

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики–процессов управления

Краткосрочная дополнительная образовательная  
программа повышения квалификации  
**«ПОДГОТОВКА МАТЕМАТИЧЕСКИХ ТЕКСТОВ В  $\text{\LaTeX}$ »**

## **Возможности оформления таблиц в $\text{\LaTeX}$**

Слушатель программы

к.ф.-м.н., доц. ф-та ПМ–ПУ  
Просолупов Е. В.

Консультант

к.ф.-м.н., доц. ф-та ПМ–ПУ  
Иванова О. А.

Санкт-Петербург  
2013

# 1 Основные средства создания таблиц

Таблица — это особая форма передачи содержания. Таблица состоит из строк и столбцов. Строки или столбцы могут иметь заголовки. Обычно заголовок есть у каждого столбца (колонки), но может потребоваться озаглавить и строки. Заголовки колонок указываются в самой верхней строке таблицы. Заголовки строк обычно указываются в самой левой колонке.

## 1.1 Окружение `tabbing`

Если таблица имеет простую структуру и известна ширина каждой колонки, для её создания можно использовать окружение `tabbing`.

Первой строкой в окружении необходимо задать ширину колонок, указывая положения границ ячеек командой «`\=`». Завершает служебную строку команда «`\kill`».

Дальше вводится содержимое таблицы. Для разделения ячеек служит команда «`\>`», а для перехода к следующей строке таблицы — команда «`\`».

Например, чтобы получить табличку вида

<b>T(n)</b>	<b>10</b>	<b>50</b>	<b>60</b>
$n$	$10^{-5}$ с.	$5 \cdot 10^{-5}$ с.	$6 \cdot 10^{-5}$ с.
$n^5$	0.1 с.	5.2 мин.	13.0 мин.
$2^n$	0.001 с.	35.7 лет	366 стол.

можно использовать следующий L<sup>A</sup>T<sub>E</sub>X-код:

---

```
\begin{tabbing}
% Эта строка только определяет ширину колонок.
MMMM \= MMMMMM \= MMMMMM \= MMMMMM \kill
% Здесь начинается первая строка содержания таблицы.
\bf T(n) \> {\bf 10} \> {\bf 50} \> {\bf 60}\\ % Конец первой строки
$n$ \> $10^{-5}$ с. \> $5\cdot 10^{-5}$ с. \> $6\cdot 10^{-5}$ с.\\
$n^5$ \> 0.1 с. \> 5.2 мин. \> 13.0 мин.\\
$2^n$ \> 0.001 с. \> 35.7 лет \> 366 стол.
\end{tabbing}
```

---

Окружение `tabbing` обладает рядом ограничений и особенностей:

- Необходимо явно задать ширину ячеек с помощью символов, или команды `\hspace{}`.
- Нет возможности вкладывать одну таблицу в другую.
- Нет возможности отображения разделительных линий между ячейками.
- Возможно использовать только в текстовом режиме.
- Таблицы, созданные с помощью окружения **tabbing**, в случае необходимости переносятся на следующую страницу по строкам.

## 1.2 Окружение **tabular**

Решить описанные выше проблемы поможет окружение **tabular**. Оно обладает большим количеством настроек, а также его возможности могут быть расширены подключением дополнительных пакетов.

При создании таблицы с помощью окружения **tabular** в первую очередь нужно задать количество колонок и выравнивание в них. Для этого у этого окружения есть обязательный параметр: последовательность из символов «l», «c», «r» и «|». Суммарное количество символов «l», «c» и «r» указывает количество колонок в таблице. При этом символ «l» указывает, что выравнивание содержимого в соответствующей колонке будет происходить по левому краю ячейки, символ «r», что выравнивание будет происходить по правому краю, а символ «c» указывает на центрирование содержимого.

Например, команда `\begin{tabular}{lrrc}` будет начинать таблицу с четырьмя колонками, где первая колонка будет выровнена по левому краю, вторая — по правому краю, а две последние колонки будут выровнены по центру.

Содержимое таблицы, как и при использовании окружения **tabbing** формируется по строкам. Содержимое ячеек в строке разделяется символом «&», а каждая строка завершается командой «`\`».

Приведем пример. Построим таблицу степеней: колонки таблицы будут отвечать за показатели степени, а строки — за основание степени. На этом примере будет видно, как выравнивание содержимого колонок по правому краю влияет на результирующий вид таблицы.

	2	3	4	5
2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024
5	25	125	625	3125

---

```

\begin{tabular}{crrrr}
  & 2 & 3 & 4 & 5 \\
2 & 4 & 8 & 16 & 32 \\
3 & 9 & 27 & 81 & 243 \\
4 & 16 & 64 & 256 & 1024 \\
5 & 25 & 125 & 625 & 3125
\end{tabular}

```

---

Очевидно, таблица будет читаться легче, если в нее добавить горизонтальные и вертикальные линии, отделяя первую строку и первый столбец. Это не сложно сделать. Символ «|» в обязательном параметре окружения **tabular** указывает промежутки между колонками, где будет нарисована вертикальная линия на всю высоту таблицы. Для указания горизонтальных линий в таблице, созданной при помощи окружения **tabular** служит команда **\hline**. При желании, можно использовать несколько символов «|» для обозначения двойных, тройных и т.д. вертикальных линий. Так же можно использовать несколько команд **\hline** подряд.

Рассмотрим таблицу из предыдущего примера, но добавим в нее несколько линий для удобства восприятия.

	2	3	4	5
2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024
5	25	125	625	3125

---

```

\begin{tabular}{c||rrrr|}
  & 2 & 3 & 4 & 5 \\
\hline
\hline
2 & 4 & 8 & 16 & 32 \\
3 & 9 & 27 & 81 & 243 \\
4 & 16 & 64 & 256 & 1024 \\
5 & 25 & 125 & 625 & 3125 \\
\hline
\end{tabular}

```

---

Помимо обязательного параметра у окружения **tabular** есть еще необязательный параметр, который отвечает за вертикальное выравнивание таблицы относительно окружающего текста, если таблица используется внутри абзаца, а не отдельно. Этот параметр может иметь одно из трех значений «t», «c» и «b», которые указывают на выравнивание по верхней строке, по центру и по нижней строке соответственно.

### Пример:

			реку.				
			руку.			реку	
Ехал Грека через	реку.	Видит Грека —	в реке	рак.	Сунул Грека	руку	в реку.
	руку.					рак	руку.
	рак.						рак.

---

```

Ехал Грека через
\begin{tabular}[t]{l}
реку. \\
руку. \\
рак.
\end{tabular}
Видит Грека --- в реке
\begin{tabular}[b]{l}
реку. \\
руку. \\
рак.
\end{tabular}
Сунул Грека
\begin{tabular}[c]{l}
реку \\
руку \\
рак
\end{tabular}
в
\begin{tabular}[t]{l}
реку. \\
руку. \\
рак.
\end{tabular}

```

---

Обратите внимание, что конец прерывание строки после последней строки таблицы, или линия в начале таблицы могут помешать правильному выравниванию:

**Пример:**

Ехал Грека через 

реку. руку. рак.
------------------------

 Видит Грека — в реке 

реку. руку. рак.
------------------------

 Сунул Грека 

реку руку рак
---------------------

 в 

реку. руку. рак.
------------------------

---

```

Ехал Грека через
\begin{tabular}[t]{|l|} \hline
реку. \\
руку. \\
рак. \\ \hline
\end{tabular}
Видит Грека --- в реке
\begin{tabular}[b]{|l|} \hline
реку. \\
руку. \\
рак. \\ \hline
\end{tabular}
Сунул Грека
\begin{tabular}[c]{|l|} \hline
реку \\
руку \\
рак \\ \hline
\end{tabular}
в
\begin{tabular}[t]{|l|} \hline
реку. \\
руку. \\
рак. \\ \hline
\end{tabular}

```

---

Если требуется создать сложное распределение элементов в одной из ячеек таблицы, можно использовать вложение одной таблицы в другую. Эта возможность позволяет создавать гораздо более сложные конструкции.

