

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ JAVA И KOTLIN

COMPARATIVE ANALYSIS OF APPLICATION EFFECTIVENESS OF THE PROGRAMMING LANGUAGES JAVA AND KOTLIN

L. Cherkesova
E. Revyakina
I. Yengibaryan
A. Revyakin

Summary. This article provides the overview and comparative analysis of Java and Kotlin programming languages developed by Oracle and JetBrains companies accordingly. Well known, that Kotlin language is considered as the successor of Java language, and is currently the face of the Android Company. The authors studied the features of languages, methods of code writing, additional development tools and features of the device of languages, the knowledge of which is necessary for the developer when designing the architecture of software applications, as well as understanding the requirements for the system being developed in the process of its implementation. Due to the fact that new network technologies are emerging every day, the development in programming language should provide for the possibility of using third-party tools for the developing of modern software, especially in our time, when domestic specialists in the field of IT technologies face the task of import substitution of software products. Using these tools will make the programmer's work more efficient, simplify and improve many processes. In order to keep up with the times, it is necessary to study quickly and implement more modern, flexible in configuration and easy-to-use developer tools in practical development. Such tools include third-party services that are an integral part of the whole project, without the use of which it is impossible to use this or that functionality. Such solutions should be convenient both for the programmer developing the software product and for users who will encounter it in practice.

Keywords: virtual machine, programming languages, manageability, development.

Черкесова Лариса Владимировна

Д.ф.-м.н
Донской Государственный Технический
Университет
chia2002@inbox.ru

Ревякина Елена Александровна

К.т.н
Донской Государственный Технический
Университет
revyelena@yandex.ru

Енгибарян Ирина Алешаевна

К.т.н
Донской Государственный Технический
Университет
eirina@live.ru

Ревякин Александр Иванович

Донской Государственный Технический
Университет
pianonafromhell@gmail.com

Аннотация. В предлагаемой статье выполнены обзор и сравнительный анализ языков программирования Java и Kotlin, разработанных компаниями Oracle и JetBrains. Как известно, язык Kotlin считается наследником языка Java, и в настоящее время является лицом компании Android. Авторами изучены особенности языков, методы написания кода, дополнительные инструменты разработки и особенности устройства языков, знания которых необходимы разработчику при проектировании архитектуры программных приложений, а также понимания требований к разрабатываемой системе в процессе её реализации. Ввиду того, что с каждым днём появляются всё новые сетевые технологии, разработка на языке программирования должна предусматривать возможность использования сторонних инструментов разработки современного программного обеспечения, особенно в наше время, когда перед отечественными специалистами в области ИТ-технологий стоит задача импортозамещения программных продуктов. Использование этих инструментов сделает работу программиста более эффективной, упростит и усовершенствует многие процессы. Чтобы не отставать от времени, необходимо быстро изучать и внедрять в практические разработки более современные, гибкие в настройке и лёгкие в эксплуатации средства разработчика. К таким средствам относятся сторонние сервисы, которые являются составной частью целого проекта, без применения которых тот или иной функционал задействовать невозможно. Подобные решения должны быть удобными как для программиста, разрабатывающего программный продукт, так и для пользователей, которые будут с ним сталкиваться на практике.

Ключевые слова: виртуальная машина, языки программирования, управляемость, разработка.

Введение

Повсеместное внедрение цифровых решений в различные сферы жизни общества и производства вызвало стремительное развитие информационных технологий, что упростило выполнения рутинных задач, позволив делегировать большинство простых заданий компьютерным системам предприятий, тем самым предоставив программисту возможность сконцентрироваться на выполнении вычислительных задач, требующих большего внимания со стороны разработчика. Зачастую такие задачи ограничены сроками их выполнения.

Кроме того, современные ИТ-специалисты должны уметь быстро и эффективно изучать основы новых языков программирования, и продолжать так же эффективно изучать их особенности в процессе разработки программных продуктов. Такие языки как JAVA, C#, C++ требуют больше времени на изучение фундаментальных основ языка, в результате чего подготовка программистов, пишущих на этих языках, занимает больше времени, чем подготовка специалистов, занимающихся разработкой на языках Python или JavaScript [1, 2].

