

# ВЫЯВЛЕНИЕ СПИСЫВАНИЯ С ПОМОЩЬЮ СЛАБО РАЗЛИЧИМЫХ ВОПРОСОВ

Шафран Ю.В.<sup>1</sup>, Кузьмин Н.М.<sup>2</sup>

<sup>1</sup>Шафран Юрий Владимирович – старший преподаватель;

<sup>2</sup>Кузьмин Николай Михайлович – кандидат физико-математических наук, доцент,  
кафедра информационных систем и компьютерного моделирования,  
институт математики и информационных технологий  
Волгоградский государственный университет,  
г. Волгоград

**Аннотация:** в работе рассмотрен подход к затруднению и выявлению списывания со шпаргалок путём составления слабо различимых вариантов вопросов с большим количеством деталей. Приведён пример для экзамена по программированию. Указаны достоинства и недостатки подхода.

**Ключевые слова:** экзамен, списывание, шпаргалки, нечестность.

## Проблема списывания на экзамене

Списывание на экзаменах – то есть использование шпаргалок – имеет очень длинную историю, обширную географию и массовый характер [1, 2, 3]. Однако борьба с ним пока не имеет хорошо разработанной научной основы [3].

Одним из выходов могло бы быть разрешение использовать шпаргалки, но есть опыты, свидетельствующие, что это не улучшает даже результаты экзаменов, не говоря уже о качестве образования [4]. И хотя в опросах учащиеся иногда говорят, что написание шпаргалок способствует усвоению материала, исследования [5, 6] этого вопроса показывают разные результаты в разных условиях.

Что касается списывания в российском образовании, то исследования [1, 3] показывают высокую терпимость к обману как со стороны студентов, так и со стороны преподавателей. Некоторые учёные [1] даже считают систему образования в некоторой степени «виновной» в развитии теневой экономики, потому что в процессе обучения люди учатся если не обходить правила самостоятельно, то, как минимум, закрывать глаза на такие действия окружающих.

Для борьбы со списыванием действенным подходом [1] представляется увеличение вероятности поимки: здесь работает формализация штрафных санкций и процедуры их применения [1, 7].

## Слабо различимые вопросы

В данной работе рассмотрен способ одновременно и затруднить один из необходимых этапов списывания, и упростить выявление списывания при рассмотрении письменной работы.

В качестве примера рассмотрим процедуру экзамена по программированию. В качестве вопроса, проверяющего как теоретические, так и практические навыки студентов можно использовать задание, предполагающее анализ фрагмента кода: студенту предлагается несколько строк кода, прочитав которые он должен выполнить следующие задания:

- перечислить использованные языковые элементы (типы данных, переменные, циклы, функции и так далее) и библиотечные классы;
- составить словесный алгоритм представленного кода;
- сформулировать решаемую задачу;
- привести пример работы, указав используемые входные и получающиеся выходные данные.

Пример двух вариантов задания на анализ кода приведён на 0 и 0. Такие фрагменты кода имеют множество деталей, которые можно незначительно варьировать в разных вариантах вопроса. В приведённом примере метод `getCounter` имеет разные типы возвращаемого значения, что сказывается на использовании его результата в методе `main`. Таким образом, при реально проведённом анализе кода перепутать приведённые варианты не представляется возможным, а описание, относящееся к другому варианту, может служить признаком недобросовестности.

```

public static void main(String[] args) {
    System.out.print("Text: ");
    String text = new Scanner(System.in).nextLine().toUpperCase();
    if (text.length() == text.codePointCount(0, text.length()))
        getCounter().apply(text).forEach(Main::print);
    else
        System.out.println("Supplementary characters are used.");
}

private static void print(char letter, int count) {
    System.out.printf("Letter '%c': %03d\n", letter, count);
}

private static Function<String, Map<Character, Integer>> getCounter() {
    return new Function<String, Map<Character, Integer>>() {
        @Override
        public Map<Character, Integer> apply(String s) {
            Map<Character, Integer> letters = new TreeMap<>();
            for (int i = 0; i < s.length(); i++)
                if (Character.isLetter(s.charAt(i)))
                    letters.merge(s.charAt(i), value: 1, Integer::sum);
            return letters;
        }
    };
}

```

Рис. 1. Вариант вопроса с Function

```

public static void main(String[] args) {
    System.out.print("Text: ");
    String text = new Scanner(System.in).nextLine().toUpperCase();
    if (text.length() == text.codePointCount(0, text.length()))
        getCounter(text).get().forEach(Main::print);
    else
        System.out.println("Supplementary characters are used.");
}

private static void print(char letter, int count) {
    System.out.printf("Letter '%c': %03d\n", letter, count);
}

private static Supplier<Map<Character, Integer>> getCounter(String s) {
    return new Supplier<Map<Character, Integer>>() {
        @Override
        public Map<Character, Integer> get() {
            Map<Character, Integer> letters = new TreeMap<>();
            for (int i = 0; i < s.length(); i++)
                if (Character.isLetter(s.charAt(i)))
                    letters.merge(s.charAt(i), value: 1, Integer::sum);
            return letters;
        }
    };
}

```

Рис. 2. Вариант вопроса с Supplier

Что касается затруднения самой процедуры списывания, то при достаточно большом количестве вариантов точное сопоставление фрагментов кода в вопросе и в шпаргалке требует внимательности и сосредоточенности на деталях или технических средств, доступных только при попустительстве списыванию.

#### Ограничения подхода

Можно выделить следующие недостатки метода:

- трудоёмкость составления вариантов вопросов;
- формулирование вопросов с множеством деталей подходит не для каждой учебной дисциплины;
- при оценивании только письменного ответа подозрение в нечестности едва ли может служить поводом к обнулению результата.

Также надо заметить, что наиболее действенным способом победить списывание в частности и нечестность обучающихся вообще может [7] только их искренний интерес к изучаемому предмету, а пробуждение этого интереса – важнейшая задача преподавателя.

#### **Список литературы**

1. *Латова Н.В., Латов Ю.В.* Обман в учебном процессе // *Общественные науки и современность*, 2007. № 1. С. 31-46.
2. *Горбов Л.В., Чигрин С.В.* Новые приемы списывания при компьютерном тестировании и способы борьбы с ними // *Академия естествознания*, 2013. № 4. С. 93-96.
3. *Иголкина Н.И.* Списывание в учебном процессе: причина или следствие педагогического конфликта // *Конфликты в современном мире: международное, государственное и межличностное измерение: международная научная конференция (Саратов, 8 апреля 2016 года)*. Москва: Издательство «Перо», 2016. С.891-895.
4. *Dickson K.L., Miller M. D.* Authorized Crib Cards Do Not Improve Exam Performance // *Teaching of Psychology*. 2005. Volume 32. Issue 4. Pp. 230-233.
5. *Wachsmann Y.* Should Cheat Sheets be Used as Study Aids in Economics Tests? // *Economics Bulletin*, 2002. Vol. 1 (1). Pp. 1-11.
6. *Erbe B.* Reducing Test Anxiety While Increasing Learning: The Cheat Sheet // *College Teaching*, 2007. Volume 55. Issue 3. Pp. 96-98.
7. *Шмелева Е.Д.* Плагиат и списывание в российских вузах: роль образовательной среды и индивидуальных характеристик студента // *Вопросы образования*, 2016. № 1. С. 84-109.