МИНОБРНАУКИ РОССИИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет *романо-германской филологии*

Кафедра *испанской филологии*

Направление *«Лингвистика»*

Профиль *«Теория и методика преподавания иностранных языков и культур»*

Проект на тему:

«Открытые данные. Регулярные выражения. Скриншоты. Вычитка отсканированных и распознанных документов. TEI. Формулярный анализ исторических документов. Сравнение двух вариантов одного текста: диффы»

Выполнила: студентка 1 курса 1 группы Касилова Анастасия Владимировна

Руководитель: Донина Ольга Валерьевна

Г. Воронеж

2018

Оглавление

Введение

Современные люди вынуждены выживать в мире, где ежедневно на нас обрушивается огромный поток данных. В большей степени от этого страдают связанные с образовательным процессом – являющиеся обучаемым или обучающим. Оптимальный подход, который даст наилучший результат, – это знание методов работы с полученной информацией, что помогает улучшить воспринимаемость. Если участники процесса подготовлены к нему, они могут создать корректную стратегию взаимодействия со сведениями, на основании чего каждый вынесет максимум пользы для себя. Метод работы с этой информацией определяется выбором ситуаций, через которые организован процесс познания новой области. В то же время важно, чтобы участники могли сами формировать нестандартные ситуации и применять в них получаемые данные. Это поможет лучше усваивать их, то есть рабочий процесс будет продуктивным, эффективным.

Человеку нужно научится правильно пользоваться информацией, сортировать её, использовать для своих личных целей, применяя более удобные методы. Как же в этом огромном потоке данных выбрать то, что нужно именно сейчас, что нужно именно тебе?

Любой пользователь мировой сети, обладающий цифровой грамотностью, знает, что существует множество программ и проектов, помогающих нам грамотно распределять и использовать информацию. С некоторыми из них мы познакомимся сейчас.

Основная часть

1.Открытые данные

Открытые данные (англ. *open data*) — концепция, отражающая идею о том, что определённые данные должны быть свободно доступны для машиночитаемого использования и дальнейшей републикации без ограничений авторского права, патентов и других механизмов контроля. Освободить данные от ограничений авторского права можно с помощью свободных лицензий, таких как лицензий Creative Commons. Если какой-либо набор данных не является общественным достоянием, либо не связан лицензией, дающей права на свободное повторное использование, то такой набор данных не считается открытым, даже если он выложен в машиночитаемом виде в Интернет.

Цели движения открытых данных похожи на другие «открытые» движения, такие как открытое программное обеспечение (open source), открытый контент (open content) и открытый доступ (open access). Рост популярности идеи об открытых данных во второй половине 2000-х годов связан, прежде всего, с запуском правительственных инициатив, таких как Data.gov.

Открытые данные часто ассоциируются с нетекстовыми материалами такими как карты, геномы, химические компоненты, математические и научные формулы, медицинские данные, данные о биологическом разнообразии. Проблемы чаще всего возникают по той причине, что эти данные могут быть коммерчески ценными или могут быть собраны в некие ценные продукты.

Доступ к данным, как и последующее их использование, контролируется организациями — государственными и частными. Контроль может быть через ограничения, лицензии, копирайт, патенты и требования оплаты для доступа или повторного использования. Сторонники идеи «открытых данных» считают, что подобные ограничения идут против общественного

блага и данные должны быть доступны без ограничений или оплаты. Также важно, что данные должны быть доступны без последующих запросов на разрешение, хотя и способы повторного использования, такие как создание продуктов на базе данных, могут контролироваться лицензией.

Открытые данные (ОД) в широком смысле — это та часть раскрываемой органами государственной власти и местного самоуправления информации, которая отвечает требованиям:

- свободы доступа;
- свободы использования;
- автоматической обработки (машиночитаемости).

Механизмом внедрения ОД в практику государственной работы в России является одноименный проект «Открытые данные» («ОД»), осуществляемый под эгидой Открытого правительства. Как важный элемент этот проект встраивается в механизм обеспечения общего принципа ***открытости правительства*** и, шире, — всего государства.

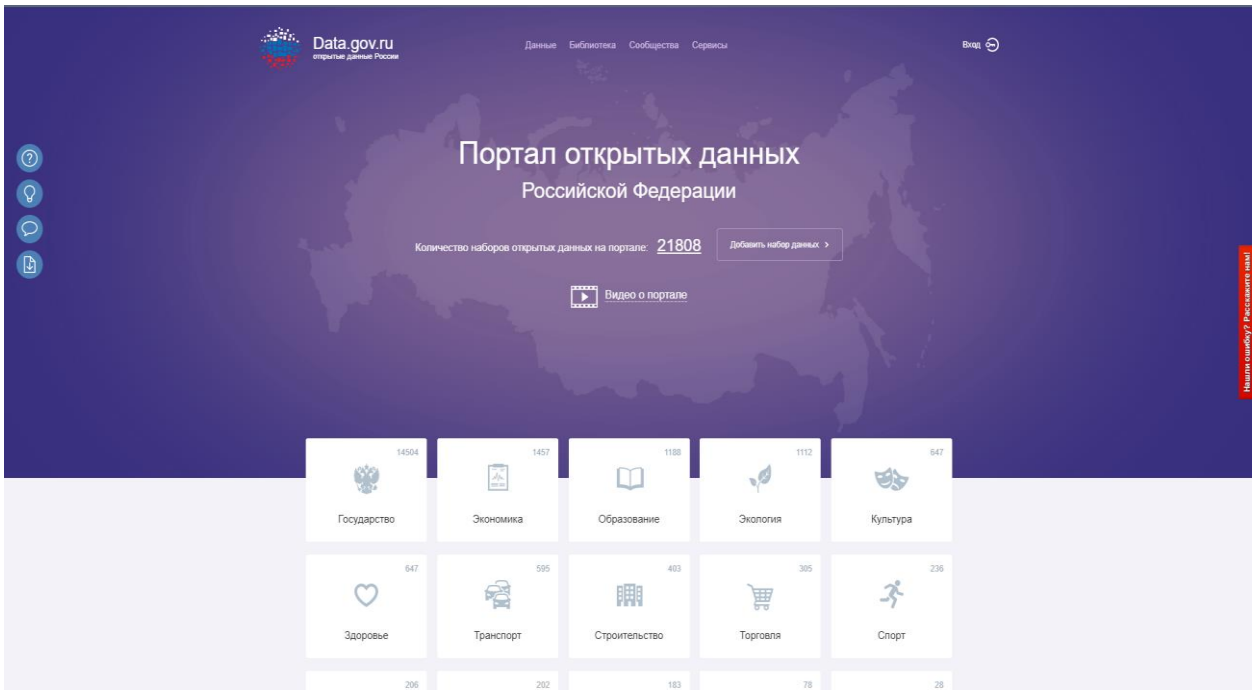


Рисунок 1. Проект "Открытые данные"

Целью проекта «ОД» является реализация экономического и социального потенциала открытых данных, находящихся в распоряжении государства, путем вовлечения их в оборот негосударственного сектора. По сути речь идет о содействии формированию ***информационного общества*** и ***экономики знаний***.

1.2 OPENDATA: зачем нужны открытые данные и что это такое

В последние годы словосочетание «открытые данные» звучит все чаще: открылся портал открытых данных Москвы, Евросоюз выделил рекордную сумму на финансирование развития концепции открытых данных, проводится Школа открытых данных и так далее. Посмотрим, что же это такое.

