

ИЗУЧЕНИЕ TEX

Предисловие. Замечательный программист и математик Дональд Кнут опередил свое время. Он разработал векторные шрифты (в формате METAFONT) за несколько лет до появления получивших всеобщее признание шрифтов в формате Postscript — но именно постскриптовские шрифты являются сейчас мейнстримом. Он разработал формат файлов (dvi — от DeVice Independent, независимый от устройства), предназначенный для представления сложных текстов в таком виде, чтоб их можно было просмотреть или напечатать под любой операционной системой, на любом компьютере или принтере, задолго до появления формата pdf — однако сейчас для этих целей используется именно pdf. А вот созданная им система компьютерной верстки TEX (произносится «тех»), для нужд которой Кнут разрабатывал эти шрифты и этот формат файлов, и по сей день жива, популярна, хотя по меркам компьютерного мира она очень стара: ее первая версия появилась еще в 1978 году, в 1989 году она приобрела современные очертания, и с тех пор мало изменилась. Издательские системы на ее базе по сию пору широко используются и сдавать позиции не собираются. Чем объясняется столь редкое в компьютерном мире долголетие?

1) Никакая другая из существующих в настоящее время издательских систем не может сравниться с TEX'ом в полиграфическом качестве текстов с математическими формулами.

2) Система TEX реализована на всех современных компьютерных платформах, и все эти реализации действительно работают одинаково.

3) Благодаря этому TEX стал международным языком для обмена математическими и физическими статьями: набрав свою статью в TEX'е, математик может послать ее по электронной почте своему коллеге, даже если отправитель работает под Windows, а получатель — с UNIX'ом или, допустим, на Макинтоше.

4) В Интернете существуют обширные архивы препринтов, в которые каждый может послать (и из которых каждый может получить) статью; все эти статьи набраны опять-таки в TEX'е.

5) Наконец, основные реализации TEX'а для всех платформ распространяются бесплатно.

Разумеется, у TEX'а есть и недостатки. Главный из них — в том, что с помощью TEX'а тяжело (хотя в принципе и возможно) готовить тексты со сложным расположением материала на странице (наподобие рекламных буклетов). Для таких приложений, практически не встречающихся в наудотехнической литературе, TEX не предназначен.

Как работать с текстом. Сам по себе TEX представляет собой специализированный язык программирования (Кнут не только придумал язык, но и написал для него транслятор, причем таким образом, что он работает совершенно одинаково на самых разных компьютерах), на котором пишутся издательские системы, используемые на практике. Точнее говоря, каждая издательская система на базе TEX'a представляет собой пакет макроопределений (макропакет) этого языка. В частности, LATEX (по-русски произносится «латех», пишется также «LaTeX») — это созданная Лесли Лэмпортом (Leslie Lamport) издательская система на базе TEX'a. Далее, если не оговорено противного, будут употребляться слова TEX и LATEX вперемешку. При первом чтении можно воспринимать их как синонимы (на самом деле мы пишем TEX, когда речь идет об общих свойствах систем на базе TEX'a, а не о специфике именно LATEX'a).

Для начала автор должен подготовить с помощью любого текстового редактора файл с текстом, оснащенным командами для TEX'a. Такие файлы по традиции имеют расширение `tex`. Дальнейшая работа протекает в два этапа. Сначала надо обработать файл с помощью программы-транслятора. В современных версиях TEX'a в результате трансляции получается `pdf`-файл. Кроме того, предусмотрена возможность получить файл с расширением `dvi` (`device independent` — не зависящий от устройства). Теперь полученный файл (его называют еще `pdf`-файлом) можно просматривать, печатать, вести поиск и т.п. Если же получился `dvi`-файл, то его необходимо обрабатывать с помощью программ, называемых `dvi`-драйверами (и входящих в поставку TEX'a): распечатывать на принтере, смотреть на экране (в таком же виде, как он появится на печати) и т.д. (для разных устройств есть разные драйверы). Неудовлетворенный результатом, автор вносит изменения в исходный файл — и цикл повторяется. В более старых версиях TEX'a генерация `pdf`-файлов не предусмотрена, но возможность создать `dvi`-файл была в TEX'e всегда.

