

Міністерство освіти і науки України  
ДВНЗ «Прикарпатський національний університет  
імені Василя Стефаника»

О. В. Махней

## Практикум з $\text{\LaTeX}$

Методичні рекомендації  
для самостійної роботи студентів  
спеціальності «прикладна математика»  
вищих навчальних закладів

Івано-Франківськ  
2018

УДК 004.915

ББК 32.973.26-018.2

М36

*Рекомендовано Вченою радою факультету математики та інформатики ДВНЗ «Прикарпатський національний університет імені Василя Стефаника» як навчальний посібник для студентів спеціальності «прикладна математика» (протокол № 4 від 26 грудня 2017 р.).*

#### **Рецензенти:**

*Мазуренко В. В.*, кандидат фізико-математичних наук (Прикарпатський національний університет імені Василя Стефаника);

*Козленко М. І.*, кандидат технічних наук (Прикарпатський національний університет імені Василя Стефаника).

**М36 Махней О. В. Практикум з LaTeX : методичні рекомендації.** Івано-Франківськ : Голіней, 2018. 36 с.

Для самостійної роботи студентів спеціальності «прикладна математика». Може бути корисним для студентів галузей знань «математика і статистика», «інформаційні технології», «природничі науки» та аспірантів і науковців.

УДК 004.915

ББК 32.973.26-018.2

© Махней О. В., 2018

## Зміст

Передмова . . . . .	4
1. Основи L <sup>A</sup> T <sub>E</sub> X . . . . .	5
2. Математичні формули . . . . .	9
3. Форматування тексту . . . . .	15
4. Оформлення тексту в цілому . . . . .	19
5. Таблиці . . . . .	20
6. Блоки . . . . .	22
7. Деякі математичні пакети . . . . .	23
8. Пакети для роботи з таблицями . . . . .	24
9. Використання кольорів . . . . .	26
10. Пакет <code>graphics</code> . . . . .	28
11. Пакет <code>pstricks</code> . . . . .	29
Список рекомендованої літератури . . . . .	35

## Передмова

$\text{T}_\text{E}_\text{X}$  (читається як «тех») — це спеціалізована мова програмування і система комп'ютерної верстки, призначена для підготовки текстів з математичними формулами.  $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$  (читається як «латех») — видавнича система, створена на основі мови  $\text{T}_\text{E}_\text{X}$ . На сьогодні  $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$  є фактично стандартом оформлення наукових публікацій з математики та суміжних галузей науки. На відміну від інших видавничих систем початковим документом  $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$ 'а є текст, написаний з використанням мови розмітки. Лише після обробки цього тексту транслятором можна побачити готовий остаточний документ для друку.

Методичні рекомендації призначено для самостійної роботи студентів спеціальності «прикладна математика» з навчальної дисципліни «Практикум з  $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$ ». На вивчення навчальної дисципліни відводиться 90 годин, у тому числі 30 годин лабораторних занять.

Методичні рекомендації містять мінімально необхідний теоретичний матеріал для виконання лабораторних робіт та вказівки для самостійної роботи з літературою. У кожній темі перш за все рекомендується використовувати джерело, вказане першим для даної теми. Всі теми містять питання для самоперевірки. Крім того, для формування практичних навичок по кожній темі необхідно виконати запропоновані практичні завдання.

## 1. Основи $\LaTeX$

При роботі з видавничою системою  $\LaTeX$  документ для друку створюють у текстовому файлі з розширенням імені «tex», який називають початковим файлом або TEX-файлом. У цьому файлі записується текст документа разом зі спеціальними символами і командами, з допомогою яких системи передаються вказівки стосовно відображення і розміщення тексту. Початковий файл можна редагувати будь-яким текстовим редактором (зокрема, зручно користуватись програмами WinEdt або TeXnicCenter), але важливо, щоб цей файл не містив форматування. Після створення початкового TEX-файлу виконують його трансляцію у DVI-файл або PDF-файл. Останні файли міститимуть готовий для друку документ.

Більшість символів у початковому файлі означають саме те, що буде надруковано. Слова відокремлюються пропусками, але їхня кількість є несуттєвою (кілька пропусків підряд сприймаються як один пропуск). Кінець рядка також сприймається як пропуск. Порожній рядок означає кінець абзацу, кілька порожніх рядків не дозволяють створити додатковий вертикальний відступ між абзацами. Початковий файл не має містити переносів у словах ( $\LaTeX$  при необхідності робить їх автоматично). При трансляції всі абзаци автоматично вирівнюються по ширині і мають абзацний відступ.

Наступні десять символів є символами керування:

{ } \$ % # & \_ ^ ~ \

Якщо їх вжити в тексті не за призначенням, то може виникнути повідомлення про помилку або замість цих символів друкуватиметься щось зовсім інше. Символ " також може викликати проблеми, зокрема після нього зникає пропуск.

Символ % означає початок коментаря, тобто всі символи, починаючи з символу % до кінця рядка ігноруються. Ігнорується також кінець рядка.

Команди бувають двох видів. Команда першого виду містить знак `\` і один символ, що не є буквою, після нього. Наприклад, команда `\%` друкує знак `%`. Аналогічно можна відображати також деякі інші зі спеціальних символів. Команда другого виду складається з символу `\` і послідовності латинських букв після нього. Великі і малі букви в записі команд розрізняються. Пропуски після команд другого виду ігноруються (але визначають кінець імені команди). Для відображення такого пропуску потрібно після імені команди поставити пару фігурних дужок `{}`. Прикладом команди другого виду є команда `\bf`, яка перемикає шрифт тексту на напівжирний. Звратною до неї є команда `\rm`.

Тут і далі у прикладах зліва наводиться запис з допомогою команд `LaTeX`'а, а справа — результат:

У групі <code>\bf ПМ-21\rm{}</code> є <code>45 \% ді%</code> це приклад вчат і <code>55 \% хлопців</code> .	У групі <b>ПМ-21</b> є 45 % дів- чат і 55 % хлопців.
---	---

Більшість команд, зокрема всі розглянуті вище, не потребує аргументів. Проте, є багато команд, для яких потрібно вказувати додаткові аргументи. Кількість і порядок аргументів залежить від команди. Кожен обов'язковий аргумент вказується в окремих фігурних дужках після назви команди. Якщо обов'язковий аргумент команди складається лише з одного символу, то фігурні дужки можна пропускати. Кожен необов'язковий аргумент задається в окремих квадратних дужках. Залежно від команди необов'язкові аргументи можуть знаходитись перед чи між обов'язковими. Якщо необов'язковий аргумент пропускається, то пропускаються і відповідні дужки.

Прикладом команд з обов'язковими аргументами є команди `\begin{ім'я_оточення}` і `\end{ім'я_оточення}`, які розпочинають і завершують оточення з заданим ім'ям. Ім'ям оточення є послідовність латинських букв. Вмістом оточення на-

зивають все те, що знаходиться між початком і кінцем оточення. Наприклад, оточення `center`:

```
\begin{center} текст \end{center}
```

використовується для центрування тексту по горизонталі.

Деякі назви команд другого виду та імена оточень завершуються символом `*`. Це означає, що дія команди чи оточення нагадає дію відповідної команди або оточення без `*`, але дечим відрізняється.

Фігурні дужки не генерують ніякого тексту, а обмежують групу всередині документа. Фігурній дужці, що відкриває групу (`{`), обов'язково має відповідати фігурна дужка, яка її закриває (`}`). Деякі команди, такі як `\bf` і `\rm`, є командами-деклараціями, і дія такої команди, заданої всередині групи, обмежується групою. Кожне оточення також є групою.