Например, изобразим блочную матрицу, где некоторые блоки приведены явно, а другие обозначены буквами.

$\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}$	$\mathbf{O}$	$\mathbf{I}$
$\mathbf{O}$	$\begin{smallmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{smallmatrix}$	$\mathbf{O}$
$\mathbf{I}$	$\mathbf{O}$	$\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}$

---

```
\begin{tabular}{c|c|c}
\begin{tabular}{cc}
1 & 0\\
0 & 1
\end{tabular}
& \mathbf{O}
& \mathbf{I}
\hline
\mathbf{O}
& \begin{tabular}{ccc}
0 & 1 & 0 \\
1 & 0 & 1 \\
0 & 1 & 0
\end{tabular}
& \mathbf{O}
\hline
\mathbf{I}
& \mathbf{O}
& \begin{tabular}{cc}
0 & 1 \\
1 & 0
\end{tabular}
\end{tabular}
```

---

Окружение **tabular** можно использовать как в текстовом, так и в математическом режиме, но при использовании в математическом режиме будут наблюдаться некоторые проблемы:

даже если вы поместите таблицу, созданную с помощью **tabular** в математическое окружение внутри **tabular** по прежнему будет текстовый режим и вся информация будет отображаться соответствующим шрифтом. Кроме того, математические формулы внутри такой таблицы также потребуют повторного перехода в математический режим.

Для использования в режиме редактирования формул, существует специальное окружение **array**, практически идентичное по возможностям окружению **tabular**.

Рассмотрим пример использования этого окружения:

	$x$	$y$	$z$	$f(x, y, z)$
$2^3$	0	0	0	1
	0	0	1	0
	0	1	0	1
	0	1	1	0
	1	0	0	0
	1	0	1	1
	1	1	0	0
	1	1	1	1

---

```

$$
\begin{array}{rrrr|c}
x~~ & y~~ & z~~ & f(x,y,z)\\
\hline
2^3 \left\{ \begin{array}{r}
0\\ 0\\ 0\\ 0\\ 1\\ 1\\ 1\\ 1
\end{array}
& \begin{array}{r}
0\\ 0\\ 1\\ 1\\ 0\\ 0\\ 1\\ 1
\end{array}
& \begin{array}{r}
0\\ 1\\ 0\\ 1\\ 0\\ 1\\ 0\\ 1
\end{array}
& \begin{array}{r}
1\\ 0\\ 1\\ 0\\ 0\\ 1\\ 0\\ 1
\end{array}
\end{array}

```



```
\end{array}
\end{array}
$$
```

---

Особенности окружения **tabular** (в отличие от окружения **tabbing**):

- Не требуется явно задавать ширину колонок, они растягиваются пропорционально содержанию.
- Можно (и нужно) задавать выравнивание по горизонтали для каждой колонки.
- Возможно создавать сложные структуры, вкладывая таблицы одну в другую.
- Можно отображать горизонтальные и вертикальные линии между ячейками и вокруг таблицы.
- Есть практически полностью аналогичное по возможностям и синтаксису окружение **array**, которое служит для создания таблиц в математическом режиме.
- Таблицы, созданные с помощью окружения **tabular** не переносятся частями на другую страницу. Чтобы создавать таблицы с возможностью переноса части таблицы на следующую страницу документа, необходимо пользоваться дополнительными пакетами.

## 2 Дополнительные возможности окружения **tabular**

Мы уже рассмотрели базовые функции окружения **tabular**, но не исчерпали все возможности его использования. Ниже мы приведем еще несколько настроек этого окружения, расширяющих диапазон использования.

Если какая-то часть описания опций выравнивания колонок (обязательного параметра окружения **tabular**) повторяется несколько раз, удобно использовать команду «\*» для указания многократного повторения последовательности символов. В первых фигурных скобках после символа «\*» нужно ввести количество повторений последовательности, а во вторых — собственно повторяемую последовательность символов.

Другими словами строка

```
«\begin{tabular}{||*{8}{rc||}}»
```

будет эквивалентна строке

```
«\begin{tabular}{||rc|rc|rc|rc|rc|rc|rc||}».
```

Приведем еще один пример. На этот раз это будет таблица значений основных булевых функций от двух переменных. Для повторяющегося семь раз описания одинаковых столбцов используем команду «**\*{7}{c|}**»:

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$	$x \rightarrow y$	$x \equiv y$	$x   y$	$x \downarrow y$
0	0	0	0	0	1	1	1	1
0	1	1	0	1	1	0	1	0
1	0	1	0	1	0	0	1	0
1	1	1	1	0	1	1	0	0

---

```

\begin{tabular}{|cc|*{7}{c|}}
\hline
x & y & x $\vee$ y & x $\wedge$ y & x $\oplus$ y & x $\rightarrow$ y & x $\equiv$ y & x | y & x $\downarrow$ y \\
\hline
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
\hline
\end{tabular}

```

---

Следующая команда параметра опций выравнивания колонок окружения **tabular** позволяет заменять вертикальный разделитель колонок (по умолчанию линия) на произвольный символ. Достаточно между значками выравнивания вместо символа «|» написать «**@{символы}**», чтобы указанные «символы» были вставлены между ячейками соответствующих столбцов в каждой строке.

Например, построим кусочек таблицы умножения для числа 2. Для указания знаков умножения и равенства зададим символы с помощью параметра окружения. Обратите внимание, как указанные символы располагаются один под другим во всех строках. Это можно использовать для представления систем однотипных выражений.

В этом же примере мы используем еще и возможность команды «**\\**» увеличивать размер вертикального отступа с помощью необязательного параметра.

$2 \times 2$	=	4
$2 \times 4$	=	8
$4 \times 8$	=	32
$8 \times 32$	=	256
$32 \times 256$	=	8 192
$256 \times 8192$	=	2 097 152

---

```

\begin{tabular}{|r@{${}\times$}l@{~~=~~}r|}
\hline
2 & 2 & 4 \\[3pt]
2 & 4 & 8 \\[3pt]
4 & 8 & 32 \\[3pt]
8 & 32 & 256 \\[3pt]
32 & 256 & 8 192 \\[3pt]
256 & 8192 & 2 097 152 \\
\hline
\end{tabular}

```

---

Еще одна вещь, которую можно настраивать в таблицах, созданных с помощью окружения **tabular**, это горизонтальные линии между строками таблицы. До сих пор мы использовали для создания горизонтальных разделителей только команду **\hline**, но существует также и команда **\cline**, в обязательном параметре которой можно указать, от какой ячейки и до какой будет проходить линия.

