
**ОПЫТ РАЗРАБОТКИ СИСТЕМЫ «АНТИПЛАГИАТ»
НА БАЗЕ ПЛАТФОРМЫ 1С 8.1 ДЛЯ ПРОВЕРКИ
ВЫПУСКНЫХ КВАЛИФИКАЦИОННЫХ РАБОТ
СТУДЕНТОВ**

А.А.Ступин

Новосибирский государственный педагогический университет

Предлагается метод решения одной из самых актуальных на сегодняшний день проблем образования – плагиата выпускных квалификационных работ студентов ВУЗов. Использование разработанной автором системы «Антиплагиат» позволяет повысить качество дипломных работ выпускников.

Ключевые слова и словосочетания: плагиат текста, шинглы, регулярные выражения, дипломная работа.

Председатель Высшей аттестационной комиссии (ВАК) Минобрнауки Михаил Кирпичников 23 марта 2006 года на интернет-конференции по проблемам аттестации научных кадров с горечью отмечал, что до трети диссертаций в России покупается за деньги, плагиат в научных работах – обычное дело. И если за прошедшее время в сфере аттестации научных кадров благодаря усилиям ВАК произошли положительные изменения, то в сфере выпускных квалификационных работ студентов ВУЗов, где положение дел ещё более плачевно, ничего не изменилось. В связи с этим задача разработки системы «Антиплагиат» для определения заимствований в текстах выпускных квалификационных работ студентов является весьма актуальной.

В связи с возникновением необходимости в реализации технологии проверки дипломных работ (текстовых документов) на наличие заимствований из ранее защищенных выпускниками был определен функционал будущей системы:

- электронные версии дипломных работ должны храниться в базе данных;
- вновь добавляемая в базу работа должна сравниваться с имеющимися в базе работами;
- на основании сравнения делается вывод, является ли добавленная в базу работа плагиатом.

В качестве платформы для разработки системы нами была выбрана «1С: Предприятие 8.1».

Для выполнения задачи хранения электронных версий дипломных работ в баз данных были созданы справочники для внесения информации о дипломных работах в базу.

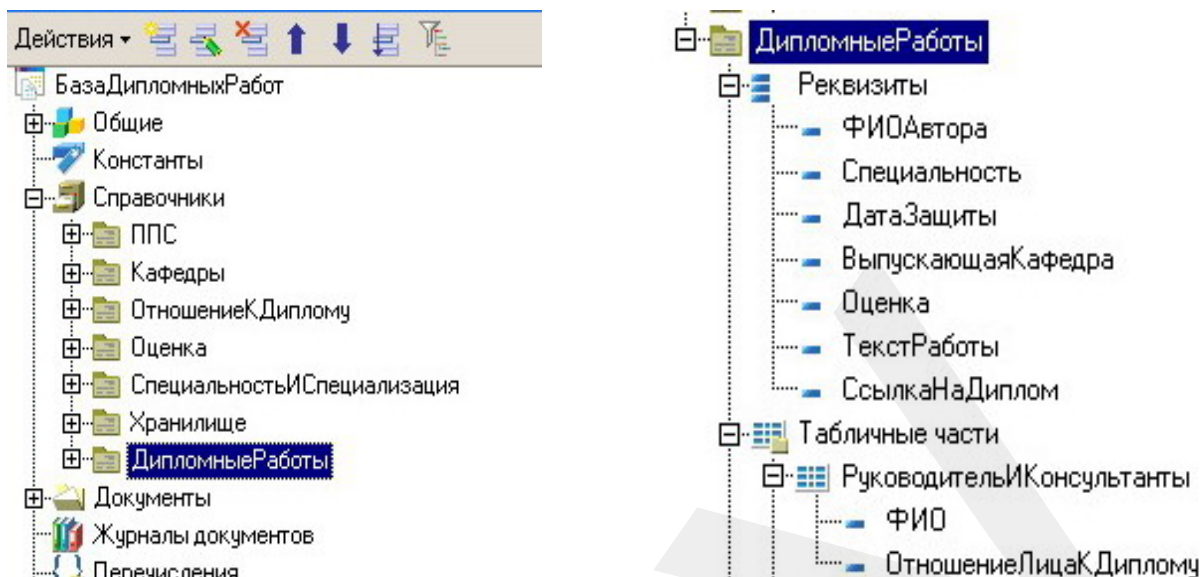


Рис. 1. Структура справочников и реквизиты справочника «Дипломные работы»

The screenshot shows the 'Справочник Дипломные работы' application window. The table displays the following data:

Код	Наименование	ФИОАвтора	Специальность	Ссылка	Шинглы...
000000074	Анализ коммерческой деятельности предприятия ООО "ТопКнига"	Юренкова Олеся Сергеевна	Экономика и управление	77	✓
000000039	Анализ коммерческой деятельности предприятия розничной торговли	Литвинов Виктор Владимирович	Экономика и управление	39	✓
000000121	Анализ конкурентоспособности И.П. Гречко в г. Черепаново	Стародубцев Александр Павлович	Экономика и управление	127	✓
000000021	Анализ конкурентоспособности предприятия по очистке питьевой воды ЗАО Компании "Росинка"	Мельникова Лилия Владимировна	Прикладная экономика	21	✓
000000040	Анализ маркетинговой деятельности предприятия розничной торговли (на примере предприятия ООО...)	Беляев Денис Сергеевич	Автомобили и автомобильное хо...	40	✓
000000035	Анализ маркетинговой среды ООО "Новосибирск-Торг" "Пятёрочка"	Спирин Александр Александрович	Экономика и управление	35	✓
000000116	Анализ маркетинговых стратегий общественного питания г. Искитима	Пихтина Елена Валерьевна	Экономика и управление	122	✓
000000195	Анализ маркетинговых стратегий предприятия ООО "Делмар"	Бабич Олег Игоревич	Экономика и управление	203	✓
000000064	Анализ развития рынка недвижимости города Новосибирска	Болдин Константин Геннадьевич	Прикладная экономика	64	✓
000000037	Анализ системы профессионального обучения мастеров по моделированию и дизайну ногтей в учебн...	Беляева Юлия Евгеньевна	Экономика и управление	37	✓
000000095	Анализ системы управления на предприятии	Кальван Александр Николаевич	Экономика и управление	101	✓
000000038	Анализ системы управления персоналом на предприятии (на примере ООО "ЭЛСИ")	Нестюшенко Елена Юрьевна	Экономика и управление	38	✓
000000085	Анализ современных форм и систем оплаты труда на примере ОАО «Аэропорт Толмачево»	Картавченко Елена Петровна	Прикладная экономика	88	✓
000000108	Анализ финансового состояния и управление финансами в ООО ЧОП "Патрон"	Абросимова Светлана Ивановна	Экономика и управление	114	✓
000000049	Анализ финансового состояния ОАО "Завод молочный "Новосибирский"	Елкина Светлана Георгиевна	Экономика и управление	49	✓
000000182	Анализ финансового состояния организации ООО "Сибирский бумажный центр"	Солодилова Оксана Евгеньевна	Экономика и управление	190	✓
000000154	Анализ финансового состояния предприятия ЗАО "Веста"	Вейт Татьяна Александровна	Экономика и управление	161	✓
000000179	Анализ финансового состояния предприятия ООО "Бердское специальное конструкторское бюро "Ве...	Худина Дарья Сергеевна	Экономика и управление	187	✓
000000207	Анализ финансово-хозяйственной деятельности малого предприятия ООО "Золотая рыбка"	Комжова Ольга Александровна	Экономика и управление	215	✓
000000023	Анализ финансово-хозяйственной деятельности на примере предприятия ОАО "Черепановский заво...	Сборщиков Максим Игоревич	Экономика и управление	23	✓
000000093	Анализ финансово-хозяйственной деятельности ООО "Росгосстрах-Сибирь"	Телкова Екатерина Сергеевна	Экономика и управление	99	✓
000000183	Анализ финансово-хозяйственной деятельности предприятия ООО "Свет-Актив"	Коврижных Ирина Павловна	Экономика и управление	191	✓

Рис. 2. Справочник «Дипломные Работы» с внесёнными в базу работами

...: Анализ финансового состояния ОАО "Завод молочн... _ □ ×

Действия ▾ [←] [↻] [→] [?]

Код: 000000049

Наименование: Анализ финансового состояния ОАО "Завод молочн...

ФИО Автора: Елкина Светлана Георгиевна

Специальность: Экономика и управление ... ×

Дата: 02.06.2008 [📅]

Выпускающая кафедра: Педагогики, психологии и предпринимательств ... ×

Оценка: Отлично ... ×

Текст работы: ДИПЛОМНАЯ ТЕМУ: «Анализ финансового состояни

В хранилище 49 49

Страница1 | Страница2

[🔍] [➕] [➖] [✖] [📄] [↑] [↓] [🔄] [🔍]

N	ФИО	Отношение лица к диплому
1	Позднякова Алла Николаевна	руководитель
2	Ордин Александр Александр...	консультант экономической части
3	Кушнир Светлана Викторовна	консультант методической части
4	Леонов Михаил Владимирович	нормоконтролёр
5	Голубева Ирина Владимиров...	рецензент

Записать | Закреть | OK

Рис. 3. Карточка дипломной работы

Для хранения электронных версий дипломных работ в базе данных использован механизм «Хранилище» платформы 1С. При сохранении дипломные работы сжимаются встроенными методами архивирования для уменьшения объёма базы.