Початковий файл документа має починатись командою `\documentclass`, яка визначає стиль документа. Ця команда розпочинає преамбулу документа, в якій можуть підключатись додаткові пакети і застосовуватись команди і параметри, які діятимуть на весь документ. Жодна буква або команда, яка генерує букву, у преамбулі зустрінись не повинна. Власне текст документа оформлюється як вміст оточення `document`. Все записане після завершення цього оточення ігнорується.

Для створення документів на українській мові при роботі в середовищі Windows рекомендується використовувати наступну структуру початкового файлу:

```
\documentclass{article}
\usepackage[T2A]{fontenc}
\usepackage[cp1251]{inputenc}
\usepackage[english,ukrainian]{babel}
...
\begin{document}
...
\end{document}
```

Параметри довжини записуються аналогічно командам — після `\` ставиться послідовність латинських букв. Параметру довжини присвоюють нове значення з допомогою знака `=`. Довжина має записуватись з використанням  $\TeX$ 'івських одиниць довжини, як-от: міліметри (mm), сантиметри (cm), пункти (pt), дюйми (in) тощо. Наприклад, `\parindent=4.5mm` задає величину абзацного відступу, рівною 4,5 мм.

$\LaTeX$  надає можливість формувати такі посилання на окремі сторінки, номери формул чи інших об'єктів, щоб програма сама підставляла замість номера потрібне число. Для цього потрібно спочатку помітити командою `\label{мітка}` потрібне місце в документі, де *мітка* — деяка послідовність латинських букв, цифр і символів пунктуації. Для посилання на номер потрібної сторінки використовують команду `\pageref{мітка}` з тією самою *міткою*. Для посилання на номер формули, елемента нумерованого списку чи іншого нумерованого об'єкта використовують команду `\ref{мітка}`. Для коректного відображення номерів потрібно після останньої зміни міток двічі виконувати трансляцію в DVI-документ чи PDF-документ.

Під час трансляції з  $\TeX$ -файлу в DVI-документ чи PDF-документ на екрані в окремому вікні відображаються повідомлення про стан трансляції і позаштатні ситуації під час трансляції, однак ці повідомлення процес трансляції не переривають. Ці самі повідомлення заносяться також в log-файл, який після трансляції за бажання можна прочитати.

Якщо під час трансляції у  $\TeX$ -файлі зустрінеться синтаксична помилка, то трансляція перерветься і на екрані в окремому вікні з'явиться повідомлення про помилку, яке після знака оклику містить назву помилки (англійською мовою), а в наступному рядку після символів «1.» знаходиться номер рядка, який викликав помилку, а також його вміст. Потрібно мати на увазі, що до помилки міг призвести попередній помилковий рядок, а не той, на який вказує  $\TeX$ .



TeX може працювати в одному з наступних режимів верстки документа: у горизонтальному, в який програма переходить після першої букви, що розпочинає абзац, у вертикальному, в якому програма перебуває після завершення абзацу або перед першим абзацом, і в математичному, в якому вона перебуває під час верстки математичної формули. У наступному розділі йтиметься про математичний режим роботи.

**Рекомендована література:** [6, с. 13–44], [2, с. 14–26], [4, с. 5–11].

### **Питання для самоперевірки.**

1. Які основні формати файлів використовуються в LaTeX?
2. З чого складається загальна структура початкового файлу?
3. Як створити групу і яке її призначення?
4. Як створити коментар?
5. Що таке команда? Наведіть приклади команд.

**Практичне завдання:** створіть засобами системи LaTeX, скориставшись структурою початкового файлу зі сторінки 7, текст з наступного абзацу і питання для самоперевірки.

До книгарні для продажу надійшли підручники з фізики і математики, причому лише 20 % з них становили підручники з фізики.

## **2. Математичні формули**

Математичні формули можуть бути внутрішньотекстовими або виключними. На відміну від внутрішньотекстових формул виключні формули розміщуються в окремих рядках. Внутрішньотекстова формула має починатись з символу  $\$$  і завершуватись цим символом. Замість символу  $\$$  для початку формули можна використовувати команду  $\backslash($ , а для її завершення —  $\backslash)$ . Виключна формула має починатись з двох символів  $$$$  і завершуватись символами  $$$$ . Альтернативне оформлення виключної формули передбачає використання команди  $\left[$  на початку формули і команди  $\right]$  в її кінці.

Для створення виключної формули зі стандартним числовим номером її потрібно оформлювати як вміст оточення `equation` без символів `$$` чи команд `\[`, `\]`. Для можливості створення майбутнього посилання на номер такої формули командами `\ref{мітка}` або `\eqref{мітка}` всередині оточення `equation` потрібно створити мітку командою `\label{мітка}`.

Для створення виключної формули з нестандартним номером чи позначенням її потрібно оформлювати з допомогою символів `$$` на початку і в кінці, а сам номер (позначення) задавати з допомогою команди `\eqno` наприкінці формули, наприклад, `\eqno (*)`.

Всередині формули всі пропуски, кінці рядків і українські букви ігноруються, а порожніх рядків там бути не повинно. Пропуски у формулах `TeX` ставить автоматично відповідно до правил математики. Всі латинські букви відображаються математичним курсивом, тобто курсивом зі збільшеними інтервалами між буквами. Символи арифметичних операцій і логічних відношень (+, -, /, =, <, >), цифри, знаки пунктуації, круглі і квадратні дужки, а також деякі інші символи набираються безпосередньо. Всередині формули можна робити текстові вставки командою `\mbox{текст}`. Ці вставки можуть містити українські букви.

Для створення нижнього індексу ставлять знак `_`, після якого у фігурних дужках записують сам індекс. Для верхнього індексу (показника степеня) використовують знак `^`, після якого у фігурних дужках вказують сам індекс (показник степеня). Якщо верхній чи нижній індекс складається лише з одного символу, то фігурні дужки можна пропускати. Для створення штриха після довільного символу використовують прямий апостроф `'`. Для створення кількох штрихів цей апостроф наводять потрібну кількість разів. Робота з індексами і штрихами ілюструється наступним прикладом, в якому також показано, як зміщувати індекси.

$$\text{\$f}_{ij}=a''_{ij}+b_i^2+c_j^j-c_{i+1}^{i+j} \quad \left| \quad f_{ij} = a''_{ij} + b_i^2 + c_j^j - c_{i+1}^{i+j} \right.$$

Дроби, які мають бути записані через косу риску, набираються безпосередньо. Однак частіше використовуються дроби, в яких чисельник розташований над знаменником. Для їх створення використовують команду

$$\text{\frac{чисельник}{знаменник}}.$$

Для створення радикалів використовують команду

$$\text{\sqrt[показник]{вираз}}.$$

$$\text{\frac{x^2+y^2}{x^3+y^3} + \frac{1}{3}\sqrt{2+\sqrt[4]{3}}} \quad \left| \quad \frac{x^2+y^2}{x^3+y^3} + \frac{1}{3}\sqrt{2+\sqrt[4]{3}} \right.$$

Існує велика кількість різних математичних символів, які можна використовувати у формулах. Всі ці символи створюють з допомогою спеціальних команд. Наприклад, команда `\alpha` дозволяє отримати грецьку букву  $\alpha$ , команда `\rightarrow` — стрілку  $\Rightarrow$ , команда `\forall` — символ  $\forall$ , команда `\sin` —  $\sin$ . Ці символи також є на панелях інструментів у програмах WinEdt і TeXnicCenter. При виборі такого символу в позицію курсора вставляється відповідна команда. Деякі символи можуть мати нижні і верхні межі, наприклад,

$$\text{\sum_{k=1}^{\infty} a_k} \quad \left| \quad \sum_{k=1}^{\infty} a_k \right.$$

У внутрішньотекстових формулах та для знаків інтегралів межі відображаються збоку, а не знизу і зверху. Для відображення меж знизу і зверху перед ними потрібно вказати команду `\limits`.

