

## ОПЫТ ПОЛУАВТОМАТИЧЕСКОЙ ПРОВЕРКИ НА ПЛАГИАТ ПРИ ВЫПОЛНЕНИИ АЛГОРИТМИЧЕСКИХ ЗАДАНИЙ

Щербина Д.Н.

ФГАОУ ВО «Южный федеральный университет»,  
УНИИБМИТ

г. Ростов-на-Дону

E-mail: dnsherbina@sfedu.ru

Навыки практического применения алгоритмов обработки данных приобретаются студентами при решении кейс-задач, где искомым алгоритм собирается из отдельных фрагментов. Фрагменты представляют собой блоки кода, снабженные комментариями и пояснениями. Из фрагментов как из кирпичиков студенты должны собрать и сдать готовый скрипт, реализующий весь алгоритм. Это творческая многокомпонентная расчетно-графическая задача предполагает многообразие исполнения: начиная от именованной переменных в программном коде и выбора цвета графика, и, кончая формулировкой выводов в свободной форме.

В отличие от традиционного бумажного дневника лабораторных работ электронная «тетрадь» поддается автоматизированной проверке на уникальность. Студенты загружают свои работы в виде файлов в формате Jupyter Notebook (с расширением .ipynb), который является расширением текстового формата JSON. Для удобства обращения со сданными файлами вводится схема именованной «kodzadania\_FamiliaIO.ipynb», где kodzadania – некоторый идентификатор задания, FamiliaIO – идентификатор студента. Для удобства работы с архивами за прошлые годы (студенты могут заимствовать работы у старшекурсников) код задания выбирается осмысленный: не просто порядковый номер вроде «hw3», а, например, «granada» - Групповой АНАлиз ДАнных. Удобство заключается в следующем: (1) включение кода задания в имя файла позволяет эффективную сортировку файлов с работами независимо от конкретной структуры папок для хранения архива за несколько лет; (2) включение фамилии студента в имя файла позволяет легко идентифицировать реального автора обнаруженного плагиата.

Следует отметить, что традиционные подходы на обнаружение заимствований не подходят для расчетно-графических работ. Большая часть текста наследуется из фрагментов, составленных преподавателем, поэтому работы разных студентов в целом очень похожи друг на друга (в процентном отношении). Принципиальные различия, отражающие самостоятельность выполнения, заключаются в конкретных полученных числовых значениях, именах переменных, формулировках промежуточных и итоговых выводов.

Проверка конкретной работы на плагиат проходит в две стадии:

1. В просматриваемой работе отмечаются информативные последовательности символов: слова с ошибками, словесные обороты из комментариев, значения вычисленных коэффициентов (должны быть уникальными при индивидуализации входных данных) и т.п.

2. Несложный скрипт проходит по всем работам с соответствующим кодом задания и выводит список тех, где встречаются искомые последовательности.

При проверке сданных работ по порядку времени загрузки на образовательный портал схема заимствований становится очевидной. Эта схема не требует больших усилий от преподавателя (проверки по 2-3 информативным последовательностям достаточно), но позволяет объективно обосновывать недобор баллов тем студентам, которые пытаются выдать чужие работы за свои, внося в них лишь небольшие косметические правки.

Следует помнить, что самостоятельная работа над алгоритмическими заданиями в отличие от тестов знаний, предполагает работу в команде. Можно лишь поощрять студентов, которые работают над заданиями парами, или консультируются друг с другом. Однако, несмотря на общее найденное решение, всегда остается место для самовыражения – при формулировке выводов, которые зависят от полученных значений (графиков), которые зависят от конкретных входных данных (разными при индивидуализации заданий).

Таким образом, представленный способ использования открытого инструментария Jupyter Notebook в качестве «электронной тетради» позволяет эффективное выявление заимствований при выполнении заданий на выработку алгоритмического мышления.