        message = "Error!"
    )
}
```

```
class CustomIssueRegistry : IssueRegistry() {
    override val api = CURRENT_API

    override val issues: List<Issue> = listOf(
        ActivityResultDetector.ISSUE
    )
}
```

```
class CustomIssueRegistry : IssueRegistry() {
    override val api = CURRENT_API

    override val issues: List<Issue> = listOf(
        ActivityResultDetector.ISSUE
    )
}
```

HTML

Result code compared to request code

../../src/main/java/MyActivity.java:20: `requestCode` should not be compared with `RESULT_OK`

```
17          super.onActivityResult(requestCode, resultCode, data);
18          if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE) {
19              CropImage.ActivityResult result = CropImage.getActivityResult(data);
20              if (requestCode == Activity.RESULT_OK) {
21                  Uri resultUri = result.getUri();
22                  try {
23                      Bitmap bitmap = getBitmap(getContentResolver(), resultUri);
```

ActivityResultDetector | Correctness | Error | Priority 5/10

Tinkoff.ru

IDE

```
@Override
protected void onActivityResult(int requestCode, int resultCode, In
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE)
        CropImage.ActivityResult result = CropImage.getActivityResu
        if (requestCode == Activity.RESULT_OK) {
            Uri resultUri = result.getUri();
                                                                  res
            profile_image.setImageBitmap(bitmap);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

requestCode should not be compared with RESULT_OK more... (Ctrl+F1)

Tinkoff.ru

Проблема

Tinkoff.ru

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    // ...
    if (Activity.RESULT_OK == requestCode) {
        Uri resultUri = result.getUri();
        // ...
    }
}
```

```
@Override
void onActivityResult(int requestCode, int resultCode, Intent data) {
    // ...
    if (Activity.RESULT_OK == requestCode) {
        Uri resultUri = result.getUri();
        // ...
    }
}
```

```
val operand =
        comparison.leftOperand as? UReferenceExpression
```

```
val operand: UReferenceExpression? =
        comparison.leftOperand as? UReferenceExpression
        getAnotherOperand(comparison, reference)
```

```
private fun getAnotherOperand(
    comparison: UBinaryExpression,
    referenceToField: UReferenceExpression
): UReferenceExpression? {
    return if (comparison.leftOperand == referenceToField) {
        comparison.rightOperand
    } else {
        comparison.leftOperand
    } as? UReferenceExpression
}
```

IDE

```
@Override
protected void onActivityResult(int requestCode, int resultC
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST
        CropImage.ActivityResult result = CropImage.getActiv
        if (Activity.RESULT_OK == requestCode) {
            Uri resultUri = result.getUri();
            Bitmap bitmap = getBitmap(getContentResolver
            profile_image.setImageBitmap(bitmap);
        } catch (IOException e) {
```

requestCode should not be compared with RESULT_OK more... (Ctrl+F1)

Tinkoff.ru

Проблема №2

Tinkoff.ru

```
fun checkActivityResult(resultCode: Int) {
    if (resultCode == Activity.RESULT_OK) {
        // ...
    }
}
```

```
val method = reference.getParentOfType(UMethod::class.java)

if (!isOnActivityResult(context, method)) {
    return
}
```

```
fun isOnActivityResult(
    context: JavaContext,
    method: UMethod
): Boolean {
    return context.evaluator.methodMatches(
        method,
        "android.app.Activity",
        false,
        "int", "int", "android.content.Intent"
    )
}
```

Отладка

Tinkoff.ru