$\rightarrow$	$\rightarrow$	$\rightarrow$	$\downarrow$
$\circlearrowleft$	$\downarrow$	$\leftarrow$	$\leftarrow$
$\downarrow$	$\leftrightarrow$	$\rightarrow$	$\downarrow$
$\rightarrow$	$\rightarrow$	$\downarrow$	$\leftarrow$

---

```

\begin{tabular}{|cccc|} \cline{2-4}
$\rightarrow$ & $\rightarrow$ & $\rightarrow$ & $\downarrow$ \\
\cline{1-3}
$\circlearrowleft$ & $\downarrow$ & $\leftarrow$ & $\leftarrow$ \\
\cline{1-1}\cline{3-4}
$\downarrow$ & $\leftrightarrow$ & $\rightarrow$ & $\downarrow$ \\
$\rightarrow$ & $\rightarrow$ & $\downarrow$ & $\leftarrow$
\end{tabular}

```

---

```

\cline{2-3}
$\rightarrow$ & $\rightarrow$ & $\downarrow$ & $\leftarrow$ \\
\cline{1-2}\cline{4-4}
\end{tabular}

```

---

Толщиной линий тоже можно управлять. Для изменения толщины линии нужно с помощью команды **setlength{}** установить значение переменной **\arrayrulewidth**.

Например, поставив код таблицы из предыдущего примера в фигурных скобках здесь:

```

{\setlength{\arrayrulewidth}{2pt}
.....
}

```

можно сделать все линии таблицы толще.

→	→	→	↓
○	↓	←	←
↓	↔	→	↓
→	→	↓	←

Иногда можно изменить толщину даже отдельных линий:

1	2	3
4	5	6
7	8	9

---

```

\begin{tabular}{|c@{\setlength{\arrayrulewidth}{3pt}\vline~}c
                @{\setlength{\arrayrulewidth}{5pt}\vline~}c|} \hline
1 & 2 & 3 \\ \hline
4 & 5 & 6 \\ \hline
7 & 8 & 9 \\ \hline
\end{tabular}

```

---

Если требуется явным образом ограничить ширину колонки, это возможно сделать с помощью опции **p{}** декларации настроек колонок. В фигурных скобках вводится желаемая ширина колонки. Если текст в некоторой ячейке не уложится в указанную ширину, он будет перенесен на следующую строку. Выравнивание содержимого в ячейках в этом случае будет по левому краю.

Например, построим таблицу значений косинусов и синусов с равномерными по ширине столбцами.

$t$	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	$\pi$	$\frac{5\pi}{4}$	$\frac{3\pi}{2}$	$\frac{7\pi}{4}$	$2\pi$
$\cos t$	1	$\frac{\sqrt{2}}{2}$	0	$-\frac{\sqrt{2}}{2}$	-1	$-\frac{\sqrt{2}}{2}$	0	$\frac{\sqrt{2}}{2}$	1
$\sin t$	0	$\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2}$	0	$-\frac{\sqrt{2}}{2}$	-1	$-\frac{\sqrt{2}}{2}$	0

---

```

\begin{tabular}{|*{10}{p{20pt}}|} \hline
\rule{0pt}{15pt} $t$ & 0 & $\frac{\pi}{4}$ & & $\frac{3\pi}{4}$ & $\pi$ & $\frac{5\pi}{4}$ & $\frac{3\pi}{2}$ & $\frac{7\pi}{4}$ & $2\pi$ \\
$\cos t$ & 1 & $\frac{\sqrt{2}}{2}$ & 0 & $-\frac{\sqrt{2}}{2}$ & -1 & $-\frac{\sqrt{2}}{2}$ & 0 & $\frac{\sqrt{2}}{2}$ & 1 \\
$\sin t$ & 0 & $\frac{\sqrt{2}}{2}$ & 1 & $\frac{\sqrt{2}}{2}$ & 0 & $-\frac{\sqrt{2}}{2}$ & -1 & $-\frac{\sqrt{2}}{2}$ & 0 \\
\\ \hline
$\cos t$ & 1 & $\frac{\sqrt{2}}{2}$ & 0 & $-\frac{\sqrt{2}}{2}$ & -1 & $-\frac{\sqrt{2}}{2}$ & 0 & $\frac{\sqrt{2}}{2}$ & 1 \\
\\ \hline
$\sin t$ & 0 & $\frac{\sqrt{2}}{2}$ & 1 & $\frac{\sqrt{2}}{2}$ & 0 & $-\frac{\sqrt{2}}{2}$ & -1 & $-\frac{\sqrt{2}}{2}$ & 0 \\
\\ \hline
\end{tabular}

```

---

Здесь будет уместно сказать пару слов об управлении текстом в ячейках таблицы. Если ввести слишком много текста в колонку таблицы с типом выравнивания **l**, **c** или **r** таблица будет растягиваться по горизонтали и в конце концов выйдет за границу документа.

Со столбцом типа **p{...}** такой проблемы нет. Когда ширина столбца оказывается недостаточной для вмещения текста в одну строку, он переносится на следующую. При этом выравнивание текста в соответствующих ячейках осуществляется по левому краю по горизонтали и по верхнему краю по вертикали.

Для прерывания строки в рамках одной ячейки таблицы можно использовать команду **\newline**. Конечно, для начала нового абзаца можно как обычно (вне таблицы) использовать пустую строку. Используя команду **\newline** несколько раз подряд, можно, кроме того, добиться большего вертикального расстояния между абзацами.

Также для добавления вертикальных отступов внутри ячейки как обычно можно использовать команду **\vskip<отступ>** указав желаемый размер отступа в качестве параметра.

Для горизонтального отступа в начале абзаца внутри ячейки таблицы можно использовать команду **\parindent=<отступ>**. Для подавления горизонтального отступа в начале абзаца, где он не нужен, можно воспользоваться командой **\noindent**.

<p>Это ячейка таблицы шириной шесть сантиметров. Продемонстрируем в нем различные размеры отступа.</p> <p>Здесь отступ от левого края составляет четверть сантиметра.</p> <p>Здесь уже пол сантиметра.</p> <p>Дальше три четверти сантиметра.</p> <p>И наконец целый сантиметр.</p> <p>В этом абзаце мы не устанавливали отступ специально, но он сохранился в размере одного сантиметра, так как был задан ранее.</p> <p>А здесь отступа нет, так как мы его подавили командой <code>\noindent</code></p>	<p>Эта колонка шириной четыре сантиметра.</p> <p>Вот мы начали новый абзац с помощью пустой строки.</p> <p>А теперь прерываем строку с помощью команды <code>\newline</code>.</p> <p>Как видно, в этом случае отступа «красной строки» нет.</p> <p>Снова начинаем абзац с помощью пустой строки. Отступ снова есть.</p>	<p>В этой ячейке продемонстрируем вертикальные отступы. Ниже стоят две команды <code>\newline</code> подряд.</p> <p>Продолжение абзаца. А дальше стоит команда <code>\vskip40pt</code>.</p> <p>В отличие от случая с командой <code>\newline</code> здесь появляется «красная строка», хотя пустой строки для начала нового абзаца мы не вставляли.</p>
--	---	---

---

```
\begin{tabular}{|p{6cm}|p{4cm}|p{5cm}|} \hline
```

Это ячейка таблицы шириной шесть сантиметров.

