

Лекция 16. Элементы программирования

Макроопределения

Макроопределения служат для создания новых команд и повышения скорости набора. Чаще всего (но не обязательно) их помещают в преамбулу документа.

Не рекомендуется злоупотреблять макроопределениями, особенно в журнальных статьях.

Простейший случай: макроопределение без параметров:

```
\newcommand{имя_команды}{определение_команды}
```

Пример:

```
\newcommand{\eps}{\varepsilon}
\newcommand\tr{\mathop{\mathrm{tr}}}
```

Формулы $\$\eps\$, $\$\tr A\$$ дают ε , $\mathrm{tr} A$$

Макроопределения с параметрами

Общий вид определения команды:

```
\newcommand{имя_команды}[N][параметр_по_умолчанию]%
{определение_команды}
```

Здесь N — целое число из диапазона 1–9, число параметров команды ($N = 0$ по умолчанию).

Формальные параметры имеют имена #1, #2, ...

Значение по умолчанию можно задать только для первого формального параметра (при вызове команды он указывается в квадратных скобках).

Макроопределения с параметрами. Примеры

Пусть в документе есть формулы:

$$\frac{\partial f(x, y)}{\partial x}, \quad \frac{\partial f(x, y)}{\partial y}, \quad \frac{\partial g(u, v)}{\partial u}, \quad \frac{\partial g(u, v)}{\partial v}$$

Определим макрокоманду:

```
\newcommand{\pdr}[4]{\frac{\partial #1(#2,#3)}{\partial #4}}
```

Формулы выше можно получить как:

```
$$
\pdr{f}{x}{y}{x}, \quad \pdr{f}{x}{y}{y}, \quad \pdr{g}{u}{v}{u}, \quad \pdr{g}{u}{v}{v}
$$
```

Здесь значение по умолчанию отсутствует.

Макроопределения с параметрами. Примеры

Определим макрокоманду со значением по умолчанию:

```
\newcommand{\Sum}[4][i]{\sum_{#1=#2}^{#3} #4_{#1}}
```

Формулы:

```
$$  
\Sum{0}{n}{a}, \quad \Sum{j}{1}{m}{C}  
$$
```

дают

$$\sum_{i=0}^n a_i, \quad \sum_{j=1}^m C_j$$

Макроопределения с параметрами. Примеры

Определим макрокоманду, которая сама переходит в математический режим, а потом возвращается из него:

```
\newcommand{\pdiff}[2]{\ensuremath{\frac{\partial^2}{\partial #2^2}}}
```

Эту команду можно вызывать в текстовом режиме: `\pdiff{f}{x}` дает $\frac{\partial^2 f}{\partial x^2}$

Переопределение макрокоманд

Чтобы переопределить существующую макрокоманду используется `\renewcommand` с тем же синтаксисом, что у `\newcommand`.

Пример:

```
\renewcommand\le{\leqslant}  
\renewcommand\kappa{\varkappa}
```

Можно переопределить команду с другим числом аргументов:

```
\renewcommand{\le}[2]{#1\leqslant #2}
```

Тогда `$(\le{a}{b})$` даст $a \leq b$.

Это не перегрузка, а просто замена!

Определение макрокоманд (продолжение)

Если вместо `\newcommand` вызвать `\providecommand`, то:

- если команда с таким именем не определена, то она определяется;
- если команда с таким именем уже существует, то переопределения не происходит и используется старый вариант команды.

Счетчики

Счетчики — служебные целочисленные переменные, для которых определены операции увеличения счетчика и уменьшения значения счетчика.

Счетчики бывают стандартные и определенные пользователем.

Стандартные счетчики связаны с некоторыми окружениями (equation, figure, table), с разделами текста (chapter, section, subsection), другими элементами (page, footnote, enumi).

Имя стандартного счетчика совпадает с именем окружения или раздела (equation, section и т.п.).

Значение счетчика в виде строки

Получить значение счетчика в виде строки можно с помощью команд:

<code>\arabic{имя_счетчика}</code>	— арабскими цифрами,
<code>\roman{имя_счетчика}</code>	— римскими строчными,
<code>\Roman{имя_счетчика}</code>	— римскими прописными,
<code>\alph{имя_счетчика}</code>	— латинская строчная буква,
<code>\Alph{имя_счетчика}</code>	— латинская прописная буква
<code>\asbuk{имя_счетчика}</code>	— кириллическая строчная буква,
<code>\Asbuk{имя_счетчика}</code>	— кириллическая прописная буква,
<code>\fnsymbol{имя_счетчика}</code>	— символ с заданным кодом

Для латинских букв значение счетчика не должно быть в диапазоне 1–26, для русских 1–28, для символов 1–9.

