

МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ  
Харківський національний університет внутрішніх справ  
Навчально-науковий інститут права, економіки та соціології  
Кафедра інформаційних систем і технологій

## **МЕТОДИ ТА ЗАСОБИ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**методичні вказівки  
до проведення практичних занять та самостійної роботи**

Розробники:  
**Ю. П. Горелов, І. В. Кобзев, О. Ф. Лановий**

Галузь знань	0501 «Інформатика та обчислювальна техніка»
Напрямок підготовки	6.050101 «Комп'ютерні науки»
Спеціальність	7.080401 «Інформаційні управляючі системи та технології»
Форма навчання	денна

Харків 2010

УДК 004.02(076)  
ББК 32.97я73р  
М 54

**Розробники:**

**Ю. П. Горелов** – канд. техн. наук, доцент, завідувач кафедри інформаційних систем і технологій ННІ ПЕС ХНУВС;  
**І. В. Кобзев** – канд. техн. наук, доцент кафедри інформаційних систем і технологій ННІ ПЕС ХНУВС;  
**О. Ф. Лановий** – канд. техн. наук, проф. кафедри інформаційних систем і технологій ННІ ПЕС ХНУВС

**Рецензенти:**

**С. Г. Удовенко** – д-р техн. наук, професор, професор кафедри ЕОМ ХНУРЕ;  
**Ю. Є. Хорошайло** – канд. техн. наук, доцент, заступник начальника з науково-методичної роботи ННІ ПМСІТ ХНУВС

Рекомендовано Методичною радою  
Харківського національного університету внутрішніх справ,  
протокол № 3 від 19.03.2008

**Методи** та засоби комп'ютерних інформаційних технологій :  
М 54 метод. вказівки до проведення практ. занять та самост. роботи /  
розроб.: Ю. П. Горелов, І. В. Кобзев, О. Ф. Лановий ; МВС України,  
Харк. нац. ун-т внутр. справ, Навч.-наук. ін-т права, економіки  
та соціології, Каф. інформ. систем і технологій. – Х. : ХНУВС,  
2010. – 84 с.

УДК 004.02(076)  
ББК 32.97я73р

# ЗМІСТ

<b>Вступ .....</b>	<b>4</b>
<b>ПРАКТИЧНЕ ЗАНЯТТЯ № 1.....</b>	<b>4</b>
Вступ до інтернет-програмування. Основи HTML.....	4
<b>ПРАКТИЧНЕ ЗАНЯТТЯ № 2.....</b>	<b>9</b>
Використання фреймів, формулярів, карт у web-проекті.....	9
<b>ПРАКТИЧНЕ ЗАНЯТТЯ № 3.....</b>	<b>15</b>
Метадані та CSS, використання фільтрів .....	15
<b>ПРАКТИЧНЕ ЗАНЯТТЯ № 4.....</b>	<b>22</b>
Створення сценаріїв на JavaScript .....	22
<b>ПРАКТИЧНЕ ЗАНЯТТЯ № 5.....</b>	<b>28</b>
Створення і використання бібліотек сценаріїв.....	28
<b>ПРАКТИЧНЕ ЗАНЯТТЯ № 6.....</b>	<b>32</b>
Створення XML-документів .....	32
<b>ПРАКТИЧНЕ ЗАНЯТТЯ № 7.....</b>	<b>35</b>
Ознайомлення з інтерфейсом редактора Flash MX.	
Робота з кольором у Flash MX.....	35
<b>ПРАКТИЧНЕ ЗАНЯТТЯ № 8.....</b>	<b>41</b>
Створення графічних зображень на основі різних методів малювання у Flash.....	41
<b>ПРАКТИЧНЕ ЗАНЯТТЯ № 9.....</b>	<b>50</b>
Анімація графічних зображень у Flash.....	50
<b>ПРАКТИЧНЕ ЗАНЯТТЯ № 10.....</b>	<b>58</b>
Інтерактивні елементи.	
Програмування за допомогою Action Script .....	58
<b>Самостійна робота .....</b>	<b>59</b>
Характеристика підручників і навчальних посібників .....	60
Методичні вказівки з вивчення дисципліни .....	62
Індивідуальні розрахункові завдання, контрольні завдання, теми для рефератів.....	77
Приклади розв'язання типових задач .....	80

# ВСТУП

Метою проведення практичних занять з дисципліни «Методи та засоби комп'ютерних інформаційних технологій» є формування у студентів систематичного і наукового підходу до методології вибору засобів комп'ютерних інформаційних технологій на фізичному, математичному і семантичному рівнях та програмної реалізації алгоритмів для вирішення складних сучасних інженерних задач. Ставиться за мету вивчити застосовувані методи сучасних інформаційних технологій. Студенти вивчають об'єктну модель документа DOM, технології HTML, DHTML, JavaScript, CSS, основи XML, Flash.

## **ПРАКТИЧНЕ ЗАНЯТТЯ № 1**

### ***Вступ до інтернет-програмування. Основи HTML***

#### **1. Мета роботи**

Вивчення мови HTML, способів виконання базових операцій, особливостей створення web-сторінок.

#### **2. Підготовка до роботи**

Перед заняттям слід повторити матеріал за курсом лекцій та за рекомендованою літературою.