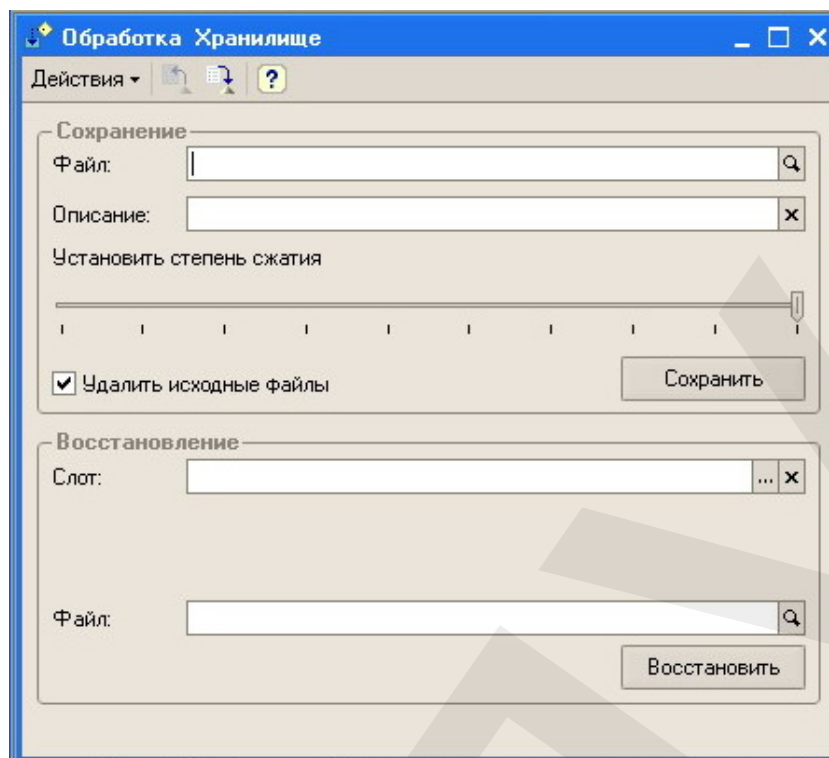


Рис. 4. Обработка «Хранилище» для сохранения электронных версий дипломных работ в базе

Функция сохранения файла реализована следующим образом:

```

Функция СохранитьФайлВХранилище(ИмяФайла)
  Файл=Новый ДвоичныеДанные(ИмяФайла);
  Сжатие=Новый
СжатиеДанных(ЭлементыФормы.ПолосаРегулирования.Значение);
  ХранилищеФайла=Новый ХранилищеЗначения(Файл,Сжатие);
  НовыйСлотХранилища= Справочники.Хранилище. СоздатьЭле-
мент());
  НовыйСлотХранилища. Наименование=ЭлементыФормы. ВыбО-
писание.Значение;
  НовыйСлотХранилища.Слот=ХранилищеФайла;
  НовыйСлотХранилища.ИмяФайла=ИмяФайла;
  НовыйСлотХранилища.Записать();
КонецФункции

```

Восстановление файла из «Хранилища» реализовано с помощью следующей процедуры:

Процедура КнопкаВосстановитьНажатие(Элемент)**Если ВыбСлот.Пустая()=Ложь Тогда****Если Не ПустаяСтрока(ВыбИмяФайлаЗаписи) Тогда****Слот=ЭлементыФормы.ВыбСлот.Значение;****Файл=Слот.Слот.Получить();****ИмяФайла=ЭлементыФормы.ВыбИмяФайлаЗаписи.Значение;****Файл.Записать(ИмяФайла);****КонецЕсли;****КонецЕсли;****КонецПроцедуры**

Для решения задачи выявления плагиата текста мы применили метод «Шинглов», предложенный Андреем Бродером в 1997 году [1]. Шингл (от англ. shingle) – некая последовательность элементов (слов) в документе (подстрока текста). Выбирать такие подстроки можно по-разному. Можно брать разный шаг, например: символ, слово, несколько слов подряд, предложение. Для получения шинглов подстроки текста должны идти «внахлёт». Далее следует понять, какого размера должны быть подстроки (выбранный размер должен свести к минимуму случайные повторы, то есть должен быть достаточно большим, но при этом оставаться достаточно малым, чтобы типичные изменения текста не сказались на результате проверки). Поскольку возможных подстрочек в тексте чересчур много, надо решить, какие запоминать, а какие отбрасывать.