Продемонстрируем в нем различные размеры отступа.

```
\parindent=0.25cm
```

Здесь отступ от левого края составляет четверть сантиметра.

```
\parindent=0.5cm
```

Здесь уже пол сантиметра.

```
\parindent=0.75cm
```

Дальше три четверти сантиметра.

```
\parindent=1cm
```

И наконец целый сантиметр.

В этом абзаце мы не устанавливали отступ специально, но он сохранился в размере одного сантиметра, так как был задан ранее.

```
\noindent
```

А здесь отступа нет, так как мы его подавили командой

```
{\bf \textbackslash noindent}
```

```
&
```

```
\parindent=0.5cm
```

Эта колонка шириной четыре сантиметра.

Вот мы начали новый абзац с помощью пустой строки.

```
\newline
```

А теперь прерываем строку

с помощью команды `{\bf \textbackslash newline}`.

Как видно, в этом случае отступа <<красной строки>> нет.

Снова начинаем абзац с помощью пустой строки. Отступ снова есть.

```
&
```

```
\parindent=0.5cm
```

В этой ячейке продемонстрируем вертикальные отступы.

Ниже стоят две команды `{\bf \textbackslash newline}` подряд.

```
\newline
```

```
\newline
```

Продолжение абзаца. А дальше стоит команда

```
{\bf \textbackslash vskip40pt}.
```

```
\vskip40pt
```

В отличие от случая с командой `{\bf \textbackslash newline}`

здесь появляется <<красная строка>>, хотя пустой строки для начала нового абзаца мы не вставляли. `\ \hline`

```
\end{tabular}
```

---

Еще одна возможность, которую удобно использовать в сочетании с таблицами, это бесконечно растяжимые пробелы. Команда `\hfill` задает такой горизонтальный пробел, который максимально расталкивает элементы, между которыми стоит, в разные стороны, а команда `\dotfill` дополнительно заполняет такой пробел бесконечным многоточием. Это полезно, например, если вы хотите создать вручную содержание для документа.

В примере ниже инструкция `p{0.7\textwidth}` указывает ширину первого столбца как 70% от ширины текстового поля, чтобы оставить пространство между заголовком и номером страницы, а инструкция `@{}` подавляет пробел между колонками таблицы.

Название параграфа	страница
Окружение <code>tabbing</code> .....	2
Окружение <code>tabular</code> .....	3
Дополнительные возможности окружения <code>tabular</code> .....	9
Расширение возможностей с помощью дополнительных пакетов	18

---

```

\begin{tabular}{p{0.7\textwidth}@{}l}
{\bf Название параграфа} \hfill {\bf страница} \\
Окружение tabbing\dotfill 2 \\
Окружение tabular\dotfill 3 \\
Дополнительные возможности окружения tabular\dotfill 9 \\
Расширение возможностей с помощью дополнительных пакетов\dotfill 18
\end{tabular}

```

---

Следующей возможностью для таблиц, созданных с помощью окружения **tabular**, является возможность объединять несколько соседних по горизонтали ячеек в одну. Для этой цели служит команда `\multicolumn{...}{...}{...}`. В качестве первого аргумента команды требуется указать число объединяемых ячеек. Вторым аргументом является стиль ячейки: выравнивание и вертикальные границы ячейки. Стиль задается теми же инструкциями, как и в аргументе команды `\tabular`.

		График дежурства						
Фамилия		Пн.	Вт.	Ср.	Чт.	Пт.	Сб.	Вс.
1.	Иванов	×			×			-
2.	Петров		×			×		-
3.	Сидоров			×			×	-

---

```

\begin{tabular}{|l|l|*{7}{c|}} \cline{3-9}
\multicolumn{2}{|c|}{} & \multicolumn{7}{c|}{График дежурства}\\ \hline
\multicolumn{2}{|c|}{Фамилия} & Пн. & Вт. & Ср. & Чт. & Пт. & Сб. & Вс.\\ \hline
1. & Иванов & $\times$ & & & $\times$ & & & - \\ \hline
2. & Петров & & $\times$ & & & $\times$ & & - \\ \hline
3. & Сидоров & & & $\times$ & & & $\times$ & - \\ \hline
\end{tabular}

```



---

Еще пример:

-							
0				1			
00		01		10		11	
000	001	010	011	100	101	110	111

---

```
\begin{tabular}{|*{8}{c|}}\hline
\multicolumn{8}{|c|}{-} \\ \hline
\multicolumn{4}{|c|}{0} & \multicolumn{4}{|c|}{1} \\ \hline
\multicolumn{2}{|c|}{00} & \multicolumn{2}{|c|}{01} & \multicolumn{2}{|c|}{10} & \multicolumn{2}{|c|}{11} \\ \hline
000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ \hline
\end{tabular}
```

---

Кроме расширения ячеек команду `\multicolumn` можно использовать для изменения стиля границ отдельных ячеек. Поскольку третий аргумент команды `\multicolumn` позволяет указать стиль правой границы (или обеих границ для левого ряда) измененной этой командой ячейки, ничто не мешает нам использовать эту возможность и без объединения ячеек.

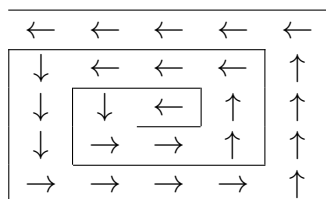
A	B	A
B	B	B
A	B	A

---

```
\begin{tabular}{|c|c|c|}\hline
A & B & A \\ \cline{1-1}\cline{3-3}
\multicolumn{1}{|c|}{B} & \multicolumn{1}{|c|}{B} & B \\ \cline{1-1}\cline{3-3}
A & B & A \\ \hline
\end{tabular}
```

---

Еще один пример использования указанной возможности.




---

```

\begin{tabular}{|ccccc|}\hline
\multicolumn{1}{c}{\leftarrow} & \leftarrow & \leftarrow & \leftarrow & \leftarrow \\
\downarrow & \leftarrow & \leftarrow & \leftarrow & \uparrow \\
\downarrow & \downarrow & \leftarrow & \uparrow & \uparrow \\
\downarrow & \rightarrow & \rightarrow & \uparrow & \uparrow \\
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \uparrow \\
\end{tabular}

```

---

Итак, в этом параграфе мы рассмотрели продвинутое возможности окружения **tabular**. Перечислим их коротко.

- Краткая запись для последовательности повторяющихся описаний колонок.
- Изменение символов разделителей между колонками (повторяющиеся выражения в каждой строке).
- Возможность проводить неполные горизонтальные линии.
- Явное задание ширины столбца.
- Объединение нескольких ячеек по горизонтали.
- Также мы сказали несколько слов об управлении текстом в ячейках и про заполнение пространства таблицы растяжимым пробелом или линией из точек (для создания содержаний).

## 3 Расширение возможностей с помощью дополнительных пакетов

### 3.1 Пакет `hhline`

Пакет **hhline** позволяет управлять видом горизонтальных линий и пересечений горизонтальных линий для каждой позиции индивидуально. При подключении этого пакета становится доступной команда `\hhline`, которую можно использовать для создания горизонтальных линий наряду с `\hline` и `\cline`, но которая позволяет указывать внешний вид линии в качестве параметра. Доступны следующие управляющие символы.