#### **3. Методичні вказівки до виконання роботи**

HTML (Hyper Text Markup Language) – мова опису документів, використовуваних у web. HTML-файли є ASCII-файлами.

3.1. Дескриптори (теги) у HTML – це спеціальні елементи розмітки, що визначають, як даний документ HTML має інтерпретуватися програмою перегляду інформації в Інтернет – браузером. Розглянемо основні дескриптори HTML.

Теги HTML бувають одно- і двокомандні. Однокомандні теги зазвичай використовуються для зміни вигляду виведеної на екран інформації. Вони задаються в такому вигляді:

`<команда [атрибут=значення [атрибут=значення]...]>текст,`

де

**команда** – команда HTML;

**атрибут=значення** – задання конкретних значень параметрам (атрибутам) команди (є не обов'язковими);

**текст** – текст, що виводиться на екран тільки для читання у вигляді, обумовленому командою та її атрибутами.

Двокомандні теги є основним засобом реалізації HTML-документів і використовуються для задання параметрів документа (заголовки, вставка посилань, об'єктів тощо). Вони спарені так, що за відкриваючим тегом має йти відповідний закриваючий тег, а між ними міститься текст чи інші теги. Два теги та частина документа, відділена ними, утворюють блок, що називається HTML-елементом.

Вони мають такий вигляд:

`<команда [атрибут=значення [атрибут=значення]...]>`

[текст][тег] [[текст][тег]...]

</команда [атрибут=значення [атрибут=значення]...]>

де / використовується у випадку, коли тег є останнім (закриваючим) у блоці тегів.

Якщо тег складається з двох команд, то першою має бути визначена команда без косої риски (початок виконання тега), а другою – команда, яка позначається косою рисою ( / ), що інформує про закінчення дії тега.

### 3.2. Базові дескриптори HTML-документа:

- <HTML> ... </HTML> – задання границь HTML-документа в цілому;
- <HEAD> ... </HEAD> – визначення заголовка (шапки) документа;
- <TITLE> ... </TITLE> – задання заголовка сторінки;
- <BODY> ... </BODY> – основний текст (тіло) web-документа, це інформація, що виводитиметься у вікні браузера;
- <!-- ... --> – усередині міститься коментар, текст якого браузери не відображають на екрані.

Загальна структура HTML-документа має такий вигляд:

```
<HTML>
<!-- ім'я файла документа – MINIMUM.HTM -->
<Head>
<Title> Текст заголовка вікна для браузера
</Title>
</Head>
<Body>
...
Текст документа
...
</Body>
</HTML>
```

### 3.3. Додаткові теги

Для створення і редагування HTML-текстів використовується ряд додаткових дескрипторів.

Основним елементом HTML документа є абзац. У тексті HTML документа знаки перекладу рядка і пробіли при обробці їх браузером ігноруються. Тег <P> визначає кінець абзацу. Тег <BR> – примусовий розрив рядка без початку нового абзацу. Тег <HR> вказує на те, що браузер має відобразити горизонтальну лінію, що йде через весь екран. Наприклад:

```
<HTML>
<!-- Другий приклад -->
<Head>
<Title> Заголовок другого прикладу </Title>
</Head>
<Body>
ХНУРЕ <BR> Текст. <P> <HR>
Кафедра ПЗ ЕОМ. <P> <HR>
</Body>
</HTML>
```

### 3.4. Заголовки

Загальний синтаксис тега визначення заголовка:

<Hx> Заголовок1 </Hx>, де x – це номер рівня від 1 до 6 ( <H1> – найбільший заголовок, <H6> – найменший).

3.5. Теги виділення тексту:

<B> ... </B> – напівжирний шрифт (Bold); <I> ... </I> – курсив (Italic);

<U> ... </U> – з підкресленням (Undersocore).

Браузери автоматично вилучають усі зайві пробіли між словами. Для виводу таблиць MS DOS, графіків і діаграм без їхнього викривлення необхідно використовувати теги <PRE> ... </PRE>. Усі пробіли, розташовані усередині цього тега, будуть відображені на екрані.

3.6. Для побудови таблиць у HTML-документі можна задати:

<TABLE BORDER=відступ WIDTH=розмір>

<CAPTION> Ім'я\_таблиці </CAPTION>

Текст\_таблиці

</TABLE>

Обов'язковий параметр BORDER= задає ширину рамки таблиці в пікселях. Необов'язковий параметр WIDTH= задає розмір таблиці у відсотках щодо розміру вікна браузера. Текст таблиці має містити дескриптори визначення рядків і граф таблиці.

Визначення рядка таблиць з двома графами:

<TR> <TD> графа1 </TD> <TD> графа2 </TD> ... </TR>

3.7. Задання списків. У HTML-документах можна задавати три види списків: упорядкований список (пронумерований); неупорядкований список (з графічними позначками); словник термінів. Списки можуть бути вкладеними.

Теги для задання списків:

<OL> ... </OL> – для упорядкованого списку. Усі елементи списку перелічуються у визначеному порядку з приписуванням перед елементом тега <LI>. При відображенні списку браузером його елементи автоматично нумеруються.

<UL> ... </UL> – для неупорядкованого списку. При відображенні браузером цих списків використовуються спеціальні символи для виділення рядків списку.