Например, в тексте “*a rose is a rose a rose*” выделенное курсивом “*is a rose*” может считаться шинглом. За *w*-shingling принимается набор уникальных шинглов длины *w*. Например, если мы ищем 4-shingling, то из “*a rose is a rose is a rose*” всего получится шесть последовательностей из 4 элементов, идущих внахлёт: { (a, rose, is, a), (rose, is, a, rose), (is, a, rose, is), (a, rose, is, a), (rose, is, a, rose) }. Отбрасываем дублирующиеся шинглы и получаем искомый 4-shingling: { (a, rose, is, a), (rose, is, a, rose), (is, a, rose, is) }.

Далее с помощью шинглов оценивают схожесть двух текстов. Для этого сначала находят shinglings для первого текста $S(A)$ и для второго $S(B)$, а потом делят число элементов в пересечении множеств

$S(A)$ и $S(B)$ на число элементов в объединении этих же множеств. В итоге получается величина от 0 до 1, которая и показывает степень схожести документов. Полученное число умножают на 100% и получают процент схожести текстов. Для идентичных документов это число равно 100%.

$$r(A, B) = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|} \times 100\%. \quad (1)$$

Для реализации данного метода текст первоначально необходимо привести к каноническому виду (буквы перевести в нижний регистр и оставить только слова из текста через 1 пробел). Кроме того, для таких больших текстов, как дипломная работа, можно откинуть мало значащие слова (например предлоги), что в результате повысит точность сравнения при намеренном изменении студентами текстов путем замены малозначащих слов (например, ООО на ОАО и т.д.). Поэтому мы решили отбрасывать слова длиной меньше трёх символов.

На языке 1С перевод текста к нижнему регистру выполняется встроенной функцией:

ТекстДляШинглов = НРег(ТекстДляШинглов);

Для решения задачи извлечения из текста только слов мы использовали регулярные выражения. Регулярные выражения — система синтаксического разбора текстовых фрагментов по формализованному шаблону, основанная на системе записи образцов для поиска [4]. Образец (от англ. pattern), задающий правило поиска, по-русски также иногда называют «шаблоном». Хотя язык 1С не поддерживает регулярные выражения, можно подключить библиотеку из VBScript, поставляющуюся вместе с MS IE и имеющуюся на каждом компьютере с операционной системой MS Windows.

Реализация данной функции:

```

Перем РегВыр, Dictionary;
Функция Извлечение(X) Экспорт
  Шаблон = "[a-zA-Za-яА-Я]+";
  // РезультатыПоиска.Очистить();
  РегВыр.IgnoreCase = False ;
  РегВыр.Global = True;
  РегВыр.Multiline = False;
РегВыр.Pattern = СокрЛП(Шаблон);
  Попытка
    Matches = РегВыр.Execute(СокрЛП(X));
Исключение
  Сообщить("Ошибка при поисках регулярного выражения");
  Возврат Ложь;
КонецПопытки;
  СЗ = Новый СписокЗначений;
Если Matches.Count() > 0 Тогда
  СчетчикЦикла = 0;
  Для СчетчикЦикла = 0 По Matches.Count()-1 Цикл
    Matches = Matches.Item(СчетчикЦикла);
    СтрокаТ = Matches.Value;
    ДлинаСлова = СтрДлина(СокрЛП(СтрокаТ));
    Если ДлинаСлова > 3 Тогда
      СЗ.Добавить(СокрЛП(СтрокаТ));
    КонецЕсли;
  КонецЦикла;
КонецЕсли;

  МассивМой = Новый Массив ();
  МассивМой.Очистить();
  МассивМой = СЗ.ВыгрузитьЗначения();
Возврат МассивМой;
КонецФункции

Попытка
  РегВыр = Новый СОМОБЪЕКТ("VBScript.RegExp");
  Dictionary = Новый СОМОБЪЕКТ("Scripting.Dictionary");
  IgnoreCase = РегВыр.IgnoreCase;
  Global = РегВыр.Global;
  Multiline = РегВыр.Multiline;

```


Исключение

```
Сообщить("Неудачная попытка создания  
""VBScript.RegExp""",СтатусСообщения.Внимание);  
КонецПопытки;
```

В результате работы функции получаем массив слов из обрабатываемого текста. Далее из полученного массива формируются 3-shingling:

```
ш=3;  
i = 0; //счетчик  
Пока i < ((ЭлементыМассиваМоего.ВГраница()+1) - (ш - 1)) Цикл  
ОдинШингл = СокрЛП(ЭлементыМассиваМоего.Получить(i))+ " " +  
СокрЛП(ЭлементыМассиваМоего.Получить(i+1)) + " " +  
СокрЛП(ЭлементыМассиваМоего.Получить(i+2));  
МассивШинглов.Добавить(ОдинШингл); // получаем массив шинглов  
i = i + 1;  
КонецЦикла;
```

Далее необходимо провести сравнение добавляемой в базу работы с имеющимися, для чего необходимо поочередно находить объединение и пересечение массива шинглов добавляемой работы и массива шинглов имеющихся в базе работ.