1.2.1 Понятие открытых данных

Концепция открытых данных базируется на представлении о том, что данные должны находиться в свободном доступе для использования и распространения без каких-либо ограничений.

Открытые данные обладают такими свойствами:

1. Доступны — без каких-либо ограничений, в удобной форме, предпочтительно через Интернет
2. Открыты для распространения — в том числе в комбинации с другими данными
3. Не содержат ограничений — могут быть использованы и преобразованы с любыми целями любым лицом или организацией. В частности, открытые данные не могут распространяться с пометками вроде «только для некоммерческого использования» или «только в ознакомительных целях».

Прежде всего, речь всегда идет о данных, связанные с управлением: о численности и структуре населения, о деятельности организаций, нормативных актах и так далее; на этой концепции основан и российский проект «Открытое правительство». А в общем случае это любые данные, скажем, база данных торговой компании о потреблении различных видов товаров, данные использования мобильной сети и т.п.

Но это общая идея. Практическое ее воплощение требует соблюдения определенных правил, иначе можно открыть данные, но так, что их будет прочитать не легче, чем шумерскую клинопись.

1.2.2 OpenData как технический формат

Казалось бы, выложи всю информацию в Интернет, в открытый доступ, и дело с концом. Но открытые данные — это не просто массивы информации, это еще и технический термин. Чтобы какая-то информация стала открытыми

данными, важно, чтобы она была выражена на некоем универсальном языке, считывать который может не человек, а машина, компьютер.

Эта важную характеристику открытых данных называют интероперабельностью. Открытые данные по своему языку и структуре должны быть организованы так, чтобы любой желающий мог их использовать. К примеру, в таком случае может быть написан простой алгоритм для сравнения плотности населения России и США, притом, что данные выложены на двух разных порталах. Алгоритм сможет считать и затем использовать эти данные без участия человека.

Таким образом, открытые данные — это характеристика того, как информация «упаковывается» и распространяется, а не информации как таковой. Создание и совершенствование такого единого языка и механизмов удобной работы с огромными массивами открытых данных — это как раз основная исследовательская проблема, которой занимаются специалисты.

В 2012 году по инициативе «создателя Интернета» Тима Бернерса-Ли был создан Институт открытых данных (Open Data Institute, ODI), который сейчас объединяет усилия и курирует работу в этой сфере. В России НП «Инфокультура» стало коммуникационным узлом этой мировой программы, по инициативе организации не так давно прошла, в частности, Школа открытых данных.

1.2.3 Успешные проекты

С помощью открытых данных уже удалось реализовать много интересных проектов. Вот лишь несколько примеров.

- «Once Upon a Crime». Ученые использовали открытые демографические данные и данные с мобильных телефонов с реальными данными о преступлениях в Лондоне. Алгоритм позволил с точностью в 68% предсказать, где именно в городе произойдет в какой-

то момент времени преступление. На основе этой информации можно эффективно распределить человеческие и технические ресурсы — которых всегда нехватает — и предотвратить преступления.

- Предсказание поведения бирж и оценки экономического самочувствия бизнеса и населения с помощью открытых данных Google. Ученые использовали данные по поисковым запросам сервиса Google Trends и индекс Доу-Джонса. Оказалось, что рост количества финансовых запросов («рынки», «портфолио», «долг», «экономика» и т.д.) связан с падением бирж. Уменьшение числа подобных финансовых запросов означало повышение оптимизма, что отражалось в росте рынков.

Рисунок 2. Индекс Доу-Джонса.

- Компания BillGuard анализирует открытые данные о судебных разбирательствах, связанных с различными случаями финансовых махинаций с кредитными и дебетовыми пластиковыми картами. Таким образом, компания создала сервис, позволяющий обезопасить себя от мошенничества в этой сфере.

Безусловно, самая широкая сфера практического повседневного применения баз открытых данных — это как раз создание приложений для конечных пользователей. На этой основе работают приложения, помогающие рассчитать маршрут на общественном транспорте, найти работу, выбрать

культурное мероприятие и многое другое. Все они эффективны при использовании массивов открытых данных, доступных в сети, то есть когда приложение «обучено» читать язык open data.

1.2.4 Возможные проблемы

Опасения, связанные с открытостью информации, вполне предсказуемые и иногда вполне обоснованы. В первую очередь, речь идет о нарушении конфиденциальности персональных данных. Хотя в самом определении открытых данных заложено, что это данные не должны относиться к индивидуумам, — если только они того не пожелают, — и персональные данные охраняются законами большинства стран, грань здесь очень тонкая. Ну, и практика показывает, что не стоит слишком полагаться на защиту от незаконного доступа к информации. Кстати, аргумент насчет защиты персональных данных работает и в обратном направлении. Историки в России хорошо знают, как много архивных дел стали недоступными в последнее десятилетие под этим предлогом.

Организации опасаются и того, что на их базах открытых данных будут наживаться недобросовестные посредники, передающие данные как некий «эксклюзив» конечному потребителю. Хотя это скорее должно привести к мысли, что пресловутый конечный потребитель мог бы получить эту информацию от владельца базы данных — минуя посредников. Либо потребителю придется заплатить за удобный для него формат представления данных.

2. Регулярные выражения

Регулярные выражения (англ. *regular expressions*, жарг. ***регэкспы*** или ***регексы***) — система обработки текста, основанная на специальной системе записи образцов для поиска. Образец (англ. *pattern*), задающий правило поиска, по-русски также иногда называют «шаблоном», «маской».

Сейчас регулярные выражения используются многими текстовыми редакторами и утилитами для поиска и изменения текста на основе выбранных правил. Многие языки программирования уже поддерживают регулярные выражения для работы со строками. Например, Perl и Tcl имеют встроенный в их синтаксис механизм обработки регулярных выражений. Набор утилит (включая редактор sed и фильтр grep), поставляемых в дистрибутивах Unix, одним из первых способствовал популяризации понятия регулярных выражений.

Регулярные выражения - это шаблоны, используемые для сопоставления последовательностей символов в строках.

Регулярные выражения представляют собой похожий, но гораздо более сильный инструмент для поиска строк, проверки их на соответствие какому-либо шаблону и другой подобной работы. Англоязычное название этого инструмента — *Regular Expressions* или просто *RegExp*. Строго говоря, регулярные выражения — специальный язык для описания шаблонов строк.

Реализация этого инструмента различается в разных языках программирования, хоть и не сильно.

Набор утилит (включая редактор sed и фильтр grep), поставляемых в дистрибутивах UNIX, одним из первых способствовал популяризации регулярных выражений для обработки текстов. Многие современные языки программирования имеют встроенную поддержку регулярных выражений. Среди них ActionScript, Perl, Java,PHP, JavaScript, языки платформы .NET Framework, Python, Tcl, Ruby, Lua, Gambas, C++ (стандарт 2011 года), Delphi, D, Haxe и другие.

Регулярные выражения используются некоторыми текстовыми редакторами и утилитами для поиска и подстановки текста. Например, при помощи регулярных выражений можно задать шаблоны, позволяющие:

- найти все последовательности символов *«кот»* в любом контексте, как то: *«кот», «котлета», «терракотовый»*;
- найти отдельно стоящее слово *«кот»* и заменить его на *«кошка»*;
- найти слово *«кот»*, которому предшествует слово *«персидский»* или *«чеширский»*;
- убрать из текста все предложения, в которых упоминается слово *кот* или *кошка*.