<DL> ... </DL> – для словника термінів. Це списки-визначення. Вони призначені для відображення списків термінів і їхніх визначень. Кожен елемент списку складається з двох частин: терміну і його визначення. Список у цілому має знаходитися усередині тега <DL> ... </DL>. Терміну передуює тег <DT>, а визначенню – <DD>.

Перед кожним елементом списку необхідно ставити однокомандний тег <LI>.

3.8. Гіпертекстові посилання (гіперзв'язки) – це покажчики на визначене місце в поточному чи іншому документі, або в зовнішньому графічному, відео чи аудіофайлі.

Посилання на інформаційні ресурси Інтернет здійснюються за допомогою URL (uniform resource locator – універсальний покажчик на ресурс). Вони мають наступні схеми доступу (необхідно для визначення протоколу):

Приклади схем доступу: http://  
ftp://

mailto://

file://

Після схеми доступу вказується адреса вузла (комп'ютера) в мережі та ім'я файла ресурсу.

Приклади:

file://host/directory/filename — ідентифікує конкретний файл;

ftp://user:password@host:port/directory/filename — FTP-сервер;

gopher://host:port/gopher-path — Gopher-сервер і пункт меню;

http://host:port/directory/filename?searchpart — сервер WWW;

mailto:user@host — поштова адреса Інтернет;

telnet://user:password@host:port/ — TELNET-сервер.

Гіпертекстові зв'язки вводяться в HTML-документ за допомогою якірного дескриптора <A> ... </A>.

Для вбудовування гіперзв'язків використовується такий загальний синтаксис:

<A HREF="адреса\_ресурсу\_посилання">

виділений\_текст\_посилання </A>,

де *адреса\_ресурсу\_посилання* – це ім'я поточного документа, шлях до іншого документа або файла, інтернет-адреса ресурсу (файла, документа тощо), з яким встановлюється гіперзв'язок;

*виділений\_текст\_посилання* (чи *маршрутизатор*) – це текст чи інший об'єкт, при натисканні на який мишею виконується перехід за адресою, зазначеною в *адресі\_ресурсу\_посилання*.

Рекомендується розміщувати значення атрибута *адреса\_ресурсу\_посилання* в лапки (або одинарні, або подвійні). Рядок у лапках не повинен містити таких самих лапок усередині себе.

Можна також опустити лапки для значень атрибутів, що складаються тільки з таких символів: символів англійського алфавіту (A...Z, a...z); цифр (0...9); проміжків часу; дефісів (-).

Існує три основних типи гіперзв'язків:

- посилання на інше місце усередині одного документа:

<A HREF="#адреса\_ресурсу"> маршрутизатор </A>

...

<A NAME="адреса\_ресурсу">

</A>

... <!-- у це місце буде здійснений перехід -->

- посилання на інший інтернет-документ (файл):

<A HREF="http://www.hitachi.com"> монітори </A>

При натисканні мишкою в тексті HTML-документа на виділений текст «монітори» відбувається автоматичний перехід у мережі Інтернет на сторінку фірми Hitachi, зазначену в адресі ресурсу.

- посилання на визначене місце іншого документа (файла):

<BODY> Для одержання даних про монітори клацніть

<A HREF="http://www.hitachi.com/video.htm#TargetName">

тут </A>

...

</BODY>

Текст тут визначає гіперзв'язок, який вказує на те, що перехід буде здійснений у визначене місце у файлі *video.htm*. При цьому у файлі *video.htm* має бути дескриптор `<A NAME="TargetName"> </A>`.

### 3.9. Графічні об'єкти

Для вбудовування посилань на графічні файли використовується тег

```
<IMG SRC="ім'я_файла" ALT="текстовий опис"  
ALIGN="вирівнювання">
```

де IMG є ім'ям тега визначення посилання на графічний файл, шлях пошуку якого визначається значенням опції SRC=. Необов'язковий атрибут ALT= містить текстовий опис, що за умовчанням відображається під картинкою. Не обов'язкова опція ALIGN= призначена для задання місця розташування виведеного текстового опису образу і може приймати такі значення: BOTTOM (знизу), TOP (зверху), MIDDLE (посередині).

Можна визначити графічний образ усередині дескриптора, що задає гіперзв'язок.

## 4. Порядок виконання роботи

1. У текстовому редакторі Notepad, або в середовищі будь-якого тегового редактора HTML написати вихідний текст документа HTML.
2. Зберегти текст у форматі HTML.
3. Переглянути документ HTML за допомогою доступного браузера (наприклад, Internet Explorer).
4. Проаналізувати якість документа, виправити помилки.

## 5. Варіанти індивідуальних завдань

Студенти обирають одну із вказаних тем:

- 1) розробити web-довідник зі створення web-сторінок;
- 2) розробити web-довідник з роботи в табличному процесорі MS Excel;
- 3) розробити web-довідник з роботи в MS Access;
- 4) розробити web-довідник з навчання основам програмування;
- 5) розробити web-довідник з роботи в середовищі візуального проектування (на вибір);
- 6) розробити web-довідник з роботи в локальній мережі університету;
- 7) розробити web-довідник з роботи з програмами електронної пошти;
- 8) розробити web-довідник для вступників до університету;
- 9) розробити web-довідник з використання електронної бібліотеки університету;
- 10) розробити web-довідник з обчислювальних мереж;
- 11) розробити web-довідник з вищих навчальних закладів м. Харкова;
- 12) розробити web-довідник з однієї з дисциплін, що викладаються в університеті (на вибір);
- 13) розробити сайт фірми;
- 14) розробити сайт своєї спеціальності;
- 15) розробити власний сайт.