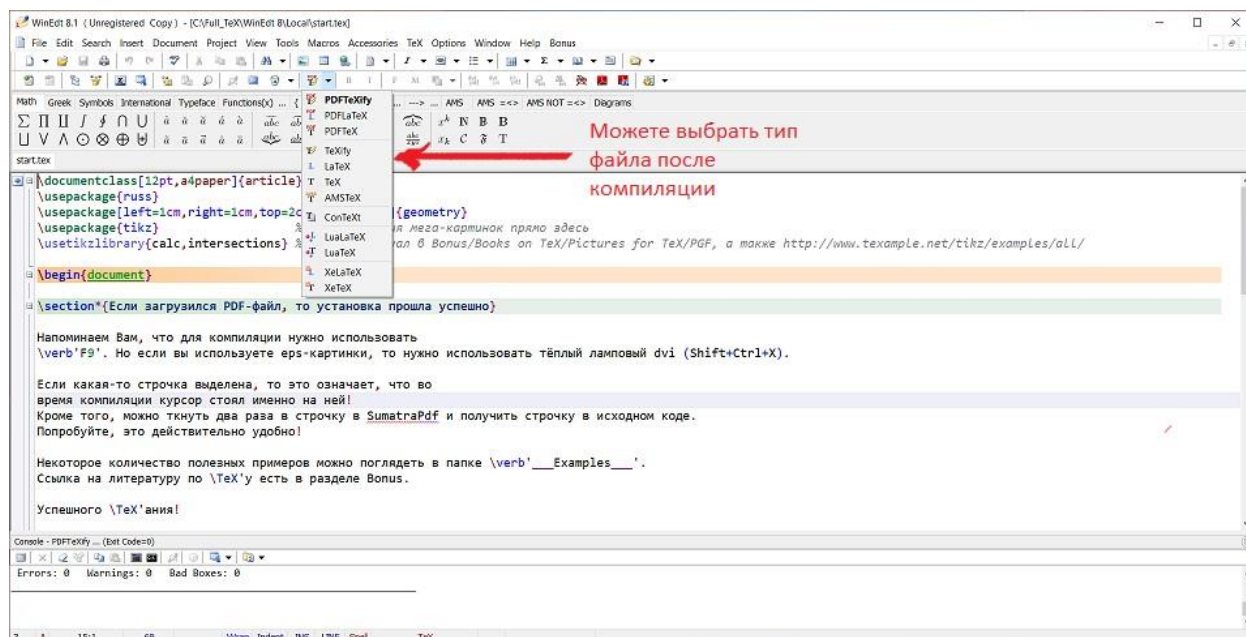


Рис. 4

Структура исходного текста

TEX-файл должен начинаться с команды `\documentclass` задающей стиль оформления документа (рис. 5)

`\documentclass{book}`

Слово **book** в фигурных скобках указывает, что документ будет оформлен, как книга: все главы будут начинаться с нечетных страниц, текст будет снабжен колонтитулами некоторого определенного вида и т. п. Кроме класса **book**, в стандартный комплект TEX'a входят классы **article** (для оформления статей), **report** (нечто среднее между **article** и **book**), **proc** (для оформления изданий типа «труды конференции») и **letter** (для оформления деловых писем так, как это принято в США). Чтобы задать оформление документа с помощью одного из этих классов, надо в фигурных скобках после команды `\documentclass` указать вместо **book** название требуемого класса. Стандартные классы можно (а иногда и нужно) менять, можно создавать и новые классы.

После команды `\documentclass` могут следовать команды, относящиеся ко всему документу и устанавливающие различные параметры оформления текста, например, величину абзацного отступа (вообще-то все эти параметры определяются используемым классом, но может случиться, что вам понадобится сделать в них изменения). Далее должна идти команда `\begin{document}`. Только после этой команды может идти собственно текст.

Часть файла, расположенная между командами `\documentclass` и `\begin{document}`, называется преамбулой. Заканчиваться файл должен командой `\end{document}`. Если после `\end{document}` в файле написано что-то еще, TEX это проигнорирует (рис. 5 и рис. 6).