При работе с массивами повторяющиеся шинглы из массивов удалить можно, например, так [2]:

```
Процедура УдалитьПовторяющиесяЭлементы(Массив)  
ТекущийИндекс = 0;  
ВсегоЭлементов = Массив.Количество();  
Пока ТекущийИндекс < ВсегоЭлементов Цикл  
Индекс2 = ТекущийИндекс + 1;  
Пока Индекс2 < ВсегоЭлементов Цикл  
Если Массив[Индекс2] = Массив[ТекущийИндекс] Тогда  
Массив.Удалить(Индекс2);  
ВсегоЭлементов = ВсегоЭлементов - 1;  
Иначе  
Индекс2 = Индекс2 + 1;
```

```
КонецЕсли;  
КонецЦикла;  
ТекущийИндекс = ТекущийИндекс + 1;  
КонецЦикла;  
КонецПроцедуры
```

Объединение массивов найти можно следующим образом [2]:

```
Для Каждого Элемент Из Массив1 Цикл  
Массив2.Добавить(Элемент);  
КонецЦикла;
```

Однако, как показала проверка, такая работа в циклах является далеко не оптимальной, с точки зрения производительности и быстродействия программы. Поэтому, для повышения быстродействия системы, мы решили сохранять полученные шинглы в табличную часть «Шинглы» справочника «Дипломные работы» и для нахождения объединения, пересечения массивов шинглов и удаления повторяющихся шинглов использовать запросы (см. рис. 5).

Таким образом реализация функции объединения массивов выглядит следующим образом:

```
Функция Объединение (Код1,Код2)  
Запрос = Новый Запрос;  
Запрос.Текст="ВЫБРАТЬ  
| ДипломныеРаботыШинглы.Шингл,  
| ДипломныеРаботыШинглы.Ссылка.Код КАК Код  
|ИЗ  
| Справочник.ДипломныеРаботы.Шинглы КАК ДипломныеРа-  
ботыШинглы  
|ГДЕ  
| ДипломныеРаботыШинглы.Ссылка.Код = &Код1  
|ОБЪЕДИНИТЬ  
|ВЫБРАТЬ  
| ДипломныеРаботыШинглы.Шингл,  
| ДипломныеРаботыШинглы.Ссылка.Код
```

|ИЗ

| Справочник.ДипломныеРаботы.Шинглы КАК ДипломныеРаботыШинглы

|ГДЕ

| ДипломныеРаботыШинглы.Ссылка.Код = &Код2";

Запрос.УстановитьПараметр("Код1",Код1); //Параметры запроса

Запрос.УстановитьПараметр("Код2",Код2);

Результат=Запрос.Выполнить();

Выборка = Результат.Выбрать();

Кол = Выборка.Количество();

Возврат Кол;