## 6. Контрольні запитання

1. Що таке WWW?
2. Дайте загальну характеристику HTML-стандарту.



3. У чому суть гіпертекстової форми представлення
4. web-документів?
5. Що таке гіперзв'язок?
6. Що таке маршрутизатор гіперпосилання?
7. Опишіть синтаксис мови HTML.
8. Назвіть основні групи тегів HTML.
9. Як задаються і відображаються заголовки?
10. Як відображаються шрифти?
11. Як задається і відображається форматований текст?
12. Як задаються і відображаються списки?
13. Як відображаються гіперпосилання?
14. Назвіть види гіперзв'язків.
15. Як відображаються графічні об'єкти?
16. Як використовувати графічний образ в якості маршрутизатора гіперпосилання?
17. Опишіть етапи створення web-документа.
18. Як відредагувати існуючий HTML-документ?
19. Що таке браузер?
20. Як відображається текст HTML-документа в текстовому редакторі?
21. Як відображається HTML-документ у вікні браузера?

## **7. Рекомендована література**

1. Янг М. Л. Internet. Полное руководство : пер. с англ. / Маргарет Левин Янг. — К. : Издательская группа BHV, 2001. — 864 с.
2. Айзенменгер Р. HTML 3.2/4.0 : справочник : пер. с нем. / Р. Айзенменгер. — М. : БИНОМ, 1998. — 368 с.
3. Соколов С. А. HTML и CSS : в примерах, типовых решениях и задачах / С. А. Соколов. — М. : Вильямс, 2007. — 416 с. — Профессиональная работа.
4. HTML, XHTML и CSS / Брайан Пфаффенбергер, Стивен Шафер, Чак Уайт, Билл Кароу. — 3-е изд. — М. : Вильямс, 2009. — 752 с. — Библия пользователя.

## **ПРАКТИЧНЕ ЗАНЯТТЯ № 2**

### ***Використання фреймів, формулярів, карт у web-проекті***

#### **1. Мета роботи**

Вивчення способів створення фреймових структур, формулярів, карт-зображень у web-документах.

#### **2. Підготовка до роботи**

Перед заняттям слід повторити матеріал за курсом лекцій та за рекомендованою літературою.

#### **3. Методичні вказівки до виконання роботи**

##### **3.1. Фрейми**

При використанні фреймової структури вікно браузера розділяється на під-вікна, у кожному з яких можна відображати довільні документи мовою HTML, організувати незалежне прокручування їхнього вмісту і змінювати розміри

фреймів у межах вікна браузера. Гіпертекстові посилання, що присутні в документі, розташованому в одному з підвікон, можуть бути використані для відношення вмісту інших фреймів.

У фрейм-утримуючих документах знаходяться дані, що визначають структуру поділу вікна браузера на підвікна. У цих документах немає інформації, яку необхідно помістити в конкретні фрейми, а тільки посилання на звичайні гіпертекстові документи мовою HTML.

У загальному випадку структура фрейм-утримуючого документа має такий вигляд:

```
<HTML>
<HEAD> ... </HEAD>
<FRAMESET COLS="30 %, 70 %">
<NOFRAMES>
Задання інформації для браузера, що не підтримує фрейми.
</NOFRAMES>
<FRAME SRC=frame1.html>
<FRAME SRC=frame2.html> </FRAMESET>
</HTML>
```

Це структура файла-фреймоутримувача.

Тег <FRAMESET> вказує, на скільки рядків або стовпчиків фреймів має розділятися вікно браузера.

У даному прикладі вікно браузера буде розділено на два фрейми, розташовані один під іншим, при цьому висота першого складе 30 % висоти вікна, а другого – 70 %.

frame1.html і frame2.html є гіпертекстовими документами. Тег <FRAMESET> описує або вертикальне розташування фреймів, або горизонтальне:

<FRAMESET ROWS="рядок розмірів фреймів"> використовуємо при горизонтальному розташуванні фреймів,

<FRAMESET COLS="рядок розмірів фреймів"> – при вертикальному розташуванні.

Тег <frame> може містити необов'язкові параметри:

SCROLLING= – наявність смуг прокручування.

SCROLLING=YES – смуги прокручування присутні.

SCROLLING=NO – смуги прокручування відсутні.

SCROLLING=AUTO – смуги прокручування присутні тільки в тому випадку, якщо документ не міститься у фреймі.

NORESIZE – користувач не може змінювати розмір фрейма.

NAME= задає ім'я фрейма, використовується в основному для того, щоб можна було оновлювати інформацію в одному фреймі, активізуючи гіпертекстове посилання в іншому.

MARGINHEIGHT= задає висоту поля фрейма в пікселях – відстань від тексту до рамки, що обмежує фрейм.

### 3.2. Графічні карти

У гіперпосиланнях як маршрутизатор можна використовувати область вбудованого в HTML-документ графічного зображення.

У HTML є можливість призначити одному малюнку кілька активних областей, кожна з яких буде зв'язана з визначеною адресою (сторінки, сайт). Для цього існує тег <MAP>.