Регулярные выражения позволяют задавать и гораздо более сложные шаблоны поиска или замены.

2.1 Результат работы

Результатом работы с регулярным выражением может быть:

- проверка наличия искомого образца в заданном тексте;
- определение подстроки текста, которая сопоставляется образцу;
- определение групп символов, соответствующих отдельным частям образца.

Если регулярное выражение используется для замены текста, то результатом работы будет новая текстовая строка, представляющая из себя исходный текст, из которого удалены найденные подстроки (сопоставленные образцу), а вместо них подставлены строки замены (возможно, модифицированные запомненными при разборе группами символов из исходного текста). Частным случаем модификации текста является удаление всех вхождений найденного образца — для чего строка замены указывается пустой.

2.2 Использование простых шаблонов

Простые шаблоны используются для нахождения прямого соответствия в тексте. Например, шаблон /abc/ соответствует комбинации символов в строке только когда символы 'abc' встречаются вместе и в том же порядке. Такое сопоставление произойдет в строке "Hi, do you know your abc's?" и "The latest

airplane designs evolved from slabcraft." В обоих случаях сопоставление произойдет с подстрокой 'abc'. Сопоставление не произойдет в строке "Grab crab", потому что она не содержит подстроку 'abc'.

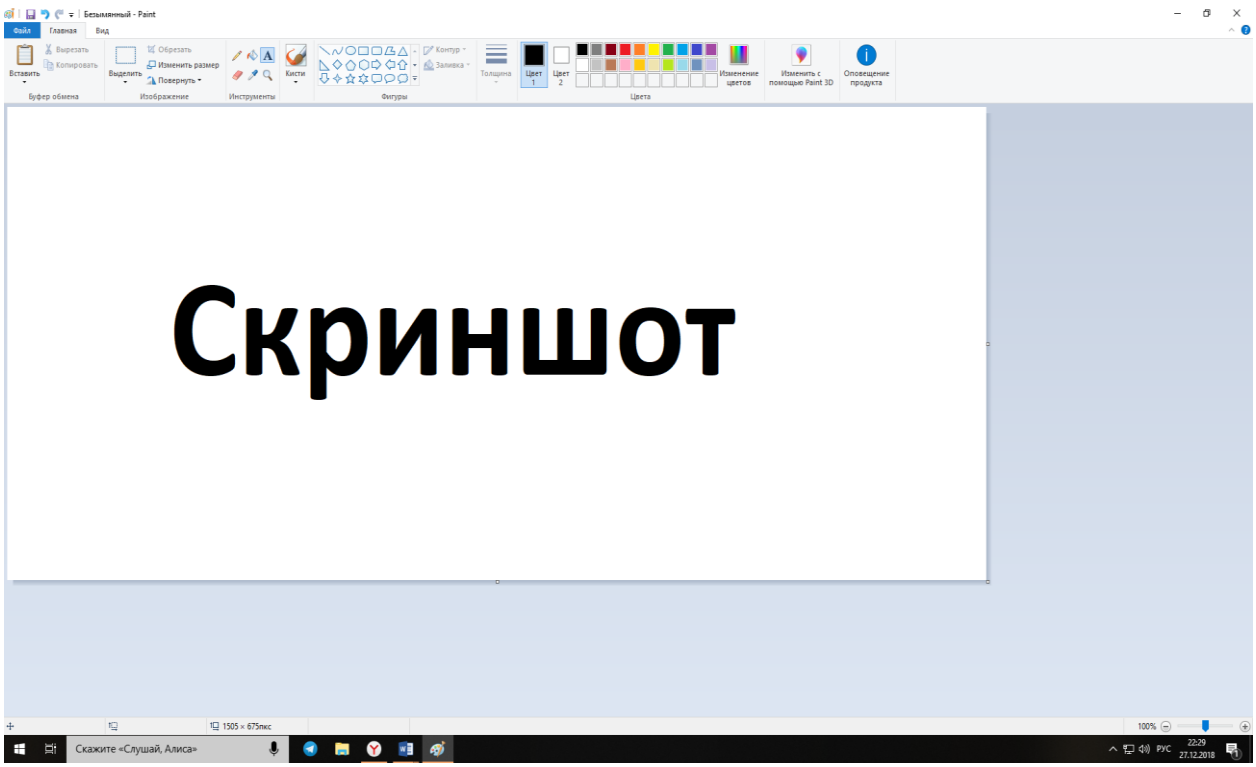
Вот полный список метасимволов:
. ^ $ * + ? { [ ] \ | ( )

Шаблон	Описание	Примеры
.	Один любой символ	1
\d	Любая цифра	7
\D	Любая **не**цифра	C
\s	Пробел, перенос строки, символ табуляции	' '
\S	Что угодно, кроме пробела, табуляции, переноса строки	f
[a-z]	Любая буква от a до z	z
[0-9]	Любая цифра от 0 до 9.	8
\w	Любая буква	c
\W	Любая **не**буква	_

Рисунок 3. Простые шаблоны

3.Снимок экрана

Снимок экрана (также **скрин**, **скриншот** (screenshot) с англ. — «снимок экрана») — изображение, полученное устройством и показывающее в точности то, что видит пользователь на экране монитора или другого визуального устройства вывода. Обычно снимок экрана создаётся по команде пользователя, с помощью встроенной функции операционной системы, или специальной программой. Намного реже снимки экрана получают с помощью внешнего устройства, такого, как фото-/видеокамера, или путём перехвата видеосигнала от компьютера к монитору.



3.1 Для чего его используют?

Скриншот — это высшая степень наглядности. Та самая картинка, которую лучше один раз увидеть, чем сто раз о ней услышать. И вот почему.

Техническая помощь

Случаи, когда при пользовании компьютером что-то пошло не так, у новичков нередки. Либо программа не работает, либо просто непонятно, куда нажимать дальше. Вместо того, чтобы долго и мучительно объяснять: “У меня там вверху открылось маленькое окошечко, а в нем написано что-то по-английски”, достаточно сделать снимок экрана и послать опытному товарищу или на тематический форум. Там, ориентируясь по увиденному, подскажут, что делать. Если бы не изобрели скриншотов, пришлось бы бежать за фотоаппаратом, фотографировать экран, выбирая ракурс без бликов, потом закачивать снимок в компьютер и отправлять по почте. Дело долгое, а результат не всегда стоящий, т. к. качество фотографии может быть низким.

Итак, основное назначение скриншота — визуальное объяснение каких-либо технических моментов. Если полистаете сайты с инструкциями, увидите, как часто его там используют.

Доказательства на все случаи жизни

Дальше, скриншот может быть использован для доказательств чего-либо. Например, делаете перевод на счет организации через Сбербанк Онлайн. Сам перевод может прийти через несколько дней, но, послав картинку с операцией, вы доказали, что оплату произвели. Некоторым фирмам этого достаточно для брони товара.

Копирайтера могут попросить прислать скриншот проверки текста на уникальность. Это оправдано в тех случаях, когда проверка осуществляется не онлайн и занимает продолжительное время.

