

University of Economics, Prague
Faculty of Informatics and Statistics
Department of Information Technologies

Study programme: Applied Informatics

Field of study: Applied Informatics

Comparative analysis of mobile applications for learning Java programming language

BACHELOR THESIS

Student : Sudarikova Svetlana

Vedoucí : Ing. Jarmila Pavlíčková, Ph.D.

Oponent : Ing. Zuzana Šedivá, Ph.D.

2017

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně a že jsem uvedla všechny použité prameny a literaturu, ze které jsem čerpala.

V Praze dne 3. května 2017

.....

Sudarikova Svetlana

Acknowledgments

First and foremost, I thank my supervisor Jarmila Pavlíčková for her insightful comments and valuable guidance extended to me.

I would also like to thank my husband for his support and encouragement.

Abstrakt

Táto bakalárská práca zkoumá možnost použití mobilních aplikací k podpoře výuky programování v Javě a zjednodušení učebního procesu pro studenty. Hlavním cílem této práce je porovnání mobilních aplikací aktuálně existujících na trhu pro výuku programování v jazyce Java a zjistit, jestli některé z ní mohou být použity jako instrument podpoující výuku programování v jazyce Java na univerzitě. Teoretická část obsahuje výzkum, cílem kterého je zjistit místo tohoto typu aplikací mezi ostatními a zjistit, jaké jsou názory na jejich efektivitu mají odborníci a uživatelé. Druhá část je praktická, obsahuje vyhodnocení dotazníků, kteří vyplňovali studenti Vysoké školy ekonomické v Praze, cílem kterého je zjistit názory a aktuální zkušenosti studentů s tímto typem aplikací. Následuje porovnání mobilních aplikací, které mohou být využity k výuce jazyka Java.

Klíčová slova

Mobilní aplikace, e-learning, Java

Abstract

The bachelor thesis explores the possibility of using mobile applications to make learning programming in Java language a better and easier experience. The main goal of the thesis is to compare mobile applications for studying Java programming language that currently exist on the market and to analyse if any of them can be used as a viable supporting tool for studying Java programming. Theoretical part of the thesis is focused around research that is meant to clarify what place does this type of mobile applications have among others and what is the current position among other researchers on the topic of their effectiveness. The second part is practical. It contains results of questioning that was carried out among the students of the University of Economics in Prague in order to figure out what their opinions and current experience with mobile applications of this type are and comparative analysis of a set of mobile applications that may be used for learning Java language.

Keywords

Mobile application, e-learning, Java

Contents

1	Introduction	3
1.1	Choice of the topic.....	3
1.2	Goals	3
1.3	Structure	3
1.4	Methodology.....	4
1.5	Limitations.....	4
1.6	Target groups.....	5
2	Theoretical basis	6
2.1	M-Learning or mobile learning.....	6
2.1.1	Definition of the term.....	6
2.1.2	Effectiveness of m-learning.....	6
2.2	Mobile applications	7
2.2.1	Definition of the term.....	7
2.2.2	Statistics	8
2.2.3	Mobile applications for studying.....	9
2.3	Java programming language.....	9
2.4	Related works	12
3	Practical part: questioning of students with programming experience	14
3.1	Preconditions.....	14
3.2	Results.....	14
3.2.1	Basic characteristics of respondents group.....	14
3.2.2	Tools and materials used for learning Java language	17
3.2.3	Beliefs and opinions surrounding mobile applications for learning Java language	21
4	Evaluation of individual mobile applications for learning Java language	26
4.1	Selection criteria.....	26
4.2	Evaluation method.....	27
4.3	Mobile applications on Android and iOS platforms	29
4.3.1	“Learn Java” (version for Android)/ “Learn Java Pro” (version for iOS) from SoloLearn Inc.	29
4.3.2	“Programming Hub, Learn to code” (version for Android)/ “Programming Hub” (version for iOS) from Nexino Labs.....	32
4.3.3	“Enki” from Enki Labs.....	34

4.4	Mobile applications on Android platform.....	36
4.4.1	“Java Programming” by Akshay Bhange	36
4.4.2	“100+ Java Programs with Output” by Code Refer/Vamsi Tallapudi	38
4.4.3	“Learn Java” by Thiyagaraaj M	40
4.5	Mobile applications on iOS platform.....	43
4.5.1	“Code Recipes” by Fedor Obratzsov.....	43
4.6	Quantitative evaluation	44
4.7	Common traits and interesting trends	44
5	Conclusion.....	46
	List of terms	47
	Bibliography	48
	List of graphs.....	51
	List of tables.....	52
	Appendix 1. Questionnaire of students (English and Czech language versions)	53
	Appendix 2. Evaluated mobile applications – design	56
	“Learn Java”/ “Learn Java Pro” from SoloLearn Inc.	56
	“Programming Hub” from Nexino Labs.....	56
	“Enki” from Enki Labs	57
	“Java Programming” from Akshay Bhange.....	57
	“100+ Java Programs with Output” by Code Refer/Vamsi Tallapudi.....	58
	“Learn Java” by Thiyagaraaj M	58
	“Code Recipes” by Fedor Obratzsov.....	59

1 Introduction

1.1 Choice of the topic

This bachelor thesis is devoted to mobile applications that may be used to help a person with studying Java language.

The choice of the topic was in no way accidental: I studied the basics of Java programming during my second year at the University of Economics in Prague. Looking backwards I can say that the way the course is structured helped me a lot during my first encounter with programming language. At the same time, there are certain limitations to what a university course can give a student: it can give a basis, an insight, but the time that can be spent on practice is rather limited. In my search for practical experience with Java language I went through a variety of books, websites, mobile applications.

In this work, I wanted to concentrate on mobile applications for two reasons. First is the flexibility this learning tool offers: it can be used everywhere and almost by everyone. The second reason is quite personal: it was a surprise to find out that mobile application can be a worthwhile learning tool, and I'd like to learn more about potential and effectiveness.

1.2 Goals

The main goal of the thesis is to compare mobile applications for studying Java programming language that currently exist on the market and to analyse if any of them can be used as a viable supporting tool for studying Java programming.

To achieve this goal I have created a questionnaire to figure out what experience with learning a programming language by the means of mobile applications do students currently have and what factors may influence their choice of such a mobile application.

After that I defined criteria for primal selection process to choose mobile applications for comparison, defined criteria for evaluation, went through evaluation process and, as a result, created a list of mobile applications that can be used to make learning programming in Java a better and easier experience.

1.3 Structure

This bachelor thesis is divided into chapters. Each chapter has its' own topic.

First chapter is introductory chapter, after that follows theoretical chapter. It contains results of my research concerning mobile applications used for studying: what place does this type of mobile applications have among others and what is the current position among other researchers on the topic of their effectiveness.

The third chapter is a more practical one, it is devoted to survey that has been carried out specifically for this work. This survey was carried out to get a fresher set of data about current situation with mobile applications and to get a more unbiased view.

The fourth chapter contains the comparison of mobile applications used for studying Java language. It is partly based on the results of survey from the third chapter.

1.4 Methodology

Theoretical part of this bachelor thesis is based on comparative analysis of information from open sources that are available freely accessible online or in-print. The main sources are scientific papers and open data published by Apple and Google companies about mobile applications they are offering through their stores.

A part of information this bachelor thesis is based on was gained by analysing answers gained from interviewing students of the University of Economics in Prague that studied programming in Java language as a part of their bachelor studies.

This study also contains a comparative analysis of several mobile applications used for studying programming in Java language. Comparison is based on criteria defined in previous chapters, the main sources of information are materials published by the owners of said applications and applications itself (i.e. practical experience).

1.5 Limitations

The idea behind this work was to help potential learners and beginners in Java language programming and offer them tools that may make the process of learning easier. This work does not contain a detailed description of Java language features. The main focus of this work is on helping to study the basics. To define what is a “basic” Java course 4IT101 “Programming in Java Language” taught at the University of Economics in Prague is used (further details about this course may be found at java.vse.cz). The specifics of Java language programming and the reasons why a person may want to study it are investigated only as far as it may be needed in order to gain the understanding what should be taken into account while choosing an instrument for studying.

The work is focused on mobile applications used a helping tool for studying Java. What exactly is considered a “mobile” application will be defined later in this work.

The work is not aimed to help parents trying to teach children a programming language by the means of mobile applications, all the applications researched in this thesis are aimed at adults.

This work may also in some way be influenced by the fact that most of the sources used in this work are in English, so there is a possibility that programmers that are outside of English-speaking community and their experiences with studying programming in Java language may not be taken fully in account.

1.6 Target groups

First and foremost, the work is aimed to help university students that are interested in IT sphere in general and Java language programming specifically by offering them new tools for learning programming logic. Mobile devices give an opportunity to create a more flexible schedule of lessons, often offer information in a smaller, “bite-sized” portions - basically, they are suited to the rhythm of modern life.

Students are not the only group that may find this work helpful. Although students are the focus of this work, anyone interested in learning Java language or seeking additional practise in it may find something useful.

The results may be used to improve existing university courses by offering additional tools to a “traditionally” organised courses.

2 Theoretical basis

2.1 M-Learning or mobile learning

2.1.1 Definition of the term

M-learning (or mobile learning) is a term that is sometimes used inappropriately. Online McMillan Dictionary sees this term as a buzzword - popular term everyone seems to be talking about and often used to attract attention - and defines m-learning as *“learning methods and materials that involve the use of mobile phones or handheld computers”*. A lot of definitions I’ve seen contain a reference to mobile devices and wireless networks, by using which learning becomes “mobile”.

Of course, there are other definitions. By UNESCO definition mobile learning is learning that *“involves the use of mobile technology, either alone or in combination with other information and communication technology (ICT), to enable learning anytime and anywhere”*. This definition is a more general one as it doesn’t contain a closed list of technologies or devices to be used. It focuses more on what characteristics mobile learning has than on what exactly is used to achieve them.

M-learning itself is a term that covers a lot of combinations of studying conditions and strategies, groups of learners and technology used. In this work, I am looking at a specific type of software (mobile applications that are aimed to be used for learning a specific skill) used as a supporting learning tool (usage has recommended and not mandatory) outside of class, during the time for individual study (or self-study).

2.1.2 Effectiveness of m-learning

Since m-learning is a very broad term, it is not easy to evaluate its` effectiveness as a whole. There are a lot of scientific studies devoted to the subject, from very specific to meta-analysis ones.

For example, it is known that one-to-one computer programs (it’s a type of study programs where every student gets a laptop and is actively encouraged to use it during studying) show very good results. The analysis of 4 such studies presented by Bebell and O’Dwyer (2010) revealed that average grades in most schools with one-to-one computer programs are much better than in ones without, students` engagement in study process is deeper than students from other groups have. But, at the same time, many studies (e.g. Fleischer, 2012) show that study results depend on the method that is used by teacher, by the way mobile devices are included in the process of learning/teaching, by the preconceptions teachers have about m-learning. So can m-learning be considered effective as a whole or not?

Several meta-studies have attempted to answer this question. One of the latest meta-studies on m-learning effectiveness was carried out by Taiwan researchers Yao-Ting

Sung, Kuo-En Chang and Tzu-Chien Liu (2015). It analyses 110 works that were published in peer-reviewed journals on the topic of m-learning between 1993 and 2013. The study showed that *“the overall effect of using mobile devices in education is better than when using desktop computers or not using mobile devices as an intervention”* (Sung, Chang and Liu, 2015). More than that, effect is greater when mobile device is handheld (in comparison to laptops).

What is interesting, latest papers (e.g. Sung, Chang and Liu, 2015) indicate that most of the attention that has been paid to mobile learning so far was concentrated around the effectiveness of content learning. That means that there may be a potential lying in the direction of perfecting higher-level skills and abilities (e.g. problem-solving, analytical thinking). American researchers Zydney and Warner (2016) in their research noticed that *“more studies are needed to assess students' higher-level cognitive outcomes, cognitive load, and skill-based outcomes such as problem solving”*. So far the effect m-learning has on developing such skills remains mostly unknown.

2.2 Mobile applications

2.2.1 Definition of the term

There are several ways how the term “mobile application” may be defined. Online Cambridge Dictionary defines mobile application as *“a software program that runs on a mobile phone”*, but there are some questions that may be asked concerning this definition. Why just mobile phones? There are other devices that are considered mobile nowadays (e.g. tablets, laptops), some of which are closer to mobile phones, whereas others are closer to stationary computers in their capabilities.

Several online sources (e.g. Technopedia, 2017) define mobile applications as *“an application software designed to run on mobile devices”*. Basically, the best way to define what “mobile application (app)” is to define 2 terms it combines: “application software (or application program)” and “mobile device”.

An application software is a computer program (or collection of programs) that is *“created for a specific purpose”* and *“used by end users”* (Definitions of Information Technology Terms, 2017) (as opposed to “system software,” which provides the infrastructure in a computer). Online version of Business Dictionary (2017) adds to this definition that it is *“complete, self-contained computer program”*, but most definitions are centred around the idea of useful program created specifically to perform tasks for end users.

Defining what “mobile devices” are is a more complicated task: the term is often used to refer to mobile phones, smartphones and tablets (devices that can be held in a hand), but, depending on context, it may also encompass laptop computers or other portable, electronic products.

Since it may be impossible to define clearly what devices belong to the “mobile”

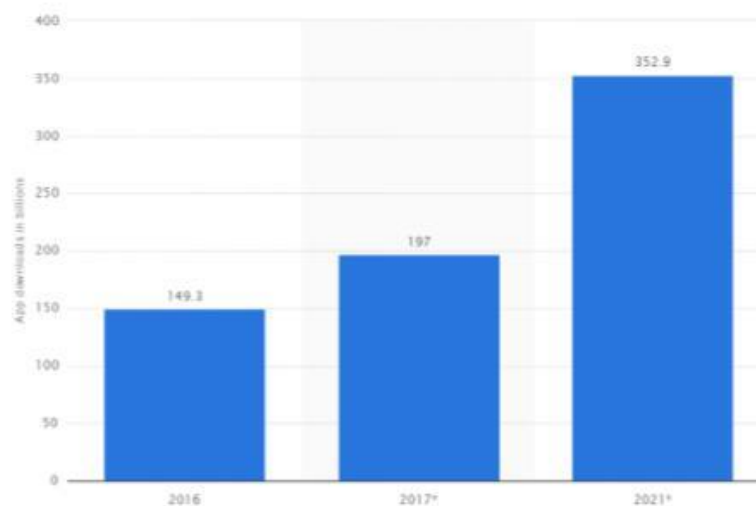
category, this bachelor thesis defines what mobile applications are not by devices they are supposed to be installed on, but by the source they come from - app stores.

App store is a digital platform used for distributing application software, usually presented in a form of *“online portal through which software programs are made available for procurement and download”* (WhatIs, 2017).

Although term “app store” itself doesn’t necessarily mean that distributed software is “mobile” (e.g. there are enterprise app stores that have different software for employees of a company), there are certain stores specifically aimed to offer applications for mobile devices. On the other side, almost all mobile applications are distributed through app stores, and the biggest ones belong to major mobile operating system vendors, because it gives them control over the software available on their respective platforms (WhatIs, 2017).

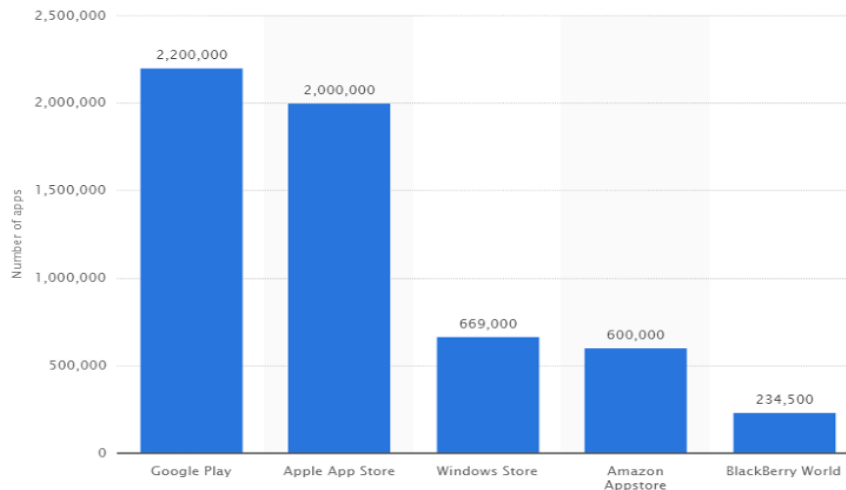
2.2.2 Statistics

The number of mobile applications has grown rapidly in the recent years. So far the experts believe that the growing trend is likely to stay the same in the future (Statista, 2017), see graph 1 (columns marked by “*” character contain estimated numbers).