(Для римских чисел отрицательные значения и ноль игнорируются.)

Значение счетчика в виде строки. Пример

Пусть значение счетчика `section` равно 9. Тогда код

```
Параграф номер~\arabic{section}  
Параграф номер~\roman{section}  
Параграф номер~\Roman{section}  
Параграф номер~\alph{section}  
Параграф номер~\Alph{section}  
Параграф номер~\asbuk{section}  
Параграф номер~\Asbuk{section}  
Параграф номер~\fnsymbol{section}
```

дает:

```
Параграф номер 9  
Параграф номер ix  
Параграф номер IX  
Параграф номер i  
Параграф номер I  
Параграф номер и  
Параграф номер И  
Параграф номер ‡‡
```

Значение счетчика в виде числа

- Установка значения счетчика в число:

```
\setcounter{имя_счетчика}{число}
```

- Текущее значение счетчика (в виде числа):

```
\value{имя_счетчика}
```

Пример:

```
\setcounter{equation}{10}
\setcounter{section}{\value{equation}}
Уравнение~\arabic{equation}
Параграф~\arabic{section}
```

дает:

Уравнение 10 Параграф 10

Создание нового счетчика

Создание нового счетчика (с инициализацией нулем):

```
\newcounter{имя_счетчика}
```

Может находиться как в преамбуле, так и в документе.

При создании счетчика автоматически создается the-команда (ее можно переопределить):

```
\newcommand{theимя_счетчика}{\arabic{имя_счетчика}}
```

Пример:

```
\newcounter{task}
\setcounter{task}{1}
Задание номер~\arabic{task}
Еще раз задание номер~\thetask
```

дает:

Задание номер 1 Еще раз задание номер 1

Изменение значения счетчика

- Увеличение значения счетчика на число N :

```
\addtocounter{имя_счетчика}{N}
```

Число N может быть отрицательным — тогда получим уменьшение значения.

- Увеличение значения счетчика на единицу:

```
\stepcounter{имя_счетчика}
\refstepcounter{имя_счетчика}
```

Второй вариант позволяет корректно работать с командами `\label` и `\ref` при создании новых команд и окружений. Он устанавливает `\ref` в `\theимя_счетчика`.

Изменение значения счетчика (пример)

Рассмотрим код:

```
\setcounter{task}{5}
\addtocounter{task}{2}
\thetask
\addtocounter{task}{-3}
\thetask
\stepcounter{task}
\thetask
```

Получим:

7 4 5

Работа со счетчиком. Пример 1

Рассмотрим код:

```
\newcounter{prob}
\newcommand{\zadacha}%
{\par\stepcounter{prob}\textbf{Задача №~\theprob}.\ }

\zadacha
Решить задачу \dots
\zadacha
Решить вторую задачу \dots
\zadacha
Решить третью задачу \dots
```

Получаем:

Задача № 1. Решить задачу ...

Задача № 2. Решить вторую задачу ...

Задача № 3. Решить третью задачу ...

Работа со счетчиком. Пример 2

Рассмотрим код:

```
\newcounter{prob}
\newcommand{\zadacha}%
{\par\refstepcounter{prob}\textbf{Задача №~\theprob}.\ }
\setcounter{prob}{3}
\zadacha\label{p1}
Решить задачу \dots
\zadacha\label{p2}
Решить вторую задачу \dots

Задача \ref{p1}\dots\ \
Задача \ref{p2}\dots\ \
```

Задача № 4. Решить задачу ...

Задача № 5. Решить вторую задачу ...

Задача 4... Задача 5...

Подчинение счетчиков

Создаваемый счетчик можно подчинить другому: тогда при изменении значения подчиняющего счетчика с помощью команд `\stepcounter` или `\refstepcounter` подчиненный счетчик сбрасывается в ноль.

Замечание: если подчиняющий счетчик изменен другими командами, то сброса не происходит.

Например, счетчик параграфов обычно подчинен счетчику глав.