```
fun UElement.asRecursiveLogString(): String
```

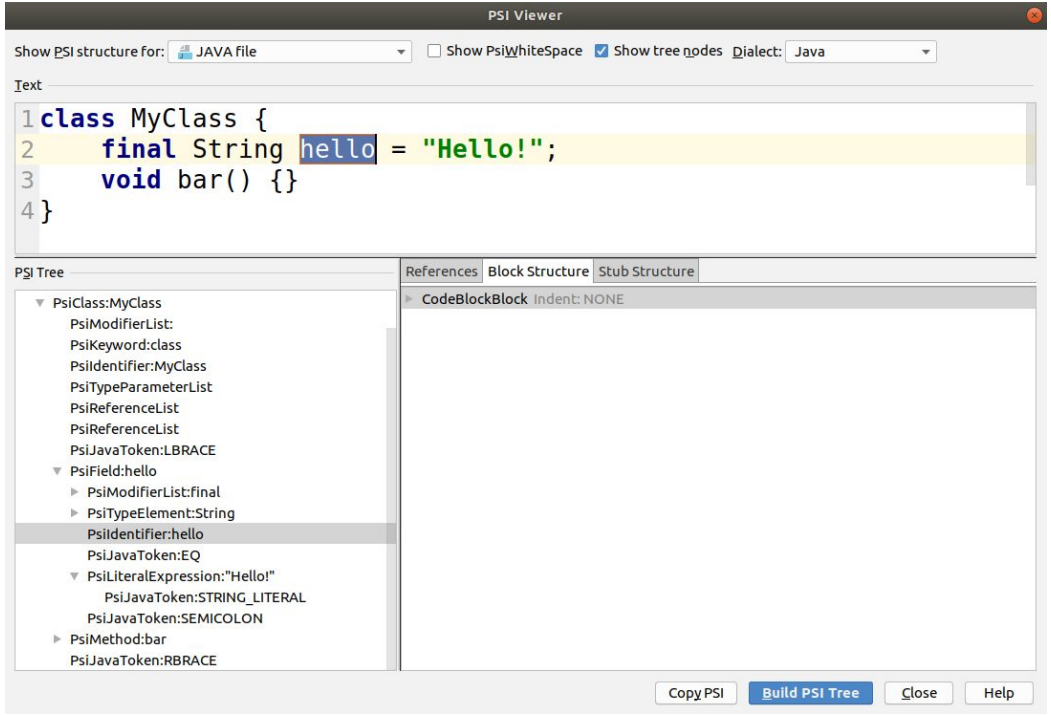
```
UMethod (name = onActivityResult)
    UAnnotation (fqName = java.lang.Override)
    UParameter (name = requestCode)
    UParameter (name = resultCode)
    UParameter (name = data)
    UBlockExpression
        UIfExpression
            UBinaryExpression (operator = ===)
                USimpleNameReferenceExpression (id = requestCode)
                UQualifiedReferenceExpression
                    USimpleNameReferenceExpression (id = Activity)
                    USimpleNameReferenceExpression (id = RESULT_OK)
```

PsiViewer plugin

Tinkoff.ru

```
# ~/.AndroidStudio3.x/config/idea.properties

idea.is.internal=true
```

View PSI structure



Tinkoff.ru

Отладка

- `-Dorg.gradle.debug=true`

Tinkoff.ru

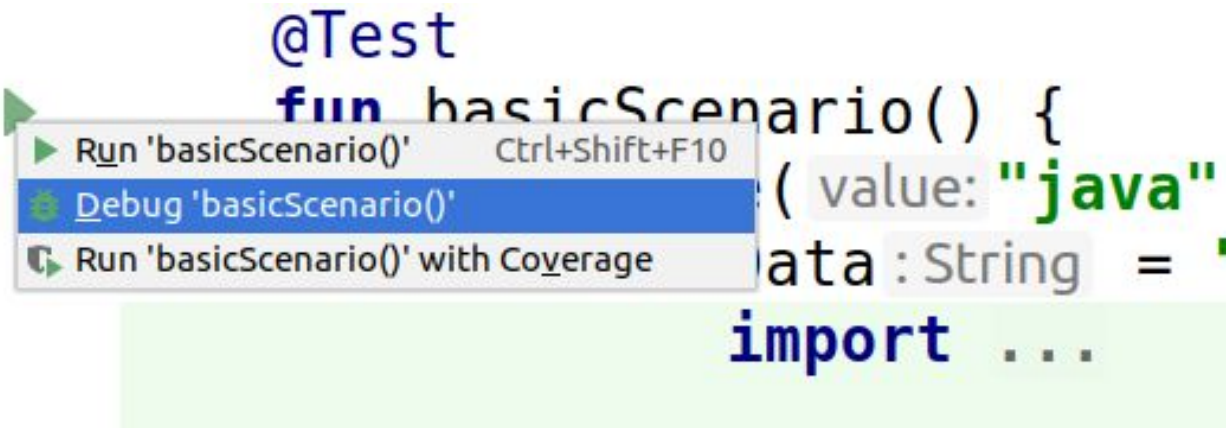
```
$ ./gradle lintDebug \
        -Dorg.gradle.debug=true \
        --no-daemon
```

Отладка

- `-Dorg.gradle.debug=true`
- `Run → Attach to Process`

Tinkoff.ru

Отладка



Tinkoff.ru

Utilities

Tinkoff.ru

SdkConstants