Graph 1. Number of mobile app downloads worldwide in 2016, 2017 to 2021, in billions (Source: Statista, 2017)

The biggest two of app stores are Google Play (2,6 million in December 2016) and Apple App Store (2,2 million apps in January 2017) by information from Statista (2017). Graph 2 shows the difference between these two app stores and their other closest competitors.



Graph 2. Number of apps available in leading app stores by June 2016 (Source: Statista, 2016)

The gap is so big that it seems unreasonable to analyse mobile applications from stores other than Google Play and Apple App Store: the probability that potential users will have mobile devices on other platforms is rather small.

2.2.3 Mobile applications for studying

“Education” category in app stores is usually not the biggest one (the biggest category is often “Games” - about 20% in three biggest app stores), but it definitely has its’ niche. In Apple App Store it has the 3rd place (after Games and Business respectively) with 8,6% of total amount of apps (Statista, 2017). Educational applications in Play Store are amounted to a similar share of 8-9% (Dogtiev, 2016).

Mobile applications used for studying is a serious trend. Experts connect their popularity with the values of a group called “millennials”. In annual report “KPCB Internet Trends Report” (2015) was revealed that 22% of this group chose to receive training over flexible hours, cash bonuses and even a company car.

When it comes to IT there is another reason why: the evolvement of new technologies is a rapid process, a specialist needs to be constantly studying new technologies if he wants to be in demand (Casuto, 2016).

The content of the category “Education” is very diverse, it includes applications for studying languages (both natural and programming), gaining skills (driving, drawing, making origami etc.), educative games for children, e-books and many others. The criteria that were used to choose what applications are going to be analysed are defined in the practical part of this work.

2.3 Java programming language

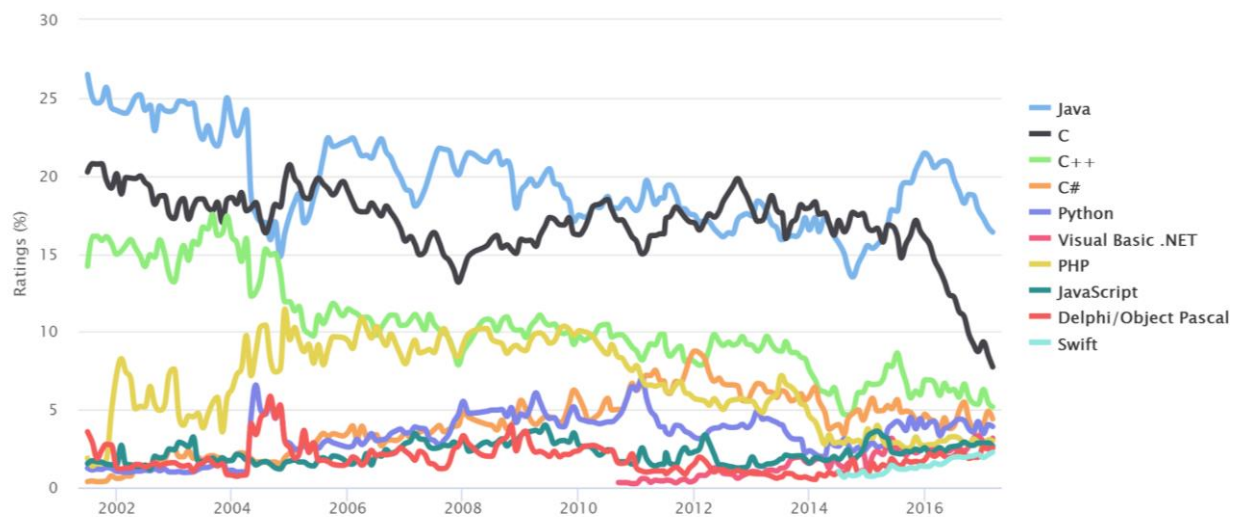
The Java programming language is characterized by its’ current owner - the Oracle Corporation - as a “*general-purpose, concurrent, strongly typed, class-based object-*

oriented” language (Oracle, 2017).

Since it was created in early 90s (project was started in 1991, first public release - Java 1.0 - happened in 1995) by Sun Microsystems. Since then Java language has gone through a series of versions. The most recent one - Java SE 8 - appeared in 2014 and gained new features (whole list of new features can be found on Oracle website) including elements from other languages and frameworks (Oliver, 2013). The Java language is developed and supported by Oracle Corporation.

Java language is one of the most common and popular languages in a modern world.

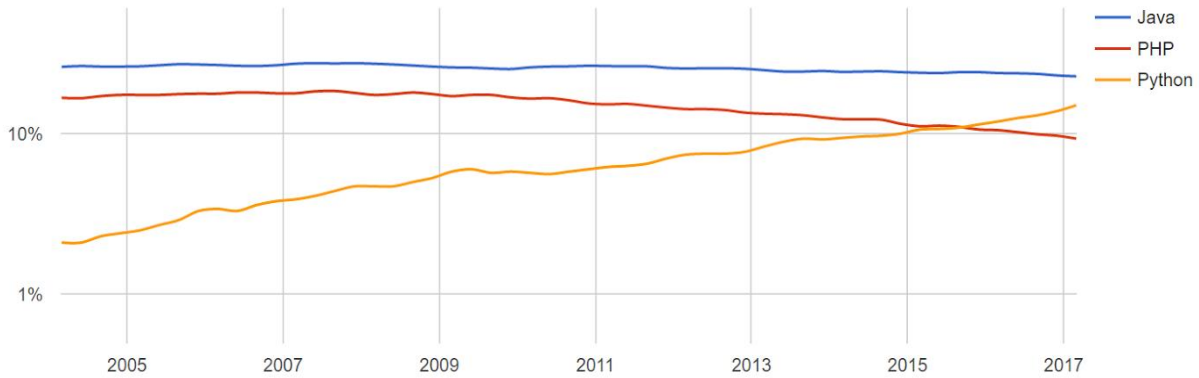
The TIOBE Programming Community index that uses popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu to calculate the ratings places Java at the 1st place (16,4% of all search queries, -4,1% change) as the most popular programming language (TIOBE, 2017).



Graph 3. The TIOBE Programming Community index from 2002 to March 2017 (Source: TIOBE, 2017)

TIOBE index is the good indicator of what people are interested in, but it doesn't answer question like "why exactly are people interested" or "what groups of people are interested".

Similar to TIOBE index is the PYPL Popularity of Programming Language index which is created by analysing how often language tutorials are searched on Google (it uses a raw data from Google Trends). This index uses just one main source of data and that makes it a bit unreliable (at least as stand-alone indicator), but in context it may be useful since it shows interest of those who has recently began learning language or is interested in learning it. According to the PYPL website, Java language is also at the 1st place by popularity (22,7% of all queries, -1,4% change to a previous year).



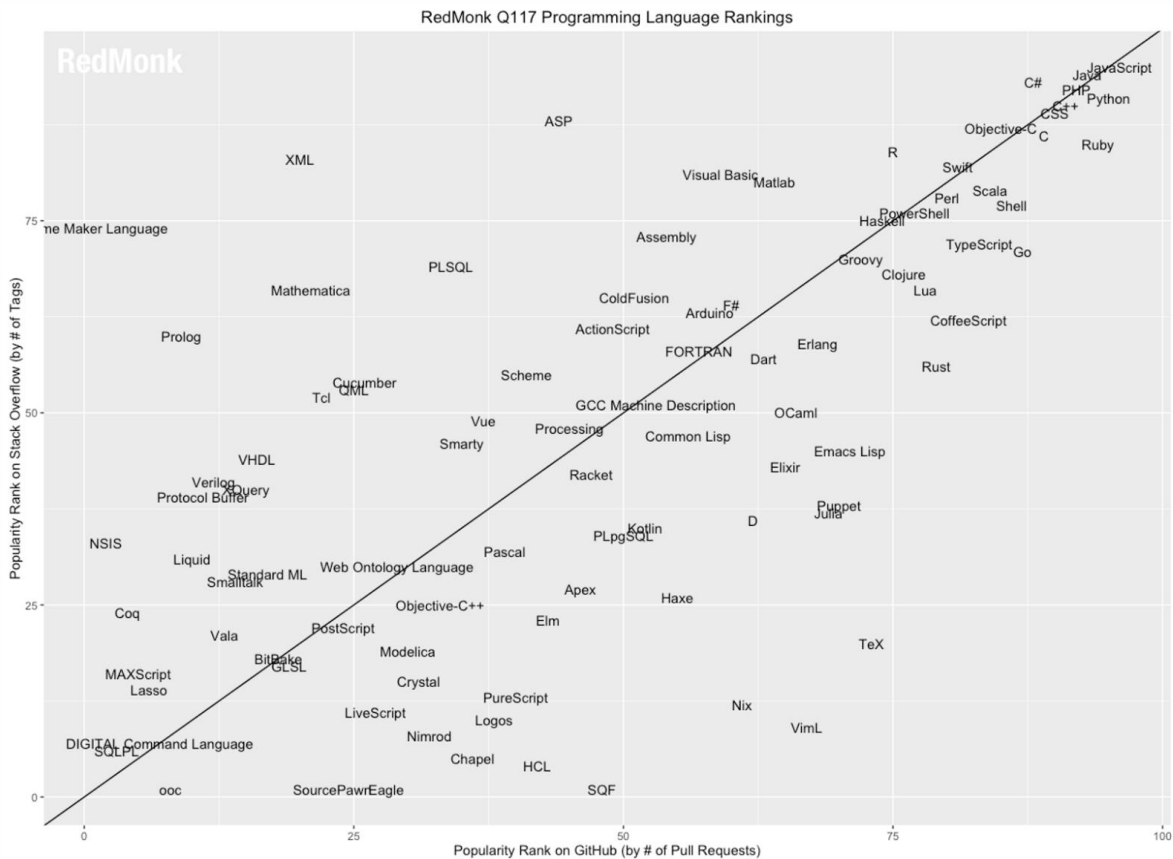
Graph 4. The PYPL Popularity of Programming Language index from 2005 to March 2017 (Source: PYPL, 2017)

What is interesting, both indexes show a slow decrease of Java language share. Partly it is connected to the natural process of new programming languages being born (e.g. Swift): those are not absolute numbers, but relative ones.

The two indexes mentioned above show Java language popularity among a wide group of people, but for figuring out how popular is Java language among the people inside programmer community may be used a ranking created by the RedMonk analyst firm. RedMonk Programming Language Rankings offer analysis of data from well-known in IT community portals GitHub and StackOverflow. The idea behind this ranking was not to create a statistically valid representation of programming language usage, but to see a language from two viewpoints: how much it is discussed (StackOverflow) and is used (GitHub).

According the RedMonk ranking (2017) Java language is the 2nd (after JavaScript) language that lies on the diagonal in the upper-right corner of it. The RedMonk ranking is published bi-annually, and previous rankings from 2015 and 2016 show the same picture (RedMonk, 2016).

There are other programming language popularity rankings and indexes that place Java somewhere around the top: the IEEE Spectrum's ranking (2nd place if predefined weightings are used, evaluation combines 12 metrics from 10 online sources) (IEEE Spectrum, 2016), TrendySkills' analysis of job postings in USA, India and 10 European countries including the Czech Republic (1st place by the amount of job postings from the beginning of 2017) (Diakopoulos and Cass, 2016) and some of the others.



Graph 5. RedMonk rating, results for 2017 (Source: RedMonk, 2017)

The popularity of Java language is caused by many reasons and factors, not the least of them is the fact that Android applications are often written in Java. The rapid growth of Play Store's number of apps alone is a strong driver behind the interest in this language and demand on new and skilled Java programmers.

2.4 Related works

To understand the true value of any academic work it is important to see it in context, to consider other works that have been written on the topic (or related topics) before. A brief overview of academic works on the topic of m-learning's effectiveness was presented in part 2.1.1. This part contains an overview of works that were written on a similar topic or somehow related by their contents to the current bachelor thesis.

I have looked through open records of three major Czech universities that have study programs in IT field – the University of Economics (Prague), Czech Technical University (Prague) and Charles University (Prague) – and tried to find any works written after 2010 that are focused on tools and instruments that are aimed to help university students learn Java programming language.

While I haven't been able to find such works that are devoted to mobile applications specifically, here are works that can also be interesting for those who are reading this work:

1. Bachelor thesis on the topic “Vizualizace objektů a datových struktur v Javě” (“Visualisation of objects and data structures in Java” – author’s translation) written by student of Czech Technical University (Prague) Martin Rejman (2010) is focused on creating an instrument for visualising specific features of Java language. This instrument can be used as teaching aid in demonstrating those features.
2. Bachelor thesis on the topic “Application for Teaching of Object-oriented Design Principles” written by student of Charles University (Prague) Marie Píchová (2010) is focused on creating instruments for education of object oriented programming. While this work isn’t, specifically, devoted to Java programming language, it is an interesting example of what supporting tools may be used for teaching and learning an object-oriented programming language.
3. The University of Economics (Prague) has several works devoted to perfecting the course 4IT101 “Programming in Java Language”:
 - a. Bachelor thesis on the topic “Navrhněte počítačovou hru pro vstupní kurzy programování” (“Design a computer game for beginner programming course” – author’s translation) by David Sedláček (2015) is focused on designing a project that is to be used for teaching students with minimal previous experience in programming.
 - b. Master thesis on the topic “Doplnění vývojového prostředí BlueJ o funkce využitelné v úvodních kurzech programování” (“BlueJ development environment extension functions used in introduction programming courses”) by Oleksandr Matviichuk (2014) is focused on updating BlueJ environment in accordance with newer features presented by Java 8.
 - c. Bachelor thesis on the topic “Alternativní výukové materiály pro vstupní kurzy programování” (“Alternative study materials for beginner programming courses”) by Závěrka Jakub (2010) is focused on perfecting BlueJ development environment and creating materials explaining some of Java features.

All the works mentioned above are focused on perfecting the way Java language is taught in universities.

3 Practical part: questioning of students with programming experience

3.1 Preconditions

Research has been carried out between students of the University of Economics in Prague by the means of questionnaire. The aim was to figure out what materials and instruments do students learning programming in Java language use, whether they use mobile applications for this or not, what are their opinion about potential mobile applications have in helping to master such a specific skill.

Before the start of the interviewing a target group was defined - people younger than 35 that studied Java programming in university. The age is limited because Java language appeared 22 years ago, so it's unlikely that there are people who studied Java language in university and are older than 35. Experience with Java language was requested specifically, so that there will be no influence caused by the possibility that interviewees learned a diverse group programming languages and their experiences also differ vastly due to this fact.

The questionnaire consisting of 13 questions (different types of questions, but mostly closed ones) was translated into Czech language, both English and Czech variants are in the Appendix 1.

