

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

Национальный исследовательский Томский государственный университет  
Томский государственный университет систем управления и радиоэлектроники  
Болгарская Академия наук  
ООО «Научно исследовательское предприятие «Лазерные технологии»

# **ИННОВАТИКА-2018**

**СБОРНИК МАТЕРИАЛОВ**

**XIV Международной школы-конференции студентов,  
аспирантов и молодых ученых  
26–27 апреля 2018 г.  
г. Томск, Россия**

*Под редакцией А.Н. Солдатова, С.Л. Минькова*

Scientific & Technical Translations



**ИЗДАТЕЛЬСТВО**

**Томск – 2018**

## **ВНЕДРЕНИЕ РОССИЙСКИХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ В ЗАРУБЕЖНЫЕ ПЛАТФОРМЫ**

**Ю.А. Чичканова, Н.В. Емельянова, А.А. Погуда**

*Национальный исследовательский Томский государственный университет  
chichkanova.1995@mail.ru*

### **THE INTRODUCTION OF RUSSIAN PROGRAMMING LANGUAGES INTO FOREIGN PLATFORMS**

**J.A. Chichkanova, N.V. Emelyanova, A.A. Poguda**

*National Research Tomsk State University*

*This article describes the Russian programming language Kotlin, which was introduced in Android Studio. Also, the article talks about the creators of this language and how it differs from Java.*

*Keywords: Kotlin, Android, Java, JetBrains, Google.*

Под языками программирования понимается система формальных обозначений для точного описания абстрактных структур данных и алгоритмов программ. Так же, как и любые языки, языки программирования имеют алфавит, словарь, синтаксис (правила конструирования выражений), семантику – разъяснение смысла конструкций из слов и выражений.

Внешняя форма языка программирования, т.е. запись текста программы, устанавливается с помощью синтаксиса языка, который определяет формальный язык. Синтаксис описывается с помощью определенных грамматических правил. Задание формального языка хотя и достаточно для установления синтаксической корректности, однако недостаточно для того, чтобы понять ее функцию и способ действия программы, которые определяются путем задания семантики, через установление значения отдельных языковых элементов.

В настоящее время программисты внедряют свои языки программирования, такие как 1С, Logo, RCML (Robot Control Meta Language), Kotlin.

Недостатков в современных платформах много, например, нужно часто вставлять стандартные куски кода (boilerplate), несмотря на наличие абстракций, многие абстракции мешают новичку в освоении концепций, т.к. непонятно, почему они созданы именно таким образом, байт-код требует установленной среды выполнения[1]. А также в связи с нецелесообразностью обновление традиционных языков программирования на устройствах, которые уже устарели. В связи с этим

приобретает актуальность сознание языков программирования лишенных основных недостатков.

Целью исследований является изучение отечественных платформ и языков программирования.

Среди исследованных языков программирования особый интерес, по мнению автора, представляет язык статически типизированный язык программирования, работающий поверх JVM( виртуальная машина Java) Kotlin.

Данный язык разработан компанией JetBrains в 2010 году. До 2017 года зарубежные компании не обращали на него внимание. Однако, в мае 2017 года компания Google сообщила, что инструменты языка Kotlin, основанные на JetBrains IDE, будут по стандарту включены в Android Studio 3.0 – официальный инструмент разработки для ОС Android [2, 3].

В Android язык встраивается с помощью Gradle, что позволяет для существующего Android-приложения внедрять новые функции на Kotlin без переписывания приложения целиком[4].

Позиционируется разработчиками как объектно-ориентированный язык промышленного уровня, а также как язык, который сможет заменить Java. При этом язык полностью совместим с Java, что позволяет разработчикам постепенно перейти с Java на Kotlin. Кроме объектно-ориентированного подхода, Kotlin также поддерживает процедурный стиль с использованием функций. Как и в языках C/C++/D, точка входа в программу – функция «main», принимающая массив параметров командной строки. Компилируется также в JavaScript и на другие платформы через инфраструктуру LLVM [5].

Синтаксис языка похож на Pascal, TypeScript, Haxe, PL/SQL, F#, Go и Scala, C++, Java, C# и D. При объявлении переменных и параметров типы данных указываются после названия (разделитель – двоеточие). Точка с запятой как разделитель операторов так же необязательна, как в Scala и Groovy; в большинстве случаев перевода строки достаточно, чтобы компилятор понял, что выражение закончилось. Программы на Kotlin также поддерживают Perl- и Unix/Linux shell-стиль интерполяции строк [5].

Для наглядного представления преимуществ языка Kotlin ниже приведен пример программы написанной на языке Kotlin(рис. 2, рис. 3) и Java(рис. 1).

```

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private TextView mHelloTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mHelloTextView = (TextView) findViewById(R.id.textView);
    }

    public void onClick(View view) {
        mHelloTextView.setText("Международная конференция Инноватика-2018!");
    }
}

```

Рис. 1. Программа, написанная на языке Java

Язык Java и Kotlin отличается, во-первых, в Kotlin расширение kt вместо class, но располагается в аналогичном пакете как у Java. Во-вторых, синтаксис кода также отличается, например, не нужно ставить точку с запятой в конце оператора. Ниже приведен пример, написанный на языке Kotlin (рис. 2).

```

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        fun onClick(v: View?) {
            textView.text = "Международная конференция Инноватика-2018!"
        }
    }
}

```

Рис. 2. Программа, написанная на языке Kotlin

В `build.gradle` пишется строчка `apply plugin: 'kotlin-android-extensions'`, позволяющая автоматически получать доступ к компоненту через его идентификатор в ресурсах. Поэтому больше не надо объявлять `ImageButton` и `TextView`. Студия импортирует нужный класс автоматически, для этого добавляется данная строка `import kotlinx.android.synthetic.main.activity_main.*`.

Kotlin позволяет заменить пары методов вида `getXX/setXX` свойствами без приставок. Если вы что-то присваиваете свойству, значит вы хотите использовать `setText()`, а если хотите прочитать значение свойства – значит вы хотите вызвать `getText()`. Вот так изящно работает Kotlin.

### Литература

1. Типы языков программирования [Электронный ресурс]. – URL: <https://tproger.ru/translations/programming-languages-types/> (дата обращения: 05.05.2018).
2. Созданный в России язык программирования Kotlin получил поддержку Google// Коммерсант [Электронный ресурс]. – URL: <https://www.kommersant.ru/doc/3303280/> (дата обращения: 05.05.2018).
3. За что Kotlin так полюбили в Google [Электронный ресурс]. – URL: <http://news.ifmo.ru/ru/science/it/news/6683/> (дата обращения: 05.05.2018).
4. Statically typed programming language for modern multiplatform applications// Официальный сайт Kotlin [Электронный ресурс]. – URL: <https://kotlinlang.org/> (дата обращения: 06.05.2018).
5. Kotlin [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Kotlin> (дата обращения: 06.05.2018).