Важным критерием быстрой разработки программного кода является его читаемость, которая во многом зависит от понятности синтаксиса. Читаемость кода играет такую же важную роль, как и его управляемость, позволяющая в короткие сроки внести изменения в программный проект, не навредив его основным компонентам. При этом не обязательно, чтобы рефакторинг проводился тем же специалистом, который писал этот код.

Выбор языка программирования зависит от множества факторов [3, 4]: это требования к скорости работы конечного продукта; к объёму занимаемой оперативной памяти; к скорости разработки программы; к кросс-платформенности; к ориентированности — на компьютер или на человека; к скорости внесения изменений, к скорости тестирования и др.

Нужно отметить, что язык программирования — это, в первую очередь, инструмент, с помощью которого решаются те или иные задачи, необходимые для корректной работы конечного продукта. Нужно учитывать скорость выполнения команд языка программирования процессором компьютера. Команды низкоуровневых языков программирования выполняются гораздо быстрее, чем программы, написанные на высокоуровневых языках программирования, синтаксис которых максимально приближен к естественным языкам.

В данной работе была поставлена цель — выполнить сравнительный анализ эффективности примене-

ния одних их самых востребованных сегодня языков программирования Java и Kotlin. Как известно, Kotlin считается наследником и преемником Java, и в настоящее время является лицом компании Android. Оба языка используют виртуальную машину Java (JVM), очень близки по синтаксису и могут использоваться для решения одних и тех же задач. Тем интереснее сравнить их между собой и показать их сильные и слабые стороны, достоинства и недостатки, преимущества и несовершенства при решении одинаковых задач. Сравнительный анализ их эффективности поможет разработчику не ошибиться в выборе.

Материалы и методы исследования

Для того, чтобы выполнить сравнение, необходимо перечислить основные критерии выбора языков программирования [5, 6]:

1. *Определить платформу разработки* — native Unix, native Windows, .Net, JVM. В рамках одной платформы писать различные части ПО на разных языках не представляет сложности.
2. *Ориентироваться на предпочтения или требования заказчика*. Например, заказчик может не использовать в своей информационной инфраструктуре определенные операционные системы такие как Windows, Unix-подобные или MacOS.
3. *Учитывать опыт работы команды разработчиков проекта с определенными информационными технологиями*. Разработку нового проекта лучше проводить на знакомой программной платформе, но с использованием незнакомого стека технологий.
4. *Знать специфику или область применения программного обеспечения* — так, например, для real time важнее всего скорость работы программы реального времени, для приложения, например, на сердечном стимуляторе — надёжность, для веб сайта — скорость разработки.

Кроме того, можно отметить следующие критерии выбора:

1. *Скорость работы конечного продукта*. Требованиями к скорости выполнения могут быть программы с большим объёмом математических вычислений, например, связанные с моделированием физических систем различной природы, с расчётами большого объёма экономических данных, выводением на экран трёхмерной графики, и прочее.
2. *Объём занимаемой оперативной памяти*. Такое требование появляется, когда ПО разрабатывается для встраиваемых систем, мобильных платформ, микроконтроллеров и др. В этих слу-