3.2 Results

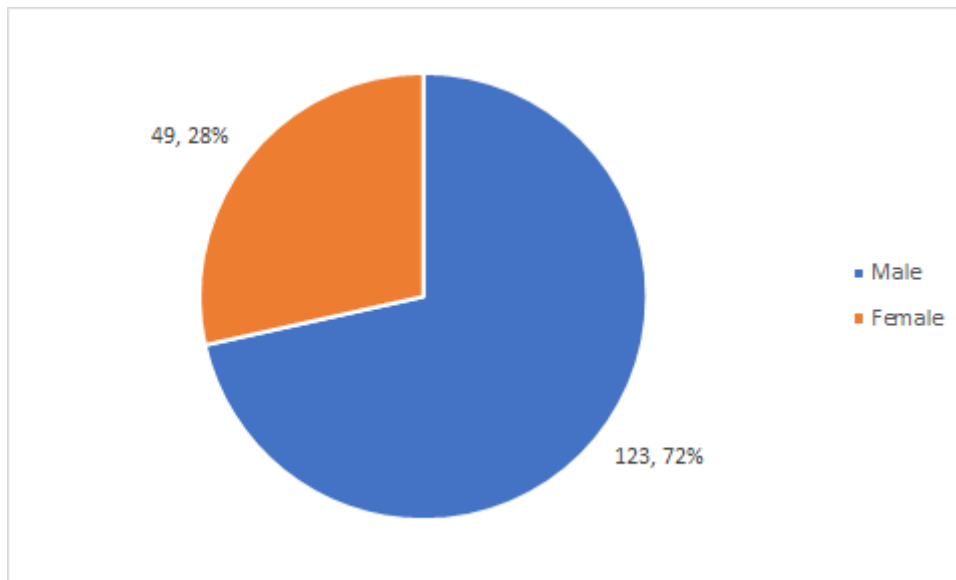
Questionnaire has been opened for a week (28.03.2017-5.04.2017), 173 respondents have answered, 172 respondents belong to a target group defined in part 3.1.

3.2.1 Basic characteristics of respondents group

First questions of the questionnaire were aimed to learn more about the group of interviewees: basic social characteristics and level of Java language programming they have.

A. Respondents by gender

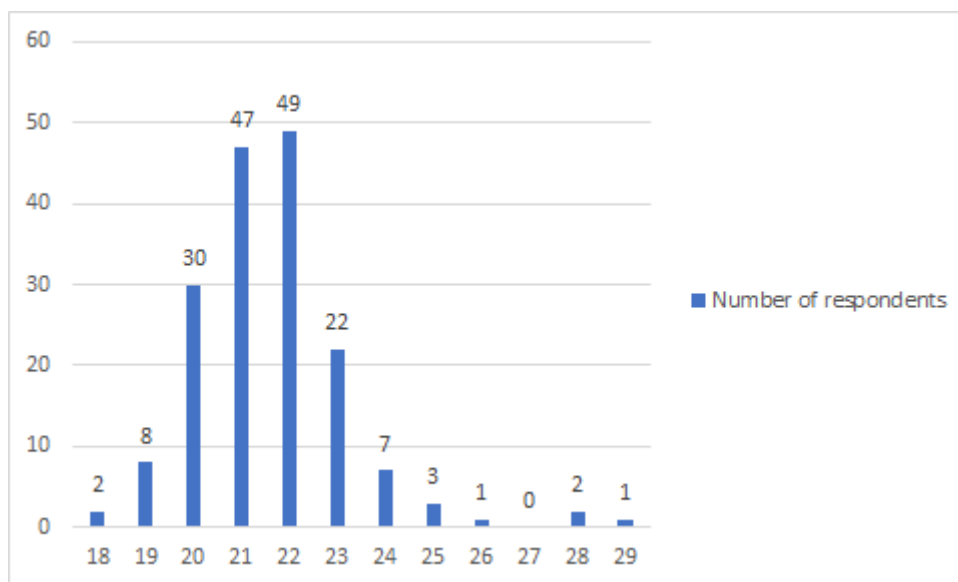
According to answers given to obligatory question 28% (49) of respondents are female (see graph 6).



Graph 6. Respondents by gender (Source: author)

B. Respondents by age

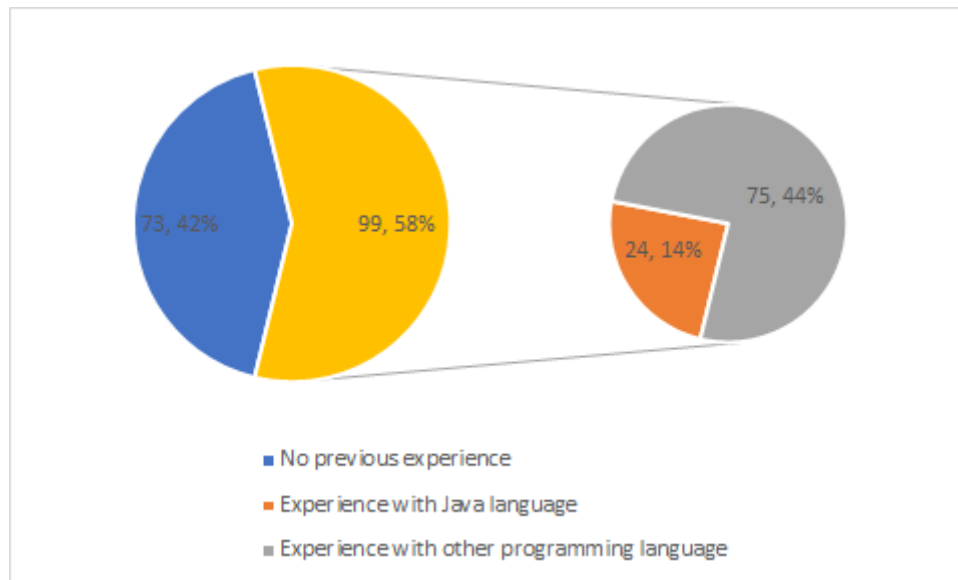
According to answers given to obligatory question 91% (156) of respondents belong to age from 19 to 23 (see graph 7). This result is expected since questionnaire was distributed mainly between the students of the University of Economics in Prague.



Graph 7. Respondents by age (Source: author)

C. Respondents' previous experience with programming

58% (99) of respondents had experience with programming before university (exact language was not specified), 14% (24) of the total number of respondents studied Java language before university (see graph 8).



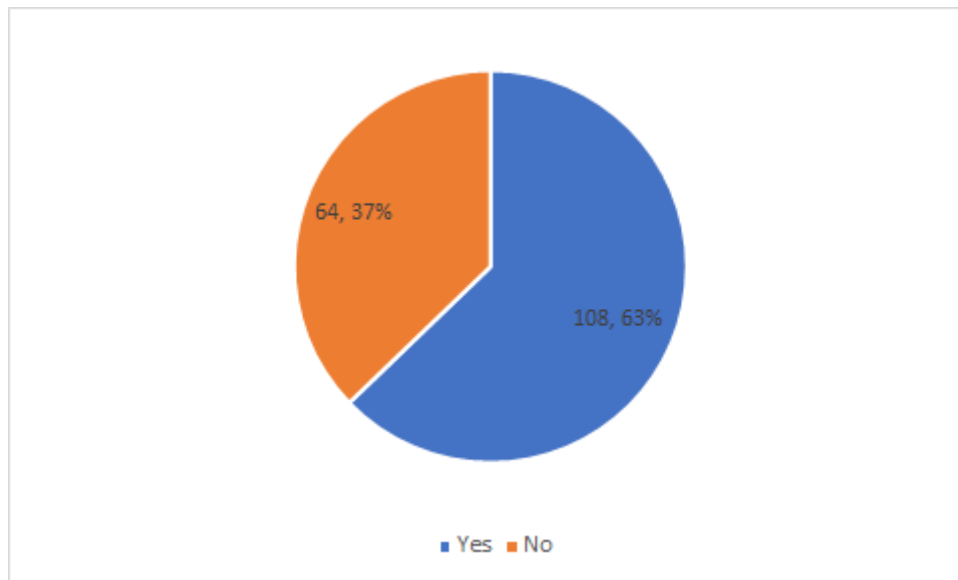
Graph 8. Respondents by previous (before university) experience with programming (Source: author)

The fact that there is a big share of students that had gained experience with programming before coming to university is indicative of several interesting moments. First, there is a big number of students in IT that have chosen that field of study deliberately and with understanding of what to expect, what to pay attention to.

What's more important, logic and basic programming concepts are similar in different programming languages, and students with previous experience in programming are able to learn new programming language quicker and with smaller effort. Such students - motivated to advance in IT field and already having experience with programming - may be interested in going forward quicker than lecture schedule allows it. In this case, mobile applications for studying programming may be very helpful allowing students to choose their own schedule of lessons.

D. Respondents by motivation for learning programming in Java

63% (108) of the total number of respondents are planning to use their knowledge of Java language programming in future (see graph 9), 10% (18) of the total number of respondents when asked about their experience with programming in Java chose answer "I've finished Java courses at university and gained some practical experience with Java language afterwards".



Graph 9. Respondents planning to use knowledge Java language programming in their future (Source: author)

This result corresponds with the fact that a big share of students come to university knowing what programming is and support the theory that there is a large group of students that come to university already planning to connect their future with IT, if not with programming exactly (i.e. IT is not a random choice).

Table 1 shows the connection between experience with programming gained (or not gained) before university and plans to use (and deepen) knowledge of Java language programming after university.

Table 1. Connection between experience with programming gained before university and plans to use knowledge of Java language programming in future, number of respondents (Source: author)

	Planning to use knowledge of Java language programming in future	Not planning to use knowledge of Java language programming in future
Had experience with programming before university (language not specified)	65	34
Didn't have experience with programming before university	43	30

3.2.2 Tools and materials used for learning Java language

Several questions of the questionnaire were aimed to learn more about how students learn Java programming: what materials and instruments (including mobile applications) do they use, what is their actual experience with mobile applications in that regard.

A. Materials and tools students use learning Java

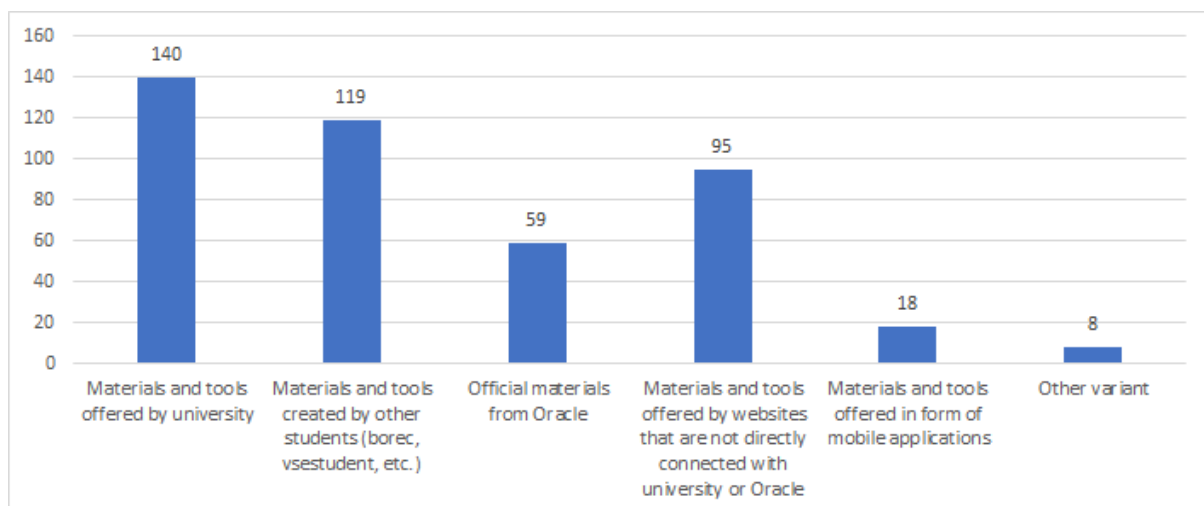
The question about types of materials respondents used while studying offered 6 variants of answer, including one “Other variant” with possibility of inputting a comment. Respondent could choose several answers.

Quick preliminary quantitative analysis shows that students use a wide variety of different material sources. Table 2 shows that several sources are used for learning Java language.

Table 2. Number of sources used for learning Java language selected by respondents (Source: author)

Number of sources used	Share of respondents with this number of sources used, %
1	16.9%
2	36.0%
3	26.7%
4 or more	20.3%

More in-depth analysis of what sources are exactly used and how often can be seen on the graph 10.



Graph 10. Materials used for learning Java language programming (Source: author)

Initially there were 18 answers “Other variant”: some respondents used this answer to give more details and name specific websites that they have used.

Most often were named 2 websites: stackoverflow.com (Q&A website containing examples of code), youtube.com (video courses and lectures). Twice was mentioned codeacademy.com - a website with different programming courses.

Among 8 answers that really didn't belong to other groups were: previous experience

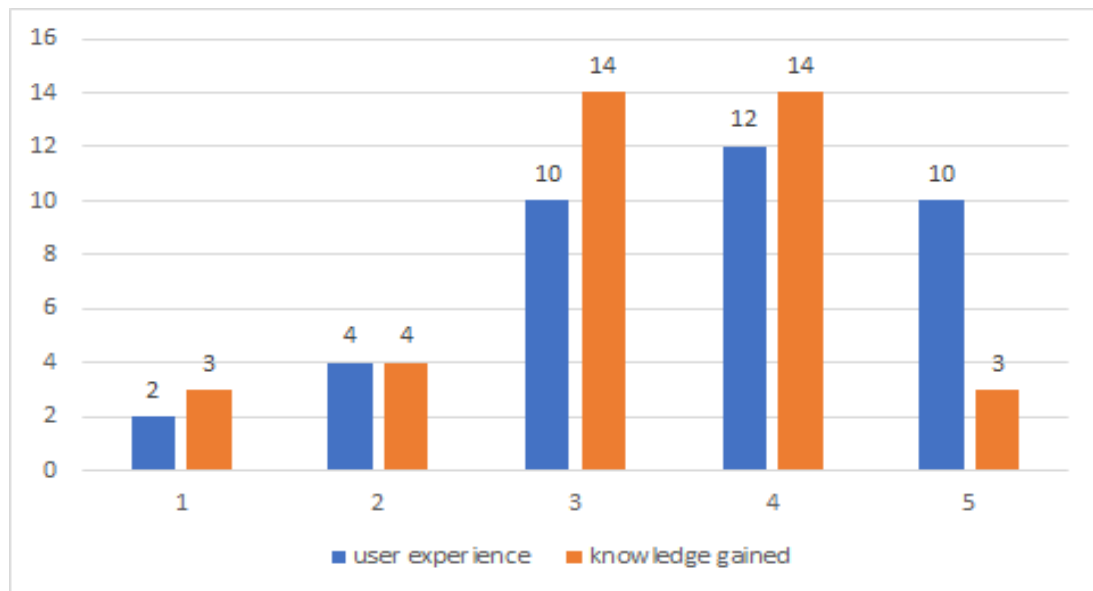
with other programming languages (3 answers) and books (2 answers).

B. Experience with mobile applications for learning Java language programming

22% (38) of respondents used mobile applications for learning programming language, in 47% (18) of those cases the language was Java (at least as one of languages).

Respondents that had experience with learning programming by using mobile applications were asked to rate their experience by two criteria: “user experience” and “knowledge gained”. Respondents rated their experience by a number from 1 to 5, where 1 meant “very negative” experience and 5 - “very positive” one.

The results may be seen on the graph 11:



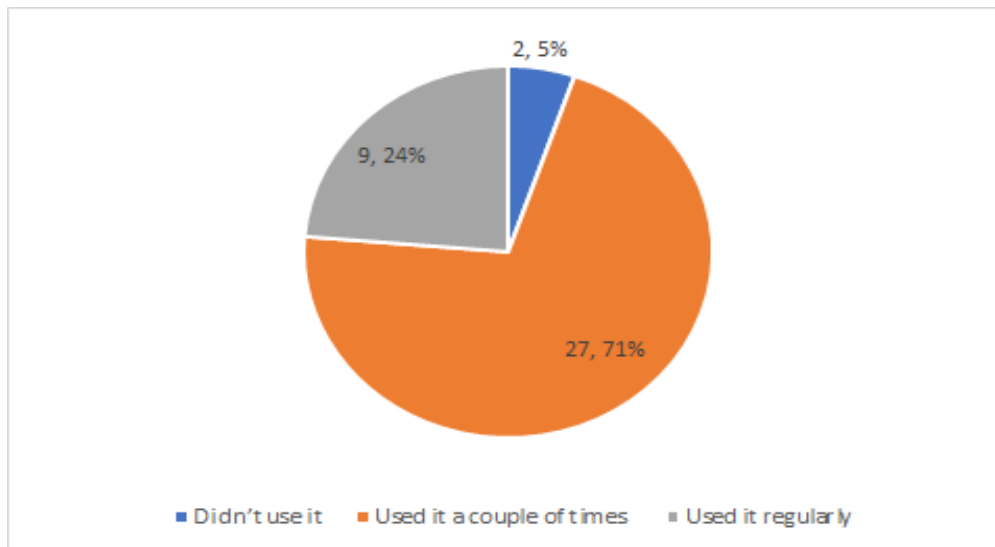
Graph 11. Previous experience with mobile applications for learning programming in Java language (1 - negative experience, 5 - positive experience) (Source: author)

The results are mostly positive, respondents evaluated their experience as average or above average for both criteria. There is certainly a place for improvement, since about a ⅓ (26% - user experience and 37% - knowledge gained) of respondents evaluated their experience as average (gave it 3 on 1-to-5 scale). Interesting is the fact that there is so little 5-positive marks for “knowledge gained” (26% - user experience and 8% - knowledge gained): that means that almost everyone who has tried learning Java this way wasn’t fully successful in achieving the main goal of using such a mobile application.