Для указания вида границы между соседними по вертикалям ячейками используется:

«—» обозначает простую горизонтальную линию;

«=» обозначает двойную горизонтальную линию;

«~» указывает, что разделительную линию в этом месте рисовать не нужно.

Общая длина линии должна совпадать с суммарным количеством символов «—», «=» и «~». Кроме того для управления пересечениями с вертикальными линиями в подходящих местах можно указывать один из следующих вариантов:

«|» обозначает вертикальную линию, которая перечеркивает двойную или одинарную горизонтальную линию. Для двойной вертикальной линии нужно использовать этот символ дважды.

«:» обозначает вертикальную линию, которая разрывает двойную горизонтальную линию. Если требуется пересечение двойной вертикальной и двойной горизонтальной линий с обоюдным разрывом, нужно поставить два двоеточия подряд.

«#» обозначает пересечение вертикальной и горизонтальной двойных линий.

«t» обозначает верхнюю горизонтальную черточку для области пересечения двойных линий. То есть, символ «t» нужно использовать, если при разрыве горизонтальной двойной линии вертикальной двойной линией сплошной должна остаться только верхняя из горизонтальных линий.

«b» обозначает нижнюю горизонтальную черточку для области пересечения двойных линий. Используется аналогично «t».

Дополнительно, для повторяющихся последовательностей в описании аргумента команды **hhline** можно использовать знакомую инструкцию «\*». Например, запись «\*{3}{-||-}» в аргументе **hhline** будет эквивалентна «-||-||-||-».

Ниже приведен пример, использующий все указанные возможности.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

---

```

\begin{tabular}{c*{4}{||c}||}
\hhline{=:t:=#:t==:t|}
1 & 2 & 3 & 4 & 5\\
\hhline{|=*{4}{:=}:|}
6 & 7 & 8 & 9 & 10\\
\hhline{=#=:t:=:b:=::=||}
11 & 12 & 13 & 14 & 15\\
\hhline{-||-||~|:=#-||}
16 & 17 & 18 & 19 & 20\\
\hhline{|=:b:=#b==#}
\end{tabular}

```

---

Вернемся к одному из наших предыдущих примеров.

	2	3	4	5
2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024
5	25	125	625	3125

Здесь видно, что по умолчанию горизонтальные и вертикальные линии накладываются не самым корректным образом. Теперь мы могли бы сделать отображение таблицы более аккуратным. Как-нибудь так:

	2	3	4	5
2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024
5	25	125	625	3125

---

```

\begin{tabular}{c||rrrr|}
\hhline{~|t|~~~~}
& 2 & 3 & 4 & 5 \\
\hhline{|=#===|}
2 & 4 & 8 & 16 & 32 \\
3 & 9 & 27 & 81 & 243 \\
4 & 16 & 64 & 256 & 1024 \\
5 & 25 & 125 & 625 & 3125 \\
\hline
\end{tabular}

```

---

Или так:

	2	3	4	5
2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024
5	25	125	625	3125

---

```

\begin{tabular}{||c||rrrr||}
\hhline{|t:=:t====:t|}
& 2 & 3 & 4 & 5 \\
\hhline{|:=:====:|}
2 & 4 & 8 & 16 & 32 \\
3 & 9 & 27 & 81 & 243 \\
4 & 16 & 64 & 256 & 1024 \\
5 & 25 & 125 & 625 & 3125 \\
\hhline{|b:=:b====:b|}
\end{tabular}

```

---

Рисунок 1 демонстрирует основные случаи применения возможностей **hhline**.

На печати	$\vdash$	$\dashv$	$\vdash, \top, \perp$	$\vDash$	$\dashv$	$\neq, \top, \perp$
В аргументе <code>\hhline</code>	$\mid-$	$-\mid$	$-\mid-$	$\mid=$	$=\mid$	$=\mid=$
На печати	$\vDash$	$\dashv$	$\neq, \top, \perp$	$\Vdash$	$\dashv$	$\nVdash$
В аргументе <code>\hhline</code>	$\vdash$	$\vdash$	$\vdash$	$\mid\mid-$	$-\mid\mid$	$-\mid\mid-$
На печати	$\Vdash$	$\dashv$	$\nVdash$	$\#$	$\#$	$\#, \top, \perp$
В аргументе <code>\hhline</code>	$\mid\vdash$	$\vdash\mid$	$\vdash\vdash$	$\#$	$\#$	$\#$
На печати	$\Uparrow$	$\Llcorner$	$\Uparrow$	$\Downarrow$	$\Uparrow$	$\Downarrow$
В аргументе <code>\hhline</code>	$\mid\vdash$	$\vdash\mid$	$\vdash\vdash$	$\vdash\mid$	$\vdash\vdash$	$\vdash\vdash$

Рис. 1: Варианты использования `hhline`

### 3.2 Пакет `multirow`

Мы уже говорили, что команда `\multicolumn` позволяет объединять несколько соседних по горизонтали ячеек. Хотелось бы получить аналогичную возможность для объединения соседних по вертикали ячеек. Подключение пакета `multirow` дает возможность использовать команду `\multirow{<число строк>}{<ширина колонки>}{<содержимое>}` или `\multirow{<число строк>}*{<содержимое>}`.

В обоих вариантах команды первым аргументом указывается количество объединяемых по вертикали ячеек, а последним — наполнение ячейки. Отличие вариантов команды заключено во втором параметре: первый вариант команды позволяет явно указать точную ширину полученной объединением ячейки, а второй вариант указывает, что ширина ячейки определяется автоматически.

Обратите внимание, что `\multirow` только располагает указанный текст по вертикали по середине предполагаемого объединения ячеек. Границы и содержимое ячеек в последующих строках будут отображаться как обычно, то есть пересекаться с расширенной ячейкой.

	b
a	
c	

---

```

\begin{tabular}{|c|c|} \hline
\multirow{2}{*{a}} & b \\ \hline
c & \\ \hline
\end{tabular}

```

---

Чтобы отображение было корректным, нужно позаботиться о том, чтобы соответствующие ячейки были пустыми, а горизонтальные линии не пересекали эту ячейку:

a	b
	c

---

```
\begin{tabular}{|c|c|} \hline
\multirow{2}{*}{a} & b\\ \cline{2-2}
& c\\ \hline
\end{tabular}
```

---

При необходимости, мы сможем изобразить даже что-то подобное.

	A		
A	A		A
		A	
	A		
		A	
A			

---

```
\begin{tabular}{|c|c|c|c|c|} \hline
\multirow{4}{*}{A} & \multicolumn{4}{c|}{A} \\ \cline{2-5}
& \multicolumn{2}{c|}{A} & \multirow{2}{*}{A} & \multirow{4}{*}{A} \\ \cline{2-3}
& \multirow{2}{*}{A} & A & & \\ \cline{1-4}
& & \multicolumn{2}{c|}{A} & \\ \hline
\multicolumn{4}{|c|}{A} & \\ \hline
\end{tabular}
```

---

Чтобы сделать ячейку имеющую размерность большую одного и по горизонтали, и по вертикали можно вложить команду `\multirow` в команду `\multicolumn`.

