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

Проблема

Проверка решений задач онлайн-курсов требует значительных временных затрат с участием человека, необходимо реализовать набор задач, проверка корректности которых потребует меньших временных затрат. В идеальном случае, с участием человека должна производиться только настройка параметров сервиса проверки задач.

Цель исследования

Создание набора автоматически проверяемых задач в рамках онлайн-курса “Мобильная разработка для Android на Kotlin”.

Задачи исследования

- Сравнение существующих аналогов онлайн-курсов;
- Изучение популярных методов оценки задач онлайн-курсов;
- Изучение потребности в системах автоматической проверки;
- Создание программного продукта, позволяющего проводить автоматическую проверку решений задач.

Актуальность исследования

Временные затраты на оценку задач являются ключевыми затратами ресурсов на обслуживание онлайн-курса. При расширении возможностей обучения, увеличении количества обучающихся, реализации интерактивных и многошаговых задач, полностью ручная проверка задач становится практически невыполнимой задачей.

Обзор предметной области

С целью изучения известных подходов к формированию задач и выявления проблем качества обучения был проведен анализ аналогов онлайн-курса “Мобильная разработка для Android на Kotlin”.

Онлайн-школа e-legion [6]

Автор: e-legion + Google

14 лекций, предоставляется сертификат об окончании обучения от Google. Распространяется бесплатно.

Become an Android Developer from Scratch [7]

Автор: Adam Lupu, Adam Schwem

121 лекция, содержит видеоматериалы общей длительностью более 12-ти часов. Обучение ведется на английском языке. Предоставляется сертификат об окончании обучения. Распространяется бесплатно.

Android Basics Nanodegree by Google [8]

Автор: Google

Более 200 лекций, содержит видеоматериалы общей длительностью 165 часов. Обучение ведется на английском языке. Предоставляется сертификат об окончании обучения. Включает в себя 5 подкурсов в рамках одного курса на одной платформе Udacity: [9], [10], [11], [12], [13]. Распространяется на основе ежемесячной подписки.

Специализация: Разработка приложений под Android [14]

Автор: Vanderbilt University

100 лекций, содержит видеоматериалы общей длительностью 80 часов. Обучение ведется на английском языке. Предоставляется сертификат об окончании обучения. Включает в себя 5 подкурсов в рамках одного курса на одной платформе Coursera. Распространяется на основе ежемесячной подписки.

Programming Mobile Applications for Android Handheld Systems: Part 1 [15]

Автор: University of Maryland

Описание:

21 лекция, содержит видеоматериалы общей длительностью 15 часов. Обучение ведется на английском языке. Предоставляется сертификат об

окончании обучения. Включает в себя 5 подкурсов в рамках одного курса на одной платформе Coursera. Распространяется на основе ежемесячной подписки.

Build a Simple Android App with Java [16]

Автор: TreeHouse

5 лекций, содержит видеоматериалы общей длительностью 3 часа. Обучение ведется на английском языке. Предоставляется сертификат об окончании обучения. Распространяется на основе ежемесячной подписки.

Критерии сравнения аналогов

В целях унификации сравнения и оценки представленных аналогов сформирован следующий набор критериев. Данный набор позволяет выявить проблемы качества обучения исследуемых аналогов.

Оценка обучающимися и 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 курсы, основанные на базе университетов, например, xMOOC [18], могут иметь специальное лицензирование от ВУЗа, иного учебного заведения или специальной экспертной группы. Данный критерий может свидетельствовать о высоком уровне подготовки организаторов курса.

Форма подачи информации

Наличие разных форм подачи информации, например текст или слайды, видеозапись, звуковой подкаст, позволяет обеспечить лучшее восприятие материалов курса обучающимися.

Актуальность информации

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

Проверочные задания

Проверочные задания анализируются по следующим критериям:

- Объем проверочных заданий:

- Содержательность проверочных заданий;
- Способы проверки: автоматизированные или ручные;
- Интерфейсы сдачи заданий;

Таблица сравнения по критериям [17]

