Математическое и компьютерное моделирование:
материалы III Международной научной конференции
(Омск, 12 ноября 2015 г.). Омск, 2015. С. 158–160. УДК 378.146

Д.М. Бречка

Омский государственный университет им. Ф.М. Достоевского, г. Омск

АВТОМАТИЗАЦИЯ ПРОВЕРКИ СТУДЕНЧЕСКИХ РАБОТ

В публикации рассматривается вопрос о частичной или полной автоматизации процесса проверки работ студентов высших учебных заведений по дисциплине «Программирование» в рамках выбранной автором методики преподавания. При проверке выполненных работ перед преподавателем встают следующие задачи:

1) необходимо проверить глубину знаний студента в вопросах, связанных с заданием;

2) необходимо проверить корректность выполнения задания.

Первая задача традиционно решается путем задавания студентам вопросов и выслушивания их ответов. Если говорить об автоматизации данного процесса, то здесь можно предложить давно известную технику тестирования с помощью специального программного обеспечения [1–3].

Вторую задачу можно разбить как минимум на следующие три подзадачи. Во-первых, необходимо проверить корректность работы программы.

Во-вторых, необходимо проверить корректность написания программного кода.

В-третьих, нужно выяснить, не была ли предлагаемая на проверку программа, уже сдана каким-либо другим студентом. Вопрос об автоматизации особенно актуален при решении третьей подзадачи, так как преподаватель не в состоянии запомнить все программы, которые ему были сданы.

Автоматизированное решение первой подзадачи можно осуществлять с помощью инструментов автоматизированного тестирования программного обеспечения. Примером такого инструмента может служить Robot Framework [4].

Решение второй подзадачи представляется наиболее сложным для автоматизации. Данный вопрос требует дополнительного изучения, но предполагается, что задача может решаться с помощью нейросетевых технологий. В качестве первичной проверки корректности программного кода можно предложить проверку на наличие ключевых слов или на порядок расположения ключевых слов в программе. Ключевыми словами могут быть операторы языка программирования или имена функций и библиотек необходимых для выполнения задания. Такая проверка может автоматически выявить недоработки в программах студентов.

Решение третьей подзадачи предлагается с помощью специального программного продукта, реализующего некоторый алгоритм проверки. Рассмотрим этот вопрос подробнее.

В первую очередь необходимо некоторое хранилище (база данных), в которое помещаются все работы, уже были представленные на проверку. В простейшем случае можно сравнивать проверяемую работу с работами, находящимися в хранилище, и искать совпадения. Сравнение можно проводить по бинарным файлам или хэш-кодам от бинарных файлов. Однако такая проверка может отсеять лишь абсолютно одинаковые работы, так как незначительное изменение файла, например, добавление пробельных символов или символов разрыва строки, приведет к изменению бинарного кода файла и хэш-кода. А компиляция одного и того же текста программы на разных компьютерах в большинстве случаев приводит к получению различных бинарных кодов исполняемых файлов.

Для осуществления более сложной проверки из текстов программ можно удалять все пробельные символы, а также комментарии и только после этого производить сравнение. Но и в этом случае достаточно произвести небольшую модификацию текста программы, например, изменить имена переменных, и алгоритм проверки совпадений уже не выявит.

Для осуществления хорошей проверки следует сравнивать не тексты программ, а структуры программ. Однако выявление структуры программы приводит к необходимости проведения синтаксического анализа текста программы, что является довольно нетривиальной задачей.

Литература

1. *Васильев В.И., Киринюк А.А., Тягунова Т.Н.* Требования к программно-дидактическим тестовым материалам и технологиям компьютерного тестирования. М.: Моск. гос. ун-т печати, 2005.

2. *Иванов Б.С.* Принципы и технология тестирования студентов: учеб.-метод. пособие. СПб.: Изд-во Политехн. ун-та, 2005.
3. *Ильина Е.А., Егорова Л.Г., Дьяконов А.В.* Технология тестирования знаний студентов с использованием системы Moodle // Математическое и программное обеспечение систем в промышленной и социальной сферах. 2011. № 1-3. С. 166–172.
4. Robot Framework. URL: `http://robotframework.org`.