Обратите внимание, что центральная ячейка имеет размеры  $2 \times 2$ .

A	A	A	A
A	A		A
A			A
A	A	A	A

---

```

\begin{tabular}{|c|c|c|c|} \hline
A & A & A & A \\ \hline
A & \multicolumn{2}{c|}{\multirow{2}{*{A}}} & A \\ \cline{1-1}\cline{4-4}
A & \multicolumn{2}{c|}{} & A \\ \hline
A & A & A & A \\ \hline
\end{tabular}

```

---

Рассмотрим более содержательный пример. Теперь мы можем упрощенно изобразить часть таблицы Менделеева.

Периоды	Ряды	Группы элементов										энерг. уровни
		I	II	III	IV	V	VI	VII	VIII			
		а б	а б	а б	а б	а б	а б	а б	б			
1	1	H									He	k
2	2	Li	Be	B	C	N	O	F			Ne	l k
3	3	Na	Mg	Al	Si	P	S	Cl			Ar	m l k
4	4	K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	n m l k
	5	Cu	Zn	Ga	Ge	As	Se	Br				Kr
.....												

```

\begin{tabular}{|c|c||*{11}{c|}|c|} \hline
\multirow{3}{*{Периоды}} & \multirow{3}{*{Ряды}} & \multicolumn{11}{c|}{Группы элементов} & \multirow{3}{*{энерг. уровни}} \\
& & I & II & III & IV & V & VI & VII & \multicolumn{2}{c}{VIII} & \\
& & а б & а б & а б & а б & а б & а б & а б & б & а & \\
& I & II & III & IV & V & VI & VII & \multicolumn{4}{c}{VIII} & \\
& \multicolumn{4}{c}{а\hfill 6} & \multicolumn{4}{c}{а\hfill 6} & \multicolumn{4}{c}{а\hfill 6} & \\
& \multicolumn{4}{c}{а\hfill 6} & \multicolumn{4}{c}{а\hfill 6} & \multicolumn{4}{c}{а\hfill 6} & \\
& \multicolumn{3}{c}{а} & \multicolumn{3}{c}{б} & \multicolumn{3}{c}{а} & \multicolumn{3}{c}{б} & \\
1 & 1 & H & & & & & & & & He & k \\
& \multicolumn{3}{c}{ } & \multicolumn{3}{c}{ } & \multicolumn{3}{c}{ } & \multicolumn{3}{c}{ } & \\
2 & 2 & Li & Be & B & C & N & O & F & & Ne & l k \\
& \multicolumn{3}{c}{ } & \multicolumn{3}{c}{ } & \multicolumn{3}{c}{ } & \multicolumn{3}{c}{ } & 

```



```

3 & 3 & Na & Mg & Al & Si & P & S & Cl
& \multicolumn{3}{c|}{ } & Ar & m l k \\ \hline
\multirow{2}{*}{4} & 4 & K & Ca & Sc & Ti & V & Cr & Mn
& Fe & Co & Ni & & n m l k \\ \cline{2-14}
& 5 & Cu & Zn & Ga & Ge & As & Se & Br
& \multicolumn{3}{c|}{ } & Kr & n m l k \\ \hline
\multicolumn{14}{|c|}{\dotfill} \\
\end{tabular}

```

Проиллюстрируем еще раз использование пакета `hhline` и наведем порядок с линиями в приведенной выше таблице.

Периоды	Ряды	Группы элементов										энерг. уровни
		I	II	III	IV	V	VI	VII	VIII			
		а б	а б	а б	а б	а б	а б	а б	б			
1	1	H									He	k
2	2	Li	Be	B	C	N	O	F			Ne	l k
3	3	Na	Mg	Al	Si	P	S	Cl			Ar	m l k
4	4	K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	n m l k
	5	Cu	Zn	Ga	Ge	As	Se	Br				Kr
.....												

```

\begin{tabular}{|c|c||*{11}{c|}|c|} \hline
\multirow{3}{*}{Периоды} & \multirow{3}{*}{Ряды}
& \multicolumn{11}{c|}{Группы элементов}
& \multirow{3}{*}{энерг. уровни} \\ \hhline{~~*{11}{-}||}
& & I & II & III & IV & V & VI & VII
& \multicolumn{4}{c|}{VIII} & \\ \hhline{~~*{11}{-}||}
& & a\hfill 6 & a\hfill 6 & a\hfill 6
& a\hfill 6 & a\hfill 6 & a\hfill 6 & a\hfill 6
& \multicolumn{3}{c|}{6} & a & \\ \hhline{|=##{11}{=}\#|=}
1 & 1 & H & & & & & & & & & & He & k \\ \hhline{--||*{11}{-}||-}
2 & 2 & Li & Be & B & C & N & O & F
& \multicolumn{3}{c|}{ } & Ne & l k \\ \hhline{--||*{11}{-}||-}
3 & 3 & Na & Mg & Al & Si & P & S & Cl

```

```

& \multicolumn{3}{c|}{ } & Ar & m l k \\ \hline{--||*{11}{-}||-}
\multirow{2}*{4} & 4 & K & Ca & Sc & Ti & V & Cr & Mn
& Fe & Co & Ni & & n m l k \\ \hline{~-||*{11}{-}||-}
& 5 & Cu & Zn & Ga & Ge & As & Se & Br
& \multicolumn{3}{c|}{ } & Kr & n m l k \\ \hline{--*{11}{-}-}
\multicolumn{14}{c|}{\dotfill} \\
\end{tabular}

```

---

### 3.3 Пакет array

При подключении пакета **array** добавляется целый ряд новых возможностей.

Новые типы колонок **m{}** и **b{}** аналогичны типу **p{}** за исключением вертикального выравнивания в ячейках такой колонки. Если в колонках типа **p{}** выравнивание происходит по верхнему краю текста, то в колонках типа **m{}** текст выравнивается по центру, а в колонках типа **b{}** — по нижнему краю текста. В фигурных скобках указывается ширина колонки.

Выравнивание происходит не так, как можно подумать по словесному описанию выше. На самом деле все ячейки таблицы в одной строке имеют одну и ту же точку выравнивания, но текст в каждой колонке фиксируется к этой точке разным местом. То есть, например, первая строка колонки типа **p** будет находиться по вертикали на том же уровне, как и самая нижняя строка колонки типа **b**. В колонке типа **m** на том же уровне будет находиться середина текста по вертикали.

Колонка типа <b>p</b> . Текст в этой колонке выравнивается по вертикали по первой строке текста. Потому этот текст окажется ниже, чем текст в соседних ячейках.	Колонка типа <b>m</b> . Здесь по центру текста по вертикали.	Колонка типа <b>b</b> . Текст в этой колонке выравнивается по вертикали по самой нижней строке текста.
--	---	---

---

```
\begin{tabular}{|p{3.5cm}|m{3.5cm}|b{3.5cm}|} \hline
```

Колонка типа `{\bf p}`.

Текст в этой колонке выравнивается по вертикали по первой строке текста.

Потому этот текст окажется ниже, чем текст в соседних ячейках.

& Колонка типа `{\bf m}`.

Здесь по центру текста по вертикали.