Курс	Отзывы	Формы подачи информации			Информация		Проверочные задания	
		текст или слайды	видео	Звукозапись или подкаст	Актуальность	Объем	Способы проверки	Интерфейсы сдачи
https://learndroid.e-legion.ru/	+	+	+	-	Последнее обновление 29.06	14 лекций, лекции 17-40 минут, тест каждые 1-2 лекции	Тест(авто)	Тест с вариантами ответов
https://www.udemy.com/become-an-android-developer-from-scratch/	+	+	+	-	Последнее обновление 2 года назад	121 лекции, 12.5 часов, тест каждые 5 лекций	Тест(авто)	Тест с вариантами ответов
https://www.udacity.com/course/android-basics-nanodegree-by-google--nd803	+	+	+	-	Обновляется регулярно	5 курсов по 5-7 частей, каждая с лекцией от 1 до 3-х часов	Тест, code review(авто)	Интерактивный тест, тест с вариантами ответов, проверка готового приложения при помощи модуля Udacity code review
https://www.coursera.org/learn/java-for-android/programming	+	+	+	-	Обновляется регулярно	3 модуля по 3-8 частей, каждая часть 3-5 лекций по 3-20 минут	Тест, code review(авто)	тест с вариантами ответов, проверка готового приложения

https://www.coursera.org/learn/android-programming	Еще не начался	+	+	-	Еще не начался	3 модуля по 3-8 частей, каждая часть 3-5 лекций по 3-20 минут	Тест, code review(авто)	тест с вариантами ответов, проверка готового приложения
https://teamtreehouse.com/library/build-a-simple-android-app-with-java	+	+	+	-	Обновляется регулярно	35 разделов по 10 минут – 4 часа, 200 часов	Тест, code review(авто)	тест с вариантами ответов, проверка готового приложения
https://developer.android.com/training/	+	+	+	-	Обновляется регулярно	20 разделов по 4-8 тем, в основном текст/изображения, подробная информация о каждом релизе	-	-

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

- Android Basics Nanodegree by Google [8]
- Build a Simple Android App with Java [16]

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;
- 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 ответ, например код программы или 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 [5].

Существует ряд основных стратегий обучающихся для решения задач на реализацию программного модуля [4], [5]:

- 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 [5], [18], [19]:

- 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: изучение лекций, текстовых материалов, подкастов. 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.

Таким образом, 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;

Технические подробности реализации

Планируется проводить все этапы проверки задач в виртуальной среде на основе Docker. Реализованы программные сценарии для сборки и запуска Android-приложений на основе Gradle и Android SDK.

Пример формулировок задания:

Реализовать Android-приложение на языке Kotlin, содержащее следующие элементы управления:

- TextView, id = textView;
- EditText, id = editText;
- Button, id = button с подписью “Swap text”;

При нажатии на кнопку, текст элемента textView заменяется на текст из editText в момент нажатия.

Реализовать Android-приложение на языке Kotlin, содержащее следующие элементы управления:

- 2x EditText, id = arg1, arg2;
- TextView, id = result;
- 4x Button, id = add, subtract, multiply, divide

Действия при нажатии на кнопки:

- add - сложение аргументов arg1 и arg2, вывод в result;
- subtract - вычитание аргументов arg1 и arg2, вывод в result;
- multiply - умножение аргументов arg1 и arg2, вывод в result;
- divide - деление аргументов arg1 и arg2, вывод в result;

При делении на ноль необходимо выводить в result текст “Div by zero” При наличии нечисловых аргументов в элементах arg1 и arg2, необходимо выводить в result текст “Wrong input”

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-distantsionnogo-avtomatizirovannogo-obucheniya
6. https://learndroid.e-legion.ru
7. https://www.udemy.com/become-an-android-developer-from-scratch
8. https://www.udacity.com/course/android-basics-nanodegree-by-google–nd803
9. https://www.udacity.com/course/android-basics-user-interface–ud834
10. https://www.udacity.com/course/android-basics-user-input–ud836
11. https://www.udacity.com/course/android-basics-multiscreen-apps–ud839
12. https://www.udacity.com/course/android-basics-networking–ud843
13. https://www.udacity.com/course/android-basics-data-storage–ud845
14. https://www.coursera.org/specializations/android-app-development
15. https://www.coursera.org/learn/android-programming
16. https://teamtreehouse.com/library/build-a-simple-android-app-with-java
17. https://docs.google.com/spreadsheets/d/10pljGIQ2SdcYoLO6nRMlm5PuSqp1MB8pZTslw6fSYoI/edit?usp=sharing
18. https://www.igi-global.com/dictionary/xmooc/40878
19. https://cyberleninka.ru/article/n/osobennosti-ispolzovaniya-nechetkih-geneticheskih-algoritmov-dlya-resheniya-zadach-optimizatsii-i-upravleniya
20. https://cyberleninka.ru/article/n/formalizatsiya-metoda-nechetkogo-poiska-v-adaptivnyh-sistemah-avtomaticheskoy-optimizatsii