Можна, наприклад, створити графічне меню з однієї великої картинки таким чином, щоб кожен елемент системи меню містив визначений URL. Розподіл посилань за картинкою описується в тегові IMG таким параметром:

```
<IMG SRC="url" USEMAP="url#map_name">
```

де USEMAP задає розташування *map\_name*. Якщо URL не зазначений, то пошук *map\_name* ведеться в поточному документі. Код схеми може виглядати так:

```
<MAP NAME="map_name">  
<AREA [SHAPE="shape"  
COORDS="x,y,... " [HREF="reference"] [NOHREF]>  
</MAP>
```

Для того, щоб задати чутливу область, існує тег <AREA>, що розташовується між тегами <MAP> і </MAP>.

Атрибути тега <AREA>:

- shape – форма області (rect – прямокутник; circle – коло; poly – багатокутник);
- href – URL посилання при клацанні на області мишкою;
- nohref – завдання неактивної області (при клацанні на ній мишкою переходу за посиланням не відбувається);
- coords – координати області.

Наприклад:

```
<MAP NAME="myname">  
<AREA SHAPE=RECT COORDS="0,0,20,20" HREF="my1.htm">  
<AREA SHAPE=RECT COORDS="40,0,60,20" HREF="my2.htm">  
</MAP>
```

У даному випадку одна з активних областей має форму прямокутника і посилань на файл *my1.htm*.

Атрибут COORDS задає координати області в пікселях. Відлік починається з нуля. Коло має три координати, прямокутник – чотири, а для багатокутника ви повинні описати кожен його кут у двох координатах.

Якщо область має форму прямокутника, наводяться дві пари координат (верхніх лівих і нижнього правого кутів) прямокутника по осях *x* і *y* у вигляді «*x,y,x,y*».

Якщо область має форму кола, наводяться координати виду «*x,y,r*», де *x,y* – координати центра кола, а *r* – його радіус.

Якщо область має вигляд багатокутника, то її координати задаються рядком типу «*x,y,x1,y1,x...,y...*», тобто кожна вершина багатокутника описується своєю координатою.

Остання пара координат автоматично замикається на першій. Для зв'язку карти з малюнком карта вміщується в розділ HEAD, а для малюнка вказується:

```

```

Тепер при клацанні мишкою в координатах першого квадрата завантажиться сторінка *my1.htm*, а в межах другого квадрата – *my2.htm*. Усі інші області малюнка виявляться неактивними. Над активною областю малюнка курсор приймає форму руки, як над звичайним посиланням.

Приклад карти:

```
<BODY>
<IMG SRC="view.jpg" ALT="Клацніть тут" USEMAP="#map">
...
<MAP NAME="map">
<AREA SHAPE="RECT" COORDS=0,0,64,64
HREF="http://www.kture.kharkov.ua/">
</MAP>
</BODY>
```

### 3.3. Форми

Форма — це інструмент, за допомогою якого HTML-документ може послати деяку інформацію в задалегідь визначену точку мережі, де інформація буде оброблена. Програми, що обробляють дані, передані формами, називають CGI-скриптами.

Форми передають інформацію програмам-оброблювачам у вигляді пар [ім'я змінної]=[значення змінної].

Імена змінних необхідно задавати латинськими літерами. Значення змінних сприймаються оброблювачами як рядки, навіть якщо вони містять тільки цифри. Форма відкривається тегом <FORM> і закінчується </FORM>. HTML-документ може містити в собі кілька форм, однак форми не повинні знаходитися одна усередині іншої. HTML-текст, включаючи позначки, може розміщуватися всередині форм без обмежень.

Атрибути:

- ACTION. Обов'язковий атрибут. Визначає, де знаходиться оброблювач форми. Наприклад, <FORM METHOD="POST" ACTION="/cgi-bin/data">

- METHOD. Визначає, яким чином (за допомогою якого методу протоколу передачі гіпертексту) дані з форми будуть передані оброблювачу. Допустимі значення: METHOD=POST і METHOD=GET. Якщо значення атрибута не встановлене, за умовчанням передбачається METHOD=GET.

- ENCTYPE. Визначає, яким чином дані з форми будуть закодовані для передачі оброблювачу. Якщо значення атрибута не встановлено, за умовчанням передбачається ENCTYPE=application/x-www-form-urlencoded.

Форма складається з одного або кількох елементів управління.

Елементи управління – це елементи інтерфейсу, що дозволяють користувачеві взаємодіяти з HTML-документами.

Тег, зв'язаний з елементами керування, – INPUT.

#### **Текстове поле**

```
<INPUT TYPE="text" NAME="var-name" VALUE="рядок за умовчанням"
SIZE="розмір" MAXLENGTH="макс_довжина">
```

Опція TYPE дозволяє вказувати, який саме тип керуючого елемента описується. Кожен тип має своє унікальне ім'я.

Опція NAME вказує ім'я змінної, у якій система зберігатиме отримане значення. Ім'я не повинне містити в собі пробілів.

Опція VALUE (необов'язкова) дозволяє вам визначити текст, що поміщається в рядок введення при відображенні форми.