Круглі, квадратні дужки та знак модуля набираються безпосередньо з клавіатури. Для фігурних дужок передбачені

команди `\{ i \}`. Подвійні вертикальні лінії (знак норми чи матриці) створюються командою `\|`. Є також команди для створення всіляких нестандартних типів дужок.

Якщо дужки чи команди для їхнього створення використовувати у формулі безпосередньо, то такі дужки матимуть розмір звичайних символів. Для збільшення розміру дужок відповідно до розміру фрагмента формули в дужках перед дужкою, що відкривається, використовують команду `\left`, а перед дужкою, яка закривається, — команду `\right`, наприклад, `\left( ... \right)`. Кожній команді `\left` має відповідати своя команда `\right` і навпаки. Якщо однієї з дужок не потрібно, то замість цієї дужки після команди `\left` або `\right` потрібно поставити крапку.

$$\int_a^b x^2 dx = \left. \frac{x^3}{3} \right|_a^b$$

Для самостійного вибору розміру дужок замість команд `\left i \right` використовують команди `\big`, `\Big`, `\bigg` і `\Bigg`, кожна наступна з яких створює дужку більшого розміру, ніж попередня.

$$\big||x-2|-|x+1|\big|$$

Для розміщення деяких знаків над символами передбачено спеціальні команди, окремі з яких для прикладу наведені в наступній таблиці для букви *a*.

<code>\hat a</code>	$\hat{a}$	<code>\ddot a</code>	$\ddot{a}$
<code>\tilde a</code>	$\tilde{a}$	<code>\bar a</code>	$\bar{a}$
<code>\dot a</code>	$\dot{a}$	<code>\vec a</code>	$\vec{a}$

Всі ці команди дозволяють ставити по одному значку над одним символом, проте є кілька команд, які працюють з цілими фрагментами формул. Зокрема, команда

`\overline{фрагмент}` утворює горизонтальну лінію над фрагментом формули.

Команди

$$\overbrace{\text{фрагмент}}^{\text{напис}},$$

$$\underbrace{\text{фрагмент}}_{\text{напис}}$$

над (під) фрагментом формули друкують горизонтальну фігурну дужку, а над (під) нею додають напис.

Команда `\stackrel{up}{line}` друкує вираз *line* в рядку, а вираз *up* — над ним.

$$\text{\$}\stackrel{f}{\to}\text{B}\$ \quad \Big| \quad A \xrightarrow{f} B$$

Для створення матриць використовують оточення `array`, яке має обов'язковий аргумент — преамбулу матриці, яка складається з послідовності букв, що описують стовпці матриці. Кожна буква відповідає одному стовпцю. У преамбулі матриці буква *s* означає центрування відповідного стовпця, буква *l* — вирівнювання стовпця по лівому краю, а буква *r* — по правому краю. Елементи в рядку матриці відокремлюються символами `&`, а рядки — командами `\\`.

$$\text{\$}\left(\begin{array}{ccc} a_{11} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{nn} \end{array}\right)\text{\$}$$

Хоча всі відступи у формулах `TeX` створює автоматично, інколи виникає потреба змінити окремі проміжки. Для створення додаткових горизонтальних відступів використовують команди `\,`, `\:`, `\;` (малі відступи від меншого до більшого), `\quad` (відступ шириною в одну букву *M*), `\qquad` (вдвічі більший відступ). Натомість команда `!\` зменшує відступ настільки, наскільки його збільшує команда `\,`.

Параметр `\arraycolsep` дозволяє змінити відстань між стовпцями матриці. Цей параметр за замовчуванням дорівнює 5 пунктам, нове значення йому потрібно присвоювати перед початком матриці. Команда `\` може мати необов'язковий аргумент, який визначає величину додаткового відступу між рядками.

Математична формула може набиратись одним з наступних стилів: стилем виключної формули, текстовим стилем, стилем для індексів і стилем для індексів до індексів. Виключна і внутрішньотекстова формули починають набиратись стилем виключної формули і текстовим стилем відповідно. У цих стилях розмір символів той самий, але індекси у виключному стилі піднімаються (опускаються) вище (нижче), а межі операцій записуються не збоку, а зверху і знизу. У стилі для індексів використовуються дрібніші шрифти і пропускаються інтервали навколо знаків операцій, а в стилі для індексів до індексів застосовуються ще дрібніші шрифти. Чисельник і знаменник дробу, створеного командою `\frac`, як і елементи матриці, набираються наступним за порядком стилем. У формулі можна явно вказати її стиль командами `\displaystyle`, `\textstyle`, `\scriptstyle` або `\scriptscriptstyle`.

При верстці  $\TeX$  за необхідності може переносити математичні формули по знакам арифметичних операцій або логічних відношень. Для заборони перенесення формули в певному місці використовують команду `\nobreak`, а для дозволу перенесення у вказаному місці — команду `\allowbreak`. Команда `*` дозволяє переносити формулу і при її перенесенні в кінці рядка друкує символ  $\times$ .

**Рекомендована література:** [6, с. 45–54, 59–80, 88–97], [2, с. 133–162], [5, с. 129–156], [8], [4, с. 31–40].

#### **Питання для самоперевірки.**

1. Чим відрізняється оформлення внутрішньотекстової формули від виключної?
2. Як забезпечити автоматичну нумерацію формул?

3. Як друкувати степені та індекси?
4. Для чого використовуються команди `\left` і `\right`?
5. Назвіть основні принципи введення матриць.

**Практичне завдання:** виконайте лабораторні роботи № 1 ([http://mif.pu.if.ua/attachments/article/14/tex\\_lab1.pdf](http://mif.pu.if.ua/attachments/article/14/tex_lab1.pdf)) і № 2 ([http://mif.pu.if.ua/attachments/article/14/tex\\_lab2.pdf](http://mif.pu.if.ua/attachments/article/14/tex_lab2.pdf)).

### 3. Форматування тексту

Усі засоби для форматування тексту, про які йтиметься у цьому розділі, стосуються горизонтального режиму, якщо не буде сказано протилежне.

Для створення спецсимволів використовують спеціальні послідовності символів, які називають лігатурами. Їх разом з деякими командами наведено в непарних стовпцях наступної таблиці. У парних стовпцях знаходяться самі спецсимволи.

--	—	<<	«	‘	“	\ldots	...
---	—	>>	»	’	”	\S	§

Команда `\symbol{c}` друкує символ з кодом  $c$ . Команда `\-` ставиться всередині слова і вказує місце можливого перенесення цього слова, використовується для перенесення слів лише так, як це потрібно автору.