КонецФункции

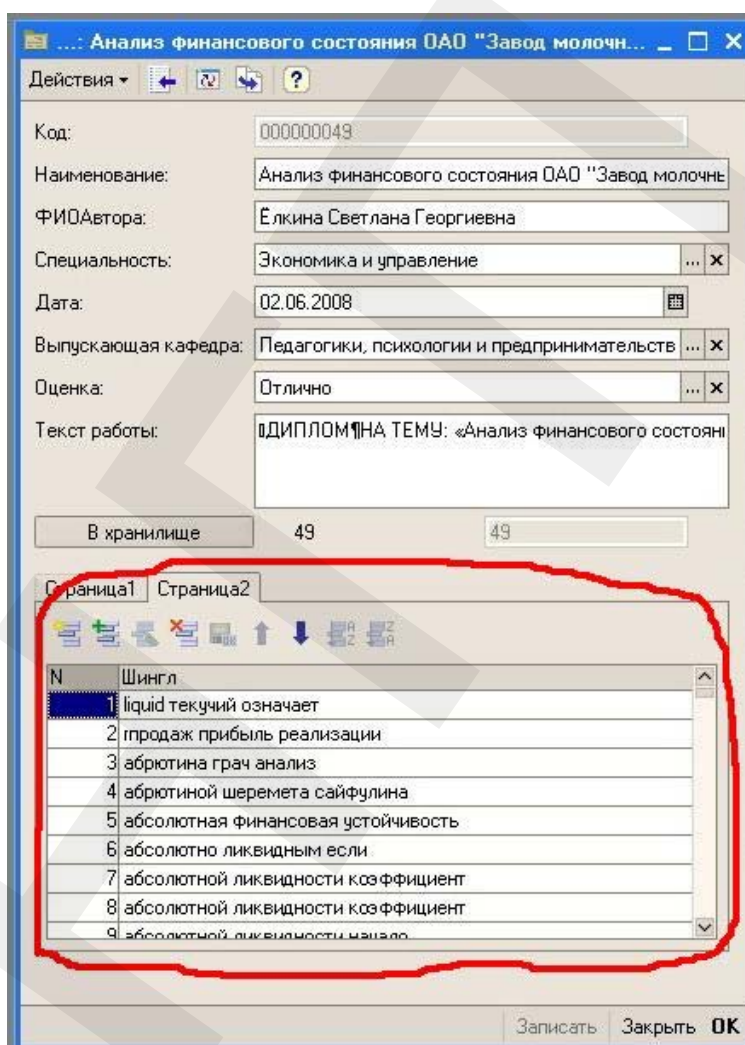


Рис. 5. Сохранённые сформированные шинглы

Оператор «ОБЪЕДИНИТЬ (UNION)» автоматически исключает дубликаты строк из вывода. Это нечто, не свойственное для SQL, так как одиночные запросы обычно содержат оператор «ВЫБРАТЬ РАЗЛИЧНЫЕ (DISTINCT)», примененный нами в функции пересечения, чтобы устранять дубликаты. Пересечение массивов получаем с помощью оператора «ВНУТРЕННЕЕ СОЕДИНЕНИЕ (INNER JOIN)», возвращающий в результирующем наборе данных записи, в которых значения в связанных полях совпадают [3].

Функция пересечения массивов принимает следующий вид:

Функция Пересечение (Код1,Код2)

Запрос = Новый Запрос;

Запрос.Текст="ВЫБРАТЬ

| **ДипломныеРаботыШинглы.Шингл,**

| **ДипломныеРаботыШинглы.Ссылка.Код,**

| **ДипломныеРаботыШинглы.Ссылка**

| **ПОМЕСТИТЬ ВремТаб1**

| **ИЗ**

| **Справочник.ДипломныеРаботы.Шинглы КАК ДипломныеРаботыШинглы**

| **ГДЕ**

| **ДипломныеРаботыШинглы.Ссылка.Код = &Код1;**

| **ВЫБРАТЬ**

| **ДипломныеРаботыШинглы.Шингл,**

| **ДипломныеРаботыШинглы.Ссылка.Код,**

| **ДипломныеРаботыШинглы.Ссылка**

| **ПОМЕСТИТЬ ВремТаб2**

| **ИЗ**

| **Справочник.ДипломныеРаботы.Шинглы КАК ДипломныеРаботыШинглы**

| **ГДЕ**

| **ДипломныеРаботыШинглы.Ссылка.Код = &Код2;**

| **ВЫБРАТЬ РАЗЛИЧНЫЕ**

| **ВремТаб1.Шингл Как Ш**

| **ИЗ**

| **ВремТаб1 КАК ВремТаб1**

| **ВНУТРЕННЕЕ СОЕДИНЕНИЕ ВремТаб2 КАК ВремТаб2**

```

|   ПО ВремТаб1.Шингл = ВремТаб2.Шингл";
Запрос.УстановитьПараметр("Код1",Код1); //Параметры запро-
са
Запрос.УстановитьПараметр("Код2",Код2);
Результат=Запрос.Выполнить();
ТаблицаЗначений = Новый ТаблицаЗначений;
ТаблицаЗначений = Результат.Выгрузить();
Кол = ТаблицаЗначений.Количество();
Возврат Кол;
КонецФункции

```

В результате функции возвращают количество элементов массивов объединения и пересечения функций. Далее проводится проверка на плагиат, во время которого пользователь видит прогресс-бар и может оценить оставшееся до конца сравнения время.

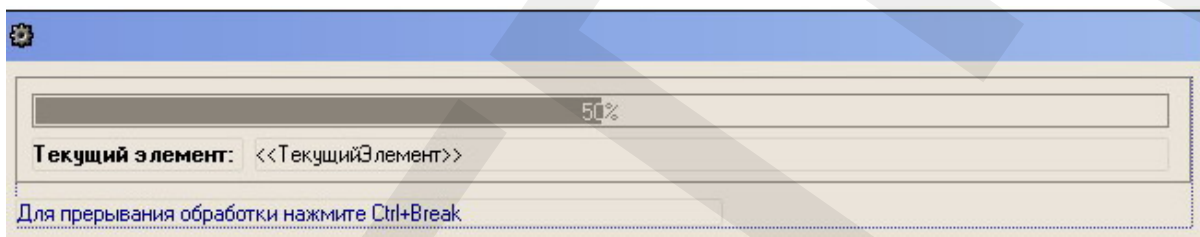


Рис. 6. Прогресс-бар

Реализация данной процедуры:

```

Процедура ПроверитьНаПлагиат()
Запрос = Новый Запрос(
"ВЫБРАТЬ
| ДипломныеРаботы.Ссылка,
| ДипломныеРаботы.Код,
| ДипломныеРаботы.Наименование
| ИЗ
| Справочник.ДипломныеРаботы КАК ДипломныеРаботы
| АВТОУПОРЯДОЧИВАНИЕ");

Результат = Запрос.Выполнить().Выбрать();
// Инициализация прогресс-бара//