Опція SIZE визначає максимальну кількість символів, що може побачити при введенні користувач. Ця опція працює разом з опцією MAXLENGTH, що визначає кількість символів, що вводиться користувачем. Якщо значення MAXLENGTH більше ніж значення SIZE і якщо користувач уведе більше символів, ніж зазначено в SIZE, то текст буде прокручуватися по горизонталі.

На формі можна використати такі елементи управління:

Елементи управління	Опис
checkbox	Керуючий елемент – перемикач (прапорець перевірки)
hidden	Прихований текст, що використовують для обміну даними між формами
password	Рядок вводу пароля
reset	Кнопка «Reset»
submit	Кнопка «Submit»
text	Рядок для вводу тексту

Приклад форми з одним рядком вводу:

```
<HTML>
```

```
<HEAD> <TITLE> Це приклад </TITLE> </HEAD>
```

```
<BODY>
```

```
<FORM METHOD="POST" ACTION="/cgi-bin/data">
```

Введіть ваші дані:

```
<INPUT TYPE="text" NAME="username" VALUE="Іванов І.І." SIZE="25"
MAXLENGTH="45">
```

```
</FORM> </BODY> </HTML>
```

Результат:

Введіть ваші дані: <input type="text" value="Іванов І.І."/>
---

### ***Введення пароля***

Рядок пароля забезпечує певний захист даних, оскільки він змінює символи, що вводяться до нього, на зірочки. Визначення рядка введення пароля дуже схоже на визначення рядка введення тексту, за винятком значення опції TYPE.

Стандартний вигляд визначення рядка введення пароля виглядає так:

```
<INPUT TYPE="password"
```

```
NAME="var-name"
```

```
VALUE="рядок за умовчанням" SIZE="25" MAXLENGTH="15">
```

Опція NAME вказує на ім'я змінної, у якій зберігатимуться введені дані.

Опція SIZE визначає максимальну кількість символів, що може побачити користувач при введенні.

### ***Введення декількох рядків тексту***

Якщо потрібно ввести великий текст, то використовується область для введення тексту. Вона визначається тегом

```
<TEXTAREA NAME="var-name" ROWS="4" COLS="80">
```

Текст, за умовчанням відображуваний в області введення

```
</TEXTAREA>
```

Опція NAME вказує на ім'я змінної, у якій зберігатимуться отримані дані, ROWS і COLS визначають розмір області в рядках і в стовпцях відповідно. Текст, який ви хочете відобразити в області введення за умовчанням, має знаходитися між початковим тегом <TEXTAREA> і завершальним </TEXTAREA>.

Приклад форми, що запитує у користувача кілька рядків тексту:

```
<TEXTAREA NAME="MyText" ROWS=8 COLS=80>
```

Текст, відображуваний за умовчанням

```
</TEXTAREA>
```

Результат: на екрані відобразиться поле вводу, що містить текст, заданий за умовчанням, і має лінійку вертикального скролінга.



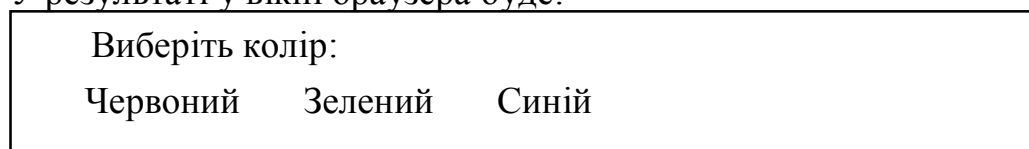
### **Перемикачі-радіокнопки**

Перемикачі-радіокнопки дуже схожі на перемикачі, з тією різницею, що одночасно може бути обрана тільки одна кнопка з групи. Для опису кнопок використовується тег <INPUT> з опцією TYPE="radio".

Кожен тег <INPUT> визначає окрему кнопку. Елементи, які мають однакове значення опції NAME, поєднуються в групу. Опція VALUE є обов'язковою. Вона описує значення, що повертається. За умовчанням жодна з кнопок не є позначеною. Можна вказати опцію CHECKED для однієї з кнопок, щоб зробити її позначеною за умовчанням:

```
<FORM METHOD="POST"
ACTION="/cgi-bin/data"> Виберіть колір:
<P> <INPUT TYPE="radio"
NAME="var-name"
VALUE="red" CHECKED> Червоний
<INPUT TYPE="radio"
NAME="var-name"
VALUE="grn"> Зелений
<INPUT TYPE="radio"
NAME="var-name"
VALUE="blu"> Синій
</FORM>
```

У результаті у вікні браузера буде:



Опцію CHECKED можна вказувати тільки для однієї кнопки в кожній групі.

### **Спливаючі меню**

Спливаюче меню дає можливість користувачеві вибрати один або кілька елементів з меню. Меню визначається усередині тега <SELECT> ... </SELECT>.

Кожен пункт меню починається тегом <OPTION>. Як правило, перший у списку елемент є елементом за умовчанням, однак можна змінити елемент за умовчанням, вказавши опцію SELECTED.

```
<FORM METHOD="POST" ACTION="/cgi-bin/data">
```

Виберіть елемент:

```
<SELECT NAME="var-name"> <OPTION> Перший <OPTION> Другий  
<OPTION> Третій <OPTION SELECTED> Четвертий <OPTION> П'ятий  
</SELECT>  
</FORM>
```

#### **4. Порядок виконання роботи**

1. Перед початком виконання роботи необхідно скласти вихідний код створеного документа відповідно до отриманого від викладача індивідуального завдання. **Можна використовувати варіанти індивідуальних завдань з першої роботи.**

2. Доповнити свій сайт web-підручником чи довідником з тестовими завданнями і можливістю контролю знань. Для цього використати форми.