Символ `~` утворює нерозривний пропуск. Для створення горизонтального проміжку спеціальної довжини використовують команду `\hspace{x plus y minus z}`, де  $x$ ,  $y$ ,  $z$  задаються в Т<sub>Е</sub>X'івських одиницях вимірювання. Цей пропуск матиме довжину  $x$ , яка може збільшуватись максимум на  $y$  і зменшуватись максимум на  $z$ . Пропуск матиме довжину  $x$ , якщо застосувати спрощений формат команди `\hspace{x}`.

Для зміни гарнітури шрифту використовують наступні команди: `\bf` — напівжирний шрифт, `\it` — курсив, `\rm` — звичайний шрифт, `\sl` — похилий шрифт, `\tt` — моноширинний шрифт, `\sf` — рублений шрифт, `\sc` — зменшені великі букви. Проте цими командами неможливо зробити, наприклад,

одночасно напівжирний і курсивний шрифт. Існують також інші команди: `\bfseries` і `\mdseries`, `\itshape`, `\slshape`, `\scshape` і `\upshape`, `\ttfamily`, `\sffamily` і `\rmfamily`. Вибираючи по одній команді з кожної з цих трьох груп, можна зробити, наприклад, моноширинний курсивний напівжирний шрифт: `\ttfamily \itshape \bfseries`. Всі ці команди зручно застосовувати всередині групи. Є також команди `\textbf{текст}`, `\textit{текст}`, `\textsl{текст}`, `\texttt{текст}` та інші, які змінюють шрифт лише свого аргументу. Такі команди також можна вкладати, наприклад, `\textbf{\textit{слово}}`.

Команда `\underline{текст}` підкреслює *текст*, а команда `\fbox{текст}` дозволяє взяти його у прямокутну рамку.

LaTeX дозволяє працювати з шрифтами відносного розміру. Це особливо зручно, коли потрібно у всьому документі, включаючи заголовки, пропорційно збільшити розмір шрифту. У наступній таблиці наведені команди для зміни розміру шрифту і приклади їх застосування. Кожна наступна команда відносно попередньої збільшує розмір шрифту на 20 %.

Команда	Приклад тексту
<code>\tiny</code>	Найменший
<code>\scriptsize</code>	Маленький (як індекси)
<code>\footnotesize</code>	Маленький (як примітки)
<code>\small</code>	Зменшений
<code>\normalsize</code>	Нормальний
<code>\large</code>	Збільшений
<code>\Large</code>	Великий
<code>\LARGE</code>	Дуже великий
<code>\huge</code>	Величезний
<code>\Huge</code>	Найбільший

Якщо команда зміни розміру шрифту дається для абзацу всередині групи, то група має завершитись після завершення



абзацу (тобто після порожнього рядка, який завершує абзац), інакше міжрядкові інтервали будуть неправильними.

Команда `\footnote[n]{p}` створює для основного тексту внизу сторінки примітку  $p$  з позначенням або номером  $n$ . Якщо необов'язковий аргумент  $n$  пропущено, то примітки нумеруються послідовними натуральними числами.

У вертикальному режимі, тобто між абзацами, може використовуватись команда `\noindent` для того, щоб наступний абзац не мав горизонтального абзацного відступу. У вертикальному режимі можна також керувати вертикальними відступами між абзацами. Команди `\smallskip`, `\medskip` і `\bigskip` створюють відповідно малий, середній і великий відступи. Команда `\vspace{x plus y minus z}` створює вертикальний відступ розміром  $x$ , який може збільшуватись максимум на  $y$  і зменшуватись не більше, ніж на  $z$ .

Для того щоб заборонити розрив сторінки в якомусь місці між абзацами, у вертикальному режимі використовують команду `\noperagebreak`. Для заборони розривів сторінок всередині абзаців призначена команда `\samespage` (її можна давати всередині групи). Звичайно, якщо обмежень надто багато, то L<sup>A</sup>T<sub>E</sub>X не зможе всіх їх виконати.

Для примусового початку нової сторінки у вертикальному режимі призначена команда `\newpage`. Для примусового переходу на новий рядок у горизонтальному режимі використовується команда `\\`.

Звичайні абзаци вирівнюються по ширині. Для центрування абзаців використовується оточення `center`, а для підготовки абзаців, вирівняних лише по лівому чи правому краю — оточення `flushleft` або `flushright` відповідно. Команда `\\` у цих оточеннях дозволяє розбивати абзац на рядки на смак автора документа. Всі ці оточення ставлять додаткові інтервали до і після абзацу. Команди `\centering`, `\raggedleft` і `\raggedright` працюють аналогічно оточенням, але не роблять додаткових відступів до і після абзаців. Їхнє місце —

всередині групи, яка містить абзац.

Для створення списку з маркерами використовують оточення `itemize`. Для друку нумерованих списків призначено оточення `enumerate`. Кожен елемент як одного так і іншого списку повинен починатись з команди `\item`. Елементи нумерованих списків нумеруються автоматично. Списки можуть бути вкладеними. На різних рівнях вкладеності використовуються різні способи нумерації і різні маркери.

<pre>\begin{enumerate} \item Микола. \item Петро. \item Віктор. \end{enumerate}</pre>	<ol style="list-style-type: none"> <li>1. Микола.</li> <li>2. Петро.</li> <li>3. Віктор.</li> </ol>
---	---

Команда `\rule[z]{x}{y}` друкує чорний прямокутник шириною  $x$ , висотою  $y$ , зміщений вгору на висоту  $z$ .

<pre>\rule{7mm}{2mm}</pre>	
----------------------------	---

**Рекомендована література:** [6, с. 98–154], [2, с. 26–62], [5, с. 93–125], [9, с. 20–29, 39–42], [4, с. 17–30].

### Питання для самоперевірки.

1. Чи можна вказати свій спосіб розбиття слова всупереч алгоритму, котрим керується програма при перенесенні слів? Як це зробити?

2. Як створити нерозривний пропуск? Як задати лапки і тире?

3. Які способи вирівнювання тексту ви знаєте? Які команди за це відповідають?

4. Як зробити текст одночасно жирним та курсивним?

5. Як створити список з маркерами та нумерований список?

**Практичне завдання:** виконайте лабораторну роботу № 3 ([http://mif.pu.if.ua/attachments/article/14/tex\\_lab3.pdf](http://mif.pu.if.ua/attachments/article/14/tex_lab3.pdf)).

## 4. Оформлення тексту в цілому

Засоби для оформлення тексту в цілому, як правило, розміщуються в преамбулі документа.

За замовчуванням весь текст набирається шрифтом розміру 10 пунктів. Команда `\documentclass` може мати обов'язковий аргумент, який розміщується перед обов'язковим. Цей аргумент може містити опції документа. Зокрема, серед них може бути опція `11pt` або `12pt`, яка визначає розмір основного шрифту документа, наприклад, `\documentclass[11pt]{article}`. Саме цей розмір відповідатиме шрифту, який вводиться командою `\normalsize`.

Крім стилю `article`, у команді `\documentclass` може використовуватись і інший стиль, наприклад, стиль `book`, при застосуванні якого, зокрема, на відміну від стилю `article`, розрізнятимуться поля на парних і непарних сторінках.