```
/* Android Class Constants */
public static final String CLASS_ACTIVITY = "android.app.Activity"; //$NON-NLS-1$
public static final String CLASS_APPLICATION = "android.app.Application"; //$NON-NLS-1$
public static final String CLASS_SERVICE = "android.app.Service"; //$NON-NLS-1$
public static final String CLASS_BROADCASTRECEIVER =
        "android.content.BroadcastReceiver"; //$NON-NLS-1$
public static final String CLASS_CONTENTPROVIDER =
        "android.content.ContentProvider"; //$NON-NLS-1$
public static final String CLASS_ATTRIBUTE_SET = "android.util.AttributeSet"; //$NON-NLS-1$
public static final String CLASS_INSTRUMENTATION = "android.app.Instrumentation"; //$NON-NLS-1$
public static final String CLASS_INSTRUMENTATION_RUNNER =
        "android.test.InstrumentationTestRunner"; //$NON-NLS-1$
public static final String CLASS_BUNDLE = "android.os.Bundle"; //$NON-NLS-1$
public static final String CLASS_R = "android.R"; //$NON-NLS-1$
public static final String CLASS_R_PREFIX = CLASS_R + "."; //$NON-NLS-1$
public static final String CLASS_MANIFEST_PERMISSION =
        "android.Manifest$permission"; //$NON-NLS-1$
public static final String CLASS_INTENT = "android.content.Intent"; //$NON-NLS-1$
public static final String CLASS_CONTEXT = "android.content.Context"; //$NON-NLS-1$
public static final String CLASS_RESOURCES = "android.content.res.Resources"; //$NON-NLS-1$
public static final String CLS_TYPED_ARRAY = "android.content.res.TypedArray"; //$NON-NLS-1$
public static final String CLASS_VIEW = "android.view.View"; //$NON-NLS-1$
public static final String CLASS_VIEWGROUP = "android.view.ViewGroup"; //$NON-NLS-1$
public static final String CLASS_NAME_LAYOUTPARAMS = "LayoutParams"; //$NON-NLS-1$
```

Tinkoff.ru

LintUtils

- `isKotlin, isJava`
- `isJavaKeyword, isKotlinHardKeyword`
- `isAnonymousClass, isStaticInnerClass, ...`
- `getMethodName`

Tinkoff.ru

UastLintUtils

- containsAnnotation
- getAnnotationValue
- isZero, isNumber, isMinusOne
- findLastAssignment
- getClassName, getQualifiedName

Tinkoff.ru

Extensions

- `toUElement, toUElementOfType`
- `getParentOfType`
- `findContainingUClass`
- `evaluateString`

Tinkoff.ru

LintFix

Tinkoff.ru

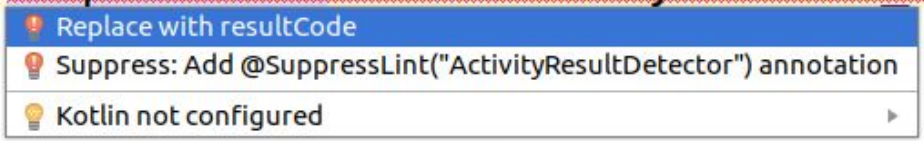
```
val paramResultCode = method.parameterList.parameters[1]

val fix = LintFix.create()
        .replace()
        .text(paramRequestCode.name)
        .with(paramResultCode.name)
        .build()
```

