

Оформление программного кода средствами \LaTeX при помощи пакета **listings**

Остапчук Анастасия Витальевна

Волжский филиал

ФГАОУ ВО “Волгоградский государственный университет”

презентация по дисциплине “Издательские системы”

Выбор пакета для оформления программного кода

Выбор пакета для оформления программного кода

Существует множество пакетов \LaTeX , позволяющих вставлять в документ программный код на разных языках программирования. Данные пакеты поддерживают форматирование кода, подсветку синтаксиса и множество дополнительного функционала.

Выбор пакета для оформления программного кода

Среди наиболее популярных, можно выделить следующие пакеты:

- **verbatim** — вставка исходного кода в документ, без подсветки синтаксиса;
- **algorithm2e** — для оформления псевдокодов и алгоритмов;
- **mcode** — для подсветки синтаксиса и вставки программного кода *MATLAB*;
- **listings**;
- **minted**.

Пакет listings

Пакет listings

Используя пакет **listings** в \LaTeX , Вы можете добавлять неформатированный текст, как если бы Вы делали это с `\begin{verbatim}`, но его основная цель — включить исходный код любого языка программирования в ваш документ. Он умеет подсвечивать код, считать строки и многое другое.

Поддерживаемые языки программирования

- | | | | | |
|--------------|----------------|-------------|---------------------|------------|
| 1 ABAP | 17 Lisp | 33 Ruby | 49 Delphi | 65 Octave |
| 2 IDL | 18 Python | 34 Caml | 50 Modula-2 | 66 Verilog |
| 3 PL/I | 19 Awk | 35 MetaPost | 51 Simula | 67 GCL |
| 4 ACSL | 20 Logo | 36 S | 52 Eiffel | 68 Oz |
| 5 inform | 21 R | 37 Clean | 53 MuPAD | 69 VHDL |
| 6 Plasm | 22 bash | 38 Miranda | 54 SQL | 70 Gnuplot |
| 7 Ada | 23 make | 39 SAS | 55 Elan | 71 Pascal |
| 8 Java | 24 Reduce | 40 Cobol | 56 NASTRAN | 72 VRML |
| 9 POV | 25 Basic | 41 Mizar | 57 tcl | 73 Haskell |
| 10 Algol | 26 Mathematica | 42 Scilab | 58 erlang | 74 Perl |
| 11 JVMIS | 27 Rexx | 43 Comal | 59 Oberon-2 | 75 XML |
| 12 Prolog | 28 C | 44 ML | 60 T _E X | 76 HTML |
| 13 Ant | 29 Matlab | 45 sh | 61 Euphoria | 77 PHP |
| 14 ksh | 30 RSL | 46 csh | 62 OCL | 78 XSLT |
| 15 Promela | 31 C++ | 47 Modelica | 63 VBScript | |
| 16 Assembler | 32 Mercury | 48 SHELXL | 64 Fortran | |

Также поддерживаются другие диалекты. Полный список Вы можете найти в документации:

<https://ctan.org/tex-archive/macros/latex/contrib/listings/>

Если Вы хотите включить псевдокод или алгоритмы, Вы также можете использовать пакеты **algorithm2e**, **algpseudocode** и **algorithm**.

Работа с пакетом listings

Практические примеры

Работа с пакетом listings

Чтобы использовать пакет **listings**, Вы должны его подключить:

```
\usepackage{listings}
```

Работа с пакетом listings

Для того чтобы вставить программный код, Вы должны использовать следующую конструкцию:

```
\begin{lstlisting}  
    Вставьте Ваш код здесь  
\end{lstlisting}
```

Работа с пакетом listings

Например, следующий \LaTeX -код позволит вставить код поиска максимального элемента массива A на языке $C++$:

```
\begin{lstlisting}
    max = A[0];
    i_max = 0;
    for(i = 1; i < n; i++)
        if (A[i] > max)
        {
            max = A[i];
            i_max = i;
        }
\end{lstlisting}
```

Работа с пакетом listings

Результат:

```
max = A[0];  
i_max = 0;  
for(i = 1; i < n; i++)  
    if (A[i] > max)  
    {  
        max = A[i];  
        i_max = i;  
    }
```

Работа с пакетом listings

Другой способ вставить код очень полезен, если Вы ещё занимаетесь его редактированием. Если Вы изменяете исходный код, то просто перекомпилируйте код \LaTeX и документ будет обновлён. Для этого используйте следующий код:

```
\lstinputlisting{source_filename.py}
```

Работа с пакетом listings

```
# считываем массив с клавиатуры
A = list(map(int, input().split()))
mx = A[0]
i_mx = A[0]
for i in range(1, len(A)):
    if A[i] > mx:
        mx = A[i]
        i_mx = i
```

В примере используется файл *Python*, но Вы можете добавлять любой файл, содержащий программный код.

