LEARNING THE KOTLIN PROGRAMMING LANGUAGE USING AN AUTOGRADING SYSTEM

NIKOLA DIMITRIJEVIĆ

Belgrade Metropolitan University, Faculty of Information Technologies, nikola.dimitrijevic@metropolitan.ac.rs

VLADIMIR MILIĆEVIĆ

Belgrade Metropolitan University, Faculty of Information Technologies, vladimir.milicevic@metropolitan.ac.rs

DEJAN CVIJANOVIĆ

OTT Solutions, Austria, dejan.cvijanovic@gss-team.com

NEMANJA ZDRAVKOVIĆ

Belgrade Metropolitan University, Faculty of Information Technologies, nemanja.zdravkovic@metropolitan.ac.rs

Abstract: In modern curricula for Computer Science (CS) and Information Technology (IT), one of the more popular classes is mobile development. With the recent rise of the Kotlin programming language as a main contender for Android development, very few courses in Higher Education Institutions (HEIs) exist which incorporate Kotlin as a main language. Furthermore, online courses which offer learning Kotlin are very limited, or require a high subscription price. More importantly, neither online, nor HEI courses offer an automatically graded system for student assignments in Kotlin.

In this paper, a tool for learning for learning the Kotlin programming language with is presented which supports automatic grading of assignments. As the Kotlin language is itself designed to interoperate fully with Java, this tool, developed by the authors, is aimed at CS and IT students who have already have basic knowledge of Java (or similar) programming languages, and want to switch to mobile development. Our solution focuses on the development of an interactive course in Kotlin meant primarily for Java developers. The assignments in the course also compare certain approaches to a solution in Java and Kotlin side-by-side. The solution is developed as a web application, not needing a separate compiler, libraries or an integrated development environment. Furthermore, to compare our solution with commercially available ones, we point out the disadvantages of currently available Kotlin courses, such as the level of previous knowledge needed, or a need for a specific development environment.

Keywords: autograding, eLearning, Kotlin, Java, mobile development

1. INTRODUCTION

Since its inception over 25 years ago, the Java programming language is currently at among the most popular programming languages [1, 2]. As a general-purpose language, Java has gained its popularity by being platform-independent, attributable to the Java Virtual Machine (JVM). The use of JVM has led to new programming languages, such as Groovy, Scala, and most notably Kotlin [3]. The increasing popularity of Kotlin is evident, as in 2017, Google announced Kotlin as one of the officially supported languages for Android development [4]. Among the reasons for choosing Kotlin, Google mentioned it is "concise, expressive, and designed to be

type and null-safe" [4]. Kotlin was developed in 2010 by JetBrains in order to improve the programming experience for the JVM [5]. It is a multiparadigm language, which supporting both object-oriented and functional programming paradigms, allowing developers to use either of a combination of [6]. In addition, Kotlin has interoperability with Java, therefore both languages can be freely mixed and allowing for gradual migration.

Although officially supported for Android development, the learning curve of Kotlin can be difficult for developers not used to Java [7]. Indeed, as more Higher Education Institutions (HEIs) offer languages such as Python as a first language for students, learning Kotlin without Java can

prove difficult [8-11]. Our motivation is to present a learning tool for seamlessly learning the Kotlin language, aimed at Computer Science (CS) and Information Technology (IT) students who have already basic knowledge of Java (or similar) programming languages, and want to switch to mobile development.

In this paper, we present a web-based application for learning the Kotlin language with autograding support, suitable to follow a mobile development course offered to CS and IT students. The web application is still in development, however major components have been identified and will be detailed below. The rest of the paper is organized as follows: Section 2 covers the Kotlin course overview, required previous knowledge, course topics and syllabus. In section 3 we propose our system with its major components. Section 4 deals with future work regarding the web application, and Section 5 concludes the paper.

2. KOTLIN COURSE OVERVIEW

Although somewhat derived from Java, Kotlin is a separate language and, while sharing some similarities with Java, its distinct differences require a separate course. In the first Subsection of Section 2, we point out the major differences in Kotlin from Java, and give a course overview in Subsection 2.

Differences and similarities

Both Kotlin and Java are a static-typed language. In addition, both languages support the object-oriented paradigm, by Kotlin also supports functional constructions. Kotlin can be used both in object-oriented and in functional programming style or in a mix of both styles [12, 13]. Kotlin allows for functions declaration outside the classes, whereas in Java, static methods are used for these purposes, resulting in classes whose instances are never created but instead only static methods are called [13]. As for data types, both languages support the basic data types such as literal constants, symbolic constants and variables. Table 1 gives an overview od supported data types [13]. Primitive data types are not object, as the String and Object reference types are. Primitive data types are object in Kotlin, but some may have special representations, e.g. numeric and Boolean types can be represented as primitives at runtime.