To interpret results of evaluation may help additional information about the extent of usage and goal respondents tried to achieve by installing such an application.

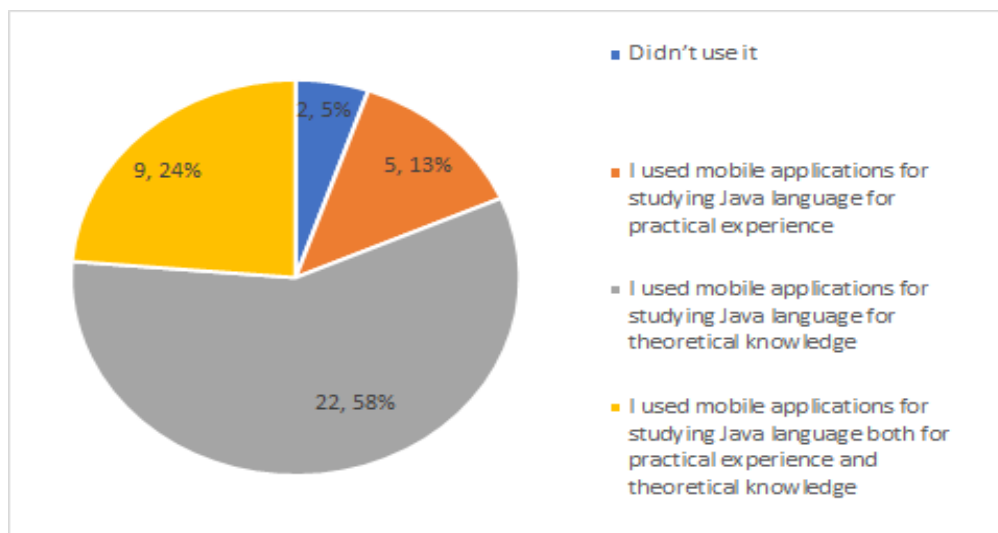
C. Usage of mobile applications for learning Java language programming

The questioning showed that only ¼ (24%, 9 of 38) of respondents who used mobile applications for learning Java did it regularly (as opposed to others who did it “couple of times”, see graph 12).



Graph 12. Usage of mobile applications for learning Java language programming - regularity (questioning of respondents who had actual experience with this type of mobile applications) (Source: author)

This result indicates that $\frac{3}{4}$ of respondents installed application, but didn't actually use it (see graph 13): either they were dissatisfied with the mobile application itself (average and above average marks given to rate experience with such mobile applications indicate that it may be at least one of the reasons) or they couldn't use it because of external factors (e.g. expected to use it in transport but were not able to do it because of the noise) or human behaviour factor (e.g. lack of motivation).



Graph 13. Usage of mobile applications for learning Java language programming - searched content (questioning of respondents who had actual experience with this type of mobile applications) (Source: author)

The results of questioning on the topic of what content respondents are searching for when they install mobile application for learning Java showed that 58% (22) of those who actually used (minimally installed) this type of mobile application looked specifically for theoretical knowledge, 24% (9) were looking both for theoretical knowledge and practical experience.

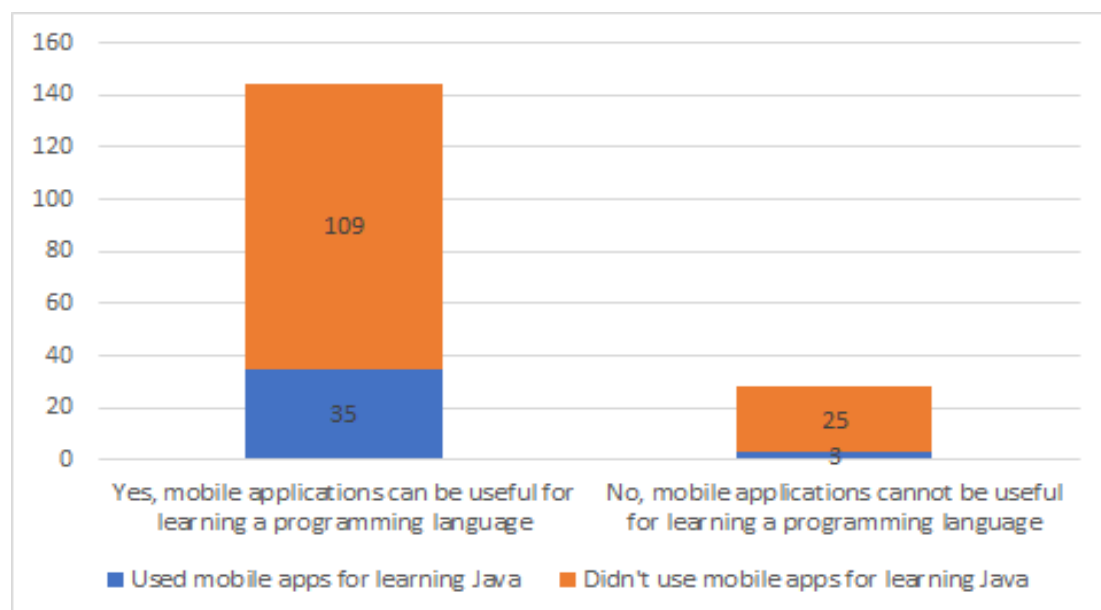
There are several possible explanations why theory is more sought for than practice: respondents may not believe in possibility of fruitful practical exercises on mobile devices, respondents may have looked for a source to consult with during practice and not practise itself or there is a possibility that respondents really felt the need to advance their theoretical knowledge first and then continue with practice. Last possibility is that respondents just were not able to successfully gain practical experience in programming because it's not comfortable to do so in mobile applications they've tried (the way question is stated it can be understood both as "what aim did you have installing such an app" and "how did you used installed app in result" - basically, the question describes the situation, but doesn't explore reasons behind it). More detailed inquiring into factors that influence the choice of mobile application for learning Java language will appear in later parts of this work.

3.2.3 Beliefs and opinions surrounding mobile applications for learning Java language

Last questions of the questionnaire were aimed to learn more about what beliefs and opinions they have on the topic of effectiveness and future mobile applications could possibly have. Every respondent was asked these questions, no matter what experience respondent had before.

A. Effectiveness and future potential

As it's shown on graph 14, 84% (144) of respondents believe that mobile applications can be useful for learning programming language (without specifying a language).



Graph 14. Respondents by opinion about effectiveness of mobile applications for learning a programming language (Source: author)

Here is a more detailed information:

Table 3. Respondents by opinion about effectiveness of mobile applications for learning a programming language

	Total amount	Used mobile apps for learning Java	Didn't use mobile apps for learning Java
Think that mobile applications may be useful for learning a programming language, number of respondents	144	35	109
Don't think that mobile applications may be useful for learning a programming language, number of respondents	28	3	25

Among those who had actually used this type of mobile application the number of respondents not believing in usefulness of this way of learning programming language is low (8%, 3 of 38). It is expected, considering the small amount of negative marks given (see graph 11). At the same time, opinion of respondents that had actually tried these applications have more basis underneath than opinions of those who didn't.

Interesting is the fact that while 84% believe in the thesis that this type of mobile applications can be useful for learning programming language, only 22% of respondents have actually used them. Either that "usefulness" isn't enough to try these mobile applications out (other materials and instruments seem more effective in comparison) or respondents just didn't think of this option when they planned their studying (learning programming through mobile applications isn't how people imagine learning programming).

To clarify this situation, respondents were asked to choose one of 4 pre-defined statements about current state of mobile applications and their potential. Table 4 contains detailed information with how answers are distributed depending on users' previous experience with mobile applications for learning Java.

Table 4. Respondents by opinion about current state and potential of mobile applications for learning a programming language, number of respondents (Source: author)

Statement	Total amount	Used mobile apps for learning Java	Didn't use mobile apps for learning Java
Mobile applications are well-suited for studying Java programming and can be used as a main (standalone) learning tool	24	5	19

Mobile applications are well-suited for studying Java programming, but as a supporting tool	109	29	80
Current mobile applications are not well-suited for studying Java, but that can possibly change in foreseeable future	28	4	24
Current mobile applications are not well-suited for studying Java, and I don't think this will change in the foreseeable future	11	0	11

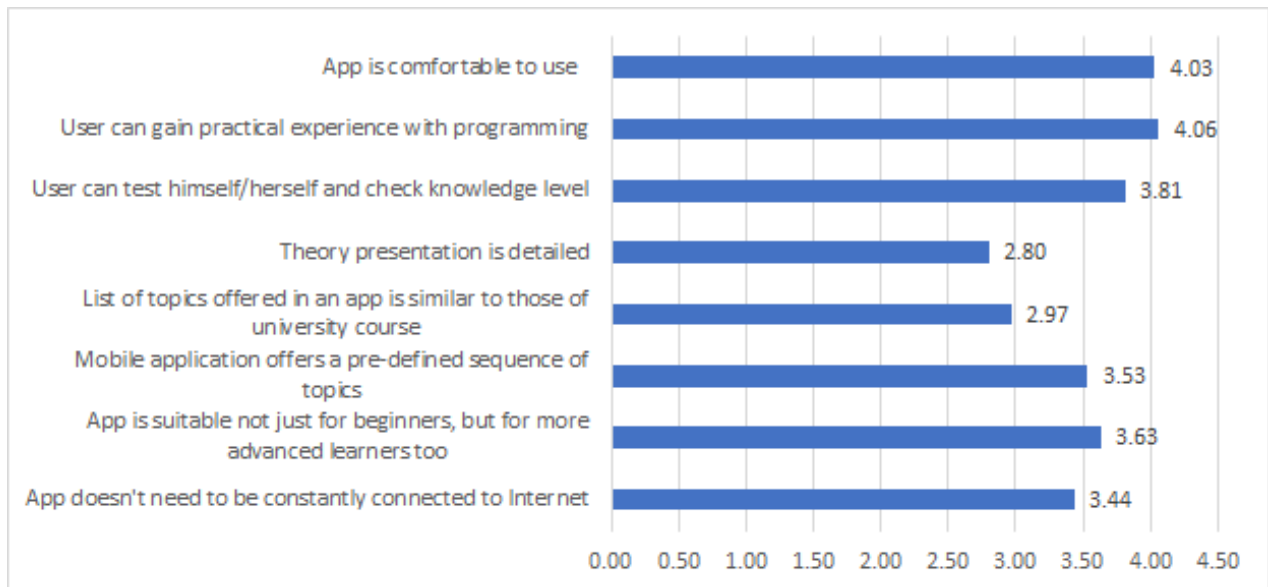
63% (109) of respondents believe that mobile applications are useful as a supporting tool, 14% (24) are ready to use mobile applications alone for this purpose.

“Sceptics” (those who are not satisfied with mobile applications for learning programming as they are now) are very interesting: only 10% (4 of 39) of sceptics have actually used mobile applications for learning programming before, and all 4 of them are inclined to believe that mobile applications have potential in that regard.

These numbers attest to the fact that despite not always impressive results of using mobile applications for learning programming, there is an interest and certain credit given to such applications.

B. Criteria for evaluation

In the last question respondents were asked to rate different variants of criteria for choosing a mobile application for learning a programming language. The question was asked to define what is more important and what is less for a person choosing a mobile application of this type. A choice of variant was formed based on criteria that are often used in reviews of mobile applications (especially mobile applications for learning languages, both programming and not).



Graph 15. Average marks given by respondents to different criteria for choosing mobile application for learning Java language programming (Source: author)

The most important criterion according to respondents' answers is "ability to gain practical experience", which is a bit surprising considering a relatively small amount of respondents (of those who had tried mobile applications for learning Java) answered that they used mobile application for gaining practical experience with programming (5% of 38 respondents sought for just practical experience, 24% - both for practical experience and theoretical knowledge). It supports the theory that the problem is not with the lack of interest in practice, but problems with realizing this intent.

On the other hand, the last place "theory presentation" doesn't mean that theory presentation itself isn't important. Previous answers (see graph 13) indicate that a big part of respondents used mobile applications for learning Java language to gain theoretical knowledge (even if partly it's due to a fact that this intent was simply easier to realize). That means that this formulation - "theory presentation is detailed" - isn't what respondents are looking for. In a context of mobile applications theory should be not just detailed (which is not a bad thing itself), but be presented in a way that is easy-to-read and understood. Basically, "theory presentation is detailed" should be understood in a context of criterion "app is comfortable to use" that is much more valuable to respondents. That may be crucial since mobile applications may be used in circumstances not ideally suited for learning (e.g. in transport).

Table 5 shows how the results look:

Table 5. Ranged list of criteria used for choosing mobile application for learning Java language (Source: author)

Place	Criterion
1	User can gain practical experience with programming
2	App is comfortable to use

3	User can test himself/herself and check his/her knowledge level
4	App is suitable not just for beginners, but for more advanced users also
5	Mobile application offers a predefined sequence of topics
6	App doesn't need to be constantly connected to Internet
7	List of topics offered in an app is similar to those of university course
8	Theory presentation is detailed

This list will be actively used in the next part of this work, where the evaluation of individual mobile applications for learning Java language programming takes place.

4 Evaluation of individual mobile applications for learning Java language

4.1 Selection criteria

Mobile applications have been chosen from a list provided by a built-in search on app store websites (Apple's AppStore and Google's Play Market). After searching by combinations of keywords "Java", "learn", "programming", "study", "language" a set of mobile applications that (from a Description) seemed to be able to satisfy the intent "to learn Java programming" has been chosen.

The selection criteria used to choose mobile applications were a bit different for Android and Apple applications because of the specifics of information presentation on the app store websites. The idea was to form a list (with 4-6 items for each platform) of applications both from Android and Apple, so that there would be a similar probability for Apple and Android users (two biggest groups of potential users, as it was stated in 2.2.2) to find something useful.

The choice of mobile applications was done in two stages: at first about 30 applications were chosen based on a list of criteria. Later requirements have been made stricter to shorten the final list: while a list of mobile applications for iOS has been formed at the first stage of selection process, amount of Android applications was too big. That is why additional condition has been added for applications on Android - number of downloads should be not less than 100 00 (Apple app store doesn't have this metric nowadays).

Here is the final list of requirements and demands mobile application had to comply to:

- A. Platform: mobile applications for mobile devices on Android and iOS
- B. Language: English and Czech
- C. Date of last update: 2 years ago or less

As it was mentioned in previous parts of this work, as an example for comparison of theoretical contents is taken a Java course 4IT101 "Programming in Java Language" taught at the University of Economics in Prague. This course includes some features (e.g. lambda expressions) that have appeared in Java 8. The date, when Java 8 became accessible to the general public, is the reason behind this limitation.

- D. Rating: more than 4 stars

Both app stores have a rating system, where maximum is 5 stars. As a standalone criterion, it's unreliable at best (a new applications may have a perfect rating, but minimal amount of downloads because owner distributes it among friends or, as an opposite example, it may have unreasonably low rating because of unsuccessful first release). The law of large numbers makes rating more "fair" and balanced.

- E. Payments: free or basic Java 8 course with practical exercises can be accessible for free

- F. Practical experience: mobile application should offer some practical experience with programming

There are a lot of mobile applications and other sources (including already mentioned above) offering explanation to concepts and principles of programming, but practice is another thing altogether. This work explores practice in Java programming that can be gained from mobile first and foremost, that's why this criterion was added to the list.