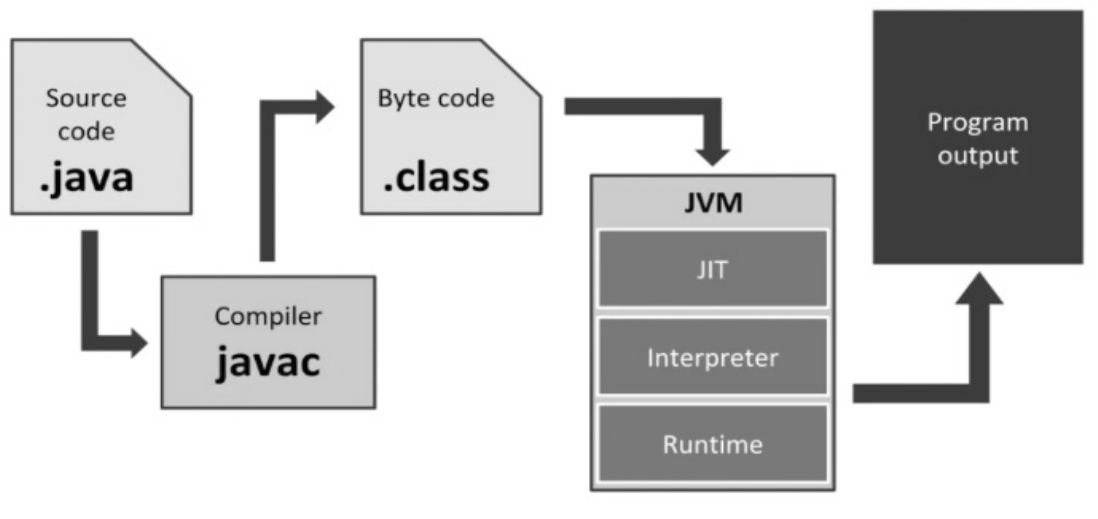


Рис. 1. Компиляция приложения, написанного на Java

чаях, чем меньше памяти расходует программа — тем лучше.

3. *Скорость разработки программного проекта.* Такое требование возникает, когда сроки разработки кода имеют существенные ограничения. Несмотря на это, в условиях стремительного развития рынка, быстрая разработка продукта важна ввиду высокой конкуренции. Значительно превышают показатель скорости разработки языки высокоуровневые, чей синтаксис максимально приближен к естественному языку. Скорость разработки возрастает при использовании готовых программных модулей. Скорость выполнения программ, написанных на таких языках, значительно ниже, чем выполнение программ, написанных на низкоуровневых языках. На рис. 1 представлен алгоритм сборки приложения, написанного на Java [6].
4. *Ориентированность на компьютер или человека.* Программа должна обладать мощным графическим интерфейсом, отвечающим требованиям дизайна. Разработка графической части зачастую требует достаточно много времени, т.к. отличается немалой сложностью. Здесь сложность возникает в том, что вывод графики — это немало математики, а значит присутствуют требования к скорости исполнения, а из-за сложности разработки присутствует необходимость в высокоуровневом языке.
5. *Кроссплатформенность.* Это возможность работы программы на различных платформах и в различных операционных системах с минимальными изменениями. В этой сфере можно выделить такие языки: *Java, C#, Flash, C++* с различными библиотеками, и другие, менее

используемые, языки. Java создавался с тем условием, что программы на данном языке должны работать на любой платформе, где есть *JVM — Java Virtual Machine*. Программы на Java вообще не требуют никаких изменений — после компиляции получается .jar файл, работающий и в операционной системе Windows, и в Mac OS, и в Unix (Linux), и др.

6. *Скорость внесения изменений, скорость тестирования.* Стремительное развитие проекта возможно, благодаря понятному синтаксису языка, простому, удобному и интуитивно понятному для человека. Такими характеристиками отличаются высокоуровневые языки программирования. Как известно, Kotlin является более высокоуровневым языком программирования, чем Java.

Для полного анализа ниже рассмотрим особенности языков программирования Java и Kotlin, их общие и отличительные черты [7]:

- ♦ Java и Kotlin являются языками со статической типизацией, что характеризует их как языки, проверка типов данных в которых выполняется в момент компиляции программы. При этом однозначно назвать Kotlin статически типизированным нельзя. Все же, в сравнении с Java, синтаксис языка позволяет реализовывать динамические конструкции.
- ♦ Оба языка распространяются свободно, как OpenSource (открытые ресурсы программного обеспечения), и являются бесплатными для пользователей.
- ♦ Эти языки компилируют высокоуровневый код, написанный разработчиком, в промежуточный байт-код, после чего происходит JIT-компиляция.

```

128 fun index(abc:ArrayList<Char>,listpart:Array<String>):ArrayList<Double>{
129     val listIndex = arrayListOf<Double>()
130     var count = 0.0
131     var indc = 0.0
132     for(k in 0 .. listpart.size-1){
133         for(i in 0 .. abc.size-1) {
134             for (j in 0 .. listpart[k].length - 1) {
135                 if (abc[i] == listpart[k][j]) {
136                     count += 1.0
137                 }
138             }
139             indc += count * (count - 1) / (listpart[k].length * (listpart[k].length - 1))//True 0.062 ~ 0.067
140             count = 0.0
141         }
142         listIndex.add(k, indc)
143         indc = 0.0
144     }
145     println(listIndex)
146     return listIndex
147 }

```

Рис. 2. Функция нахождения индекса совпадений в рамках криптографического анализа алгоритма Виженера, написанная на языке Kotlin

ция, в машинный код, исполняемый виртуальной Java машиной (JVM).

- ◆ Оба языка являются интероперабельными, что позволяет использовать файлы Java и Kotlin в одном проекте или пакете JAR.
- ◆ Как и Java, Kotlin является высокоуровневым и объектно-ориентированным языком программирования, что позволяет работать с кодом, как с объектом материального мира.
- ◆ Оба языка являются кроссплатформенными, а, значит, разработка программного обеспечения может проводиться на операционных системах Android, Windows, Linux и др.

Ниже перечислены отличительные особенности языков программирования Java и Kotlin [7]. В реализации этих языков можно выделить следующие различия:

- ◆ Язык Java был разработан компанией Sun Microsystems, активы которой в настоящее время принадлежат Oracle с 1995 года. Во многом, этот язык устарел.
- ◆ Язык Kotlin был представлен компанией JetBrains в 2011 году, выложен в открытый доступ в 2012 году, официально поддержан на ежегодном мероприятии разработчиков Google (Google I/O) в 2017 году, и быстро завоевал популярность в среде разработчиков [8, 9].
- ◆ Компания Google заявляет, что 70% из 1000 лучших приложений для Android на сегодня-

ний день разработаны на языке Kotlin, лицом которого он является. Многие приложения находятся в процессе переноса с Java на Kotlin, например, приложение Google Home еще не полностью переписано на Kotlin, но по состоянию на июнь 2020 года около 30% статистике кодовой базы на язык Java уже было переписано на Kotlin. Процесс миграции с Java на Kotlin еще не завершился, но компании активно переписывают свои программные продукты, написанные на Java, на язык Kotlin. Сейчас, при поддержке разработки под Android, компания Google придерживается стратегии «Kotlin прежде всего». Это в чём-то похоже на ситуацию с разработкой приложений под операционную систему iOS, где произошел аналогичный переход от языка Objective-C к языку Swift.

- ◆ Особенности версий: по состоянию на 9.06.2022 г. актуальной версией Kotlin является версия 1.7.0. Несмотря на выход версии Java 17, большой популярностью среди разработчиков до сих пор пользуется удачная версия языка Java 8 (или 1.8)
- ◆ Скорость: Java превосходит Kotlin на 12–15% для чистых сборок, иными словами, код на Kotlin компилируется медленнее, чем код, написанный на Java. Однако в случае частичныхборок с включенной инкрементной компиляцией, другими словами, с компиляцией только небольших

```

274 @NotNull
275 public static final ArrayList index(@NotNull ArrayList abc, @NotNull String[] listpart) {
276     Intrinsic.checkNotNullParameter(abc, paramName: "abc");
277     Intrinsic.checkNotNullParameter(listpart, paramName: "listpart");
278     ArrayList listIndex = new ArrayList();
279     double count = 0.0;
280     double inde = 0.0;
281     int k = 0;
282     for(int var8 = listpart.length; k < var8; ++k) {
283         int i = 0;
284         int var10 = abc.size() - 1;
285         if (i <= var10) {
286             while(true) {
287                 int j = 0;
288                 int var12 = listpart[k].length() - 1;
289                 if (j <= var12) {
290                     while(true) {
291                         Character var10000 = (Character)abc.get(i);
292                         char var13 = listpart[k].charAt(j);
293                         if (var10000 != null) {
294                             if (var10000 == var13) ++count;
295                         }
296                         if (j == var12) {break;}
297                         ++j;
298                     }
299                 }
300                 inde += count * (count - (double)1) / ((double)(listpart[k].length() * (listpart[k].length() - 1)));
301                 count = 0.0;
302                 if (i == var10) {
303                     break;
304                 }
305                 ++i;
306             }
307         }
308         listIndex.add(k, inde);
309         inde = 0.0;
310     }
311     System.out.println(listIndex);
312     return listIndex;
313 }