Если на работе имеете дело с программой, в которую есть доступ нескольким сотрудникам, данные за свою смену сохраняйте как изображение, желательно с захватом нижней части экрана, где показана дата и время. Убережет от многих спорных ситуаций. Или от случайных потерь информации.

Больной вопрос — мошенничество в интернете. Заказывая или предоставляя услугу, мы рискуем не получить нужного результата. Скриншот переписки может быть представлен в качестве судебного доказательства. А вынесенный на публику, например, в сети ВКонтакте, он предостережет других от взаимодействия с обманщиком.

Путешественникам

Путешественники тоже найдут применение этому инструменту.
К примеру, вам надо узнать подробный маршрут. Открываете карты Яндекс или Гугл, отмечаете там все точки, через которые нужно проследовать, делаете снимок экрана и сохраняете. При желании его, как обычную

картинку, можно распечатать. С распечатанными картами удобно передвигаться по местности, где не работает связь, например, в горах.

Для хобби

Читаете книги онлайн? Не всегда есть возможность сохранить их в удобном варианте, но важные места можно сохранить при помощи скриншота. Особенно удобно, когда есть возможность одновременно выделить маркером нужные строки. С помощью каких программ удобнее всего сделать скриншот на компьютере, читайте в нашем руководстве.

Любители кинофильмов при помощи скриншота могут нарезать себе солидную коллекцию любимых кадров. Гораздо лучше сохранить полюбившиеся эпизоды во время просмотра, чем потом искать по запросу в Яндексе.

3.2 Получение снимка экрана

Простейший способ получения снимка экрана для операционных систем Microsoft Windows — использование клавиши PrtScr (для всего экрана) или сочетания клавиш Alt+PrtScr (для текущего окна) на клавиатуре. При этом снимок копируется в буфер обмена операционной системы и может быть затем вставлен и при необходимости отредактирован в любом графическом редакторе, например, в Paint, входящем в стандартный набор приложений Windows.

4.Вычитка отсканированных и распознанных документов

После обработки документа сканером получается графическое изображение документа (графический образ). Но графический образ еще не является текстовым документом. Человеку достаточно взглянуть на лист бумаги с текстом, чтобы понять, что на нем написано. С точки зрения компьютера, документ после сканирования превращается в набор разноцветных точек, а вовсе не в текстовый документ. Проблема распознавания текста в составе

точечного графического изображения является весьма сложной. Подобные задачи решают с помощью специальных программных средств, называемых средствами распознавания образов. Реальный технический прорыв в этой области произошел лишь в последние годы. До этого распознавание текста было возможно только путем сравнения обнаруженных конфигураций точек со стандартным образцом (эталоном, хранящимся в памяти компьютера). Авторы программ задавали критерий «похожести», используемый при идентификации символов.

Подобные системы назывались OCR (Optical Character Recognition — оптическое распознавание символов) и опирались на специально разработанные шрифты, облегчавшие такой подход. Естественно приходилось сталкиваться с произвольным и, тем более, сложным шрифтом, программы такого рода начинали давать серьезные сбои.

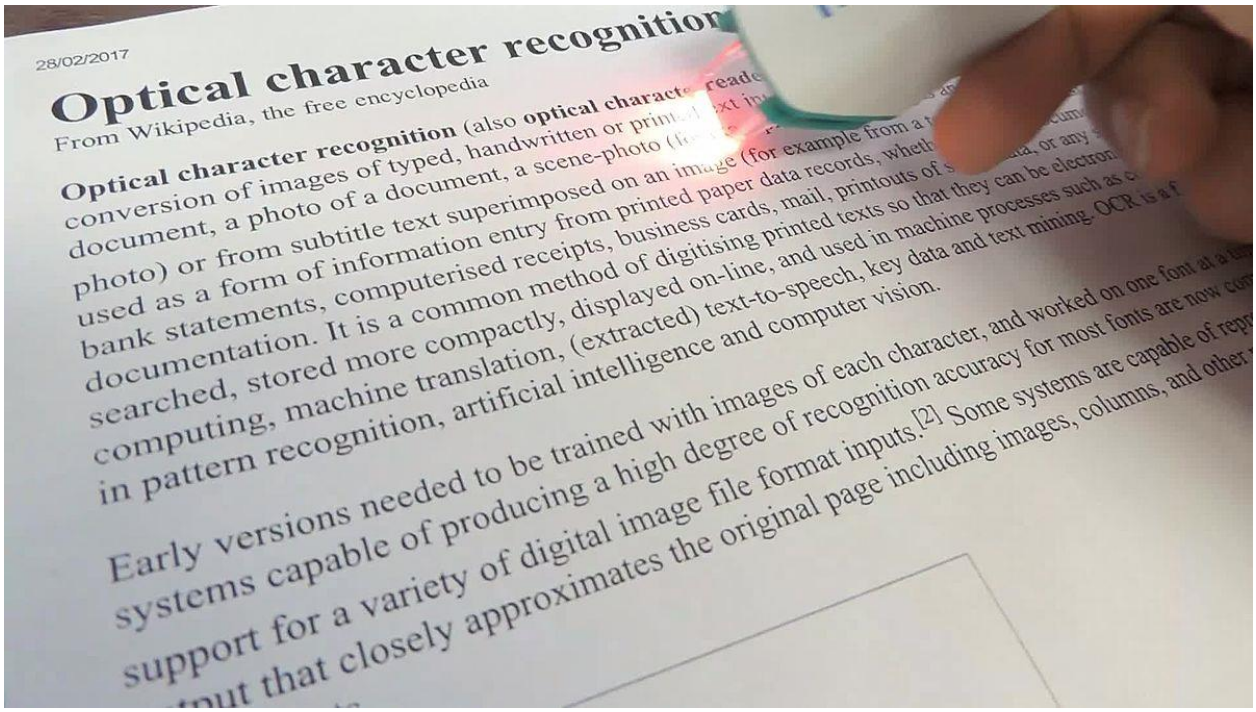


Рисунок 4. Optical Character Recognition

Современные научные достижения в области распознавания образов буквально перевернули представление об оптическом распознавании символов. Современные программы вполне могут справляться с различными

(и весьма вычурными) шрифтами без перенастройки. Многие распознают даже рукописный текст.

Поскольку потребность в распознавании текста отсканированных документов достаточно велика, неудивительно, что имеется значительное число программ, предназначенных для этой цели. Так как разные научные методы распознавания текста развивались независимо друг от друга, многие из этих программ используют совершенно разные алгоритмы.

Эти алгоритмы могут давать разные результаты на разных документах. Например, упоминавшиеся выше системы OCR способны распознавать только стандартный специально подготовленный шрифт и дают на этом шрифте наилучшие результаты, которые не может превзойти ни одна, из более универсальных программ.

Современные алгоритмы распознавания текста не ориентируются ни на конкретный шрифт, ни на конкретный алфавит. Большинство программ способно распознавать текст на нескольких языках. Одни и те же алгоритмы можно использовать для распознавания русского, латинского, арабского и других алфавитов и даже смешанных текстов. Разумеется, программа должна знать, о каком алфавите идет речь.

Нас, прежде всего, интересуют программы, способные распознавать текст, напечатанный на русском языке. Такие программы выпускаются отечественными производителями. Наиболее широко известна и распространена программа FineReader. Мы подробно остановимся именно на этой программе, обеспечивающей высокое качество распознавания и удобство применения.