In general, Kotlin types are divided into nullable and non-nullable. The base class Any is a super-type for all type, and it cannot contain a value of null [12,13]. However, Kotlin has explicit support for nullable types, and is represented as a question mark after the type name, allowing the variable to contain null. For the Any example, the type Any? Is used if a null value is necessary.

Both Java and Kotlin support object creation as instances of a class using constructors. Whereas Java supports multiple constructor declarations, Kotlin has a primary constructor, which is declared outside the class body, and a secondary constructor within the class body. The role of the primary constructor is to initialize the class, while the role of the secondary constructor is to incorporate additional logic. As for operations and expressions, statements, the full comparison with tables is given in [13]. Overall, although similar, Kotlin's synthase is shorter and intuitive, which can increase the productivity of the programmer.

Table 1: Data type similarities between Java and Kotlin.

Data Type	Java	Kotlin
Integer	byte, short, int, long	Byte, Short, Int, Long
Floating point	float, double	Float, Double
Boolean	boolean	Boolean
Alphanumeric (character)	char	Char
Alphanumeric (string)	string	String
Null object	null	null
Base class	Object	Any

Apart from similarities in different data types in Java and Kotlin found in Table 1, the differences between the two programming languages is shown in Table 2. Namely, Table 2 highlights the properties of Kotlin which Java does not have [14].

Table 2: Properties of Kotlin programming language which are different from Java.

Property
Lambda expressions + Inline functions
Extension functions
Null-safety and smart casts
String templates
Properties
Primary constructors
First-class delegation
Type inference for variable and property types
Singletons and companion objects
Declaration-site variance & Type projections
Range expressions
Operator overloading

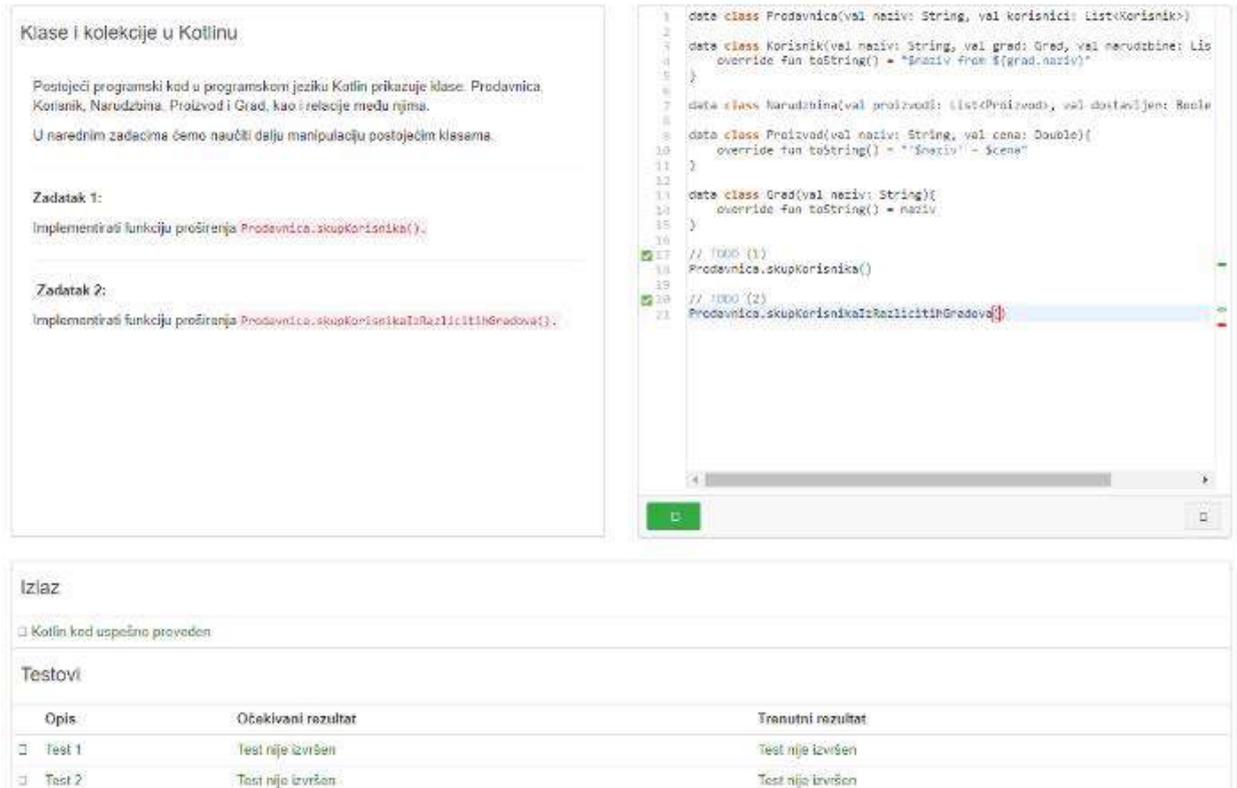- Conventions, with topics as comparison, range, and loops;

