

# Методика формирования задач для автоматической проверки в рамках МООС “Мобильная разработка для Android на Kotlin”

## Аннотация

В данной работе описывается модель формирования набора задач, позволяющая сократить временные затраты на проверку задач в рамках онлайн-курсов, в том числе, при расширении возможностей обучения: увеличении количества обучающихся, реализации интерактивных и многошаговых задач и т.д. на примере курса “Мобильная разработка для Android на Kotlin”. Процедура проверки включает в себя сборку, компиляцию, запуск и интерпретацию результатов работы решения обучающегося.

## Введение

## Обзор предметной области

### Онлайн-школа Android-разработчиков

<https://learndroid.e-legion.ru>

Author: e-legion + Google

Description:

14 бесплатных лекций от экспертов из e-Legion, OK.RU, Tinkoff, JetBrains, Kaspersky Lab, Parallels. Лекции подойдут как новичкам с базовыми знаниями Java, так и уже опытным разработчикам. Сертификат об окончании школы от Google. Курс завершился.

### Become an Android Developer from Scratch

<https://www.udemy.com/become-an-android-developer-from-scratch>

Description:

“Become An Android Developer From Scratch” is designed to be the best first step to launching your career as an Android Developer. Whether you’ve never programmed before or are coming to mobile development for the first time, if you’re truly inspired to learn Android Development, this is where you need to start. Лекции: 121 Видео: 12,5 ч. 237618 студентов. Языки: Английский + Субтитры. Сертификат об окончании.

### Android Basics Nanodegree by Google

<https://www.udacity.com/course/android-basics-nanodegree-by-google-nd803>

Description:

No programming experience? No problem! Learn how to develop Android apps in this efficient, guided program, where you'll have access to your own personal mentor, a network of project reviewers, and a rich community of learners to support you at every step. You'll also learn the basics of the Java programming language, which you'll use to build your own portfolio of Android apps.

Включает в себя 5 подкурсов в рамках одного курса на одной платформе Udacity:

- <https://www.udacity.com/course/android-basics-user-interface-ud834>
- <https://www.udacity.com/course/android-basics-user-input-ud836>
- <https://www.udacity.com/course/android-basics-multiscreen-apps-ud839>
- <https://www.udacity.com/course/android-basics-networking-ud843>
- <https://www.udacity.com/course/android-basics-data-storage-ud845>

### **Специализация Разработка приложений под Android**

<https://www.coursera.org/specializations/android-app-development>

Author: Vanderbilt University

Description:

This Specialization enables learners to successfully apply core Java programming languages features & software patterns needed to develop maintainable mobile apps comprised of core Android components, as well as fundamental Java I/O & persistence mechanisms. Learners who successfully complete this Specialization will be well-prepared to master the more advanced material in the subsequent "Mobile Cloud Computing with Android" Specialization.

Включает в себя 5 подкурсов в рамках одного курса на одной платформе Coursera. Целиком курс платный, можно бесплатно получить начальные фрагменты.

### **Programming Mobile Applications for Android Handheld Systems: Part 1**

<https://www.coursera.org/learn/android-programming>

Author: University of Maryland

Description:

This course introduces you to the design and implementation of Android applications for mobile devices. You will develop an app from scratch, assuming a basic knowledge of Java, and learn how to set up Android Studio, work with various Activities and create simple user interfaces to make your apps run smoothly.

### **Build a Simple Android App with Java**

<https://teamtreehouse.com/library/build-a-simple-android-app-with-java>

Author: TreeHouse

## Description:

This course covers the very basics of Android development. We will build a simple app that will serve up some fun facts when you tap on a button. We introduce you to programming in Android, a tool for Android development called Android Studio, and some very basic concepts of the Android Software Development Kit, or SDK. By the end you will have a good idea of how a basic app works, and you will be armed with the knowledge to start building more.

Курс платный.

## Критерии сравнения аналогов

### Оценка обучающимися (отзывы)

Оценки курса обучающимися, а также их отзывы, являются общедоступной информацией, исходя из которой можно сделать предварительный вывод о качестве онлайн-курса, реже - получить аргументированный анализ аспектов онлайн-курса.

### Аккредитация курса

Онлайн курс (как правило xMOOC) может иметь специальное лицензирование от ВУЗа, иного учебного заведения или специальной экспертной группы. Данный критерий не является обязательным, однако может свидетельствовать о высоком уровне подготовки организаторов курса.

### Форма подачи информации

Наличие разных форм подачи информации (текст/слайды, видеозапись, звуковой подкаст) позволяет обеспечить лучшее восприятие материалов курса обучающимися.

### Актуальность информации

Вследствие постоянного развития инструментов и языков программирования для мобильной разработки, обучение устаревшим принципам и инструментам может дать неверное представление о данной области.

### Проверочные задания

- Объем проверочных заданий;
- Содержательность проверочных заданий;
- Способы проверки (автоматизированные, ручные, etc);
- Интерфейсы сдачи заданий;

### Таблица сравнения по критериям

Применительно к рассматриваемому списку курсов, наибольшее соответствие критериям демонстрируют следующие онлайн-курсы:

- <https://teamtreehouse.com/library/build-a-simple-android-app-with-java>
- <https://www.udacity.com/course/android-basics-nanodegree-by-google-nd803>