What is understood under the term "practical experience with programming"? The minimum that has been stated is reading code. It is questionable whether reading examples of somebody else's code can be considered a "real" programming practise, but it's a good way to learn different programming techniques and adopt some of the best practises, to get acquainted with way of thinking that is used in programming.

- G. Number of downloads (criterion is used just for mobile applications on Android): minimal requirement was set at 100 000 downloads.

- H. Mobile application is aimed for teaching adult people Java language programming.

4.2 Evaluation method

To evaluate a mobile application for learning Java language programming, a list of 7 basic points was formed. Every item of that list is a viewpoint, from which application is evaluated.

First point of the list is aimed to provide basic information, there is no point to compare applications between themselves based on this information. This information is offered so that reader could receive some basic statistics and learn more about applications' history.

Points 2-6 of the list are rated from 1 to 3, where 1 is "below expectations" and 3 is "above expectations". Last point - "motivational elements" - is rated from 0 to 1, since not every application has those elements and they are not essential to the process of learning Java language (even if they may be of help).

Table 6. 7 evaluation points of mobile applications for learning Java language programming (Source: author)

No	Characteristics	Is evaluated
1.	Basic characteristics: - Title and author(s), current version - Platform(s) - Current rating in app store - Current number of downloads (if it's available)	Not evaluated

2.	<p>Description:</p> <ul style="list-style-type: none"> - Main logical parts (content characterization) - Target groups (beginners, more advanced users, etc.) - Online/offline/combination of first two 	<p>Above-standard evaluation for:</p> <ul style="list-style-type: none"> - menu contains useful options most apps don't have, - application is aimed both for beginners and more advanced users, - can be used offline
3.	<p>Theoretical contents:</p> <ul style="list-style-type: none"> - Organisation of lessons: fixed/flexible sequence of topics, form of presenting information, etc. - Teaching value of theoretical material itself: topics covered, detalization of lectures, interesting and useful details and tips. 	<p>Above-standard evaluation for:</p> <ul style="list-style-type: none"> - application has a predefined sequence of topics, - application contains useful tips, - basic university course's topics are fully covered and there are a lot of additional information on other topics.
4.	<p>Practical exercises:</p> <ul style="list-style-type: none"> - Organization of exercises: fixed/flexible sequence of topics, form of practical exercise, etc. - Tasks themselves: different levels of difficulty, teaching value, etc. 	<p>Above-standard evaluation for:</p> <ul style="list-style-type: none"> - application requires actual entering (i.e. writing) programming code, - there are unusual and useful forms of gaining practical experience.
5.	<p>Knowledge checks:</p> <ul style="list-style-type: none"> - Placing of knowledge checks: around learning blocks or in other way - Triggering events: finish of learning block, by user's choice, happens at certain time intervals, etc. - Material knowledge checks are aimed for: newly learnt block, previously learnt material, by user's choice, etc. 	<p>Above-standard evaluation for:</p> <ul style="list-style-type: none"> - user can go through testing process and evaluate his/her current level and progress, - app contains different types of effective knowledge checks, - user can adjust knowledge checks settings according to his wishes.
6.	<p>User experience:</p> <ul style="list-style-type: none"> - Individual elements - Whole application - Navigation logic 	<p>Is evaluated based on personal experience and reviews in app stores</p>
7.	<p>Motivational elements:</p> <ul style="list-style-type: none"> - Reminders - Goal setting - System of achievements - Ability to monitor progress 	<p>Is evaluated as bonus feature - additional 1-2 points depending on the scale motivational elements are used.</p>

All the mobile applications that fit the criteria stated in 4.1 were gathered by 1.02.2017 and described below, one-by-one, following the table above. The quantitative evaluation

of chosen mobile applications appears in a table with collective results that is placed after all applications' descriptions. All the illustrations meant to show how application design looks like are in the Appendix 2 of this thesis (source: official webpages on app stores owned by Apple and Google).

Evaluation has been carried out based on open materials available in app stores (Google and Apple owned), developers' websites and my own experience. All the mobile applications that are described in part 4 of this thesis were downloaded and tried out. As it was already mentioned before, as an example of an introductory Java course was used course 4IT101 "Programming in Java Language" taught at the University of Economics in Prague (the contents of the course are presented by address java.vse.cz).

4.3 Mobile applications on Android and iOS platforms

4.3.1 "Learn Java" (version for Android)/ "Learn Java Pro" (version for iOS) from SoloLearn Inc.

Basic characteristics:

- Current version: 3.3.8 - Android version, 2.3 - iOS versions;
- Current rating: 4,7 - Android version, 4+ - iOS versions;
- Current number of downloads: between 1,0 and 5,0 millions - Android version.

Description:

SoloLearn Inc. is a well-known creator of mobile applications for learning different programming languages: Python, C++, JavaScript, etc. According to information from the company's website (www.sololearn.com), the application has more than 15 million users (Java course itself contains more than 2,2 million users). Applications made by SoloLearn are oriented on creating a community between learners of programming language: applications have a lot of "social" elements that motivate user to involve friends and seek help from those who are already a part of community. Nonetheless, user can enable offline version in settings if he wants to.

Mobile application has the following main parts (i.e. parts that are directly related to learning programming):

a) Tutorial

The main part of application (it is placed on a "Home" screen) that offers user a sequence of theoretical "bite-sized lessons". "Bite-sized lessons" is a specific format of presenting theoretical information in which small portions of theoretic texts are followed by a quick training task in form of test, example or practice exercise. The structure of the course is suitable for those who has never programmed before.

b) Code playground

Code playground is a place where user can try out his own code or an example from Java

tutorial (examples from Tutorial section have a link to Playground for a quick practice) and see the actual output created by a running program. User can see code created by other users and share his own, can save his code and return to it later. To make entering and reading code easier the application automatically highlights the keywords, offers hints, closes the brackets. User can find examples by keyword search, code examples can be rated positively or negatively by other users.

c) Q & A discussions

Q & A discussions is basically a forum where users discuss programming-related questions. Navigation through topics is made through a system of tags, additional information (number of comments, author, etc.) is shown near every topic.

d) Glossary

Glossary contains a very brief list of terms user is supposed to learn in a Tutorial. Terms are divided into groups by topics, but not ranged by alphabet.

Theoretical contents:

Theoretical contents of the application are presented by Tutorial, Q & A discussions and Glossary sections of the applications. Tutorial contains 6 modules that consist of 70 lessons supported by 152 quiz questions (some of those quiz questions may be seen as small practical tasks). Topics have a predefined sequence, but user can skip some modules by taking a shortcut.

As it was mentioned before, contents of the course are suitable for a beginner and cover the bigger part of basic Java course (some minor topics are not present e.g. lambda expressions). At the same time, from the list of topics covered by this application it becomes clear that it has nothing to offer to more experienced programmers: no useful tips and tricks, no further material - this course has no value after you've gone through it once. More than that, it is doubtful that the community SoloLearn relies so much on can be of real help since it consists mostly of beginner level programmers - what would more experienced programmer do there?

My personal experience shows that the hardest theoretical parts are the first ones: clueless user doesn't know what is important and what is not, course's creators sometimes don't know where to start and what to explain first. In that regard, this application was almost perfect: while first lessons were purely theoretical and a bit boring to me, the flow of explanation seemed logical to me, nothing major hasn't been lost (even when the format of "bite-sized lessons" itself forces creators not to go into details unless it's necessary). The chosen format is - in a way - an optimization to mobile devices, and that affects how easy it is to acquire new knowledge and skills.

Another positive moment is a close connection between theory and practice in this app. Despite the low-quality level of some practical tasks, the fact that user can try out what he has just learnt costs a lot. Motivated student who has some previous experience can gain a lot from this format of small read-and-try out blocks.

Glossary part of the application is poorly made: it contains only basic definitions

without links to the tutorial lessons, navigation through this part is lacking.

Practical exercises:

Practical tasks of the course can be found in Tutorial and Playground parts of the application.

Practical tasks in Tutorial part are not doing what they were created for. These tasks follow a specific theoretical lesson and created in this place of the application to help user both to remember (by repeating) the lesson and to check the knowledge that has been gained. Unfortunately, the way exercises are formulated makes it easy to go through them without understanding material. Very often knowledge checks are not focused around places where most of mistakes occur (e.g. in a topic with loops one task checked that user knew how to write command “System.out.print()” right, but not loop itself). These tasks do not check the understanding of material and just offer very basic syntax learning and training. Basically, these practical tasks not only not helping user in learning real programming, but also creating a false sense of self-worth as a programmer.

Practical exercises in Playground part are not actually a set of tasks. Although user can try out coding independently, there are no actual tasks he can try to figure out. User can try out examples from Tutorial that have already been explained there, look through and modify the code created by other users or he can think of something himself (herself). User has a very promising instrument in his hands, but he is unlikely to use its’ full potential within the bounds created by the application. This application can be improved greatly by adding some coding challenges, puzzles and tasks to already existing Playground.

Knowledge checks:

Question of knowledge checks has already been touched above. The application is based on constant reviewing of material learnt. Smaller knowledge checks are placed after every block to control how user has understood new material appear, bigger checks in form of quiz appear in the end of each module. If user wants to check his knowledge of previous chapters, he can return to those lessons, but there is no other form of checking the overall level of knowledge.

Despite a large amount of different question types (writing a right answer, choosing a right variant, dragging a right word to a right place), knowledge checks are not actually checking much and are very easy to get through.

The overall feeling left by these checks is that they are not actually meant to check anything in the first place, it’s just a system of positive reinforcement that is meant to create emotional connection between users and the application. Actual programming skills in this context are just a secondary goal.

User experience:

The application has very easy-to-understand structure and navigation system. It looks

good and pleasant to use. A lot of reviews mention that it's "easy to use" and has a "nice user interface".

Motivational elements:

The application in questions relies greatly in its' community and its' communication with users, so it's really unsurprising that motivational elements is a very actively used tool in this application. User is constantly reminded about the importance of practice, the whole interface is created in a way that user always sees his/her progress: how far is he/she, how much blocks and questions are left to the next level-up. Every user can see himself/herself in Leadership list and his closest neighbours.

4.3.2 "Programming Hub, Learn to code" (version for Android)/ "Programming Hub" (version for iOS) from Nexino Labs

Basic characteristics:

- Current version: 3.2.4 - Android version, 1.1.2 - iOS versions;
- Current rating: 4,3 - Android version, 4+ - iOS versions;
- Current number of downloads: between 1,0 and 5,0 million - Android version.

Description:

Nexino Labs is a company that specializes in creating mobile applications for learning different programming languages. According to information from app stores, Nexino Labs can help user with learning more than 17 languages. The application itself isn't aimed uniquely for learning Java language - user can choose different programming languages after installation has been finished. Programming Hub isn't just a mobile application, but also a website (programminghub.io).

Contents of Java language course are suitable for a beginner, but there are certain elements of the application (like part containing questions often asked at interviews) that may be helpful to more experienced programmers. According to information on both app store pages, the application doesn't need to be constantly connected to Internet (but that's not exactly the truth - Compiler doesn't work without Internet connection).

Application is meant to be used by beginner programmers, though some parts of this application may be useful for a more advanced programmer who is trying to refresh his knowledge of Java basics and prepare for an interview.

Mobile application has the following main parts (i.e. parts that are directly related to learning programming):

a. Course

Course contains a non-interactive tutorial of Java language that is similar to an e-book. Course contains both theory and examples (code and output) to demonstrate it.

b. Example programs

Examples in this part of the application are also non-interactive: code and the output are without any additional comments. Examples are divided into 14 groups for an easier search, but standard search tool is also available. Examples are not optimized to be read from small screens, so reading code may be uncomfortable and require a lot of scrolling. In the end of every group of examples there is a button - user can send a request to developers to add an example illustrating a specific question.

c. Compiler

Compiler is a place where user can try out his own code and see the output. Compiler has no tasks inside, it's a playground without aim or rules (except, of course, the rules that are naturally dictated by the programming language itself).

To make entering and reading code easier the application automatically highlights the keywords, sometimes it may hint where the problem may be.

d. Interview questions

This part contains popular questions that should check user's knowledge of how Java language theoretically works. As it's clear from the title, the format "Q & A" is used. User can add some questions to favourites and then go through a list of favourites instead of the whole one.

Theoretical contents:

Theoretical contents of the application are presented by Course part of the application. This part contains 14 groups of topics that are all accessible to user from the start (user can go from beginning, choose his own sequence of topics or use this part just as reference material).

Application's topics cover the bigger part of basic Java course, but there are certain notable absences here: not only Java 8 elements are missing, but also several design patterns (e.g. Singleton), classes (e.g. Math), work with text strings, etc. It looks like this application was meant to introduce to user main Java concepts (what is class, method, constructor, inheritance, etc.), but not to go into more specific details. Course part of the application is more suitable to be used as reference material (though it's value is quite questionable when there are just basic concepts), than as a textbook: the way material is presented to the user is sometimes strange, like there is too much and not enough details at the same time. I had difficulties with following the inside logic of single topics sometimes.

Part of the problem here is in the fact that material wasn't really optimized for mobile devices. It looks like creator of the app tried to make theoretical material more "readable" by using bullet-points, but went a bit overboard with it and that worked against them (and a potential user).

Another minor detail that doesn't help user understand how Java language programming works is the fact that examples of programs don't have notes or other type of commentary. If user couldn't understand the example, he's unable to do anything with it. Creators offer the option of sending feedback, but the waiting time is

unclear in this case.

Practical exercises:

There are two major ways in which user can practise programming using this application. First one is reading code: user can go through examples in Course and Example programs parts of the application. Second one is trying out actual creation of new code in Compiler part of it, but there are no actual tasks, challenges or puzzles in that part - just pure experimentation.

Knowledge checks:

Interview questions part may be considered a form of knowledge check: user sees the question, tries answering it and then clicks to see the right answer. That may be a relatively good way to check the knowledge of theory, but unfortunately in this application answers to some questions may seem a bit obvious (e.g. question “What is object cloning?” has the answer “Object cloning is used to create the exact copy of an object”) or not explaining anything (e.g. question “Why method overloading is not possible by changing the return type in Java?” has the answer “Because of ambiguity”)

User experience:

The application has a very minimalistic interface, so navigation is easy because of the app’s simplicity (a lot of users in their reviews on app store mentioned that it’s “easy to use” and it has a “simple interface”). The overall user experience is not great, the code sometimes does not fit on the page and you have to scroll it horizontally.

Motivational elements:

None.

4.3.3 “Enki” from Enki Labs

Basic characteristics:

- Current version: unknown - Android version, 3.14 - iOS versions;
- Current rating: 4,6 - Android version, 4+ - iOS versions;
- Current number of downloads: between 1,0 and 5,0 million - Android version.

Description:

Enki application is a bit different from most of other mobile applications that are meant to help beginner programmers learn Java language. First, application is aimed for users of different programming experience: when user opens the application for the first time, he chooses languages he wants to learn and level he’s currently at.

The idea behind this application is not in a long self-study sessions, but in small workouts. Everyday user is offered a set of small lessons called a “workout”: after a short lesson usually follows a small question or task aimed to check newly gained knowledge. After answering user rates each lesson by choosing between “Already

knew” or “Learned something” options - supposedly, it somehow influences lessons offered in future workouts. After workout is finished, user can either go through lessons from previous workouts (they are gathered in Insights tab) or play one of the games (Games tab).

Insights tab contains previous lessons tagged for easier search. User can bookmark some of them if he needs it.

Games tab is more interesting one, it contains various games. In the beginning, there are two of them (Guess the class and Thread safe?), later more games are opened (Java 8 or earlier?).

Enki application doesn't require constant Internet connection, user can download new lessons beforehand.

Theoretical contents:

Theoretical contents of this application are very rich: unique content, lessons are small, but written well. They are useful and contain not just a standard information as it is often presented in different tutorials, but tips and tricks that may be useful to a more experienced programmer.

The negative side is that there is no standard course where user can see the sequence of topics and access at least the starting ones. Lessons are chosen randomly (but within a level chosen in the beginning) and that seriously limits its' usage. The application with brilliantly chosen lessons can be used just as a supporting tool.