3. Налаштувати текст документа в браузерах MS IE і Netscape Navigator.

4. Підготувати докладний звіт про виконану роботу.

#### **5. Контрольні запитання**

1. У чому суть гіпертекстової форми представлення web-документів?

2. Які особливості синтаксису мови гіпертекстових посилань?

3. Які групи тегів ви знаєте?

4. Як на основі HTML відображаються заголовки, шрифти, списки, задалегідь форматований текст, гіперпосилання, фрейми, діалогові форми, меню, карти зображення?

5. Які способи використання стилів у документі web?

#### **6. Рекомендована література**

1. Айзенменгер Р. HTML 3.2/4.0 : справочник : пер. с нем. / Р. Айзенменгер. — М. : БИНОМ, 1998. — 368 с.

2. Айзекс С. Dynamic HTML / С. Айзекс. — СПб. : BHV, 1999. — 496 с.

3. Соколов С. А. HTML и CSS : в примерах, типовых решениях и задачах / С. А. Соколов. — М. : Вильямс, 2007. — 416 с. — Профессиональная работа.

4. HTML, XHTML и CSS / Брайан Пфаффенбергер, Стивен Шафер, Чак Уайт, Билл Кароу. — 3-е изд. — М. : Вильямс, 2009. — 752 с. — Библия пользователя.

### **ПРАКТИЧНЕ ЗАНЯТТЯ № 3**

#### **Метадані та CSS, використання фільтрів**

##### **1. Мета роботи**

Вивчення особливостей стильового оформлення web-проекту, способів створення таблиць стилів, використання фільтрів CSS, управління web-документом за допомогою мета-тегів.

## 2. Підготовка до роботи

Перед заняттям слід повторити матеріал за курсом лекцій та рекомендованою літературою.

## 3. Методичні вказівки до виконання роботи

### 3.1. Використання таблиць стилів

Способи визначення (опис) CSS:

1) Визначення стилю за допомогою тега безпосередньо в документі.

<STYLE> Текст опису стилю </STYLE>

Ім'я\_об'єкта {властивість: значення; властивість: значення}

Приклад: BODY {font-size: 14pt}

2) Розміщення опису стилю в самому тегові.

<ім'я\_тегу style="властивість: значення; властивість: значення... ">

Приклад: <h1 style="color: red">

3) Імпорт опису стилю в документ із зовнішнього файлу.

Опис стилю може зберігатися в зовнішньому текстовому файлі, що є набором рядків з описувачами відповідного формату:

ІМ'Я\_КОНТЕЙНЕРА {опис стилю}

Файл: *css.htm*, але краще *ім'я\_стилю.css*

Посилання на опис стилю, розташованого за межами документа, може здійснюватися двома способами:

- за допомогою тега <link>.

Контейнер <link> призначений для зазначення відносних зв'язків між документами. Має два обов'язкових атрибути:

```
<HEAD>
```

```
<link rel="URL">
```

```
<link rev="URL">
```

```
<link type="text/CSS" rel="url"/"stylesheet" href="url">
```

```
</HEAD>
```

- за допомогою оператора @import:

```
@import: url (http://.../1.css)
```

```
H1 {color: red}.
```

### **Використання тега <style>**

Тег <style> може використовуватися як самостійний контейнер або бути атрибутом іншого контейнера. Якщо тег <style> використовується як самостійний, то він має такий синтаксис:

```
<style>
```

```
selector {властивість: значення; властивість: значення...}
```

```
</style>
```

В описі стилю зазвичай задають стилі для тегів html, стилі для класів об'єктів (задаються атрибутом class), стилі для об'єктів із заданим унікальним ідентифікатором (для цього використовується атрибут ID, що має унікальне значення), стилі для класів тега <A>.

```
A{color: black} // чорний колір гіперпосилань
```

```
.samesize{font-size: 12pt} // розмір шрифту для всіх об'єктів, у яких один з атрибутів class="samesize"
```



Атрибут *class* дозволяє застосовувати однакове оформлення до різнотипних об'єктів.

Якщо необхідно задавати однакові стилі неоднорідним тегам розмітки, їх перелічують через кому:

*h1, h2, h3. myorigin {color: green}*

Якщо необхідно застосувати вкладені (каскадні) визначення стилів, тоді ім'я селектора задається за таким синтаксисом:

*ім'я\_контейнера1 ім'я\_контейнера2...ім'я\_контейнера N {...}*, де N – показник ступеня вкладеності.

Наприклад:

Для вкладеного тегу, що визначає список

```
<ul>
<li><A href...><img src=...></img></A>
<li>
</ul>
```

визначення стилю має вигляд

```
ul li A img {src: url}.
```

Стилі контейнерів можуть успадковуватися вкладеними об'єктами, при цьому об'єкти успадковують стилі тільки в тому випадку, якщо вони не мають стилю за умовчанням і якщо для них не перевизначені властивості батьківського контейнера.

```
<style>
P {color: green}
I {font-size: 16pt}
</style>
```

Розміщений усередині параграфа контейнер `<I>` Текст `</I>` відобразитиметься зеленим кольором.

```
<style>
P {color: green}
I {font-size: 16pt color: red}
</style>
```

У даному ж випадку контейнер `<I>` відобразатиметься червоним кольором, тому що відбулося перевизначення.