## Выбор метода решения

В результате изучения и анализа существующих онлайн-курсов, а также сравнения способов и систем для проверки задач, можно сделать вывод, что для автоматизации проверки необходима разработка набора задач и реализация программного модуля для существующего программного продукта, позволяющего проводить автоматическую проверку решений в рамках онлайн-курсов. Кроме того, необходимо сформулировать упрощенный подход к проверке задач, поскольку он позволяет значительно сократить статью временных затрат на обслуживание онлайн-курса. Из рассмотренных существующих подходов к формулировке задания ни один не позволяет добиться полной автоматизации проверки задач. Таким образом, решение должно представлять собой набор задач, сформированных по определенному шаблону, позволяющему проводить проверку задач без участия человека, и обладать следующими свойствами:

- Возможность решения пользователем вне зависимости от его локальных условий (ОС, вычислительной мощности компьютера, скорости интернет-соединения и т.д.);
- Низкая вычислительная сложность для возможности реализации на удаленном сервере;
- Стойкостью к попыткам вмешательства в работу системы или получения решения некорректным образом (взлому);
- Открытая формулировка (задания не содержат готового ответа, каждому пользователю необходимо записывать ответ самостоятельно);

## Описание метода решения

В рамках данной работы рассматривается исключительно алгоритм формирования задач для автоматической проверки. Применительно к рассматриваемой предметной области, можно выделить три основных формата задач:

- Задачи на знание теоретических сведений: перечислить свойства и/или методы объектов, описать взаимное влияние работы компонентов, проверка корректности высказывания и т.д.
- Задачи на формирование частного ответа: есть входные данные А и Б, необходимо сформировать запрос, который решает поставленную задачу;
- Задачи на реализацию программного модуля, который решает поставленную задачу и отвечает указанным требованиям: имеет указанные компоненты, элементы управления с определенными идентификаторами и т.д.

Наиболее значимыми для формирования устойчивых навыков являются задачи на реализацию программных модулей. Существует несколько методов проверки корректности таких задач, в частности, открытая проверка, то есть сравнение ответа, данного обучающимся, с заранее установленным ответом.

Существует ряд основных стратегий обучающихся для решения задач на реализацию программного модуля:

- Строитель: обучающийся постепенно разрабатывает программу, постепенно расширяя ее функциональность, добавляя новые концепты или улучшая существующие корректные фрагменты.
- Манипулятор: обучающийся мало изменяет заданный шаблон кода, не добавляет и не удаляет существующие концепты, надеясь, что программа заработает.
- Сокращающий: обучающийся удаляет существующие программные компоненты, сокращая или сохраняя уровень корректности программы.
- Борец: обучающийся пытается произвести все возможные изменения кода, затрачивая значительное время на решение.

Дополнительные затруднения вызывает тот факт, что многошаговые задачи иногда содержат произвольный элемент творчества. Например, для задачи создания функциональной модели для одной предметной области разными обучающимися могут быть сформированы разные модели, что может быть обосновано тем, что они по-разному определяют, называют и декомпозируют функции. В случае обнаружения совпадений моделей у разных обучающихся можно говорить о плагиате.

По способу проверки корректности решения многошаговые задачи делятся на 2 основные методики:

- Поиск по шаблону: осуществляется проверка полного совпадения промежуточного решения с шаблоном. В рамках данной методики правильное, но не совпадающее с шаблоном, решение не будет засчитано.
- поиск по схожести: осуществляется поиск ответов, схожих по функциональности, в частности, по результату выполнения.

Методика состоит в построении модели активно-пассивного слежения. В рамках такой модели преподавателем задается ход решения задачи, а также автоматически проверяется и оценивается каждый шаг обучающегося, отображая последовательность применения источников информации в процессе решения задачи (лекций, текстовых материалов, подкастов и т.д.). Строится алгоритм решения, который позволяет определить, как обучающиеся пришли к решению, ошибочному или верному. Все действия обучающегося при решении могут быть отображены в виде карты траектории поиска. Обучающиеся получают подсказки и помощь только по запросу в случае необходимости. Идея состоит в том, что формулировка задачи и ответ, а также результаты и состояния всех промежуточных этапов, записываются с целью выявления этапа, на котором обучающимся допущена ошибка.

В рамках проверки задач для онлайн-курса “Мобильная разработка для Android на Kotlin” необходимо выделить следующие этапы:

- Проверка корректности компиляции и сборки программного продукта;

- Проверка соответствия языка исходного кода программы языку, установленному в рамках курса;
- Проверка наличия необходимых компонентов и элементов управления с указанными идентификаторами;
- Проверка корректности выходных данных;

## Выводы

В данной статье изучена проблема проверки задач онлайн-курсов, а также предложена модель формирования задач, основанная на методике активно-пассивного слежения. Выделены этапы проверки корректности задач в рамках онлайн-курса “Мобильная разработка для Android на Kotlin”, проведен анализ ряда аналогов, а также описаны основные стратегии обучающихся для решения задач. На основе критериев сделаны выводы о преимуществах предложенного способа по сравнению с существующими аналогами, в частности, ожидается сокращение временных затрат на проверку задач. Таким образом, разработка набора задач в соответствии со сформированной в рамках статьи методикой позволит достигнуть поставленной цели – сократить временные затраты на проверку задач в рамках онлайн-курсов.

## Список литературы

1. <https://www.science-education.ru/ru/article/view?id=11491>
2. <http://www.it-education.ru/2016/section/234/96319/index.html>
3. <https://cyberleninka.ru/article/n/otsenka-kachestva-onlayn-kursov>
4. <https://cyberleninka.ru/article/n/massovye-otkrytye-onlayn-kursy-kak-tendentsiya-razvitiya-obrazovaniya>
5. <https://cyberleninka.ru/article/n/metodiki-proverki-mnogoshagovyh-zadach-v-usloviyah-smeshannogo-i-distantionnogo-avtomatizirovannogo-obucheniya>