```
val paramResultCode = method.parameterList.parameters[1]

val fix = LintFix.create()
        .replace()
        .text(paramRequestCode.name)
        .with(paramResultCode.name)
        .build()
```

```
context.report(
        issue = ActivityResultDetector.ISSUE,
        location = context.getLocation(comparison),
        message = "Error!",
        quickfixData= fix
)
```

LintFix

```
override fun onActivityResult(
    requestCode: Int,
    resultCode: Int,
    data: Intent?
) {
    if (requestCode == Activity.RESULT_OK) {

    }
}
```



Tinkoff.ru

Тесты

Tinkoff.ru

```
val testData = """
    import android.app.Activity;
    import android.content.Intent;

    class MyActivity extends Activity {
        @Override
        public void onActivityResult(<...>) {
            if (requestCode == Activity.RESULT_OK) {
            }
        }
    }
""".trimIndent()
```

```
val expectedText = """
    |src/MyActivity.java:7: Error: Error!
    |        if (requestCode == Activity.RESULT_OK) {
    |            ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    |1 errors, 0 warnings
""".trimMargin()
```

```
@Test
fun basicScenario() {
    val testData = "<...>"
    val expectedText = "<...>"
    TestLintTask.lint()
            .files(java(testData))
            .issues(ActivityResultDetector.ISSUE)
            .run()
            .expect(expectedText)
}
```

```
val expectedFixDiffs = """
    |Fix for src/MyActivity.java line 7: Replace:
    |@@ -7 +7
    |-          if (requestCode == Activity.RESULT_OK) {
    |+          if (resultCode == Activity.RESULT_OK) {
""".trimMargin()
```

```
@Test
fun quickfix() {
    val testData = "<...>"
    val expectedFixDiffs = "<...>"
    TestLintTask.lint()
            .files(java(testData))
            .issues(ActivityResultDetector.ISSUE)
            .run()
            .expectErrorCount(1)
            .expectFixDiffs(expectedFixDiffs)
}
```

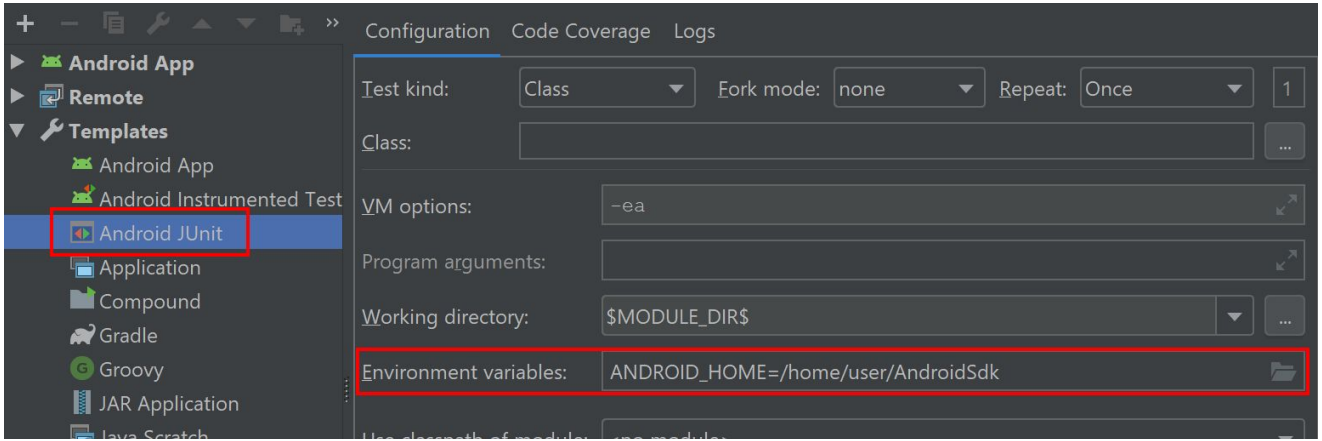
Решение проблем

Tinkoff.ru