Image 2: The UI of Kotlin learning web-based application.

| Data classes |
| Separate interfaces for read-only and mutable collections |
| Coroutines |

Course overview

As stated in the Introduction, without the basic knowledge of Java, Kotlin can be difficult to learn and master. Furthermore, being a relatively new programming language, courses in Kotlin are few and far between. Apart from the official documentation [12], the courses offered online are either intermediate level such as the developers.android courses [15-17], or require a paid subscription fee such as the course on Codecademy [18].

We base our proposed course on Java courses taught at Belgrade Metropolitan University (BMU), as well as the Kotlin Koans online web application, developed by JetBrains as an introduction to Kotlin [19]. The syllabus is as follows:

- Introduction, with named and default arguments, string templates, nullable objects and the nothing type;

- Classes, covering data classes, smart and sealed classes, and extension functions;

- Collections, Properties, and Builders;

- Generics functions;

- Common mistakes for Java developers, with often encountered situation where the differences in the languages can lead to programming errors.

PROPOSED SYSTEM

Our proposed application starts with the actual lesson, which can be presented in text or video format, followed by multiple assignments within the lesson [20]. The User Interface (UI) diagram of the application is presented in Image 1, while the actual UI screen in shown in Image 2.

139
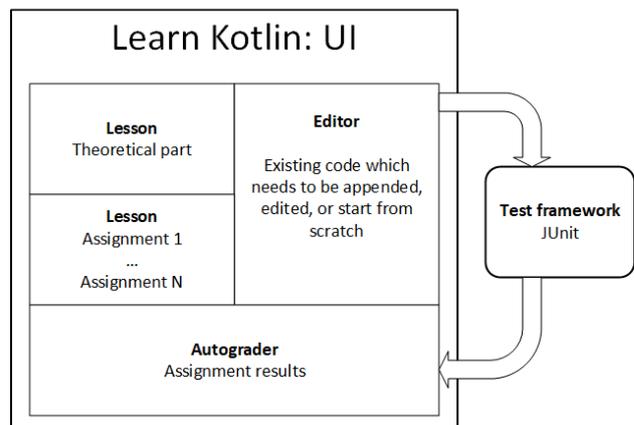
Image 1: Proposed UI for the Kotlin learning application.

The upper part of the UI screen is divided into three areas. The upper left area presents the lesson, the lower left area presents the assignments, while the right side represents the code editor. According to the given assignment, a student can append existing code, edit existing code, write new code from scratch. Every assignment consists of a group of tests to check the correctness of the student's solution.

Finally, the lower part of the screen is the output of the autograder, which runs multiple tests after the student submits their code. For this purpose, we have applied the JUnit framework for Kotlin [21].

4. FUTURE WORK

To further increase the interactivity with course takers, we plan to add a quiz at the beginning of the whole course, establishing a baseline knowledge for the Java programming language. Furthermore, apart from the code-based assignments, we plan to add simple multiple-choice quizzes at the end of each lesson to accompany the theoretical part of each lesson. These quizzes can also contain simple lines of code in Java and Kotlin, with the aim to find the correct approach to a specific problem.

Finally, we plan to combine the baseline knowledge and quiz and assignment results, forming an adaptive learning path for course takers.

5. CONCLUSION

In this paper we have proposed web application for learning the Kotlin programming language. As of writing this paper, both online solutions with which we compared our own application offer only intermediate courses for Kotlin for experienced Java developers, often not pointing out the main differences and common mistakes. Our proposed course, although aimed at those already familiar with Java, offers a simpler learning curve than those encountered by courses found online. Furthermore, with the inclusion of detailed theoretical knowledge, clips and/or animations, the offered lessons would feel lest like a step-by-step tutorial, but rather a like a full course for CS and IT students.