Работа с пакетом listings

Если код не подсвечивается, значит компилятор не смог определить язык на котором написан код. Вы можете определить язык вручную следующей командой:

```
\lstinputlisting[language=Python]{source_filename.py}
```

Работа с пакетом listings

Результат с подсветкой синтаксиса:

```
# считываем массив с клавиатуры  
A = list(map(int, input().split()))  
mx = A[0]  
i_mx = A[0]  
for i in range(1, len(A)):  
    if A[i] > mx:  
        mx = A[i]  
        i_mx = i
```

Настройка конфигурации пакета listings

Настройка конфигурации пакета listings

Настройка цветовой схемы:

```
\usepackage{listings}
\usepackage{color}

\definecolor{dkgreen}{rgb}{0,0.6,0}
\definecolor{gray}{rgb}{0.5,0.5,0.5}
\definecolor{mauve}{rgb}{0.58,0,0.82}
```

Настройка конфигурации пакета listings

```
\lstset{
  % Язык программирования
  language=Python,
  % С какой стороны нумеровать
  numbers=left,
  % Стилй который будет использоваться
  % для нумерации строк
  numberstyle=\tiny\color{gray},
  % Шаг между линиями. Если 1, то будет
  % пронумерована каждая строка
  stepnumber=2,
  numbersep=5pt,
  ...
}
```

Настройка конфигурации пакета listings

```
...  
% Цвет подложки. Вы должны добавить  
% пакет color - \usepackage{color}  
    backgroundcolor=\color{white},  
    showspaces=false,  
    showstringspaces=false,  
    showtabs=false,  
% Добавить рамку  
    frame=single,  
    rulecolor=\color{black},  
% Tab - 2 пробела  
    tabsize=2,  
% Автоматический перенос строк  
    breaklines=true,  
...
```

Настройка конфигурации пакета listings

```
...  
% Переносить строки по словам  
    breakatwhitespace=true,  
% Показать название подгружаемого файла  
    title=\lstname,  
% Стилй ключевых слов  
    keywordstyle=\color{blue},  
% Стилй комментариев  
    commentstyle=\color{dkgreen},  
% Стилй литералов  
    stringstyle=\color{mauve}  
}
```

Настройка конфигурации пакета listings

Получим красивое оформление исходного кода:

source_filename.py

```
# считываем массив с клавиатуры
2 A = list(map(int, input().split()))
mx = A[0]
4 i_mx = A[0]
for i in range(1, len(A)):
6     if A[i] > mx:
            mx = A[i]
8     i_mx = i
```


Настройка конфигурации пакета listings

Вы также можете указать область для файла:

```
\lstinputlisting[language=Python,  
firstline=37, lastline=45]{source_filename.py}
```

или

```
\lstinputlisting[language=Python,  
linrange={37-45,48-50}]{source_filename.py}
```

Недостатки пакета listings

Проблема с UTF-8

При попытке вставить в код текст на кириллице (не в комментарий, а именно в код — текст должен был выводиться в графическом окне) будет получено сообщение об ошибке вида:

```
Package inputenc Error: Unicode char \u8:  
?\lst@FillFixed@ not set up for use
```

При этом кодировка основного документа — *UTF-8*, что указано в преамбуле.

Проблема с UTF-8

Сообщения об ошибках исчезнут, если указать в настройках `\lstset` использование не-ASCII символов:

```
\lstset{extendedchars=\true}
```

Важно: именно `\true`, а не `true`, как указано, например, на [Wikibooks](#).

Однако при этом остаётся проблема с русскоязычными комментариями, в которых слова оказываются склеены. Окончательно решает эту проблему следующий пакет.

Пакет minted

minted — альтернатива ставшему популярным пакету **listings**. Он использует внешнюю библиотеку *Python Pygments* для выделения кода, которая по состоянию на 2021 год может похвастаться более чем 500 поддерживаемыми языками и текстовыми форматами.

Поскольку пакет полагается на внешний код *Python*, для его установки требуется больше дополнительных шагов, чем для обычного пакета \LaTeX , поэтому, пожалуйста, ознакомьтесь с его репозиторием на GitHub и руководством.

**БЛАГОДАРЮ ЗА
ВНИМАНИЕ!**