```

Рис. 3. Функция нахождения индекса совпадений на языке Java

изменений, Kotlin не уступает в скорости компиляции программного кода языку Java, а в некоторых случаях этот процесс происходит немного быстрее.

- ◆ Количество строк кода: код, написанный на языке Kotlin, выглядит намного лаконичнее, компактнее и понятнее, по сравнению с громоздким кодом, написанном на языке Java. Такая разница может достигать 50% и более. В теории, размер файлов приложений, написанных на языке Kotlin, можно уменьшить в 3 раза. Язык Java является объёмным и сложным, в понимании написанного

на нём кода, в то время как Kotlin представляется более современным и лаконичным языком программирования.

Для сравнения, на рис. 2 приведена функция нахождения индекса совпадений в рамках криптографического анализа алгоритма Виженера, написанная на языке Kotlin [10].

На рис. 3 продемонстрирована функция нахождения индекса совпадений, конвертируемая в программный код, написанный на языке Java.

```

1025 public final static index(Ljava/util/ArrayList;[Ljava/lang/String;)Ljava/util/ArrayList;
1026 @Lorg/jetbrains/annotations/NotNull;() // invisible
1027 // annotable parameter count: 2 (visible)
1028 // annotable parameter count: 2 (invisible)
1029 @Lorg/jetbrains/annotations/NotNull;() // invisible, parameter 0
1030 @Lorg/jetbrains/annotations/NotNull;() // invisible, parameter 1
1031 L0
1032 ALOAD 0
1033 LDC "abc"
1034 INVOKESTATIC kotlin/jvm/internal/Intrinsics.checkNotNullParameter (Ljava/lang/Object;Ljava/lang/String;)V
1035 ALOAD 1
1036 LDC "listpart"
1037 INVOKESTATIC kotlin/jvm/internal/Intrinsics.checkNotNullParameter (Ljava/lang/Object;Ljava/lang/String;)V
1038 L1
1039 LINENUMBER 129 L1
1040 L2
1041 NEW java/util/ArrayList
1042 DUP
1043 INVOKESPECIAL java/util/ArrayList.<init> ()V
1044 L3
1045 LINENUMBER 129 L3
1046 ASTORE 2
1047 L4
1048 LINENUMBER 130 L4
1049 DCONST_0