Practical exercises:

There are two main types of exercises: small quiz questions placed after every lesson and games. Quiz questions are usually very small, connected to the lesson's contents and can appear in many forms: fill missing parts of code in examples, write out the output, choose the right answer, etc.

Games in this application are more aimed at memorizing things: what classes are considered thread-safe in current version of Java and what are not, what elements have appeared in Java 8 and what elements existed before that, etc. It is known that later new types of games become available to the user: find a mistake in a code, fill the gaps in a code, etc.

Knowledge checks:

The whole application is based on constant reviewing of material, that's why there is a question asking if the lesson contained something new and useful. Lessons with quiz question at the end and games are aimed to check what user already knows (or remembers from previous lessons) and offer a new piece of information.

User can always see his progress on the main screen, where statistics are shown. When user gets too successful, he is offered to change his level and go further.

User experience:

Despite the fact that this application differs in navigation system from almost all reviewed in my thesis, it is easy to understand. Design is complex at the first glance (partly because main screen is basically a dashboard), but it's very easy to get used to it. Previous lessons can be easily found by the tag or a keyword.

Motivational elements:

According to reviews, daily reminders to workout are highly appreciated by users. The way statistics are shown (with "streaks" - series of uninterrupted workouts) also motivate users not to miss days of practice.

Inside the application there are also minor motivational elements: for example, if user pauses in the middle of a workout, he will see estimated time left till the end of workout.

The overall feeling left by motivational elements in this application is that they are done cleverly and annoyingly: they support the concept of workout, but do not push user too much.

4.4 Mobile applications on Android platform

4.4.1 "Java Programming" by Akshay Bhange

Basic characteristics:

- Current version: 2.0.1;
- Current rating: 4,4;
- Current number of downloads: between 500,0 thousand and 1,0 million.

Description:

Akshay Bhange is an author of 3 mobile applications for learning programming languages.

Application can be used offline, in this case user is unable to compete with other users in a quiz game. Main focus of this application are beginner programmers, the application could possibly be of use for more advanced programmers (e.g. to prepare for job interviews), but not favourable reviews of theoretical contents will probably stop them.

Mobile application has the following main parts (i.e. parts that are directly related to learning programming):

a. Tutorial

Course contains a non-interactive tutorial of Java language that reminds a book. Course contains both theory and examples (code and output) to demonstrate it.

b. Programs

Examples in this part of the application are also non-interactive: code and the output it would have without any additional comments, for easier navigation is offered a standard search tool, which is a very good thing since those examples are not in alphabetical order. Unlike examples in Tutorial, code formatting here is much better

(keywords are highlighted, rows formatting exists).

c. Important Questions

Important Questions are somewhat like interview questions that appeared in previous applications, but where interview questions were focused on details programmers often miss (i.e. “what is the difference between A and B”) these questions are more about definitions (i.e. “what is A”) and less about things beginner programmers often miss.

d. Quiz

Quiz is created in form of a game: answer right and get points (points multiply if there are several right answers in a row), answer wrong and lose a life. Questions are mostly about theoretical, but some are asked in a form of “what will this code write out?” To compete with other users a logging in through Google Games account is used.

Theoretical contents:

Theoretical contents of the application are presented in Tutorial section. This part contains 13 major topics that are all accessible to user (user can go from first to last topic or choose his own sequence of topics).

Some of the topics of a basic Java course are missing (Java 8 elements, specific classes like Math, etc.), there are no more advanced topics or tips or tricks that may be interesting to more advanced programmers.

The theoretical contents of the course are presented to a user in a form that make it easy-to-perceive: small blocks of text are well-written and have understandable internal logic. The main problem is the way examples are written: formatting is poor and almost non-existent, code looks like normal text which makes it hard to read. A small number of examples in Tutorial part (some topics have “templates” of code instead of normal code examples) may be compensated by examples in Programs part, but those don’t have any comments explaining how this code works.

Another problem with theoretical contents of the application was reported by the users: there are mistakes in theory. Mistakes were mostly found in Quiz part of the application, but Tutorial part has them too.

Practical exercises:

Programs part of the application contains examples of programs. User is unable to try out those examples by the means of this application, reading code is the option available to users.

Examples of code have output that shows what will be written out as a result, but sometimes it may be not enough. There is no explanation to a task itself (e.g. in “Pascal triangle” or “Floyd Triangle” user may not know what those triangles are) or to the code (why it is written this way and not the other).

Knowledge checks:

Interview questions part may be considered a form of knowledge check: user sees the

question, but not the answer (until he clicks on it), so he can check himself. As it was mentioned before, interview questions in this application are aimed to check basic knowledge, but not the things that are often missed.

Another form of knowledge check is Quiz. It is gamified, so user may learn by playing, not really noticing that he is learning. Minus of this quiz is that user cannot choose difficulty level, so beginner may learn answers that basically mean nothing to user of his level. More important problem with the Quiz is that it has mistakes in it, and that basically nullifies its' usefulness.

User experience:

The application is very minimalistic in its' design and navigation system is not perfect (it is not that hard to alphabetize program examples and link them to theoretical parts). There is some problem with connection to Google Store that makes experience of using this application not pleasant: sometimes Google connection screen or full-screen ad just randomly appears and disrupts learning process.

Motivational elements:

Only ability to compete with other players.

4.4.2 "100+ Java Programs with Output" by Code Refer/Vamsi Tallapudi

Basic characteristics:

- Current version: 1.2;
- Current rating: 4,1;
- Current number of downloads: between 100,0 and 500,0 thousand.

Description:

Vamsi Vallapudi is an author of blog with tips and tricks concerning Android development (www.coderefer.com). This application is his attempt to help beginner developers learn Java language and demonstration of his programming skills.

The application is very simple, there are no big interactive elements (like compiler that will show user real code output), exactly like its' name states it is mainly a set of programs with output. Because of that the application doesn't need connection to the Internet and can be used offline.

Mobile application has the following main parts (i.e. parts that are directly related to learning programming):

a. 100+ Java programs

Java programs is the main part of the application. Programs have output and basic commentary inside the code. Code is made easy-to-read: keywords are highlighted, formatting is helpful, but user sometimes will have to scroll horizontally through example if he has a small mobile device. Examples are written out in a random sequenced list, but standard search tool is available.

b. Java interview questions

This part of the application is presented by a list of basic interview questions that are mostly centered around definitions. This list has a rather poor formatting and no option to see answer by clicking, answers and questions are shown as one text, together.

Theoretical contents:

There is no specific part in this application that is devoted to explaining theory. Interview questions may be considered a way to refresh existing theoretical knowledge.

If we are evaluating a set of topics code examples are meant to illustrate, it should be said that those examples are not (mostly) meant to illustrate specific theoretical topics: there are examples showing how to solve typical programming tasks, examples on topics like cycles and if-expressions and a set of very specific tasks on working with Strings and Date formats. Some of topics that should be in a basic Java course are not reflected in this application (e.g. examples on features that have appeared in Java 8).

Practical exercises:

The aim of the application (as it's clear from its' name) is to offer user examples of someone else's code so he can practice reading it. No more interactive options (like compiler) are offered to a user, but it's compensated by the number of programs collected in this application.

A list of programs starts with the basic "Hello, World", goes through a list of standard examples that can be found almost in all of the applications described here (bubble sort, Fibonacci sequence, "for" loops, how to check a number if it's odd or even, how to check word if it's a palindrome, different actions with matrix, etc.)

Some examples are in the list twice (e.g. "Hello, World" is there twice), and while in some cases it would be nice to have more examples than one (e.g. Fibonacci sequence can be programmed in a number of ways), those examples have a very similar code and it looks like author just copied examples from different sources. It would be nice to see a more serious work in that direction, though: group examples with similar topics, add examples that show how a task can be solved differently (e.g. Fibonacci sequence: using recursion, cycles, etc.) and explain what positive and negative sides have each of them. This application sometimes has minimal commentary inside the code, but explanation of the task itself is missing (and a more detailed explanation of code is sometimes necessary too).

Some code examples seem a bit strange for an entirely different reason. There are several examples like "Formatting ... using SimpleDateFormat" that seemed strange to me: why write out all the variants? 2-3 examples are enough to understand the principle, all other similar examples look like an attempt to add value to the application by adding examples that are not helpful.

Knowledge checks:

Interview questions part may be considered a form of knowledge check, but in case of

that application formatting makes it uncomfortable to use: user sees question and answer at the same time, so he doesn't have time to think of an answer himself.

User experience:

The application is very simple, design is rather primitive. Navigation could be better (examples could be in alphabetic order and divided into groups), but screen with ads that sometimes appears is annoying, which was noticed by users in reviews section.

Motivational elements:

None.

4.4.3 "Learn Java" by Thiyagaraaj M

Basic characteristics:

- Current version: 6.2;
- Current rating: 4,3;
- Current number of downloads: between 100,0 and 500,0 thousands.

Description:

Thiyagaraaj M is the author of a series of successful mobile applications for learning different programming languages called "Little Drops". All applications from this developer have similar design and highly rated by users.

The application is aimed to be used mainly by beginners (the author specifically mentions that it's designed with beginners in mind in the app store description), but more advanced programmers can also find something useful here: the application contains a lot of unique texts and exercises that may be useful for refreshing user's knowledge and some material that is outside of a basic Java course (e.g. GUI). The application can be used offline, most of the content are of non-interactive nature.

Mobile application has the following main parts (i.e. parts that are directly related to learning programming):

a. Java idea bubbles

Java idea bubbles contains summarized information regarding a group of concepts (the author divided those concepts into 4 main groups: basics, conditional, oop, java collections) that are explained in a form that reminds a wikipedia page: just basic terms, minimum of explanations and code examples to illustrate them.

This part of application may be useful for those who wants to quickly refresh their knowledge of this language or get acquainted quickly with theoretical basics.

b. Java overview

Java overview is similar to a previous part (Java idea bubbles), but here theoretical information is structured a bit differently. In this part, theoretical information is presented in a form that is closer to analytical review: minimum of examples, Java basics are explained more deeply than in previous part, sometimes even with an

overview of when this particular element appeared in Java and what are positive and negative sides of using it.

This part of application will be interesting to those who want a deeper understanding of Java language and its' background.

c. Java tutorials

Java tutorials part contains theoretical material in a form that reminds a e-book. Big blocks of text that is devoted not only to Java, but also author's thoughts about things that may also be useful to beginner Java programmer (XML, GUI components). The sequence of topics sometimes is a bit unconventional and doesn't always follow the same sequence that is usually chosen by other applications, there are some unnecessary long author's thoughts on the things that are not directly connected to learning Java programming - author's personality has a great influence on the form in which material is presented. In the end of each chapter there is a small non-interactive quiz.

From a technical point of view, this e-book is a very easy to read despite how long blocks of text can be here: keywords are highlighted not only in examples, but in a text too, every chapter has a brief introduction and contents listed in the beginning, chapters are linked by references. Theoretical material in this part is also linked with practical exercises from the next part (programs and exercises).

d. Programs and exercises

This part is divided into sections: Java programs, exercises and quiz.

Java programs section is a set of code examples. These examples are not interactive, user can't change code and see how the output changes. Examples are grouped by topics; every example is linked to a specific chapter from tutorial. Examples of programs have introduction explaining the task that the program is aimed to solve and comments that clarify the code itself.

Exercises, like it is clear from the title, are the tasks that user may try to solve and a detailed explanation of how they can be solved. Exercises are grouped by topics.

Java quiz is a list of questions that user is meant to answer after reading a tutorial chapter.

e. Java reference

Java reference is a glossary or dictionary that contains definitions for Java concepts, main terms, keywords, etc.

Theoretical contents:

It is clear from the beginning, that theoretical contents of this application are very rich: information is presented in different forms (user can choose if he wants a brief overview or a long course filled with details), it is clearly not simply taken from another similar application, but carefully written and adapted for different types of users.

This variety of forms is both a good and a bad thing: while it is always nice to have a choice, it is not clear to the beginner what part of the application is exactly for him (her)

and what is the difference between various formats of presenting information.

The sequence of topics differs from part to part, depending on the scope covered in each of them. The author may sometimes miss some non-essential topic from a basic Java course, but in some things he goes over the boundaries set for a “standard beginners course”.

User can access any topic he or she wants, but since the main theoretical part - the tutorial - is written in a form of a book, it's more interesting to follow it in an order offered by the book. Theory presented in tutorial is closely linked with practical exercises and examples.

Practical exercises:

Practical exercises in this application appear in two forms: Java programs (i.e. a set of static code examples) and exercises (tasks and challenges for users to figure out).

Examples of programs that allow user to practise reading code appear in every mobile application for learning Java language, but here they are well-made: user gets an explanation of how the code works and he always knows what part of theory he may refer to in case something isn't clear.

Exercises in a form of a puzzle or a challenge for a user is a wonderful idea. Even if user won't try coding himself (too bad that this application doesn't have a compiler to try out different ideas), he will begin thinking in Java language. It is more than a simple code reading, it's how programming starts. In case user gets stuck, there is also a solution and a very detailed explanation of how this task could be solved (and sometimes it is solved in different ways).

Knowledge checks:

Knowledge may be checked in 2 forms: quiz and exercises.

Quiz questions are linked to chapters of tutorial and may be a bit too specific for users that just wanted to check a general level of Java language knowledge, but a good way to test one's understanding of tutorial material. User can see two variants of the same list of questions: with or without answers.

Exercises are, probably, the most accurate way to check programming knowledge, if not the quickest. Unfortunately, since exercises are closely connected to tutorial topics and chapters, exercises may be too specific or contain requirements to what should be used for solving the task.

User experience:

Although the developer made almost everything possible to make text and code more readable, there are some problems with navigation. It is not always clear why the contents of the application are structured this way and there are too many sub-levels for an easy orientation in this application. Links that send user from one part to another are meant to make navigation easier, but in fact they sometimes create an opposite

effect.

Nonetheless, all elements are always grouped by topics for easier search, user can add bookmarks.

Motivational elements:

None.

4.5 Mobile applications on iOS platform

Two applications that fit initial criteria in 4.1 were left out later: “Learn Java” by Le Chi Hong became unavailable by the time I began reviewing applications in detail, “Mimo: learn to code on the go” (also known as “Javvy”) becomes a paid application after first 4 introduction topics (that wasn’t mentioned in description) and the contents of the application are aimed to interest a much younger users than university students.

4.5.1 “Code Recipes” by Fedor Obratsov

Basic characteristics:

- Current version: 1.3;
- Current rating: 4+.

Description:

“Code Recipes” is a mobile application containing examples of code for 5 programming languages at once. It exists in 2 variants: for iPad and iPhone. Java language examples are unlocked in a free version of the application (together with Swift language).

The application is very simple, the main part of the application is a list of topics containing code examples. Examples are not interactive: user can’t change the code and see how output will be changed. Nonetheless, everything is done in order to make reading code easier.

Theoretical contents:

While there are no explanations of theory that is behind each code example, the examples are grouped by topics that are close to the contents of a basic Java course. There are several topics with examples devoted to new features that appeared in Java 8.

Practical exercises:

Examples of programming code are grouped by topic and each topic has a set of examples to demonstrate the topic from different angles, which is a very nice thing: some things can be better seen in comparison.

The somewhat negative moment is that there is a lack of commentary explaining why the code is written the way it is, but it is compensated by the fact that every example has a link to a forum on the author’s website (www.code-samples.ru/java) where users may discuss problems and questions they have with it.

Knowledge checks:

None.

User experience:

The application is made in a simple, but very comfortable style. Users in their reviews usually positively mark how easy navigation is and a big number of examples.

Motivational elements:

None.

4.6 Quantitative evaluation

To summarize the information given in previous part, all quantitative evaluations and marks were united together in the following table.

Table 7. Quantitative evaluation of reviewed mobile applications for learning Java language programming (Source: author)

Application	Platform	Description	Theoretical contents	Practical exercises	Knowledge checks	User experience	Motivational elements	Result
"Learn Java" / "Learn Java Pro" from SoloLearn Inc.	iOS, Android	3	2	1	1	3	1	11
"Programming Hub" from Nexino Labs	iOS, Android	2	1	3	1	1	0	8
"Enki" from Enki Labs	iOS, Android	3	2	3	3	2	1	14
"Java Programming" from Akshay Bhange	Android	2	1	2	1	1	0	7
"100+ Java Programs with Output" by Code Refer/Vamsi Tallapudi	Android	2	1	1	1	1	0	6
"Learn Java" by Thiagaraja M	Android	3	3	3	2	2	0	13
"Code Recipes" by Fedor Obratsov	iOS	2	1	2	1	2	0	8

4.7 Common traits and interesting trends

There are certain common traits and trends that become clear after going through several mobile applications that are supposed to help beginner programmers learn Java language.