Ширина тексту на сторінці визначається параметром довжини `\textwidth`. Параметр `\oddsidemargin` задає ширину лівого поля непарних сторінок у стилі `book` і всіх сторінок у стилі `article`. Відлік поля починається не від краю аркуша, а від точки, віддаленої від краю аркуша на один дюйм (25,4 мм). Для парних сторінок у стилі `book` аналогічну роль виконує параметр `\evensidemargin`. Висота тексту на сторінці визначається параметром `\textheight`. Розмір верхнього поля задається параметром `\topmargin`. Відлік верхнього поля теж починається від точки, розміщеної на відстані один дюйм від краю аркуша. Крім того, зверху виділяється місце на колонтитул, навіть якщо його немає. Змінювати всі ці параметри треба у преамбулі. Наприклад, якщо ліве поле сторінок у стилі `article` має бути рівним 2 см, то у преамбулі потрібно писати `\oddsidemargin=-5.4mm`.

Колонтитулом на сторінці є інформація, яка розміщується над чи під основним текстом сторінки. Колонтитул найчастіше може містити номер сторінки, назву розділу, книги

або статті. Стиль оформлення колонтитулів дозволяє змінити команда `\pagestyle{st}`. Її обов'язковий аргумент *st* може бути одним з наступних: **empty** — колонтитули порожні, **plain** — номери сторінок ставляться внизу по центру сторінок, **headings** — номери сторінок ставляться зверху з краю сторінок, верхній колонтитул може також містити назву розділу, створеного стандартними командами для заголовків розділів, **myheadings** — оформлення колонтитулів здійснюється редактором або автором. Команда `\thispagestyle{st}` задає оформлення лише поточної сторінки і, на відміну від команди `\pagestyle`, ставиться не в преамбулі, а всередині тексту, який потрапляє на сторінку з нестандартним колонтитулом.

Для створення заголовків розділів документа можна користуватись командами `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph`, `\subparagraph`, обов'язковим аргументом яких має бути текст заголовка. Цей текст друкуватиметься як заголовок, а також заноситиметься до колонтитулів і до змісту.

**Рекомендована література:** [6, с. 155–175], [2, с. 63–86], [5, с. 62–66, 71–75], [9, с. 7–20].

### **Питання для самоперевірки.**

1. Як змінити оформлення колонтитулів?
2. Як задають поля сторінок?
3. Як зробити розмір основного тексту в документі рівним 12 пунктам?
4. Назвіть основні команди для розбиття тексту на розділи та підрозділи.

**Практичне завдання:** змініть поля і ширину сторінки виконаної лабораторної роботи № 3, вилучіть нумерацію сторінок.

## **5. Таблиці**

Для створення таблиць звичайно використовують оточення **tabular**, яке працює аналогічно оточенню **array** для побу-

дови матриць. Обов'язковий аргумент оточення — преамбула таблиці, що є, в найпростішому випадку, послідовністю букв, які описують стовпці таблиці. У преамбулі таблиці буква `c` означає центрування вмісту відповідного стовпця, буква `l` — вирівнювання стовпця по лівому краю, а буква `r` — по правому краю. Клітинки в рядку таблиці відокремлюються символами `&`, а рядки — командами `\\`. Пропуски до і після символів `&` ігноруються. Кожна клітинка таблиці утворює групу, тому зміни шрифтів всередині клітинки забуваються при виході з неї.

Для того щоб розмістити у клітинці таблиці цілий абзац тексту, треба у преамбулі таблиці замість букви `r`, `l` або `c` для потрібного стовпця написати `r{x}`, де  $x$  — ширина відповідного стовпця з вказуванням одиниць вимірювання.

Можна будувати таблиці з лініями. Горизонтальні лінії на всю ширину таблиці задаються командами `\hline`, які застосовуються між рядками таблиці. Команда `\cline{x-y}` дозволяє провести горизонтальну лінію на частину таблиці (від стовпця з номером  $x$  до стовпця з номером  $y$ ). Вертикальні лінії задаються у преамбулі таблиці вертикальними рисками `|` між буквами.

Для створення напису на кілька рядків таблиці (об'єднання кількох клітинок по горизонталі) використовують команду `\multicolumn{x}{y}{z}`, де  $x$  — кількість клітинок, що об'єднуються,  $y$  — преамбула об'єднаної клітинки, а  $z$  — текст, який записується в об'єднану клітинку.

```
\begin{tabular}{|p{22mm}|c|}
\hline \multicolumn{2}{|c|}
{Населення міст, тис. чол.}
\\ \hline Місто & Населення
\\ \hline Львів & 728\\
Хмельниць- & 269\\
кий & 269\\
Тернопіль & 218\\ \hline
\end{tabular}
```

Населення міст, тис. чол.	
Місто	Населення
Львів	728
Хмельниць- кий	269
Тернопіль	218

**Рекомендована література:** [6, с. 212–225], [2, с. 106–113], [5, с. 283–290].

### Питання для самоперевірки.

1. Як створити таблицю з невидимими границями? Що таке преамбула таблиці? З яких символів вона може складатись?

2. Як провести подвійну вертикальну лінію між стовпцями таблиці?

3. Як створити горизонтальну лінію між рядками таблиці?

4. Як у таблиці задати стовпець потрібної ширини?

5. Як об'єднати кілька клітинок у таблиці в одну клітинку?

**Практичне завдання:** створіть засобами  $\text{\LaTeX}$ 'а будь-яку нескладну таблицю.

## 6. Блоки

Кожна буква для  $\text{\TeX}$ 'а — це блок, який характеризується висотою, шириною і глибиною. Він має також базисну лінію та точку відліку на ній (див. рис.).



Блоки розміщуються поруч так, щоб базисна лінія наступного блока була продовженням базисної лінії попереднього блока. Кожен рядок тексту теж утворює блок, блоком є також таблиця і навіть сторінка. Для роботи з блоками призначені спеціальні команди. Команда  $\text{\mbox}\{text\}$  утворює блок з тексту *text*. Наслідком цього буде те, що частина слова або словосполучення, що є аргументом команди  $\text{\mbox}$ , не буде переноситись на наступний рядок, а проміжки між словами

у цьому словосполученні не стискатимуться і не розтягатимуться. Команда `\makebox[width]{text}` дозволяє додатково вказати ширину блока  $width$ . Команда `\framebox[width]{text}` працює аналогічно до команди `\makebox`, але додатково бере текст у рамку.

Якщо потрібно створити блок, у якому міститься цілий абзац тексту заданої ширини  $width$ , то використовують команду `\parbox{width}{text}`. Для того щоб підняти текст  $text$  відносно базисної лінії на величину  $x$ , використовують команду `\raisebox{x}[y][z]{text}`. Якщо  $x < 0$ , то  $text$  опускається на  $|x|$ . Додатково також можна вказати висоту  $y$  і глибину  $z$ , які матиме  $text$  з точки зору програми.

**Рекомендована література:** [6, с. 267–288], [5, с. 210–219], [9, с. 30–31].

#### Питання для самоперевірки.

1. Як зверстати текст у абзац заданої ширини?
2. Як підняти (опустити) кілька символів у рядку?
3. Що таке базисна лінія блока?
4. Що таке точка відліку блока?

**Практичне завдання:** виконайте лабораторну роботу № 4 ([http://mif.pu.if.ua/attachments/article/14/tex\\_lab4.pdf](http://mif.pu.if.ua/attachments/article/14/tex_lab4.pdf)).

## 7. Деякі математичні пакети

Додаткові бібліотеки команд видавничої системи L<sup>A</sup>T<sub>E</sub>X називають пакетами. Ці пакети значною мірою розширюють можливості видавничої системи. Для підключення пакета з ім'ям  $name$  використовують команду `\usepackage{name}`, наприклад, `\usepackage{amsmath}`. Пакети зазвичай підключають у преамбулі документа.