```

```

Всего=Результат.Количество();
ФормаПроцесса=ПолучитьФорму("ФормаПроцесса");
ФормаПроцесса.Заголовок="Обработка данных...";
ФормаПроцес-
са.ЭлементыФормы.ИндикаторПроцесса.МаксимальноеЗначение=Всего;
ФормаПроцесса.Открыть();
н=0;
//-----
ТаблицаЗначений = Новый ТаблицаЗначений;
// Создать нужные колонки.
ТаблицаЗначений.Колонки.Добавить("КодДокумента");
ТаблицаЗначе-
ний.Колонки.Добавить("СсылкаНаДокументСовпадения");
ТаблицаЗначений.Колонки.Добавить("ПроцентСовпадения");
//-----
Пока Результат.Следующий() Цикл
// Установка текущих значений прогресс-бара//
н=н+1;
ОбработкаПрерыванияПользователя();
ФормаПроцес-
са.ЭлементыФормы.ТекущийЭлемент.Заголовок=Результат.Ссылка;
ФормаПроцесса.ЭлементыФормы.ИндикаторПроцесса.Значение=н;
// Далее непосредственно обработка данных...
Код2 = Результат.Код;
СсылкаНаДокумент = Результат.Ссылка;
Если Не Код1 = Код2 Тогда
Сообщить("Код1 = " + Код1 + ", Код2 = " + Код2);
Об = Объединение (Код1,Код2);
РезультатСравнения = (Пер/Об)*100;
// Сообщить("Образец совпадает с документом " + СсылкаНаДоку-
мент + " на " + РезультатСравнения + "%");
//-----
Если РезультатСравнения > 30 Тогда
НоваяСтрока = ТаблицаЗначений.Добавить();
НоваяСтрока.КодДокумента = Код2;
НоваяСтрока.СсылкаНаДокументСовпадения = СсылкаНаДоку-
мент;
НоваяСтрока.ПроцентСовпадения = РезультатСравнения;
КонецЕсли;

```



```

КонецЕсли;
КонецЦикла;
ФормаПроцесса.Заккрыть(); // Закрываем форму процесс-бара
ТабличноеПолеДанные = ТаблицаЗначений.Скопировать();
ЭлементыФормы.ТабличноеПолеОтображение.Значение = Таблич-
ноеПолеДанные;
ЭлементыФормы.ТабличноеПолеОтображение.СоздатьКолонки();
КонецПроцедуры

```

В результате вычлняем список дипломных работ из базы, с которыми вновь добавляемая работа совпадает более чем на 30% и производим визуальное сравнение вновь добавленного образца с этими работами. Для этого создаем обработку, в которой производим визуальное сравнение, где цветом выделяются одинаковые, измененные и добавленные куски текста.

```

Функция Сравнить() Экспорт
T1 = Новый ТекстовыйДокумент();
T1.УстановитьТекст(Текст1);
Имя1 = ПолучитьИмяВременногоФайла("ТХТ");
T1.Записать(Имя1);
T2 = Новый ТекстовыйДокумент();
T2.УстановитьТекст(Текст2);
Имя2 = ПолучитьИмяВременногоФайла("ТХТ");
T2.Записать(Имя2);
Ср = Новый СравнениеФайлов();
Ср.ПервыйФайл = Имя1;
Ср.ВторойФайл = Имя2;
Ср.ИгнорироватьПустоеПространство = Истина;
Ср.УчитыватьРазделителиСтрок = Ложь;
Ср.УчитыватьРегистр = Ложь;
Ср.СпособСравнения = СпособСравненияФай-
лов.ТекстовыйДокумент;
Ср.ПоказатьРазличия();
КонецФункции