Пример: показывает TEX-файл, составленный по всем правилам. Ничего интересного в результате его обработки не напечатается, но уж зато и сообщений об ошибках вы не получите.

```
\documentclass{article}
```

```
\begin{document}
```

Проба пера.

```
\end{document}
```

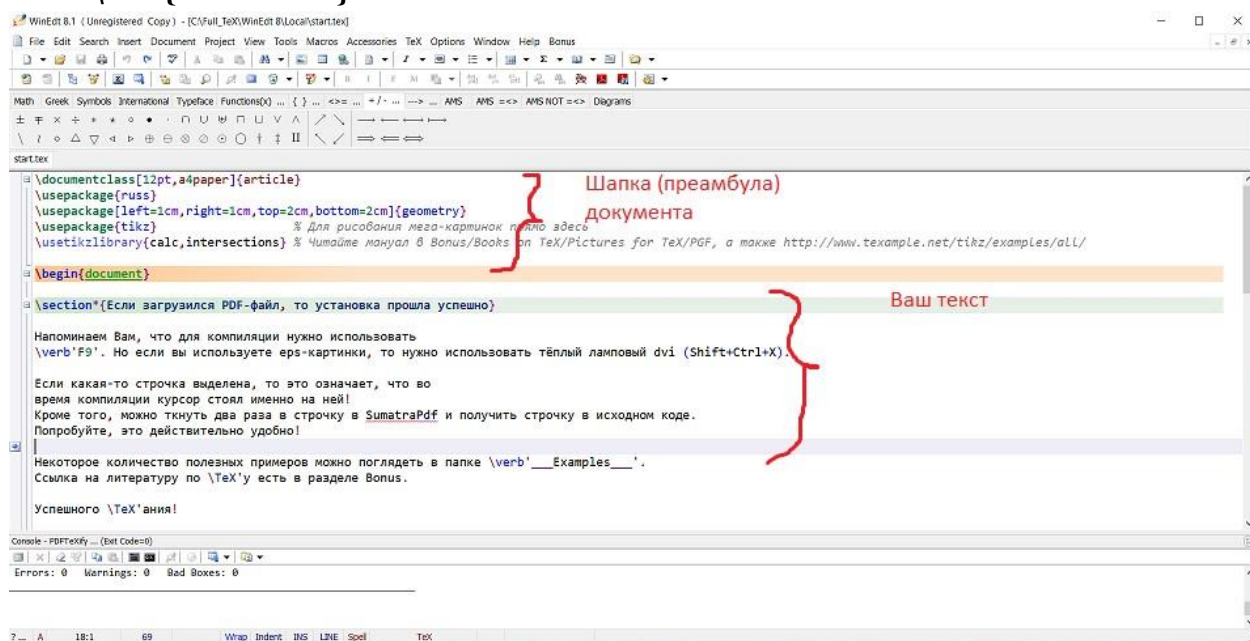


Рис. 5

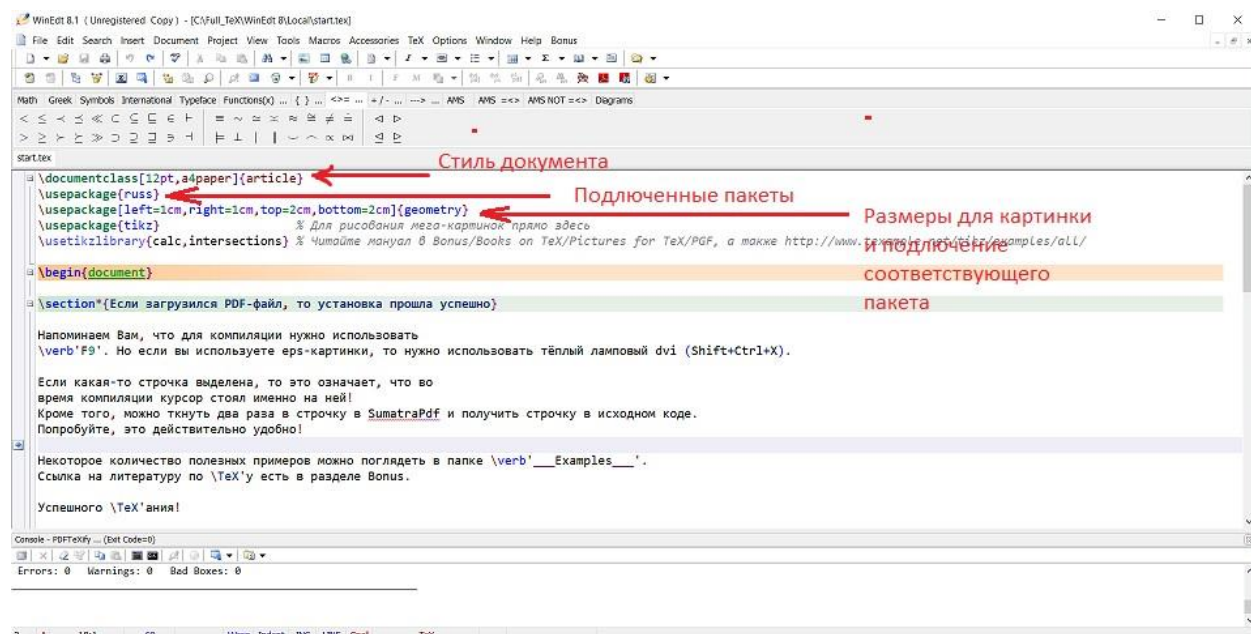


Рис. 6

Спецсимволы. Большинство символов в исходном тексте прямо обозначает то, что будет напечатано (если в исходном тексте стоит запятая, то и на печати выйдет запятая). Следующие 10 символов: { } \$ & # % _ ^ ~ \ имеют особый статус; если вы употребите их в тексте *просто так*, то скорее всего получите сообщение об ошибке (и на печати не увидите того, что хотелось). Печатное изображение знаков, соответствующих первым семи из них, можно получить, если в исходном тексте поставить перед соответствующим символом без пробела знак \ (по-английски он называется .backslash.).

ПРИМЕР: Курс тугрика повысился на 7\%, и теперь за него дают \\$200. (набрано в *Texe*) Курс тугрика повысился на 7%, и теперь за него дают \$200.
(получилось)

Если символ % употреблен в тексте не в составе комбинации \%, то он является символом комментария: все символы, расположенные в строке после него (и сам %), TEX игнорирует. С помощью символа % в исходный текст можно вносить пометки для себя.

О смысле остальных спецсимволов. Фигурные скобки ограничивают группы в исходном файле. Знак доллара ограничивает математические формулы. При наборе математических же формул используются знаки _ и ^ («знак подчеркивания» и «крышка»). Знак ~ обозначает «неразрывный пробел» между словами. Со знака \ начинаются все TEX'овские команды.