Пакети `amsmath`, `amssymb` і `amsfonts` надають додаткові можливості для оформлення математичних формул. Зокрема, оточення `gather` з пакета `amsmath` працює подібно до оточення `equation`, але, на відміну від останнього, дає можливість відображати виключні формули у кілька рядків.

При цьому рядки відокремлюють командами `\\` і вони автоматично отримують номер. Для того рядка, який не потрібно нумерувати, використовують команду `\nonumber`. Оточення `gather*` працює подібно, але взагалі не нумерує рядки. Для подвійних, потрійних і кратних інтегралів пакет `amsmath` має команди `\iint`, `\iiint` і `\idotsint` відповідно. Команда `\xrightarrow{k}` друкує стрілку  $\rightarrow$  з написом  $k$  над нею, причому довжина стрілки залежить від довжини напису.

Пакет `amssymb` містить велику кількість команд для створення різноманітних математичних символів.

Пакет `amsfonts` надає доступ до двох математичних шрифтів. Команда `\mathfrak{K}` розміщує у формулі готичну букву  $\mathfrak{K}$ . Команда `\mathbb{K}` друкує у формулі ажурну букву  $\mathbb{K}$ .

Для створення каліграфічної латинської букви  $\mathcal{K}$  використовують команду `\mathcal{K}`. Пакет `eucal` з опцією `mathscr` (`\usepackage[mathscr]{eucal}`) дає можливість з допомогою команди `\mathscr` користуватись ще одним каліграфічним шрифтом. Зокрема, команда `\mathscr{K}` створює букву  $\mathcal{K}$ .

**Рекомендована література:** [5, с. 172–209], [6, с. 55–59, 63–64, 72–88], [8].

### Питання для самоперевірки.

1. Назвіть основні математичні пакети. Як підключити пакет?
2. Як друкувати готичні букви?
3. Як набирати виключні формули у кілька рядків?
4. Для чого призначений пакет `eucal`?

**Практичне завдання:** навчіться користуватись готичним і каліграфічним шрифтом та набирати довгі виключні формули, які займають кілька рядків.

## 8. Пакети для роботи з таблицями

Пакет `array` дає можливість використовувати у преамбулі таблиці кілька додаткових символів. Крім стандартних сим-



волів `l`, `r`, `c`, `p{x}` і `l`, у преамбулі можна використовувати `m{x}` (працює подібно до `p{x}`, але рядки таблиці будуть додатково центруватись по вертикалі відносно клітинок даного стовпця), `b{x}` (працює подібно до `p{x}`, але рядки таблиці будуть вирівнюватись по вертикалі по нижньому краю відносно клітинок даного стовпця), `>{y}` (ставиться перед визначенням стовпця і містить команди `y`, які мають виконуватись на початку кожного елемента цього стовпця), `<{y}` (ставиться після визначення стовпця і містить команди `y`, які мають виконуватись у кінці кожного елемента цього стовпця). Наприклад, текст у всіх клітинках стовпця, описаного як `>{\(}c<{\)}`, буде зверстаний у математичному режимі (як формули) і центрований по горизонталі.

```
\begin{tabular}{|l|>{\centering
\arraybackslash}m{2cm}|} \hline
Річка & Площа басейну, км$^2$\\
\hline Дніпро & 504\\
Дністер & 72,1\\ \hline
\end{tabular}
```

Річка	Площа басейну, км <sup>2</sup>
Дніпро	504
Дністер	72,1

Для друку довгих таблиць використовують оточення `longtable` з пакета `longtable` замість оточення `tabular`. При роботі з цим оточенням, як преамбула таблиці, так і сама таблиця, описуються так само, як і при роботі з оточенням `tabular`. При цьому можна використовувати всі можливості для оформлення таблиць, які надає пакет `array`.

Оточення `longtable` виводить центровану на сторінці таблицю. На початку опису такої таблиці (після її преамбули) може бути розміщено до чотирьох груп спеціальних рядків, зокрема для створення заголовків таблиці. Група рядків, яка завершується командою `\endfirsthead`, буде надрукована лише на початку таблиці. Група рядків, після якої стоїть команда `\endhead`, буде надрукована у верхній частині всіх сторінок таблиці, крім першої. Група рядків, яка завершується коман-

дою `\endfoot`, друкуватиметься у нижній частині всіх сторінок таблиці, крім останньої. Група рядків, після якої знаходиться команда `\endlastfoot`, буде надрукована лише в кінці таблиці.

Для роботи з таблицями можна використовувати й інші пакети, які надають спеціальні можливості. Зокрема, пакет `tabularx` дає можливість створювати таблиці заданої ширини, автоматично вибираючи ширину стовпців.

**Рекомендована література:** [5, с. 290–305], [6, с. 226–233], [9, с. 33–36], [11, с. 102–109].

### Питання для самоперевірки.

1. Які пакети дають додаткові можливості для оформлення таблиць? У чому ці додаткові можливості полягають?

2. Як створити заголовок таблиці, який буде автоматично повторюватись на наступній сторінці?

**Практичне завдання:** додавши кілька рядків до таблиці з лабораторної роботи № 4, збільште її так, щоб вона займала більше однієї сторінки, заголовок таблиці має автоматично дублюватись на початку наступних сторінок.

## 9. Використання кольорів

Для друку кольорових текстів призначений пакет `color`. Він підтримує моделі кольорів, наведені у наступній таблиці.

Модель кольорів	Визначення кольору	Приклад кольору
rgb	Три десяткові числа від 0 до 1, які задають інтенсивності червоного, зеленого і синього кольорів відповідно	0.3, 0.825, 1
смук	Чотири десяткові числа від 0 до 1, які задають інтенсивності блакитного, пурпурного, жовтого і чорного кольорів відповідно	0.3, 0.8, 0, 1

Модель кольорів	Визначення кольору	Приклад кольору
gray	Одне десяткове число від 0 до 1, яке задає інтенсивність сірого кольору	0.4
named	Назва кольору (з 68 назв)	SeaGreen
Відсутня	Назва базового кольору (з 8 назв)	green

Колір *clr* тексту *text* відносно моделі *model* вказується командою `\textcolor[model]{clr}{text}` або декларацією `\color[model]{clr}`. Коротку назву *name* кольору можна задати командою `\definecolor{name}{model}{clr}`. Цю назву слід використовувати у командах `\textcolor`, `\color` та інших без вказування моделі.

Команда `colorbox[model]{clr}{text}` зафарбовує фон тексту *text* кольором *clr*, відповідним моделі кольорів *model*. Команда `fcolorbox[model]{fclr}{clr}{text}` додатково будувє прямокутну рамку навколо тексту кольором *fclr*. Команда `\pagecolor[model]{clr}` дозволяє задати колір фону цілої сторінки.

Для розфарбовування таблиць, крім пакета `color`, використовують пакет `colortbl`. Команду `\columncolor[model]{clr}` розміщують у преамбулі таблиці в аргументі символу `>` перед позначенням стовпця, фон якого потрібно зафарбувати кольором *clr*. Команду `\rowcolor[model]{clr}` використовують для зафарбовування фону рядка таблиці кольором *clr*. Її місце — на початку рядка таблиці. Якщо у таблиці є об'єднані клітинки, колір їхнього фону задається у преамбулах цих клітинок (у командах `\multicolumn`).