4.1 Программа FineReader

Программа FineReader выпускается отечественной компанией ABBYY Software (http://bitsoft.su). Эта программа предназначена для распознавания текстов на русском, английском, немецком, украинском, французском и многих других языках, а также для распознавания смешанных текстов.

Программа имеет ряд удобных возможностей. Она позволяет объединять сканирование и распознавание в одну операцию, работать с пакетами документов (или с многостраничными документами) и с бланками. Программу можно обучать для повышения качества распознавания неудачно напечатанных текстов или сложных шрифтов. Она позволяет редактировать распознанный текст и проверять его орфографию.

4.2 Обработка отсканированных изображений

После сканирования необходимо просмотреть все страницы и убедиться, что нет явных огрехов. Например, иногда по недосмотру книга неровно легла на стекло сканера и часть текста на какой-либо странице не отсканировалась, или были вовсе пропущены некоторые страницы. После этого можно архивировать отсканированные изображения и приступать к обработке. Поскольку сканирование — физически самый трудоёмкий этап, рекомендуется держать резервную копию всех исходных сканов (такими, какими они были до обработки) на случай какого-либо сбоя.

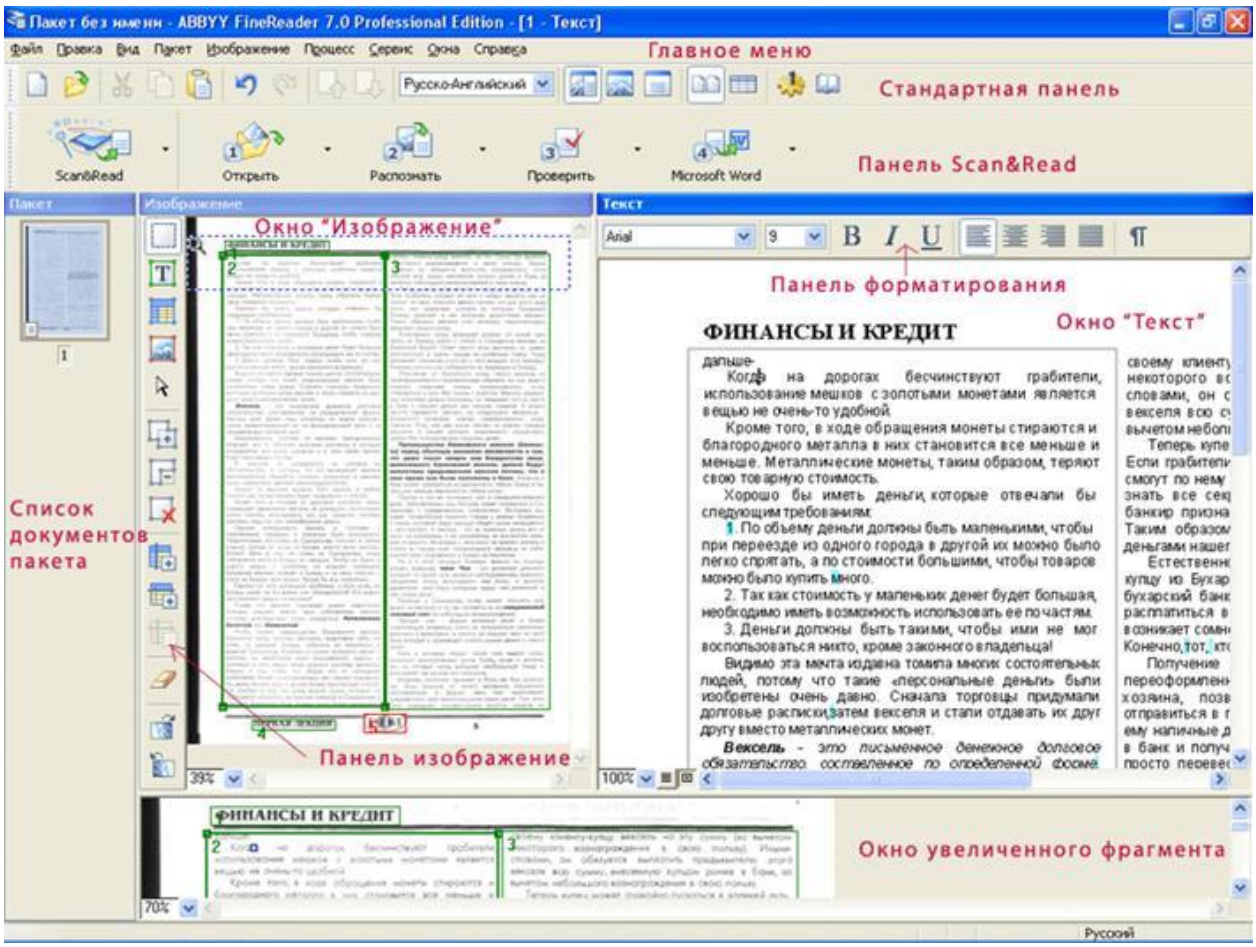


Рисунок 5. Работа в FineReader

Энтузиасты подготовили полные подробные инструкции по обработке отсканированных изображений и созданию электронных книг — смотрите ссылки внизу страницы. Каковы главные задачи обработки? Они зависят от того, ставим ли мы целью создание векторного файла или растрового файла. Для создания векторного файла производится распознавание (OCR) текста и его дальнейшее редактирование вручную в текстовом процессоре (таком, как MS Word или Adobe Pagemaker). Конечным продуктом обычно является сверстанная книга в формате PDF. Для создания растрового файла необходима доводка графических изображений до высокой степени сжатия и качества, а распознавание (OCR) производится лишь начерно, без вычитки и правки текста, в самом конце процесса. Обработка графических изображений производится обычно в пакетном режиме, так что не требуется обрабатывать каждую страницу вручную в Photoshop’е или другом графическом редакторе.

Поэтому затраты времени на создание растровой электронной книги гораздо меньше, чем на создание векторной книги.

Графическая обработка сканов состоит из следующих основных шагов:

- преобразование серых сканов в черно-белые (если исходные сканы были серыми в 300 dpi, то после этого получаются черно-белые в 600 dpi)
- разрезание разворотов на два изображения отдельных страниц (если книгу сканировали в развороте)
- поворот изображения каждой страницы, чтобы текст стал по возможности горизонтальным
- отрезание ненужных тёмных полос на краях, создание ровных и одинаковых для всех страниц белых полей
- вычищение «грязи» на страницах (включая помарки от руки, штапмы и прочее)

Эти шаги частично автоматизированы в программе ScanKromsator (Windows) и описаны в инструкции «Scan and Share» (смотрите ссылки внизу страницы). Однако если эта программа показалась для вас слишком сложной, вы можете воспользоваться Scan Tailor (ссылка на неё внизу страницы).

После создания окончательной чистовой версии книги делается распознавание текста (OCR). Распознавание текста на большинстве языков можно производить как коммерческой версией Djvu Document Editor (для DJVU), так и широко распространённой программой FineReader (для PDF). Имеется также бесплатный софт (утилита DjvuOCR) для вставки OCR-слоя в DJVU-файлы после распознавания в программе FineReader. По опыту, FineReader дает лучшее качество распознавания, чем Djvu Document Editor (который использует движок IRIS). Ознакомительные или демо-версии этих программ можно получить на официальных сайтах производителей.