Знаки # и & используются в более сложных конструкциях TEX'а, о которых сейчас говорить преждевременно.

Отметим еще, что символы $< > |$ в тексте употреблять можно в том смысле, что сообщения об ошибке это не вызовет, но напечатается при этом нечто, совсем на эти символы не похожее. Подлинное место для этих символов, так же как и для символов = и +, — математические формулы, о которых речь пойдет позже.

Группы. Важным понятием TEX'а является понятие группы. Чтобы понять, что это такое, рассмотрим пример. При обработке TEX'ом исходного файла набор текста в каждый момент идет каким-то вполне определенным шрифтом (он называется текущим шрифтом). Изначально текущим шрифтом является, обычный прямой шрифт. Если нужно слово или несколько слов написать полужирным шрифтом (см. пример), то заключаем в фигурные скобки это слово вместе с соответствующей командой: `{\bf это}` — это и есть группа.

ПРИМЕР: Полужирным шрифтом набрано только `{\bf это}` слово; после скобок все идет, как прежде. *(набрано в Texe)*

Полужирным шрифтом набрано только **это** слово; после скобок все идет, как прежде. *(Получилось)*

Команды и их задание в тексте (см. рис. 7, 8, 9, 10). Задание печатного знака процента с помощью последовательности символов `\%` — пример важнейшего понятия TEX'а, называемого командой. С точки зрения их записи в исходном тексте, команды делятся на два типа.