Команда `\arrayrulecolor[model]{clr}` дозволяє змінити колір горизонтальних і вертикальних ліній таблиці. Область дії цієї команди починається в точці, де вона зустрічається. Тому можна змінювати колір ліній всієї таблиці чи її частини, розміщуючи цю команду перед таблицею, у аргументі симво-

лу > перед потрібним стовпцем у преамбулі таблиці або перед черговим рядком таблиці.

**Рекомендована література:** [5, с. 258–261, 305–307], [10].

**Питання для самоперевірки.**

1. Які моделі кольорів підтримує  $\LaTeX$ ? Опишіть їх.
2. Як змінити колір тексту, колір фону тексту, колір фону цілої сторінки?
3. Як змінити колір тексту і фону в рядку (у стовпці) таблиці?

**Практичне завдання:** поекспериментуйте зі зміною кольорів тексту і фону.

## 10. Пакет `graphics`

Пакет `graphics` дозволяє імпортувати в документ фотографії і рисунки, створені іншими програмами, а також обертати і масштабувати блоки. Рисунки можуть бути збережені в різних форматах (BMP, JPEG, TIFF, GIF, PNG та інших), але найкраще використовувати формат EPS (Encapsulated PostScript), який підтримує і растрові, і векторні рисунки. Перетворити растровий рисунок у формат EPS можна, наприклад, з допомогою пакета PhotoShop.

Графічне зображення вставляється в документ командою `\includegraphics[ldx,ldy][rux,ruy]{file}`, де *ldx*, *ldy* — координати лівого нижнього кута рисунка, а *ru**x*, *ru**y* — координати його правого верхнього кута, *file* — ім'я файлу з рисунком. Файл з рисунком має знаходитись у папці з TEX-файлом документа. Якщо рисунок заданий у форматі EPS, то необов'язкові аргументи можна не вказувати, в іншому випадку вони стають обов'язковими. За замовчуванням *ldx* = 0mm, *ldy* = 0mm. Координати *ldx* < *ru**x*, *ldy* < *ru**y* записують з використанням TEX'івських одиниць вимірювання (mm, cm, in тощо), вони визначають розмір області, яка відводиться для зображення. Якщо ця область виявиться замалою для рисунка, то він накладеться на оточуючий текст чи поля сторінки.

Якщо область буде завеликою, то залишиться вільне місце.

Команда `\includegraphics*[ldx,ldy][rux,ruy]{file}` дозволяє при вставці обрізати частину рисунка, якщо він є завеликим для виділеної області.

Команда `\resizebox{xl}{yl}{obj}` масштабує блок (текст, рисунок, формулу чи таблицю) *obj* так, щоб його ширина стала рівною *xl*, а висота — *yl*. Якщо замість *xl* чи *yl* поставити знак оклику **!**, то масштабування буде пропорційним.

Команда `\scalebox{xs}[ys]{obj}` масштабує блок *obj* відповідно до вказаних множників *xs* (по горизонталі) і *ys* (по вертикалі). Якщо *ys* не вказано, то масштабування буде пропорційним. Якщо *xs* чи *ys* є від'ємним, то додатково відбувається дзеркальне відображення блока по горизонталі чи по вертикалі.

Команда `\reflectbox{obj}` буде дзеркальне відображення блока *obj* по горизонталі.

Команда `\rotatebox{angle}{obj}` розвертає блок *obj* на кут *angle* за годинниковою стрілкою навколо точки відліку блока. Кут вказується у градусах.

**Рекомендована література:** [5, с. 230–257], [7], [3, с. 47–73], [1, с. 55–60].

### Питання для самоперевірки.

1. Як вставити фотографію або рисунок в документ?
2. Як обрізати краї рисунка?
3. Як розвернути блок (текст, формулу або рисунок) на заданий кут?
4. Як масштабувати блок?

**Практичне завдання:** виконайте лабораторну роботу № 5 ([http://mif.pu.if.ua/attachments/article/14/tex\\_lab5.pdf](http://mif.pu.if.ua/attachments/article/14/tex_lab5.pdf)).

## 11. Пакет `pstricks`

Пакет `pstricks` і кілька пов'язаних з ним пакетів призначені для створення рисунків з векторною графікою засобами мови PostScript. Трансляція документів з такими рисунками

безпосередньо у формат PDF є недопустимою. Потрібно спочатку виконувати трансляцію у формат DVI, потім — у формат PS, і лише після цього — у формат PDF.

Рисунок зазвичай має бути вмістом оточення `pspicture`, яке виділяє для рисунка місце на сторінці. Формат оточення `pspicture` є наступним:

```
\begin{pspicture}(x_1,y_1)(x_2,y_2)
...
\end{pspicture}
```

де  $x_1, y_1$  — координати лівої нижньої вершини, а  $x_2, y_2$  — координати правої верхньої вершини виділеної прямокутної області. Координати, як і інші довжини, для пакета `pstricks` можуть бути абсолютними і відносними. Абсолютні довжини і координати задають з вказуванням одиниць вимірювання, таких як mm, cm, pt, in. Відносні довжини і координати наводять без одиниць вимірювання. За замовчуванням відносні довжини і координати вимірюються у сантиметрах.

Більшість команд для створення графічних об'єктів мають наступний формат:

команда[опції]{аргументи}( $x_1, y_1$ )( $x_2, y_2$ )...( $x_n, y_n$ )

Опції задають у вигляді рівностей «назва=значення», наприклад, опції `linewidth=0.5mm`, `linecolor=red` вказують товщину лінії та її колір. Аргументи є необов'язковими (їх можна пропускати разом із фігурними дужками), вони можуть вказувати деякі деталі оформлення графічних об'єктів. Координати точок ( $x_i, y_i$ ) задають відносно точки (0, 0) з оточення `pspicture`.

Для створення найпростіших графічних об'єктів використовують наступні команди:

`\psdots[opts]( $x_1, y_1$ )( $x_2, y_2$ )...( $x_n, y_n$ )` будує точки з заданими координатами ( $x_1, y_1$ ), ( $x_2, y_2$ ), ..., ( $x_n, y_n$ ) (розмір,

вигляд і колір символів, які позначають точки, визначається опціями *opts*);

`\psline[opts]{arrows}(x_1,y_1)(x_2,y_2)\dots(x_n,y_n)` проводить ламану через точки з заданими координатами (крім опцій *opts* можуть вказуватись стрілки *arrows* на кінцях ламаної);

`\pspolygon[opts](x_1,y_1)(x_2,y_2)\dots(x_n,y_n)` відображає багатокутник з вершинами у точках із заданими координатами (замкнену ламану);

`\pscurve[opts]{arrows}(x_1,y_1)(x_2,y_2)\dots(x_n,y_n)` створює незамкнену криву, яка проходить через точки з заданими координатами;

`\psccurve[opts](x_1,y_1)(x_2,y_2)\dots(x_n,y_n)` будує замкнену криву, яка проходить через точки з заданими координатами;

`\psframe[opts](x_1,y_1)(x_2,y_2)` створює прямокутник із лівою нижньою і правою верхньою вершинами у точках  $(x_1, y_1)$  і  $(x_2, y_2)$  відповідно (сторони прямокутника паралельні осям координат);

`\pscircle[opts](x_0,y_0){radius}` відображає коло з центром у точці  $(x_0, y_0)$  і радіусом *radius*;

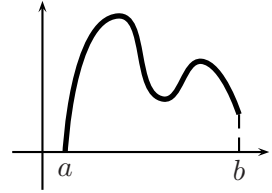
`\psarc[opts]{arrows}(x_0,y_0){radius}{a1}{a2}` проводить дугу кола з центром у точці  $(x_0, y_0)$  і радіусом *radius* від кута *a1* до кута *a2* (кути вимірюються проти годинникової стрілки у градусах).