```

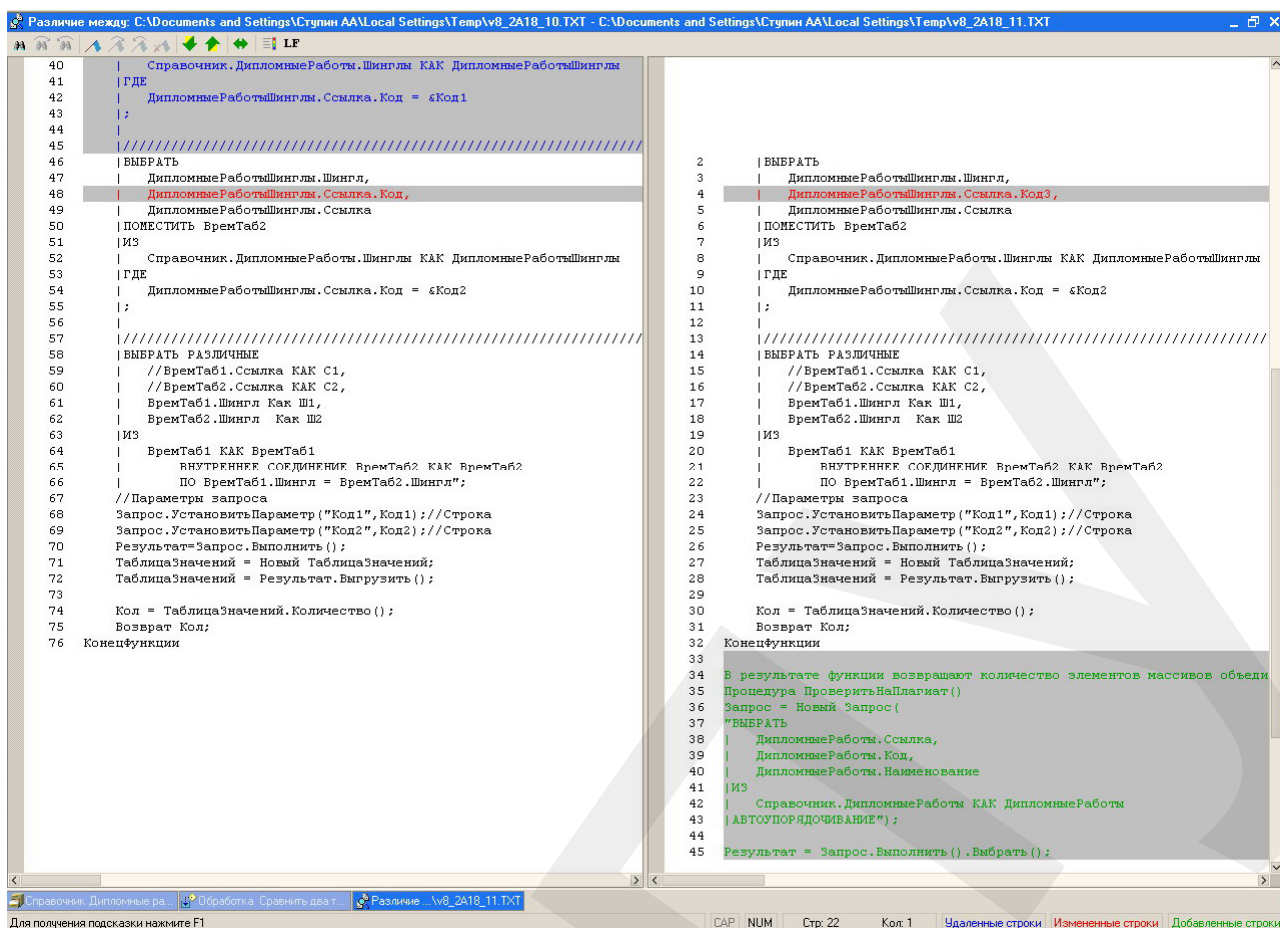


Рис. 7. Результат работы обработки для визуального сравнения двух текстов

По результатам сравнения делается вывод, является ли работа плагиатом, и если является, такая работа не допускается к защите.

Данная разработка внедрена на факультете технологии и предпринимательства НГПУ. Её эксплуатация показала высокую эффективность и повысила качество выпускных квалификационных работ студентов.

Библиографический список

1. **Broder, A.Z.** Syntactic Clustering of the Web / A.Z. Broder, S.C. Glassman, M.S. Manasse, G. Zweig // Computer Networks and ISDN Systems. – Volume 29 – Issue 8–13 – September, 1997 – P. 1157–1166.
2. **Габец, А.П.** 1С: Предприятие 8.1. Простые примеры разработки / А.П. Габец. – М., 1С-Паблишинг, 2008. – 383 с.
3. **Клайн, К.** SQL: Справочник / К. Клайн. – М., Кудиц-Образ, 2006. – 832 с.
4. **Фридл, Д.** Регулярные выражения / Д. Фридл. – СПб., Питер, 2001. – 352 с.

"ANTIPLAGIAT" SYSTEM DEVELOPMENT EXPERIENCE BASED ON THE 1C 8.1 PLATFORM FOR TESTING GRADUATE QUALIFYING WORKS OF STUDENTS

A.A. Stupin

The paper proposes a solution method for one of the most actual problems of education – plagiarism of graduate qualifying works of students of institutions of higher education. The use of the system "Anti-plagiat" developed by the author allows increasing quality of graduation works of students.

Key words: text plagiarism, shingles, regular expressions, graduation work.