& Колонка типа `{\bf b}`. Текст в этой колонке выравнивается по вертикали по самой нижней строке текста.\\ \hline

```
\end{tabular}
```

---

Много способов использования можно найти для возможности добавлять в начало и в конец каждой ячейки таблицы наперед заданный текст. Описания `<{ }` и `>{ }` позволяют вставлять любой текст в начале и/или в конце каждой ячейки: `>{ }` вставляется перед типом требуемой колонки, а `<{ }` после него. В фигурных скобках указывается нужный текст.

Например, эту возможность можно использовать так.

В этой колонке будет некоторый текст. И в каждой ячейке автоматически будет установлена «красная строка» в начале каждого абзаца.	5см	Температура -10 градусов
А во второй колонке таблицы к каждому значению добавляется постфикс «см».	38см	Температура 28 градусов
В третьей колонке нам достаточно указать число, чтобы получить правильное предложение	112см	Температура 36,6 градусов

---

```
\begin{tabular}{|>{\parindent=0.75cm}p{4cm}|c<{см}|>{Температура }c<{ градусов}|} \hline
```

В этой колонке будет некоторый текст.  
И в каждой ячейке автоматически будет установлена  
<<красная строка>> в начале каждого абзаца.  
& 5 & -10 \\ \hline  
А во второй колонке таблицы к каждому значению  
добавляется постфикс <<см>>.  
& 38 & 28 \\ \hline  
В третьей колонке нам достаточно указать число,  
чтобы получить правильное предложение  
& 112 & 15 \\ \hline  
\end{tabular}

---

Другим способом использования вставки текста в ячейку может служить задание единого  
стиля текста для колонки таблицы. Например так:

Это обычный столбец для сравнения. Тут текст как текст.	<b>В этом столб- це весь текст по умолча- нию выделен жирным.</b>	<i>А тут все бу- дет курсивом. Аналогично можно было бы использо- вать и другие настройки для ячейки.</i>
Текст	<b>Текст</b>	<i>Текст</i>
Текст	<b>Текст</b>	<i>Текст</i>

---

```
\begin{tabular}[t]{|p{3cm}|>{\bfseries}p{3cm}|>{\itshape}p{3cm}|}\hline
Это обычный столбец для сравнения. Тут текст как текст. &
В этом столбце весь текст по умолчанию выделен жирным. &
А тут все будет курсивом. Аналогично можно было бы
использовать и другие настройки для ячейки.\\ \hline
Текст & Текст & Текст \\ \hline
Текст & Текст & Текст \\ \hline
\end{tabular}
```

---

Еще одно нововведение — опция `!{}`. Она действует аналогично опции `@{}`, но не вызывает подавления пробелов по бокам от разделяющего ячейки символа.

Например:

1x2	x	3
4x5	x	6
7x8	x	9

---

```
\begin{tabular}{|c@{x}c!{x}c|} \hline
1 & 2 & 3 \\ \hline
4 & 5 & 6 \\ \hline
7 & 8 & 9 \\ \hline
\end{tabular}
```

---

Кроме прочего, у нас есть возможность управлять высотой строки в таблице. С этой целью с помощью команды `\setlength{\extrarowheight}{...}` нужно задать значение переменной `\extrarowheight`. Указанное расстояние будет добавляться к высоте каждой строки таблицы.

$t$	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	$\pi$	$\frac{5\pi}{4}$	$\frac{3\pi}{2}$	$\frac{7\pi}{4}$	$2\pi$
$\cos t$	1	$\frac{\sqrt{2}}{2}$	0	$-\frac{\sqrt{2}}{2}$	-1	$-\frac{\sqrt{2}}{2}$	0	$\frac{\sqrt{2}}{2}$	1
$\sin t$	0	$\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2}$	0	$-\frac{\sqrt{2}}{2}$	-1	$-\frac{\sqrt{2}}{2}$	0

Вы помните наш пример с таблицей синусов и косинусов. Горизонтальные линии здесь буквально накладываются на содержимое ячеек. Но теперь это легко исправить.

$t$	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	$\pi$	$\frac{5\pi}{4}$	$\frac{3\pi}{2}$	$\frac{7\pi}{4}$	$2\pi$
$\cos t$	1	$\frac{\sqrt{2}}{2}$	0	$-\frac{\sqrt{2}}{2}$	-1	$-\frac{\sqrt{2}}{2}$	0	$\frac{\sqrt{2}}{2}$	1
$\sin t$	0	$\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2}$	0	$-\frac{\sqrt{2}}{2}$	-1	$-\frac{\sqrt{2}}{2}$	0

---

```
{\setlength{\extrarowheight}{5pt}
\begin{tabular}{|*{10}{m{20pt}}|} \hline
$t$ & 0 & $\frac{\pi}{4}$ & & & & & & & & \\
$\frac{\pi}{2}$ & & $\frac{3\pi}{4}$ & & $\pi$ & & & & & & \\
$\frac{5\pi}{4}$ & & $\frac{3\pi}{2}$ & & $\frac{7\pi}{4}$ & & $2\pi$ & & & & \\
\\ \hline
```

```

 $\cos{t}$  & 1 &  $\frac{\sqrt{2}}{2}$  & 0 &  $-\frac{\sqrt{2}}{2}$  &
 $-1$  &  $-\frac{\sqrt{2}}{2}$  & 0 &  $\frac{\sqrt{2}}{2}$  & 1
\\ \hline
 $\sin{t}$  & 0 &  $\frac{\sqrt{2}}{2}$  & 1 &  $\frac{\sqrt{2}}{2}$  & 0 &
 $-\frac{\sqrt{2}}{2}$  &  $-1$  &  $-\frac{\sqrt{2}}{2}$  & 0 \\ \hline
\end{tabular}
}

```

---

Вернемся к одному из прошлых примеров:

Ехал Грека через 

реку.
руку.
рак.

 Видит Грека — в реке 

реку.
руку.
рак.

 Сунул Грека 

реку
руку
рак

 в 

реку.
руку.
рак.

Как вы помните, выравнивание таблицы по вертикали отклоняется от предполагаемого совпадения первой/последней строки таблицы с текстом вокруг. Чтобы избежать этого искажения, вместо команды `\hline` можно использовать `\firsthline` для первой горизонтальной линии в таблице и `\lasthline` для последней горизонтальной линии.

Ехал Грека через 

реку.
руку.
рак.

 Видит Грека — в реке 

реку.
руку.
рак.

 Сунул Грека 

реку
руку
рак

 в 

реку.
руку.
рак.

---

```

Ехал Грека через
\begin{tabular}[t]{|l|} \firsthline
реку. \\
руку. \\
рак. \\ \hline
\end{tabular}
Видит Грека --- в реке
\begin{tabular}[b]{|l|} \hline
реку. \\

```

```

реку. \\
пак. \\ \lasthline
\end{tabular}
Сунул Грека
\begin{tabular}[c]{|l|} \hline
реку \\
реку \\
пак \\ \hline
\end{tabular}
в
\begin{tabular}[t]{|l|} \firsthline
реку. \\
реку. \\
пак. \\ \hline
\end{tabular}

```

---

### 3.4 Пакет dcolumn

Если вам требуется вставить в документ таблицу с данными, включающими нецелые числа, встает вопрос понятного и красивого их отображения. Чтобы решить эту проблему, в пакете **dcolumn** вводится новый тип колонки таблицы:

**D**{<разделитель в tex>}{<разделитель в dvi>}{<знаков после запятой>}.

Здесь первый параметр указывает, каким символом будет отделяться дробная часть в записи чисел в ячейках в tex-файле, а второй параметр указывает, на что нужно заменить символ из первого параметра при формировании выходного файла. Первый и второй параметр могут совпадать, если ничего заменять не нужно, но могут и отличаться. Второй параметр может принимать любой математический код.