Имеется также возможность автоматически добавить гипертекстовые ссылки в оглавление и индекс DJVU-книги. Это делает бесплатная утилита Djvu Hyperlink Editor.

Также в Djvu-книгу можно добавить оглавление в виде иерархического дерева с помощью бесплатной утилиты Djvu Bookmarker.

4.3 Форматы DJVU и PDF

Формат DJVU позволяет сжимать растровое изображение несколько лучше, чем PDF, просматривается несколько быстрее, а также более удобен в технической обработке. Например, есть простые и бесплатные программные средства для редактирования гиперлинков, закладок и OCR-слоя в DJVU, но таких средств нет для PDF. Также файлы DJVU более устойчивы к сбоям, чем PDF, и менее зависимы от версии просмотрщика, поскольку формат DJVU гораздо проще. Недостаток DJVU: возможность внести искажения при сильном сжатии и большое количество разных режимов сжатия приводят к тому, что сделать некачественный файл начинающему пользователю довольно легко. Также DJVU-файлы (по теперешнему стандарту) позволяют делать гиперлинки на другую страницу того же документа, но не на другой файл, не на сайт интернета, и не на выбранное место на данной странице (это можно делать в PDF). Однако формат DJVU несложен, документирован и содержит гибкий механизм добавления метаинформации: к каждой странице можно добавлять произвольную информацию в виде нескольких пар `key=value`. Поэтому в принципе можно сделать всё это и многое другое (например проверку md5sum или криптографическую подпись) средствами формата DJVU.

Главное достоинство формата PDF — широкая совместимость (у всех есть бесплатный Acrobat Reader или его аналоги) и тот факт, что большинство людей пока ничего не знают о формате DJVU. Однако, надо заметить, что программы для просмотра DJVU тоже бесплатные и требуют гораздо

меньших ресурсов компьютера, чем Acrobat Reader. Недостатки PDF в основном технические, но они существенны. Главный недостаток — невозможность определить разрешение растра, находящегося внутри PDF. Это приводит к сильным потерям в качестве изображения при попытках улучшить качество неоптимально сделанного растрового PDF-файла. Неоптимальные PDF-файлы могут иметь размеры 100—200 КБ на страницу и даже более. Оптимальный растровый PDF тратит от 10 до 20 КБ на страницу, что примерно на 30—50 % больше, чем DJVU.

4.4 Приспособления для сканирования

Получать изображение документа можно сканером или фотоаппаратом. Качественных различий между ними нет, но и у сканеров, и фотоаппаратов есть свои достоинства и недостатки.

Достоинства фотоаппаратов	**Их недостатки**
• Скорость сканирования — мгновения. • Можно снимать где угодно, хоть прямо в библиотеке. • Фотографировать можно не только изображение на бумаге.	• Низкое разрешение; необходимость почастной съёмки (и склейки частей) для получения высокого. • Может быть широкоугольное искажение (выпуклость или дисторсия) изображения, пагубность которого особо проявляется на иллюстрациях. • Сложно настраивать для достижения высокого качества.

Достоинства сканеров	Их недостатки
• Высочайшее качество цветопередачи. • Высокая разрешающая способность. • Полное отсутствие искажений в случае плоских и плотно прилагаемых документов	• Относительно низкая скорость. • Величина и громоздкость, за исключением ручных сканеров. • Большие различия моделей, приводящие к невозможности описания простого алгоритма настройки. • Ограниченность размера; иногда — невозможность качественно отсканировать большой лист по частям. • Часто — требование плотного прилегания разворота книги к сканеру, что её повреждает.

5.TEI

TEI (Text Encoding Initiative)- инициатива по кодированию текстов: разработана в Центре электронных текстов Вирджинии в 1989 г. как инструмент при процессе оцифровке, который идентифицирует электронный ресурс и его печатный источник посредством метаданных, размещаемых внутри самого электронного ресурса.

Инициатива кодирования текста представляет собой сообщество, занимающееся вопросами обработки текста в академической области цифровых гуманитарных наук. Формат используется многими проектами по всему миру.

Например:

British National Corpus — https://corpus.byu.edu/bnc/

The Oxford Text Archive — http://ota.ox.ac.uk

Perseus project - http://perseus.uchicago.edu

Epydoc — http://epydoc.sourceforge.net

The Sword Project — http://www.crosswire.org/sword/index.jsp и др.

5.1 Цели кодирования текста:

- ✓ Доступ и хранение
- ✓ Распространение

  -поиск и просмотр

  -взаимодействие и переносимость между различными источниками
- ✓ Анализ

  -лингвистический анализ

  -тематическое моделирование
- ✓ Визуализация

Рисунок 6. Пример кодирования текста 1

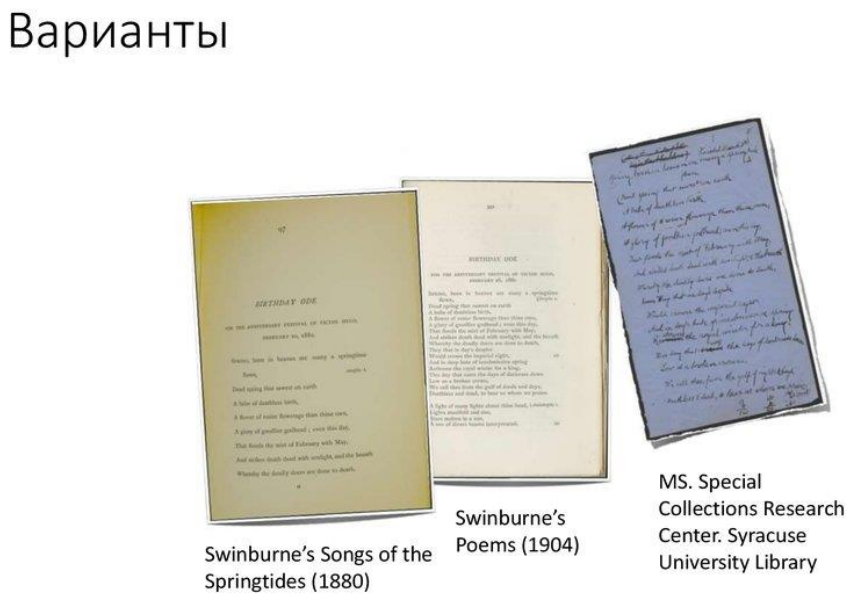


Рисунок 7. Пример кодирования текста 2

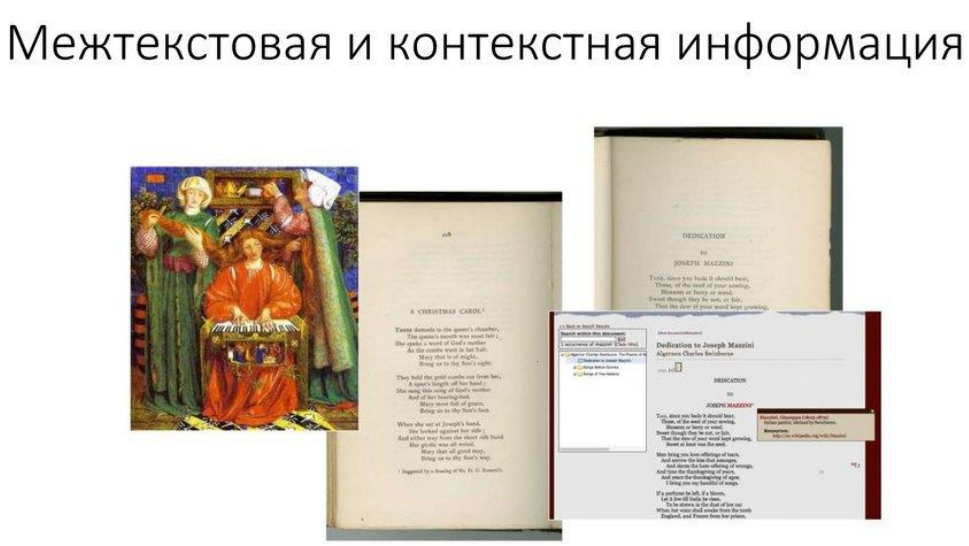


Рисунок 8. Пример кодирования текста 3

6.Формулярный анализ

Формулярный анализ — исторический источниковедческий метод анализа формуляра актового источника.