First of all, all applications may be divided into two groups: made by standalone programmers and made by companies.

Applications made by companies are usually better made: they have nice designs with a lot of interactive and complex elements, unusual parts like games and compilers. These applications require active interaction from users: the principle of “bite-sized lessons” is often implemented, motivational elements like reminders are often used, users are pushed to become a part of the community.

Applications made by standalone users are often much simpler: minimalistic design with mostly static elements, similar parts (like interview questions) with content they sometimes borrow material from each other. These applications are not always bad, some of them are made with big dedication and interest, but it is hard to compete with a team of programmers.

What is interesting, the fact that an application was made by a company doesn't automatically mean that it is more effective: creators of those apps are sometimes too interested in attracting and keeping their users and not enough interested in how progress in their application is connected to real evolvement of programming skills.

Second result is an expected one: there is a bigger number of applications for learning Java for Android platform than for iOS, which is easily explained by the fact that Android developers often use this language. What is more unexpected, most of those applications are quite recent, appearing in the last year or two. One of possibilities is that there have appeared a great number of courses that are focused not just on Java, but on development of mobile applications for Android platform.

5 Conclusion

In this work I compared and analysed a set of mobile applications for learning programming in Java language that currently exist on the market in order to see if any of them are fit to be used as a viable supporting tool for studying Java programming.

To achieve this goal, a questioning of students (main target group of this work) was carried out, its' results were used to create a set of criteria for evaluation of each mobile applications from different viewpoints. Evaluation that took place was based on materials available on app store pages and my own experience with those mobile applications.

While none of the applications got the maximum, there are certain leaders: "Enki" from Enki Labs and "Learn Java" by Thiyagaraaj M that may possibly be used, but mostly as supporting tools. Full results of this analysis will be posted on website java.vse.cz, so that students of University of Economics (Prague) and other interested parties could access and use them.

List of terms

Programming language	A simple language system designed to facilitate the writing of computer programs (Collins English Dictionary, 2017)
IT (Information Technology)	The technology involving the development, maintenance, and use of computer systems, software, and networks for the processing and distribution of data (Merriam-Webster Dictionary, 2017)

Bibliography

BEBELL D. and O'DWYER L. M. Educational outcomes and research from 1:1 computing settings. Journal of Technology, Learning, and Assessment, Volume 9, Number 1, January 2010. [online] [cit. 2017-03-28] Available: <<http://ejournals.bc.edu/ojs/index.php/jtla/article/view/1606/1463>>

Business Dictionary: application software [online] [cit. 2017-03-27] Available: <<http://www.businessdictionary.com/definition/application-software.html>>

Cambridge Dictionary: mobile application [online] [cit. 2017-03-26] Available: <<http://dictionary.cambridge.org/dictionary/english/mobile-application>>

CASUTO S., 13.04.2016. Five Trends Driving The Growth Of Mobile Learning In Corporate Education. Forbes. [online] [cit. 2017-03-27] Available: <<https://www.forbes.com/sites/theyec/2016/04/13/five-trends-pointing-to-mobile-learning-as-the-future-of-corporate-education/#78285803141c>>

Collins English Dictionary: programming language [online] [cit. 2017-05-01] Available: <<https://www.collinsdictionary.com/dictionary/english/programming-language>>

Definitions of Information Technology Terms: Application Software [online] [cit. 2017-03-27] Available: <<https://www.defit.org/application-software/>>

DIAKOPOULOS N. and CASS S., 26.07.2016. Interactive: The Top Programming Languages 2016. IEEE Spectrum. [online] [cit. 2017-03-27] Available: <<http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2016>>

DOGTIEV A., 29.02.2016. App Store Statistics Roundup. BusinessOfApps. [online] [cit. 2017-03-27] Available: <<http://www.businessofapps.com/app-store-statistics-roundup/>>

FLEISCHER, H. What is our current understanding of one-to-one computer projects: a systematic narrative research review. Educational Research Review, Volume 7, Issue 2, June 2012. [online] [cit. 2017-03-28] Available: <<http://www.sciencedirect.com/science/article/pii/S1747938X11000509>>

KPCB: Internet Trends 2015 – code conference [online] [cit. 2017-03-27] Available: <<http://www.kpcb.com/blog/2015-internet-trends>>

MATVIICHUK, O. Doplnění vývojového prostředí BlueJ o funkce využitelné v úvodních kurzech programování. Prague, 2014. Thesis supervisor Rudolf Pecinovský.

McMillan Dictionary: m-learning or mobile learning [online] [cit. 2017-03-28] Available: <<http://www.macmillandictionary.com/buzzword/entries/m-learning.html>>

Merriam-Webster Dictionary: information technology [online] [cit. 2017-05-01] Available: <<http://www.merriam-webster.com/dictionary/information%20technology>>

O'GRADY, 17.03.2017. The RedMonk Programming Language Rankings: January 2017. RedMonk [online] [cit. 2017-03-27] Available: <<http://redmonk.com/sograde/2017/03/17/language-rankings-1-17/>>

OLIVER A. C., 25.07.2013. Love and hate for Java 8. Javaworld. [online] [cit. 2017-03-27] Available: <<http://www.javaworld.com/article/2078836/java-se/love-and-hate-for-java->

8.html>

Oracle. Java SE Documentation [online] [cit. 2017-03-27] Available:
<<http://docs.oracle.com/javase/7/docs/technotes/guides/language/>>

Oracle: What's New in JDK 8 [online] [cit. 2017-03-27] Available:
<<http://www.oracle.com/technetwork/java/javase/8-whats-new-2157071.html>>

PÍCHOVÁ, M. Application for Teaching of Object-oriented Design Principles. Prague, 2010. Thesis supervisor Pavel Ježek.

PYPL PopularitY of Programming Language [online] [cit. 2017-03-27] Available:
<<http://pypl.github.io/PYPL.html>>

REJMAN, M. Vizualizace objektů a datových struktur v Javě. Prague, 2010. Thesis supervisor Jan Stoklasa.

SEDLÁČEK, D. Navrhněte počítačovou hru pro vstupní kurzy programování. Prague, 2015. Thesis supervisor Rudolf Pecinovský.

Statista: Annual number of global mobile app downloads 2016-2021 [online] [cit. 2017-03-27] Available: <<https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/>>

Statista: Most popular Apple App Store categories in December 2016, by share of available apps [online] [cit. 2017-03-27] Available: <<https://www.statista.com/statistics/270291/popular-categories-in-the-app-store/>>

Statista: Number of apps available in leading app stores as of June 2016 [online] [cit. 2017-03-27] Available: <<https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>>

Statista: Number of available applications in the Google Play Store from December 2009 to March 2017 [online] [cit. 2017-03-27] Available:
<<https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>>

Statista: Number of available apps in the Apple App Store from July 2008 to January 2017 [online] [cit. 2017-03-27] Available: <<https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/>>

SUNG, Y.T., CHANG K.E., LIU T.C. The effects of integrating mobile devices with teaching and learning on students' learning performance: A meta-analysis and research synthesis. Computers & Education, Volume 94, March 2016. [online] [cit. 2017-04-04] Available:
<<http://www.sciencedirect.com/science/article/pii/S0360131515300804>>

Technopedia: Mobile Application (Mobile App) [online] [cit. 2017-04-04] Available:<<https://www.techopedia.com/definition/2953/mobile-application-mobile-app>>

TIOBE Index for March 2017 [online] [cit. 2017-03-27] Available:
<<https://www.tiobe.com/tiobe-index/>>

TrendySkills: Trends for languages from 1.1.2017 [online] [cit. 2017-03-27] Available:
<<http://trendyskills.com/#trends>>

UNESCO: ICT in education: Mobile learning [online] [cit. 2017-03-28] Available:
<<http://www.unesco.org/new/en/unesco/themes/icts/m4ed/>>

WhatIs.com: app store [online] [cit. 2017-03-27] Available:
<<http://searchmobilecomputing.techtarget.com/definition/app-store-application-store>>

ZÁVĚRKA, J. Alternativní výukové materiály pro vstupní kurzy programování. Prague, 2010.
Thesis supervisor Rudolf Pecinovský.

ZYDNEY J. M. and WARNER Z. Mobile apps for science learning: Review of research. Computers & Education, Volume 94, March 2016. [online] [cit. 2017-04-04] Available:
<<http://www.sciencedirect.com/science/article/pii/S0360131515300737>>

List of graphs

Graph 1. Number of mobile app downloads worldwide in 2016, 2017 to 2021, in billions (Source: Statista, 2017).....	8
Graph 2. Number of apps available in leading app stores by June 2016 (Source: Statista, 2016).9	
Graph 3. The TIOBE Programming Community index from 2002 to March 2017 (Source: TIOBE, 2017).....	10
Graph 4. The PYPL Popularity of Programming Language index from 2005 to March 2017 (Source: PYPL, 2017)	11
Graph 5. RedMonk rating, results for 2017 (Source: RedMonk, 2017).....	12
Graph 6. Respondents by gender (Source: author)	15
Graph 7. Respondents by age (Source: author)	15
Graph 8. Respondents by previous (before university) experience with programming (Source: author)	16
Graph 9. Respondents planning to use knowledge Java language programming in their future (Source: author)	17
Graph 10. Materials used for learning Java language programming (Source: author).....	18
Graph 11. Previous experience with mobile applications for learning programming in Java language (1 - negative experience, 5 - positive experience) (Source: author).....	19
Graph 12. Usage of mobile applications for learning Java language programming - regularity (questioning of respondents who had actual experience with this type of mobile applications) (Source: author)	20
Graph 13. Usage of mobile applications for learning Java language programming - searched content (questioning of respondents who had actual experience with this type of mobile applications) (Source: author)	20
Graph 14. Respondents by opinion about effectiveness of mobile applications for learning a programming language (Source: author)	21
Graph 15. Average marks given by respondents to different criteria for choosing mobile application for learning Java language programming (Source: author).....	24

List of tables

Table 1. Connection between experience with programming gained before university and plans to use knowledge of Java language programming in future, number of respondents (Source: author)	17
Table 2. Number of sources used for learning Java language selected by respondents (Source: author)	18
Table 3. Respondents by opinion about effectiveness of mobile applications for learning a programming language.....	22
Table 4. Respondents by opinion about current state and potential of mobile applications for learning a programming language, number of respondents (Source: author)	22
Table 5. Ranged list of criteria used for choosing mobile application for learning Java language (Source: author)	24
Table 7. Quantitative evaluation of reviewed mobile applications for learning Java language programming (Source: author)	44

Appendix 1. Questionnaire of students (English and Czech language versions)

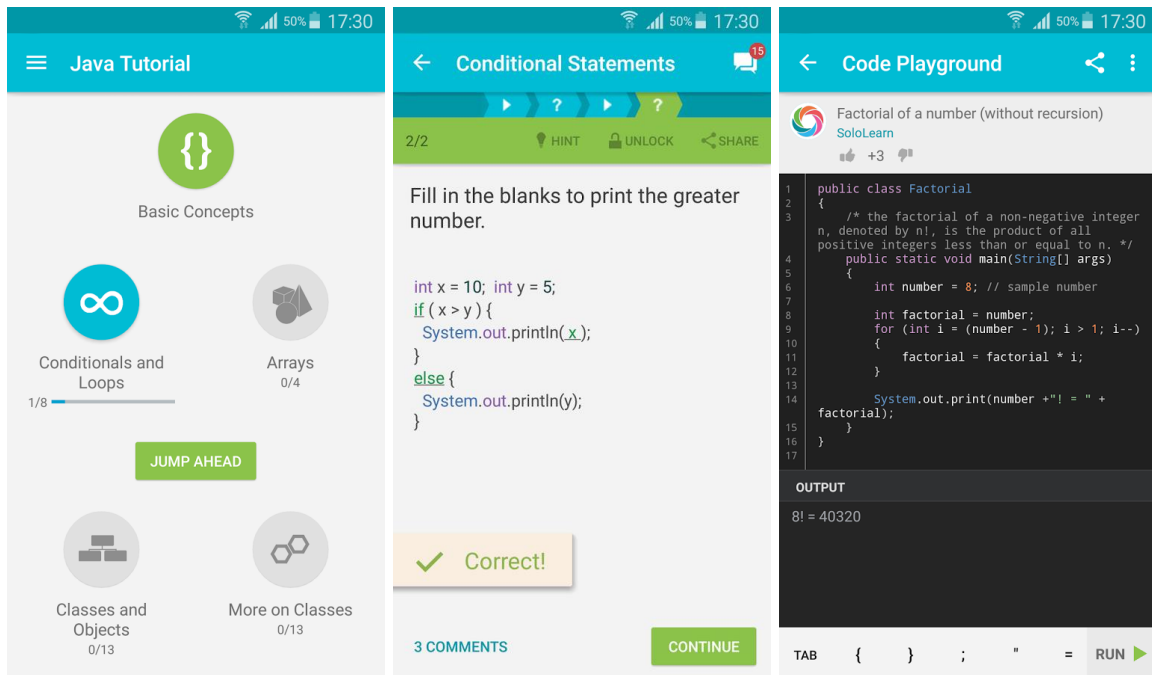
1	Please, choose your gender: <Male, Female>	Prosím Vás, vyberte, jaké máte pohlaví <Muž, Žena>
2	Please, choose your age: <16 or less, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41 or more>	Prosím Vás, vyberte, kolik je vám let <16 a méně, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41 a více>
3	Did you have experience with programming before university? <Yes, No>	Měl(a) jste zkušenosti s programováním před univerzitou? <Ano, Ne>
4	Please, rate your experience with Java language programming (choose the variant that describes your situation the best): < -I'm in the process of studying Java at university, and i have no previous experience with it -I'm in the process of studying Java at university, and i have studied Java before -I've finished only Java <u>courses</u> at university and have no further experience with it -I've finished Java <u>courses</u> at university and gained some practical experience with Java language afterwards >	Prosím vás, vyberte tvrzení, které nejlepším způsobem popisuje vaše zkušenosti s programováním v jazyce Java: < -Učím se Javu teď na univerzitě, předtím jsem tento jazyk nestudoval(a) -Učím se Javu teď na univerzitě, ale seznámil(a) jsem se s tímto jazykem již dříve -Mám za sebou vysokoškolský kurz programování v Javě, žádné jiné zkušenosti s tímto jazykem nemám -Mám za sebou vysokoškolský kurz programování v Javě, kromě toho mám i praktické zkušenosti s tímto jazykem >
5	Are you planning to use (or are you already using) the knowledge of Java language for work? <Yes, No>	Plánujete využít své znalosti programování v Javě v praxi? <Ano, Ne>
6	What materials and tools did (do) you use studying Java (please, choose one or several answers): < -Materials and tools offered by university -Materials and tools created by other students (borec, vstudent, etc.) -Official materials from Oracle -Materials and tools offered by websites that are not directly connected with university or Oracle -Materials and tools offered in form of mobile applications >	Jaké materiály jste používal(a) ke studiu Javy (vyberte i několik odpovědí): < - Materiály a instrumenty nabízené vysokou školou - Materiály a instrumenty vytvořené jinými studenty (borec, vstudent, apod.) - Oficiální materiály od firmy Oracle - Materiály a instrumenty nabízené na webových stránkách, které nejsou spojené s vysokou školou nebo firmou Oracle - Materiály a instrumenty nabízené mobilními aplikacemi >