Команда `\rput[refpoint]{angle}(x_0,y_0){obj}` розміщує об'єкт *obj* у точці з координатами  $(x_0, y_0)$  відносно точки прив'язки *refpoint*. Об'єктом *obj* може бути текстовий напис, формула, таблиця, інший рисунок чи будь-який інший блок. За замовчуванням точкою прив'язки є центр об'єкта. Об'єкт може бути розвернутим за годинниковою стрілкою на кут *angle*.

```

\begin{pspicture}(-0.4,-0.5)(3,2)
\psline{->}(-0.4,0)(3,0)
\psline{->}(0,-0.5)(0,2)
\pscurve[linewidth=2pt](0.3,0)
(1,1.8)(1.6,0.7)(2.1,1.2)(2.6,0.5)
\psline[linestyle=dashed](2.6,0)
(2.6,0.5) \rput(0.3,-0.23){$a$}
\rput(2.6,-0.23){$b$}
\end{pspicture}

```



Для зафарбовування або заштриховування замкнених фігур (багатокутник, замкнена крива, коло та деякі інші) використовують спеціальні опції. Для зафарбовування або заштриховування фігур, обмежених простими незамкненими лініями, використовують команду `\pscustom` разом з опціями для зафарбовування.

Для багатьох операцій потрібно додатково, крім пакета `pstricks`, використовувати ряд допоміжних пакетів. Зокрема, пакет `pst-plot` дозволяє будувати графіки функцій з явним чи параметричним заданням, а також графічно відображати дані. Графік явно заданої функції на проміжку  $[xmin, xmax]$  буде команда `\psplot[opts]{xmin}{xmax}{func}`. Функцію `func` потрібно записувати на мові PostScript з використанням символу `x` як незалежної змінної. При обчисленнях на мові PostScript спочатку задають операнди, а потім — операції. Кожен операнд розміщується у стеку. Стек працює за принципом: «першим прийшов — останнім вийшов», тобто спочатку зі стеку зчитується останнє записане в нього число. Операції і приклади їх використання наведені у наступній таблиці:

Операція	Бере зі стеку	Кладе у стек	Приклад	Результат
add	$a, b$	$a + b$	2 3 add	5
sub	$a, b$	$a - b$	2 3 sub	-1
mul	$a, b$	$ab$	2 3 mul	6



Операція	Бере зі стеку	Кладе у стек	Приклад	Результат
div	$a, b$	$a/b$	3 2 div	1.5
idiv	$a, b$	$[a/b]$	3 2 idiv	1
mod	$a, b$	$a \bmod b$	5 3 mod	2
exp	$a, b$	$a^b$	2 3 exp	8
abs	$a$	$ a $	-2 abs	2
neg	$a$	$-a$	2 neg	-2
floor	$a$	$[a]$	2.8 floor	2
sqrt	$a$	$\sqrt{a}$	4 sqrt	2
sin	$a$	$\sin a$	90 sin	1
cos	$a$	$\cos a$	90 cos	0
atan	$a, b$	$\arctg \frac{a}{b}$	2 2 atan	45
log	$a$	$\lg a$	100 log	2
ln	$a$	$\ln a$	5 ln	1.60943791

Наприклад, для побудови графіка функції  $y = x^2 - 1$  на проміжку  $[-2, 3]$  використовують команду:

```
\psplot{-2}{3}{x x mul 1 sub}
```

Пакет `pst-coil` дає можливість досить легко будувати звивисті і зигзагоподібні лінії. Пакет `pst-grad` дозволяє виконувати градієнтне зафарбовування замкнених фігур з поступовою зміною кольору. Пакет `pst-text` призначений для виконання складних операцій над текстом, таких як розміщення тексту вздовж кривої, оформлення тексту тінню, зафарбовування певним чином букв, відсікання ліній у формі букв тексту. Допустимо виконувати такі операції над формулами замість звичайного тексту. Пакет `pst-3d` дає можливість працювати з тривимірною графікою. Пакет `pst-node` призначений для спрощення створення вершин з іменами і з'єднування їх з додаванням міток. Пакет `pst-tree` дозволяє будувати деревовидні діаграми.

**Рекомендована література:** [3, с. 122–200].

**Питання для самоперевірки.**

1. Які пакети призначені для створення рисунків?
2. Як виділити область для розміщення рисунка?
3. Назвіть основні команди для створення найпростіших графічних об'єктів.
4. Яка опція змінює товщину лінії?
5. Як додати до рисунка текстовий напис?

**Практичне завдання:** виконайте лабораторні роботи № 6 ([http://mif.pu.if.ua/attachments/article/14/tex\\_lab6.pdf](http://mif.pu.if.ua/attachments/article/14/tex_lab6.pdf)) і № 7 ([http://mif.pu.if.ua/attachments/article/14/tex\\_lab7.pdf](http://mif.pu.if.ua/attachments/article/14/tex_lab7.pdf)).

## Список рекомендованої літератури

1. Балдин Е. М. Компьютерная типография LaTeX. СПб. : БХВ-Петербург, 2008. 304 с.
2. Беляков Н. С., Палош В. Е., Садовский П. А. TeX для всех: Оформление учебных и научных работ в системе LaTeX. М. : Книжный дом «ЛИБРОКОМ», 2009. 208 с.
3. Гуссенс М., Ратц С., Миттельбах Ф. Путеводитель по пакету LaTeX и его графическим расширениям. Иллюстрирование документов при помощи TeX'a и PostScript'a / пер. с англ. под ред. А. И. Лебедева. М. : Мир : Бином ЛЗ, 2002. 621 с.
4. Кренивч А. П., Бородин В. А. Видавнича система LaTeX : методичні вказівки до лабораторних занять з дисципліни «Практикум на ЕОМ». К. : ВПЦ «Київський університет», 2007. 49 с.
5. Котельников И. А., Чеботаев П. З. LATEX по-русски. 3-е издание, перераб. и доп. Новосибирск : Сибирский хронограф, 2004. 496 с.
6. Львовский С. М. Набор и верстка в системе LaTeX. 3-е издание, испр. и доп. М. : МЦНМО, 2003. 448 с.
7. Сюткин В. Включение рисунков в LaTeX2e. 2001. 12 с.
8. Сюткин В. Набор математических формул в LaTeX2e. 2002. 46 с.
9. Сюткин В. Справочник по командам LaTeX2e. 2002. 49 с.
10. Сюткин В. Цвет в LaTeX2e. 2001. 8 с.
11. Ткачук В. М., Ткачук О. М. Практикум на ЕОМ : навчальний посібник. Частина 1 : Видавнича система LaTeX. Івано-Франківськ : Вид-во ПНУ ім. В. Стефаника, 2012. 178 с.

Підписано до друку 04.01.2018. Формат 60×84/16.  
Папір офсетний. Друк цифровий.  
Гарнітура Computer Modern Roman. Умовн. друк. арк. 2,1.  
Тираж 100. Зам. № 37 від 04.01.2018.

Віддруковано: Приватний підприємець Голіней О. М.  
76000, м. Івано-Франківськ,  
вул. Галицька, 128,  
тел.: (0342) 58-04-32.