The addition of an advanced autorgrader allows our solution to hide the answers, encouraging students to think about the problem and not getting the right solution for the specific test or tests. As the mobile development continues, we plan to add more specific learning topics for Kotlin Android developers.

ACKNOWLEDGMENT

REFERENCES

[1] IEEE Spectrum: Top Programming Languages," 2021, [Online] Available: https://spectrum.ieee.org/top-programming-languages/ (Accessed: Aug 2021).

[2] TIOBE index, 2021 [Online]. Available: https://www.tiobe.com/tiobe-index/java/ (Accessed: Aug 2021).

[3] R.G. Urma, "Alternative Languages for the JVM. A look at eight features from eight JVM languages," 2014, [Online] Available: https://www.oracle.com/technetwork/articles/java/architect-languages-2266279. (Accessed Sep 2019).

[4] V. Oliveira, L. Teixeira, and F. Ebert, 2020, "On the adoption of Kotlin on android development: A triangulation study," in Proc. IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 206-216, 2020.

[5] A. Breslav, "History of Kotlin Kotlin for Java Developers – Coursera," [Online] Available: https://www.coursera.org/lecture/kotlin-for-java-developers/history-of-kotlin-K8pZr. (Accessed Aug 2021).

[6] M. Flauzino, J. Veríssimo, R. Terra, E. Cirilo, V. H. Durelli, and R. S. Durelli, "Are you still smelling it? A comparative study between Java and Kotlin language," in Proc. VII Brazilian symposium on software components, architectures, and reuse, pp. 23-32, 2018.

[7] T. Hagos. "Kotlin Basics," in Learn Android Studio 3 with Kotlin, Apress, Berkeley, CA, 2018, pp. 33-61.

[8] C. Lo, Y. Lin, and C. Wu, "Which Programming Language Should Students Learn First? A Comparison of Java and Python," in Proc. 2015 International Conference on Learning and Teaching in Computing and Engineering, 2015, pp. 225-226.

[9] C. Dierbach, "Python as a first programming language," Journal of Computing Sciences in Colleges, vol. 29, no. 3 pp. 73-73, 2014.

[10] V. Leping, M. Lepp, M. Niitsoo, E. Tõnisson, V. Vene, and A. Villems, "Python prevails," in Proc. International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, 2009, pp. 1-5.

[11] M. Martinez and B. G. Mateus. "How and Why did developers migrate Android Applications from Java to Kotlin? A study based on code analysis and interviews with developers,", arXiv preprint arXiv:2003.12730, 2020.

[12] Kotlin Language Documentation, 2021. [Online] Available: https://kotlinlang.org/docs/kotlin-docs.pdf (Accessed Aug 2021).

[13] D. Gotseva, Y. Tomov, and P. Danov, "Comparative study Java vs Kotlin," In Proc. 27th National Conference with International Participation (TELECOM, pp. 86-89, 2019.

[14] What Kotlin has that Java does not, 2021. [Online] Available: https://kotlinlang.org/docs/comparison-to-java.html#what-kotlin-has-that-java-does-not (Accessed Aug 2021).

[15] Kotlin language training, , 2021. [Online] Available: https://developer.android.com/courses (Accessed Aug 2021).

[16] Kotlin Bootcamp for Programmers, 2021. [Online] Available: https://developer.android.com/courses/kotlin-bootcamp/overview (Accessed Aug 2021).

[17] Kotlin for Java developers, 2021. [Online] Available: https://developer.android.com/courses/pathways/kotlin-for-java (Accessed Aug 2021).

[18] Learn Kotlin, 2021. [Online] Available: https://www.codecademy.com/learn/learn-kotlin (Accessed Aug 2021).

[19] Kotlin Koans, 2021 [Online] Available: https://play.kotlinlang.org/koans/ (Accessed Aug 2021).

[20] N. Dimitrijević, D. Cvijanovic and S. Jovanović, "Analysis and improvements of existing e-learning systems and a draft of an e-learning system for data structures problems," in Proc. The Ninth International Conference on e-Learning, Belgrade Metropolitan University, pp. 94-98, 2018.

[21] Test code using JUnit in JVM, 2021 [Online] Available: https://kotlinlang.org/docs/jvm-test-using-junit.html/ (Accessed Aug 2021).