Положительное число в третьем параметре инструкции **D**{ }{ }{ }. указывает, под какое количество знаков после запятой нужно оставить место справа в ячейке таблицы. Если количество знаков после запятой во введенном числе превосходит этот параметр, дробная часть числа может выйти за правую границу ячейки. Пространство слева от запятой определяется автоматически по максимальной целой части числа в этой колонке.

Если мы не знаем заранее, сколько знаков после запятой может встретиться, можно указать отрицательное число в качестве третьего параметра. В этом случае десятичная точка будет центрирована по горизонтали в данной ячейке, а справа и слева от нее будет оставлено одинаковое пространство, достаточное для вмещения как наибольшей целой части из всех чисел в этой колонке, так и наибольшей дробной части.

В примере ниже четыре раза повторен один и тот же столбец чисел. Первые две колонки имеют тип **r** и **c** соответственно, числа в них выровнены без учета десятичной точки.

Третья колонка не меняет знак десятичной точки и предполагает два знака после запятой. Можно видеть, что большая дробная часть не уместится в колонке такого типа. Четвертая колонка присутствует в примере только для вмещения этой вылезавшей дробной части.

Наконец, четвертый столбец чисел расположен в колонке, которая заменяет знак десятичной точки на запятую и допускает любой размер дробной части.

Последние четыре строки демонстрируют обработку содержимого ячейки, которая не содержит символа указанного в качестве первого параметра **D{ }{ }{ }**. Число 256 отображается корректно. В числе 4,25 использован неверный символ отделения дробной части и все число воспринимается в числовых колонках как целая часть. А «слово» вовсе не отображается в числовых колонках, так как не является математическим выражением. Мы можем сделать его видимым, если воспользуемся командой типа **\text{ }** или изменим тип ячейки с помощью команды **\multicolumn{ }{ }{ }**.

2.5	2.5	2.5		2,5
4.25	4.25	4.25		4,25
8.125	8.125	8.125		8,125
16.625	16.625	16.625		16,625
32.3125	32.3125	32.3125		32,3125
64.15625	64.15625	64.15625		64,15625
128.78125	128.78125	128.78125		128,78125
256	256	256		256
4,25	4,25	4,25		4,25
слово	слово	слово		слово

---

```

\begin{tabular}{|r|c|D{.}{.}{2}|p{1cm}|D{.}{,}{-1}|} \hline
2.5 & 2.5 & 2.5 & & 2.5 \\
4.25 & 4.25 & 4.25 & & 4.25 \\
8.125 & 8.125 & 8.125 & & 8.125 \\
16.625 & 16.625 & 16.625 & & 16.625 \\
32.3125 & 32.3125 & 32.3125 & & 32.3125 \\
64.15625 & 64.15625 & 64.15625 & & 64.15625 \\
128.78125 & 128.78125 & 128.78125 & & 128.78125 \\
\hline \hline
256 & 256 & 256 & & 256 \\
4,25 & 4,25 & 4,25 & & 4,25

```



```

слово & слово & слово && слово \\
& & \text{слово} && \multicolumn{1}{c|}{слово} \\ \hline
\end{tabular}

```

---

Поскольку вторым параметром директивы **D{ }{ }** является любое математическое выражение, это можно использовать для получения более необычных эффектов.

2,5	$2 + 0,5$	2 целых и 5 после запятой
4,25	$4 + 0,25$	4 целых и 25 после запятой
8,125	$8 + 0,125$	8 целых и 125 после запятой
16,625	$16 + 0,625$	16 целых и 625 после запятой
32,3125	$32 + 0,3125$	32 целых и 3125 после запятой

---

```

\begin{tabular}{|D{.}{,}{4}|D{.}{~+~0,}{4}|
D{.}{\text{ целых и }}{18}<\text{ после запятой}}|}\hline
2.5 & 2.5 & 2.5 \\
4.25 & 4.25 & 4.25 \\
8.125 & 8.125 & 8.125 \\
16.625 & 16.625 & 16.625 \\
32.3125 & 32.3125 & 32.3125 \\
\hline
\end{tabular}

```

---

Еще одна возможность, которую предоставляет пакет **dcolumn**, это создание собственных типов колонки на основе существующих типов и конструкций. Для создания новых типов используется следующая команда:

**\newcolumntype{<символ>}[<аргументов>]{<описание>}**.

Приведем несколько примеров деклараций типов:

```

\newcolumntype{d}[1]{D{.}{\cdot}{#1}}
\newcolumntype{e}{|c|}
\newcolumntype{4}{|p{4cm}}
\newcolumntype{n}{*{3}{r|}}

```

Первая колонка имеет ширину 4 сантиметров.	текст	текст	1·25	1	2	3
Это обеспечивает специально созданный тип колонки «4».	текст	текст	32·3125	4	5	6

---

```

\begin{tabular}{4eed{5}|n} \hline
Первая колонка имеет ширину 4 сантиметров.
& текст & текст & 1.25 & 1 & 2 & 3 \\ \hline
Это обеспечивает специально созданный
тип колонки {\bf<<4>>}.
& текст & текст & 32.3125 & 4 & 5 & 6 \\ \hline
\end{tabular}

```

---

## 4 Заключение

В заключение можно добавить, что это, конечно, далеко не все возможности, которые можно использовать в  $\text{\LaTeX}$  для оформления таблиц, но, надеюсь, это краткое введение позволит удовлетворить большинство потребностей, которые возникают в работе.

## Источники

1. Лекции Ивановой Ольги Александровны по оформлению математических текстов в  $\text{\LaTeX}$ .
2. Е.М. Балдин.  $\text{\LaTeX}$  в России. <http://www.inp.nsk.su/~baldin/LaTeX/lurs-tables.pdf>
3. Документация по  $\text{\LaTeX}$ . Например, <http://www.math.harvard.edu/computing/latex/tetex/>
4.  $\text{\LaTeX}$  для начинающих. <http://latex.tostudents.ru/>
5. Собственные эксперименты для определения точного результата определенных комбинаций команд.

# Содержание

<b>1</b>	<b>Основные средства создания таблиц</b>	<b>2</b>
1.1	Окружение <b>tabbing</b> . . . . .	2
1.2	Окружение <b>tabular</b> . . . . .	3
<b>2</b>	<b>Дополнительные возможности окружения <code>tabular</code></b>	<b>9</b>
<b>3</b>	<b>Расширение возможностей с помощью дополнительных пакетов</b>	<b>19</b>
3.1	Пакет <b>hhline</b> . . . . .	19
3.2	Пакет <b>multirow</b> . . . . .	22
3.3	Пакет <b>array</b> . . . . .	26
3.4	Пакет <b>dcolumn</b> . . . . .	31
<b>4</b>	<b>Заключение</b>	<b>34</b>