Первый тип — команды, состоящие из знака `\` и одного символа после него, не являющегося буквой. Именно к этому типу относятся команды `\{, \}, \dots, \%`.

Команды **второго типа** состоят из `\` и последовательности букв, называемой именем команды (имя может состоять и из одной буквы). Например, команды `\TeX, \LaTeX` генерируют эмблемы систем TEX, LATEX и LATEX2". В имени команды, а также между `\` и именем, не должно быть пробелов; имя команды нельзя разрывать при переносе на другую строку.

В именах команд прописные и строчные буквы различаются. Например, `\large, \Large` и `\LARGE` — это три разные команды (они задают различные размеры шрифта).

После команды первого типа (из `\` и не-буквы) пробел в исходном тексте ставится или не ставится в зависимости от того, что вы хотите получить на

печати. После команды из \ и букв в исходном тексте обязательно должен стоять либо пробел, либо символ, не являющийся буквой (это необходимо, чтобы TEX смог определить, где кончается имя команды и начинается дальнейший текст).

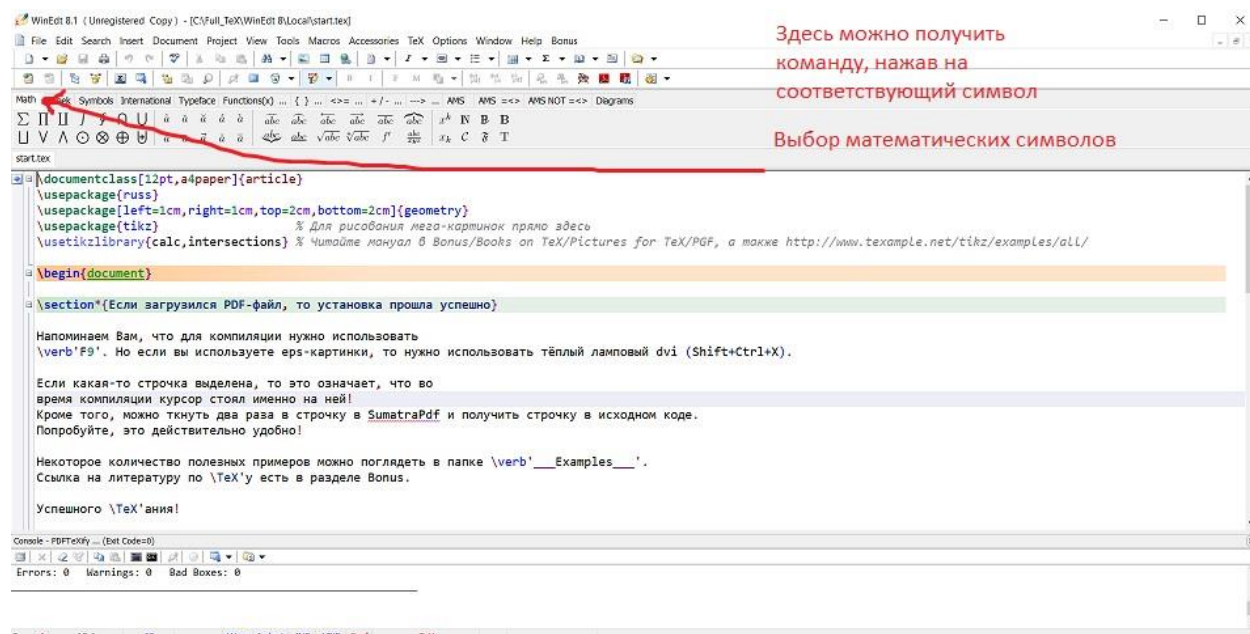