Создание подчиненного счетчика:

```
\newcounter{подчиненный_счетчик}[подчиняющий_счетчик]
```

Подчинение счетчиков. Пример

Рассмотрим код:

```
\newcounter{subtask}[task]
\setcounter{task}{5}
\setcounter{subtask}{2}
\thetask.\thesubtask
\stepcounter{task}
\thetask.\thesubtask
```

Получим:

5.2

6.0

Переопределим:

```
\renewcommand{\thesubtask}{\thetask.\arabic{subtask}}
\thesubtask
```

Получим:

6.0

Окружения

Окружения служат для объединения сложной последовательности команд. Работа с окружением выглядит следующим образом:

```
\begin{имя_окружения}[параметры]
  Тело окружения
\end{имя_окружения}
```

Мы уже сталкивались с окружениями `equation`, `tabular`, `table`, `center`, `array` и многими другими.

Новое окружение создается командой:

```
\newenvironment{имя_окружения}[N]{начальные_команды}%
{конечные_команды}
```

Здесь N — число параметров окружения (как в командах). Начальные и конечные команды выполняются до и после выполнения тела.

Переопределение окружения: команда `\renewenvironment`

Создание окружения. Пример

Опишем окружение для создания экзаменационных билетов.

```
\newenvironment{bilet}[1]%
{\begin{minipage}[t][13.5cm][t]{16cm}% высота и ширина
{Группа 1200\hfill Семестр III
\noindent
\begin{center}\bfseries
БИЛЕТ №{#1}
\end{center}}}%
{\end{minipage}}
```

Использование окружения:

```
\begin{bilet}{3} % Билет 3
Тело билета
\end{bilet}
```

Создание окружения. Пример (продолжение)

Группа 1200

БИЛЕТ № 3

Семестр III

1. Лейблы списки. Прямые вставки узлов.
2. Другая классы. Способы объявление друзей.

Группа 1200

БИЛЕТ № 4

Семестр III

1. Новые принципы структурного программирования.
2. Объявление и описание класса. Члены класса. Объекты класса.

Переменные для описания длины

В отличие от счетчиков это переменные вещественного типа.

Создаются командой:

```
\newlength{\имя_переменной}
```

Пример:

```
\newlength{\myspace}
```

Присваивание значений двумя способами (с указанием «клея» \pm):

```
\myspace = 1cm plus 2mm minus 1mm  
\setlength{\myspace}{1cm plus 2mm minus 1mm}
```

Имя переменной длины является глобальным, но все команды для работы со значениями локализованы внутри блока.

Некоторые стандартные переменные длины

<code>\textwidth</code>	% ширина текстового поля
<code>\textheight</code>	% высота текстового поля
<code>\parindent</code>	% абзацный отступ
<code>\baselineskip</code>	% расстояние между строками
<code>\oddsidemargin</code>	% смещение бокового поля
<code>\topmargin</code>	% смещение верхнего поля

Переменные длины. Преобразование в пробелы

Чтобы преобразовать значение переменной в горизонтальные или вертикальные пробелы, используем команды

```
\hspace{\myspace}  
\vspace{\myspace}
```

Например:

горизонталь-1 `\hspace{\myspace}` горизонталь-2

вертикаль-1 `\\ \vspace{\myspace}` вертикаль-2

дает:

горизонталь-1	горизонталь-2
вертикаль-1	вертикаль-2

Работа с переменными длины. Умножение

Код:

```
\myspace = 1cm plus 2mm minus 1mm  
слово1\hspace{\myspace}слово2  
  
слово1\hspace{3.5\myspace}слово2  
  
\newlength{\newspace}  
\newspace = 3.5\myspace  
слово1\hspace{\newspace}слово2
```

Получаем:

слово1	слово2
слово1	слово2
слово1	слово2

Работа с переменными длины. Сложение

Код:

```
\myspace = 1cm plus 2mm minus 1mm  
слово1\hspace{\myspace}слово2
```

```
\addtolength{\myspace}{1cm}  
слово1\hspace{\myspace}слово2
```

Получаем:

```
слово1      слово2  
слово1      слово2
```

Работа с переменными длины. Измерение текста

Можно из текста получить его размеры, записав их в параметр длины:

```
\settowidth{параметр}{текст}  
\settoheight{параметр}{текст}
```

Например:

```
\newlength{\myw}  
\settowidth{\myw}{ТЕКСТ}  
ТЕКСТ
```

```
\hspace{\myw} ТЕКСТ
```

```
\hspace{2\myw} ТЕКСТ
```

```
ТЕКСТ  
      ТЕКСТ  
      ТЕКСТ
```

Сложные вычисления

Для сложных вычислений (в том числе, с математическими функциями) имеются специальные пакеты: calc, calculator, calculus и другие.