Под актом в узком смысле в источниковедении понимается договор между двумя контрагентами. Это могут быть международные договоры, (например, древнейшими русскими актовыми источниками являются Договоры Руси с

Византией), различные грамоты: о продаже (купчая грамота), об обмене (меновая грамота), о закладывании земли (закладная грамота), о пожаловании, например, князем монастыря (жалованная грамота) и т. д. Как правило, акты одного типа имеют устоявшуюся модель построения, **формуляр**. Формулярный анализ основан на сравнении формуляров конкретных актовых документов, в частности для их более точного датирования (формуляр изменялся во времени) и классификации по группам. **Условный формуляр**, то есть идеальная модель средневекового акта, делится на следующие части:

Начальный протокол

- Инвокация — посвящение высшим силам или государю.
- Интитуляция — указание, от кого исходит документ.
- Инскрипция — указание, кому документ адресован.
- Салютация — приветствие.

Основная часть

- Аренга — преамбула, мотивы создания документа.
- Промульгация — предуведомление о сути документа.
- Наррация — суть дела.
- Диспозиция — предложения и вопросы.
- Санкция — запрет на нарушение условий.
- Корроборация — удостоверение документа, подпись и печать.

Конечный протокол (эсхатокол)

- Датум — дата и место написания.
- Аппрекация — заключение-благопожелание.

В конкретном акте могут присутствовать не все перечисленные части, кроме того, они могут следовать в другом порядке. Кроме того, при формулярном анализе текст акта может делится на «статьи» или «клаузулы» (понятия не совсем идентичны), законченные по мысли выражения: сакральные,

мотивировочные, содержащие обращение, уведомительные, процессуальные, удостоверительные, описательные, указные, просительные, договорные. Статьи делятся на обороты, обороты на элементы. Кроме того, в актах выделяют «формулы» — устойчивые выражения, «реалии» — имена, топонимы и т. д. и «описания» — оригинальные выражения, не являющиеся штампами.

7.Сравнение двух вариантов одного текста: диффы

В вычислительной технике **diff** — утилита сравнения файлов, выводящая разницу между двумя файлами. Эта программа выводит построчно изменения, сделанные в файле (для текстовых файлов). Современные реализации поддерживают также двоичные файлы. Вывод утилиты называется «diff», или, что более распространено, **патч**, так как он может быть применён с программой patch. Вывод других утилит сравнения файлов также часто называется «diff»

7.1 История

Утилита diff была разработана в начале 1970-х годов для операционной системы Unix, которая была плодом работы AT&T Bell Labs, в Мюррей Хилл (Нью-Джерси). Финальная версия, распространяемая с 5-й версией Unix в 1974, была полностью написана Дугласом Макилроем.

7.2 Использование

diff вызывается из командной строки с именами двух файлов в качестве аргументов: diff *original new*. Вывод команды представляет собой изменения, которые нужно произвести в исходном файле original, чтобы получить новый файл new. Если *original*и *new* — каталоги, то diff автоматически будет применён к каждому файлу, который существует в обоих каталогах. Все примеры в этой статье используют следующие два файла, *original* и *new*:

original:

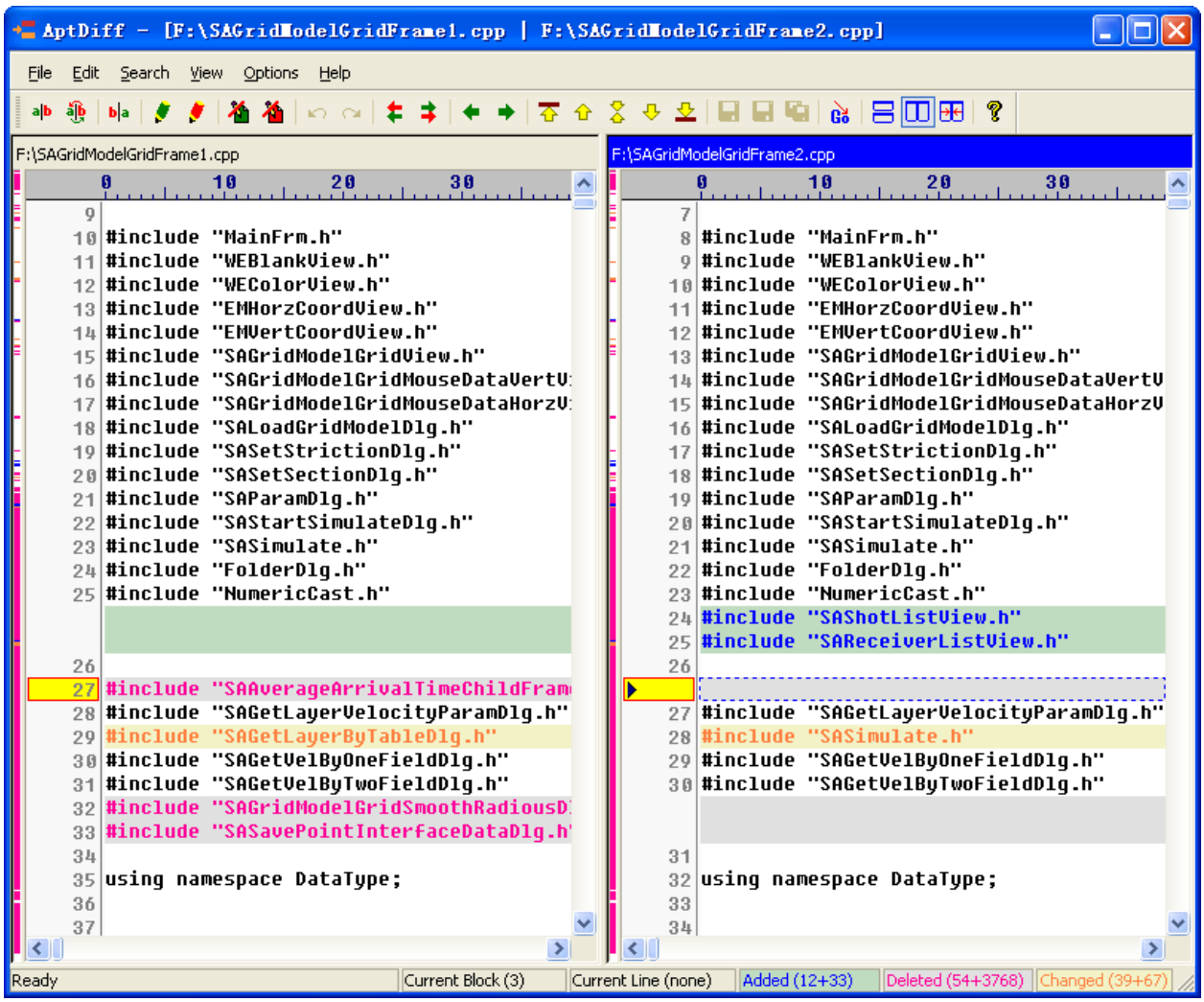
```
1 Эта часть документа
2 оставалась неизменной
3 от версии к версии. Если
4 в ней нет изменений, она
5 не должна отображаться.
6 Иначе это не
способствует
7 выводу оптимального
8 объёма произведённых
9 изменений.
10
11 Этот абзац содержит
12 устаревший текст.
13 Он будет удалён
14 в ближайшем будущем.
15
16 В этом документе
17 необходима провести
18 проверку правописания.
19 С другой стороны,
ошибка
20 в слове - не конец света.
21 Остальная часть абзаца
22 не требует изменений.
23 Новый текст можно
24 добавлять в конец
документа.
```