```

Рис. 4. Байт-код функции индекса совпадений

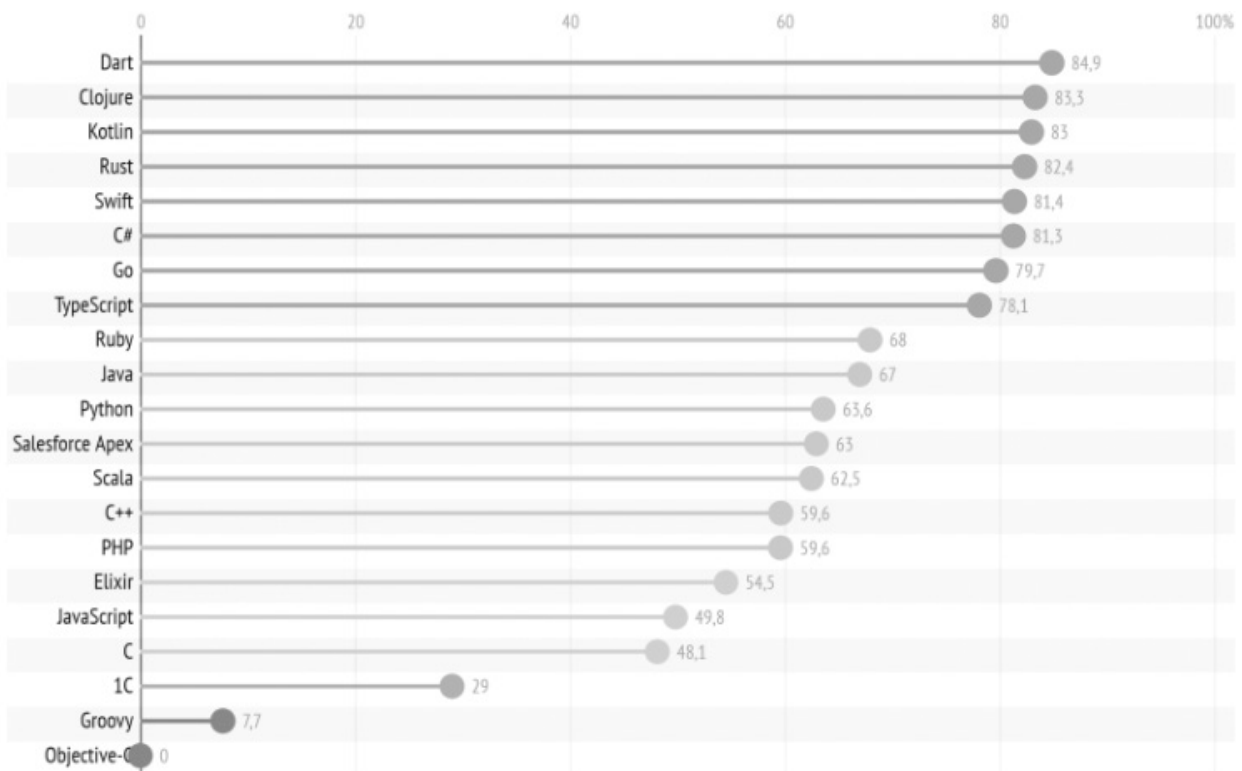


Рис. 5. Индекс предпочтения языков программирования ИТ-разработчиками

Таблица 1. Универсальность языков программирования Java и Kotlin

Области применения	Java,%	Kotlin,%	Остальные языки программирования,%
Back-end (Программно-аппаратная часть сервиса, набор средств реализации ПО)	29.4	0.9	69.7
GameDev (Разработка компьютерных игр)	0	0	100
Front-end (Пользовательский интерфейс)	0	0	100
Mobile (Разработка программных приложений для мобильных устройств)	4.9	31.3	63.8
Data Processing (Обработка данных)	4.1	0	95.9
Full Stack (разработка полного цикла ПО, включающая в себя и Front-end, и Back-end разработку, создание web-приложения с нуля)	8.2	0.2	91.6
Embedded (Встраиваемое в устройство ПО)	0	0	100
Desktop (программа на стороне клиента, запускаемая на устройстве пользователя)	0	0	100
DevOps (методология процесса разработки ПО, при которой программисты, тестировщики и системные администраторы разрабатывают программный продукт быстрее и эффективнее)	0	0	100

Таблица 2. Сравнительный анализ языков программирования Java и Kotlin

Критерии сравнения	Java	Kotlin
Кроссплатформенность	+	+
Читаемость кода	-	+
Универсальность	-	+
Скорость работы конечного продукта	+	-
Поддержка сообщества	-	+
Полнота и надежность кода	-	+

Результаты исследования и их обсуждение

Из рис. 2 и 3 очевидно, что одна и та же функция занимает на языке Kotlin всего 20 строк кода, а на языке Java — 39 строк кода — практически, в два раза больше. Однако здесь речь идёт именно о читаемости кода и об удобстве работы с ним, так как байт-код проекта будет для этих языков неразличимым. На рис. 4 приведена часть объявления той же функции в байт коде.