```
java.lang.AssertionError: This test requires an Android SDK:
No SDK configured.
```

Запуск тестов

- Run → Edit configurations



Tinkoff.ru

```
class MyClass {
    @MyAnnotation
    fun foobar() {
    }
}
```

```
UFile (package = )
    UClass (name = MyClass)
        UAnnotationMethod (name = foobar)
            UBlockExpression
        UAnnotationMethod (name = MyClass)
```

```
UFile (package = )
    UClass (name = MyClass)
        UAnnotationMethod (name = foobar)
            UBlockExpression
        UAnnotationMethod (name = MyClass)
```

Решения

- скопировать код в тест (stubs)

Tinkoff.ru

```
val myAnnotation = "annotation class MyAnnotation"
TestLintTask.lint()
        .files(kotlin(testData), kotlin(myAnnotation))
        .issues(ISSUE)
        .run()
        .expect(expectedText)
```

```
val myAnnotation = "annotation class MyAnnotation"
TestLintTask.lint()
        .files(kotlin(testData), kotlin(myAnnotation))
        .issues(ISSUE)
        .run()
        .expect(expectedText)
```

```
UFile (package = )
    UClass (name = MyClass)
        UAnnotationMethod (name = foobar)
            UAnnotation (fqName = MyAnnotation)
            UBlockExpression
        UAnnotationMethod (name = MyClass)
```

```
UFile (package = )
    UClass (name = MyClass)
        UAnnotationMethod (name = foobar)
            UAnnotation (fqName = MyAnnotation)
            UBlockExpression
        UAnnotationMethod (name = MyClass)
```

Решения

- скопировать код в тест (stubs)
- подключить JAR

Tinkoff.ru

```
val targetPath = "libs/my-lib.jar"
val producer = object : TestFile.BytecodeProducer() {
    override fun produce() = // read bytes from resources
}
val myLib = bytecode(targetPath, producer)
TestLintTask.lint()
    .files(kotlin(file), classpath(targetPath), myLib)
    .issues(ISSUE)
    .run()
    .expect(expectedText)
```

Недостатки

Tinkoff.ru

Недостатки

- мало документации
- высокий порог входа
- рефакторинг может принести проблем
- требуется поддержка
- баги

Tinkoff.ru

Заключение

Tinkoff.ru

Заключение

- чем раньше выявим баг - тем раньше его исправим
- то, что можно автоматизировать - нужно автоматизировать
- lint - не панацея

Tinkoff.ru

Полезные материалы

- *Kotlin Static Analysis with Android Lint* - Tor Norbye

  https://www.youtube.com/watch?v=p8yX5-lPS6o
- *Coding in Style: Static Analysis with Custom Lint Rules*

  https://www.youtube.com/watch?v=jCmJWOkjbM0
- lint-dev - Google Groups

  https://groups.google.com/forum/#!forum/lint-dev

Tinkoff.ru

Полезные материалы

- lint-api

  https://bit.ly/2WKaPYW
- lint-checks

  https://bit.ly/2If9XqW
- UAST, PSI

  https://github.com/JetBrains/intellij-community

Tinkoff.ru

Полезные материалы

- Android Lint performance probe

  https://bit.ly/32Vqrv3

Tinkoff.ru

Вопросы?

t.me/italankin
github.com/italankin