new:

```
1 Это важное
замечание!
 2 Поэтому оно должно
 3 быть расположено
 4 в начале этого
 5 документа!
 6
 7 Эта часть документа
 8 оставалась
неизменной
 9 от версии к версии.
Если
10 в ней нет изменений,
она
11 не должна
отображаться.
12 Иначе это не
способствует
13 выводу
оптимального
14 объёма информации.
15
16 В этом документе
17 необходимо
провести
18 проверку
правописания.
19 С другой стороны,
ошибка
20 в слове - не конец
света.
21 Остальная часть
абзаца
22 не требует
изменений.
23 Новый текст можно
24 добавлять в конец
документа.
25
26 Этот абзац содержит
27 важные дополнения
28 для данного
документа.
```

Команда diff original new производит следующий *нормальный дифф-вывод*:

```
0a1,6
> Это важное замечание!
> Поэтому оно должно
> быть расположено
> в начале этого
> документа!
>
8,14c14
< объёма произведённых
< изменений.
<
< Этот абзац содержит
< устаревший текст.
< Он будет удалён
< в ближайшем будущем.
---
> объёма информации.
17c17
< необходима провести
---
> необходимо провести
24a25,28
>
> Этот абзац содержит
> важные дополнения
> для данного документа.
```



В этом традиционном формате вывода **a** означает *добавлено* (от англ. *add*), **d** — *удалено*, **c** — *изменено*. Перед буквами a, d или c стоят номера строк исходного файла, после них — номера строк конечного файла. Каждая строка, которая была добавлена, удалена или изменена, предваряется угловыми скобками.

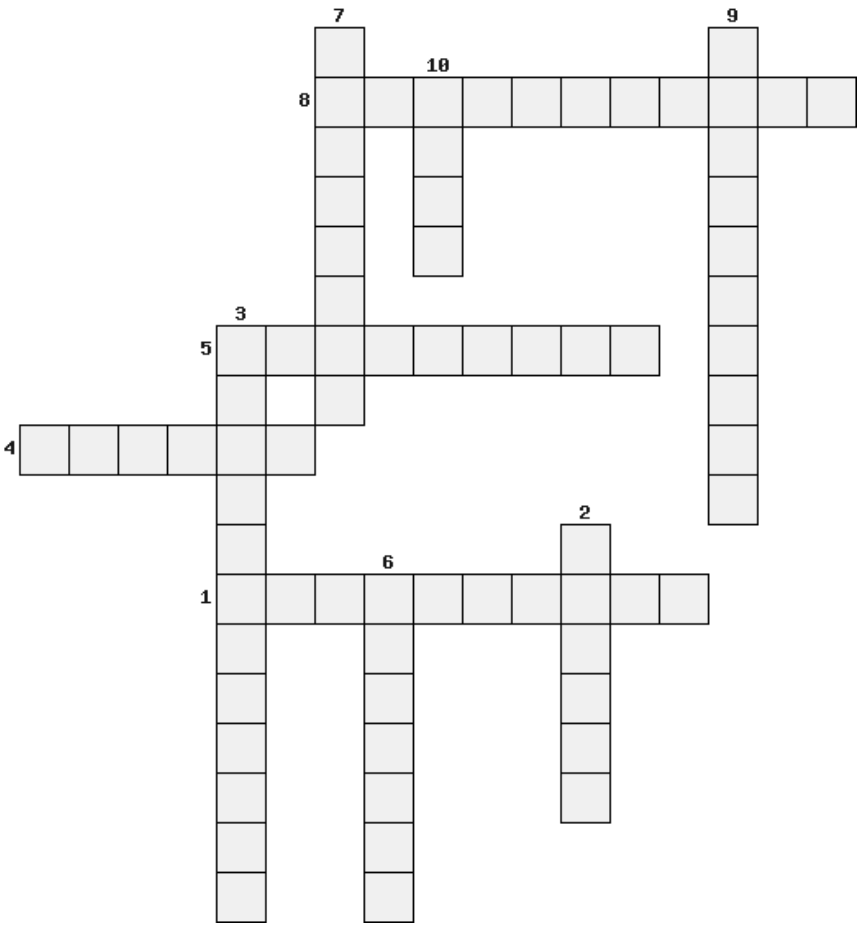
По умолчанию, общие для исходного и конечного файлов номера строк не указываются. Строки, которые перемещены, показываются как добавленные на своём новом месте и удалённые из своего прошлого расположения.

Закрепление материала

Задание №1

Вопросы:

1. Как называются выражения-шаблоны, образцы, задающие правило поиска?
2. Одна из основных целей кодирования текста. Бывает лингвистический, формулярный.
3. Благодаря этому процессу происходит считывание информации с бумажного носителя на компьютер.
4. Что такое PDF, DJVU?
5. Одна из главных миссий OpenData - ... доступ к любым файлам в сети.
6. Что из себя представляет diff?
7. Как называется снимок экрана?
8. Процесс представления информации в виде кода.
9. Самый важный интернет-ресурс в мире.
10. Утилита сравнения файлов, выводящая разницу между двумя файлами.



Ответы: 1 регулярные; 2 анализ; 3 сканирование; 4 формат; 5 свободный; 6 утилита; 7 скриншот; 8 кодирование; 9 информация; 10 дифф.

Задание №2

Назовите основные цели кодирования текста.

Ответ: Доступ и хранение, Распространение, Анализ, Визуализация

Задание №3

Какие на ваш взгляд плюсы и минусы имеет проект OpenData? Что стоит учесть или изменить?

Список используемых источников

1. Справочник и ресурсы по регулярным выражениям https://www.regular-expressions.info
2. Открытые данные России https://data.gov.ru
3. Материалы сайта Wikipedia https://ru.wikipedia.org
4. Интерактивный график индекса Доу-Джонса https://ru.investing.com/indices/us-30-advanced-chart
5. Пособие для чайников, как делать скриншот https://iklife.ru/dlya-novichka/chto-takoe-skrinshot.html
6. Материалы сайта FB.ru http://fb.ru