Рис. 7

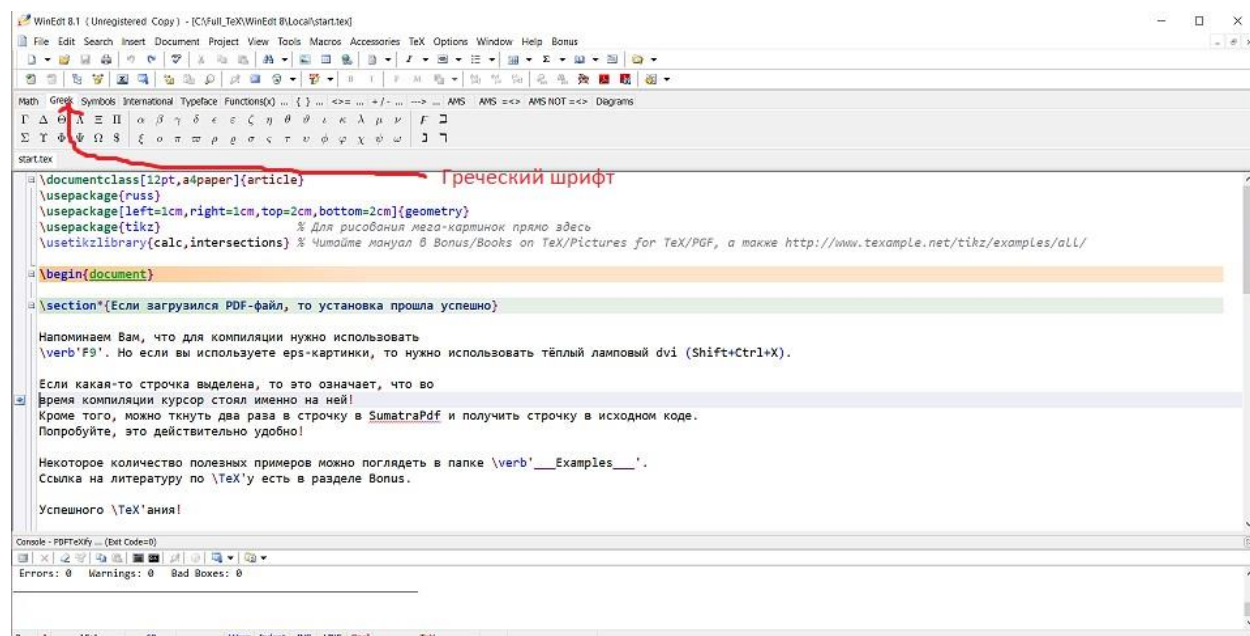


Рис. 8

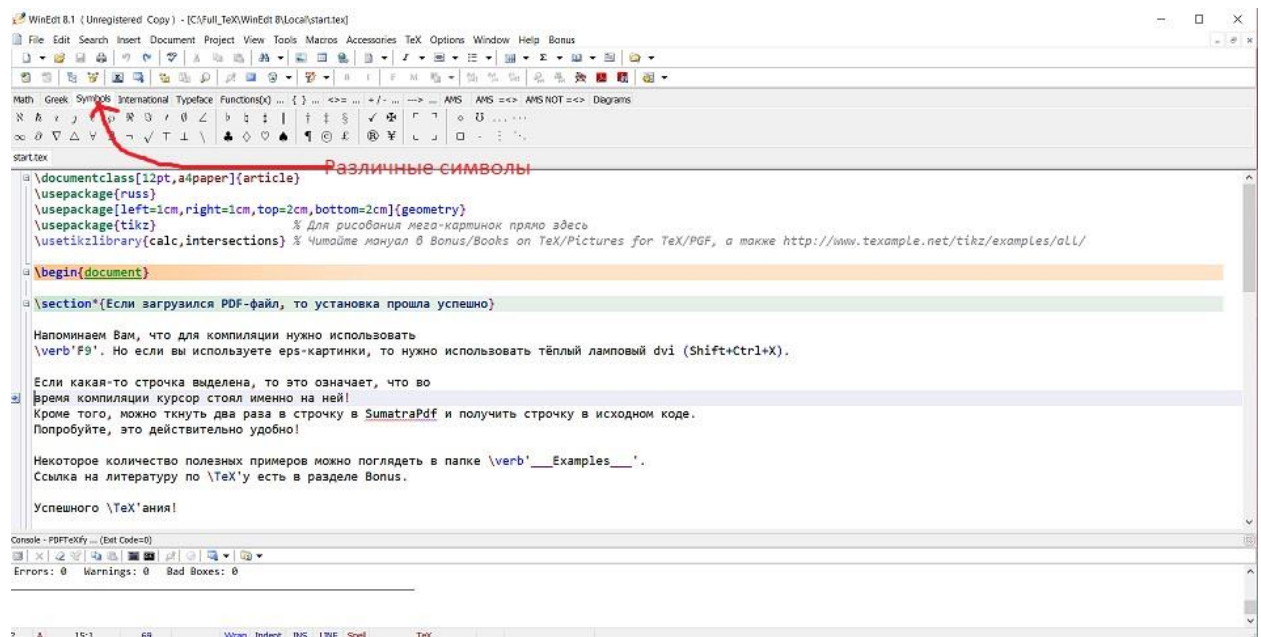


Рис. 9

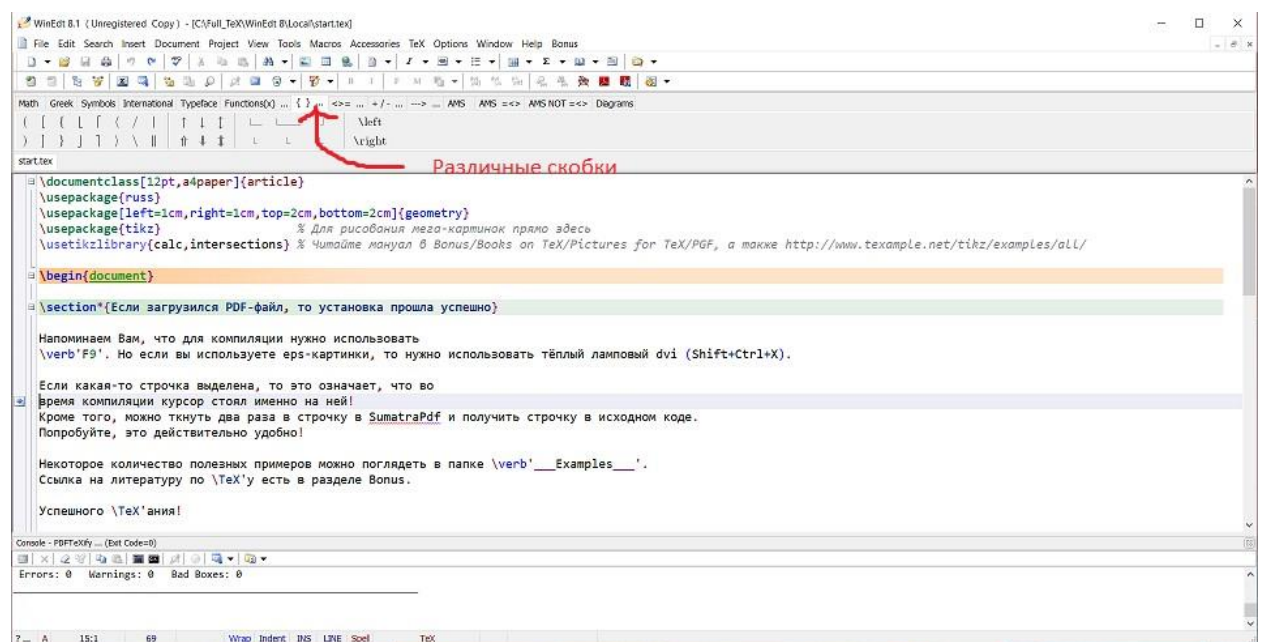


Рис. 10

ПРИМЕР: показывает еще один TEX-файл с подключенными пакетами.

```

\documentclass[12pt]{article}
\usepackage{cp1251}{inputenc} % следующие две строки используются для
\usepackage[russian]{babel} % руссификации AmSLaTeX
\usepackage{graphicx}
\pagestyle{plain}
\usepackage{amsmath,amsfonts,amssymb,euscript}
\begin{document} Текст. Формулы.
\end{document}

```