По данным опросов, разработчиков на языке Kotlin в настоящее время примерно в пять раз меньше, чем программистов, использующих Java. 8% разработчиков предпочитают Kotlin, а более чем 40% — привычный им язык Java [11]. Однако эти опросы также показывают, что Kotlin нравится знакомым с ним программистам, гораздо больше, чем Java, и быстро наращивает аудиторию. Типичным является сравнение Java с телегой, а Kotlin — с космической ракетой.

На рис. 5 представлены показатели индекса предпочтения ИТ-разработчиками языков программирования (ЯП) в 2022 году [12, 13].

Оценка универсальности целевых языков программирования приведена в таблице 1.

Из проведенной оценки видно, что язык программирования Java более универсален, чем Kotlin. Оценка языков программирования Java и Kotlin представлена в таблице 2.

Выводы и заключение

Исследования показали, что язык программирования Kotlin превосходит своего предшественника, язык Java по ряду показателей, и является хорошим решением при выборе языка разработки для больших программных проектов.

Язык программирования Java уступает языку программирования Kotlin в лаконичности, однако превосходит его по скорости выполнения программного кода, что

является существенным преимуществом, особенно, когда требуется разрабатывать программные средства, в которых скорость вычислений имеет приоритетное значение.

ЛИТЕРАТУРА

1. Свейгарт Э. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ.— М.: Изд. «Вильямс», 2016.— 592 с.
2. Атенсио Л. Функциональное программирование на JavaScript: как улучшить код JavaScript–программ.— М.: Диалектика, 2018.— 304 с.
3. Блох Дж. Java: эффективное программирование.— М.: Диалектика, 2019.— 464 с.
4. Дашнер С. Изучаем Java EE. Современное программирование для больших предприятий.— СПб.: Питер, 2018.— 384 с.
5. Куценко С.М., Дубовиков И.И. Сравнительный анализ языков программирования // Учёные записки ИСГЗ.— 2019.— Т. 17.— № . 2.— С. 170–178.
6. Иванов С.О., Ильин Д.В., Большаков И.Ю. Сравнительное тестирование языков программирования // Вестник Чувашского университета.— 2017.— № . 3.— С. 222–227.
7. Зарипов А.Р., Сафиуллина Ф.Ф. Сравнительный анализ языков программирования Kotlin и Java // Сборник трудов молодых ученых УВО «Университет управления «ТИСБИ».— Казань. 2019.— С. 134–138.
8. Солонько М.К. Язык программирования Kotlin // Вестник науки и образования.— 2020.— № . 7–1 (85).— С. 25–27.
9. Кочешков М.А., Киришин И.И. Особенности и перспективы языка программирования Kotlin // Colloquium–journal.— 2019.— № . 10–2.— С. 113–115.
10. Прендота А.В., Балакшин П.В. Браузерная платформа для языка программирования Kotlin // Сборник трудов VII Конгресса молодых ученых.— 2018.— С. 181–183.
11. Вейлшери Л.С. Наиболее широко используемые языки программирования среди разработчиков по всему миру в 2022 г.— Режим доступа: <https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/> (дата обращения 30.07.2022)
12. Чернов В.В. Исследование современного языка программирования Kotlin // С. 68.
13. Хакимова Т.В. Обзор информационных технологий, используемых в обучении программированию на языке Kotlin // Modern Science.— 2020.— № . 2–2.— С. 354–358.

© Черкесова Лариса Владимировна (chia2002@inbox.ru), Ревякина Елена Александровна (revyelena@yandex.ru),
 Енгибарян Ирина Алешаевна (eirina@live.ru), Ревякин Александр Иванович (pianonafromhell@gmail.com).
 Журнал «Современная наука: актуальные проблемы теории и практики»