		>
7	Did you use mobile applications for studying Java language? <Yes, No>	Používal(a) jste mobilní aplikace pro podporu učení se programování v Javě? <Ano, Ne>
8	Please, rate your experience with mobile applications you have used for studying Java: <Two 1-5 ratings for "user experience" a "knowledge gained">	Prosím Vás, ohodnoťte vaše zkušenosti s mobilními aplikacemi podporujícími výuku programovacího jazyku Java: <Dva 1-5 hodnocení pro "uživatelské pohodlí" a "nabytí znalosti a zkušenosti">
9	How often did you use mobile applications for studying Java language? < - Didn't use it - Used it a couple of times - Used it regularly >	Jak často jste používal(a) mobilní aplikace podporující výuku Javy < - Nepoužíval(a) jsem mobilní aplikace - Použil(a) jsem mobilní aplikace jen pár krat - Používal(a) jsem mobilní aplikace pravidelně >
10	Please, choose the statement that describes your situation best: < - Didn't use it - I used mobile applications for studying Java language for practical experience - I used mobile applications for studying Java language for theoretical knowledge - I used mobile applications for studying Java language both for practical experience and theoretical knowledge >	Prosím vás, vyberte tvrzení, které nejlepším způsobem popisuje vaši situaci: < - Nepoužíval(a) jsem mobilní aplikace - Používal(a) jsem mobilní aplikace pro výuku Javy především za účelem nabytí praktických zkušeností - Používal(a) jsem mobilní aplikace pro výuku Javy především za účelem nabytí teoretických znalostí - Používal(a) jsem mobilní aplikace pro výuku Javy za účelem nabytí praktických zkušeností a teoretických znalostí najednou >
11	Do you think that mobile applications can be useful for studying programming language? <Yes, No>	Myslíte si, že mobilní aplikace mohou být užitečné při studiu programovacího jazyka? <Ano, Ne>
12	What characteristics do you find important for mobile application for studying Java programming: < Rate criteria from 1-5: - App doesn't need to be constantly connected to Internet - App is suitable not just for beginners, but also for more advanced users - Mobile application offers a predefined sequence of topics - List of topics offered in an app is similar to those of university course >	Prosím,vás, ohodnoťte kritéria výběru mobilních aplikací podporujících výuku Javy podle jejich důležitosti: < Hodnocení 1-5 pro kritéria: - Aplikace nepotřebuje připojení k Internetu - Aplikace je vhodná nejen pro začátečníky, ale i pro pokročilé - Aplikace má v sobě nabízené pořadí témat - Seznam témat je schodný s vysokoškolským kurzem - Výklad teorie je velmi podrobný - Uživatel může zkontrolovat své znalosti >

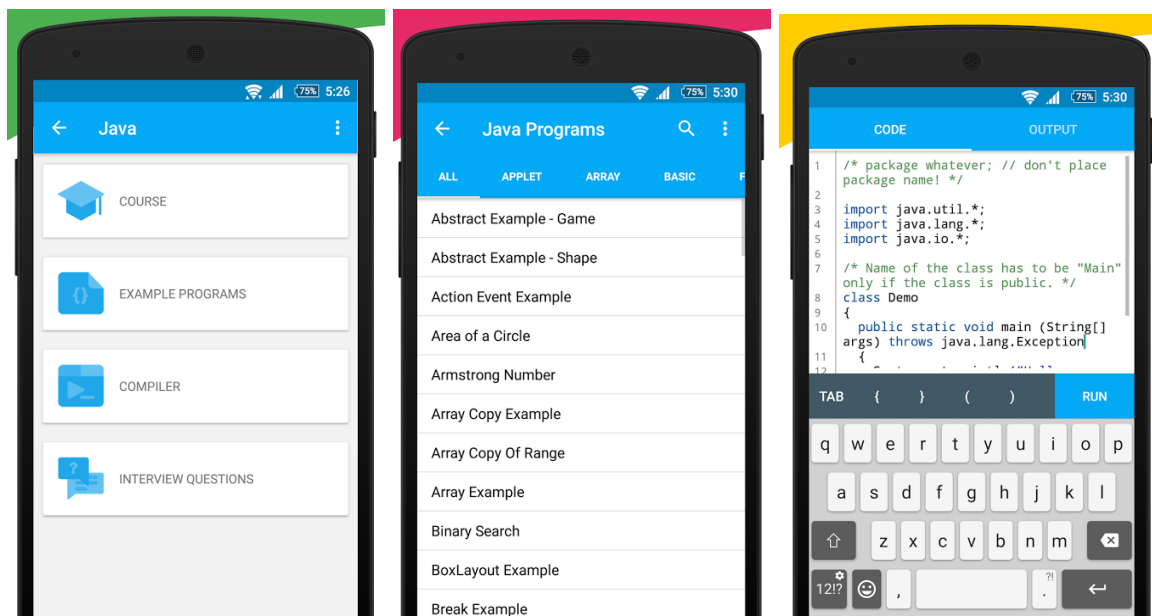
	<ul style="list-style-type: none"> - Theory presentation is detailed - User can test himself/herself and check his/her knowledge level - User can gain practical experience with programming - App is comfortable to use <p>></p>	<ul style="list-style-type: none"> - Uživatel může získat praktické zkušenosti s programováním - Aplikace je uživatelsky příjemná <p>></p>
13	<p>Please, choose the statement that you are most certain about:</p> <p><</p> <ul style="list-style-type: none"> - Mobile applications are well-suited for studying Java programming and can be used as a main (standalone) learning tool - Mobile applications are well-suited for studying Java programming, but as a supporting tool - Current mobile applications are not well-suited for studying Java, but that can possibly change in foreseeable future - Current mobile applications are not well-suited for studying Java, and I don't think this will change in the foreseeable future <p>></p>	<p>Prosím vás, vyberte tvrzení, které nejlepším způsobem popisuje váš názor na mobilní aplikace pro výuku programování:</p> <p><</p> <ul style="list-style-type: none"> - Mobilní aplikace jsou dobré na výuku programování a mohou být použité samostatně, jako hlavní instrument výuky - Mobilní aplikace jsou dobré na výuku programování, ale spíš jako podporující instrument - Aktuální mobilní aplikace nejsou dobré na výuku programování, ale to se může změnit v budoucnosti - Aktuální mobilní aplikace nejsou dobré na výuku programování, a není velká pravděpodobnost, že to se může změnit v budoucnosti <p>></p>

Appendix 2. Evaluated mobile applications – design

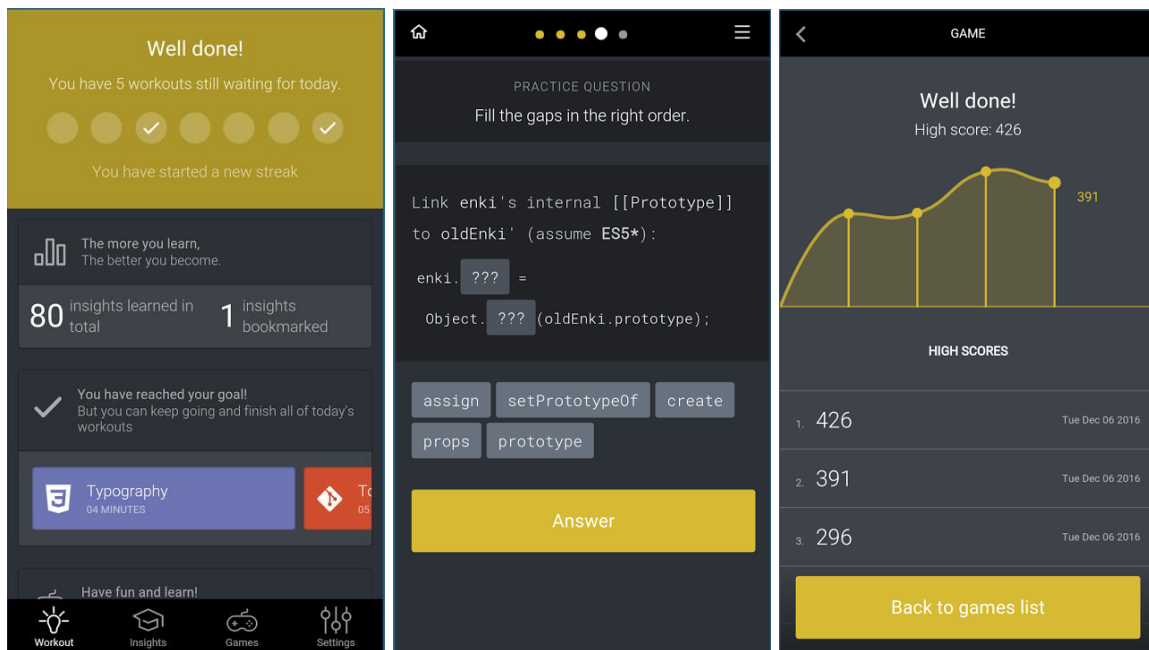
“Learn Java”/ “Learn Java Pro” from SoloLearn Inc.



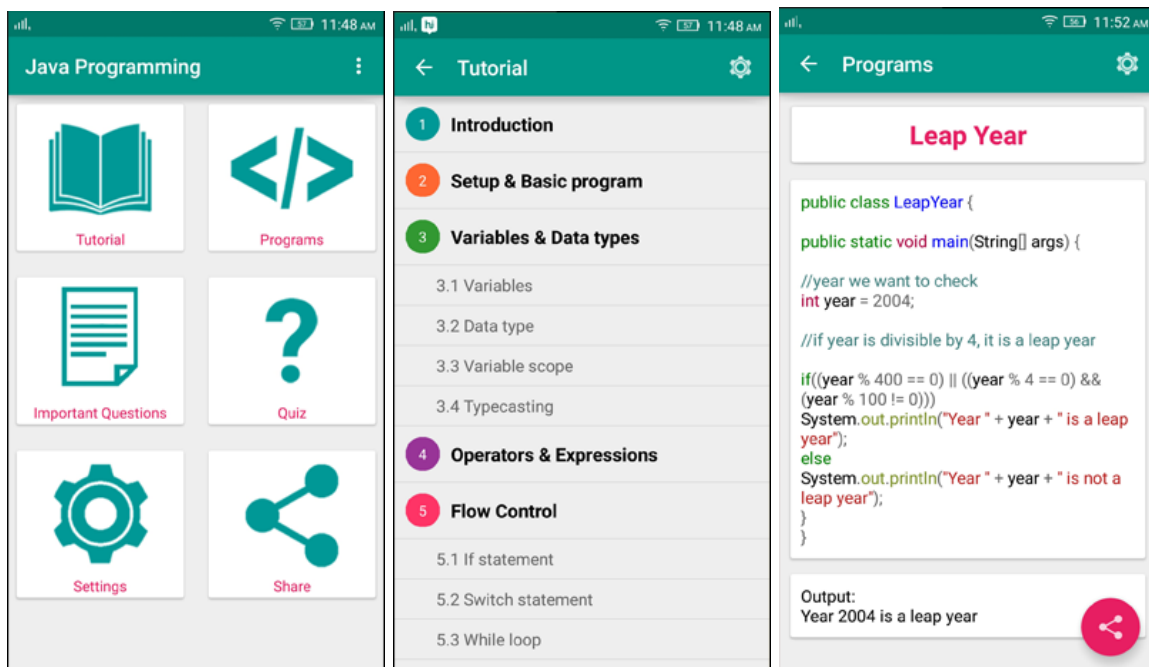
“Programming Hub” from Nexino Labs



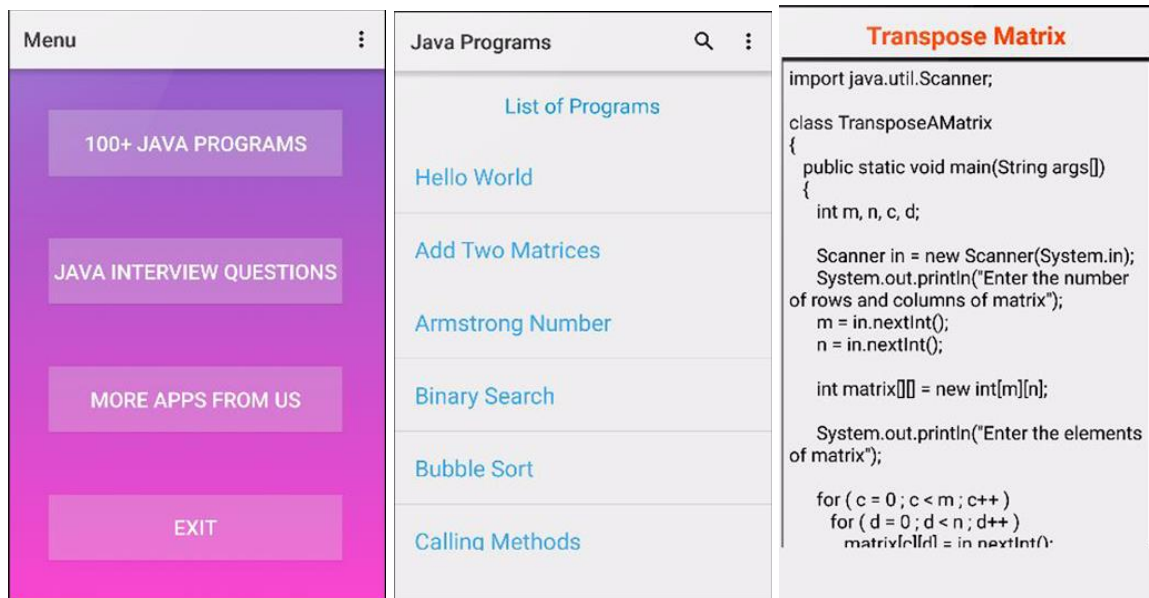
“Enki” from Enki Labs



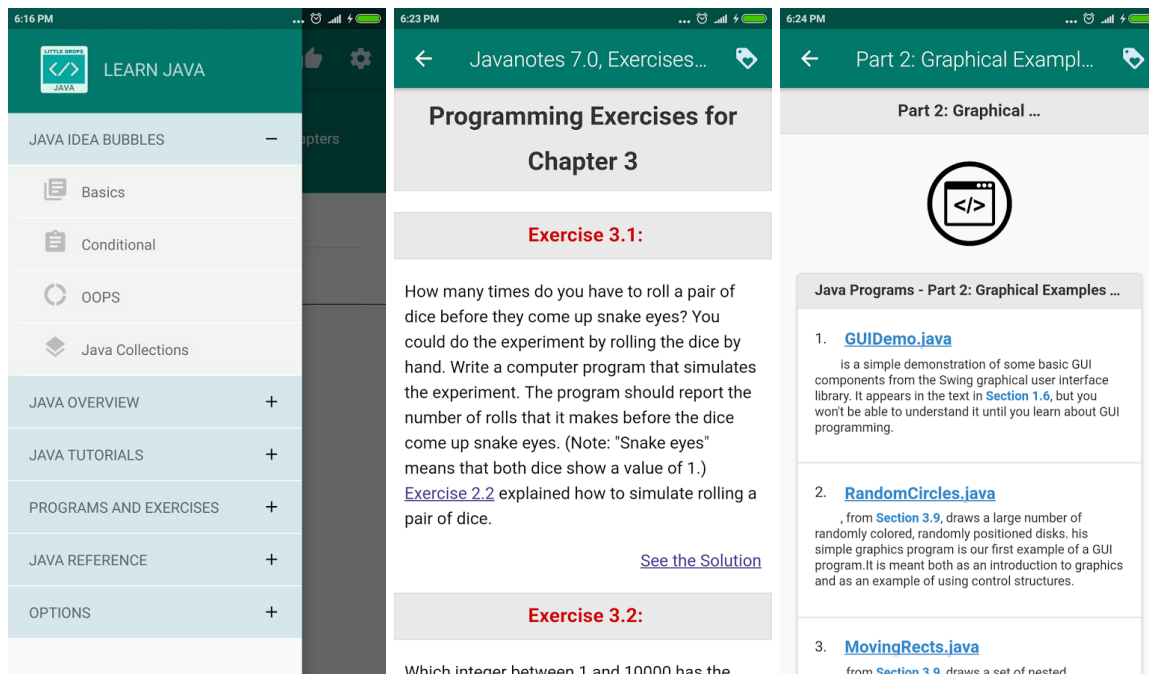
“Java Programming” from Akshay Bhange



“100+ Java Programs with Output” by Code Refer/Vamsi Tallapudi



“Learn Java” by Thiyagaraaj M



“Code Recipes” by Fedor Obraztsov