### ***Посилання на зовнішній опис стилю***

При посиланні на зовнішній опис можливі різноманітні комбінації опису стилів. У цьому випадку діють правила старшинства стилів:

- 1) Спочатку застосовуються стилі за умовчанням (стилі установки браузера).
- 2) Стилі за умовчанням визначаються приєднаними стилями (тег `<link>` у розділі `<head>`).
- 3) Приєднані стилі визначаються описувачами в тегові `<style>`.
- 4) Стилі тега `<style>` визначаються атрибутом `style` у будь-якому тегові.

### ***Блокові і рядкові елементи***

Блоковий елемент розмітки відображається всередині прямокутного фрагмента, що знаходиться на окремому рядку і відділений від попереднього і наступного фрагментів. Рядковий елемент не займає окремого рядка, і навколо

нього знаходяться інші елементи. Заголовок – це блоковий елемент, а курсив – рядковий. Блокові елементи можна вкладати, а рядкові вкладати і перетинати.

Для стильового узагальнення рядкових і блокових елементів виступають теги `<div>` і `<span>`.

Контейнер `<div>` є універсальним блоковим елементом. Його властивості:

- завжди відокремлюється від інших елементів порожнім рядком;
- не несе ніякого значеннєвого навантаження, а є роздільником сторінок.

Якщо не задана *css*, то контейнер `<div>` просто створює порожній рядок тексту;

• контейнер `<div>` дозволяє застосовувати атрибути стилю, пов'язані з границею блоків (відступ блока від границь старшого елемента).

Якщо браузер не підтримує *css*, то застосовувати `<div>` не рекомендується.

Контейнер `<span>` є узагальненим рядковим контейнером розмітки. Застосовується для заміни тегів `<I>`, `<B>`, `<U>` та інших тегів, пов'язаних з форматом шрифту.

`<I> Текст </I>` замінюється на `<span style="font-style: italic"> Текст </span>`

`<U> Текст </U>` замінюється на

`<span style="font-decoration: underline"> Текст </span>`

або на `<span class="decor">`, а в таблиці стилів

`span.decor{font-decoration: underline}`

Застосування тега `<span>` обмежується браузерами, що підтримують *css*.

### **Властивості CSS**

Усі змінювані властивості поділяються на 5 груп.

1) Властивості для визначення кольору і фону.

<b>Властивість</b>	<b>Опис властивості</b>	<b>Значення властивості</b>
<b>Color</b>	Визначає колір	color: колір / transparent (прозорий)
<b>Background-color</b>	Визначає колір фону	background-color: колір / transparent
<b>Background-image</b>	Фонове зображення	background-image: URL / "none"
<b>Background-repeat</b>	Для повторення фонового зображення	background-repeat: repeat / repeat_x / repeat_y / no_repeat
<b>Background-attachment</b>	Дозволяє або забороняє рух картинки зі скролом	background-attachment: fixed / scroll
<b>Background-position</b>	Визначає положення фонові картини	background-position: bottom / center / top / left / right / % від ширини екрана

2) Властивості шрифтів і тексту.

<b>font-family</b>	Назва шрифту	Font-family: Arial, TimesNewRoman тощо
<b>font-style</b>	Вид шрифту	Font-style: normal / italic
<b>font-variant</b>	Управляє регістром шрифту	Font-variant: normal / caps / small

<b>font-size</b>	Розмір шрифту	Font-size: 12pt / 12px
<b>font-weight</b>	Жирність шрифту	Font-weight: bold
<b>Letter-spacing</b>	Установлює відстань між символами	Letter-spacing: normal / довжина в пікселях
<b>vertical-align</b>	Вирівнювання рядкових елементів	Vertical-align: sub / super

3) Властивості списків.

**List-style-type: square/circle** – визначення стилю маркерів у списку.

4) Властивості блокових елементів.

<b>margin-top</b> <b>-right</b> <b>-left</b> <b>-bottom</b>	Зовнішні відступи блокового елемента
<b>padding-top</b> <b>-right</b> <b>-left</b> <b>-bottom</b>	Внутрішні відступи блокового елемента
<b>border-top-width</b> <b>border-right-width</b> <b>border-left-width</b>	Для визначення ширини відповідних границь блокового елемента
<b>border-width</b>	Для визначення товщини всіх границь

5) Властивості позиціонування об'єктів.

**height** – висота елемента.

**width** – ширина елемента.

**overflow** – визначення видимості елемента, якщо вміст перевищує розміри границь елемента. Може мати значення scroll, hidden, visible, auto.

3.2. Використання фільтрів CSS (у Internet Explorer)

DHTML-фільтри:

1) Фільтр FlipH перевертає об'єкт горизонтально.

Синтаксис: `STYLE="filter:Flip"`

2) Фільтр FlipV перевертає об'єкт вертикально.

3) Фільтр Glow – об'єкт сяє.

Синтаксис: `STYLE="filter:Glow(Strength=strength, Color=color;)"`,

де color – колір, яким засяє текст, strength – сила, з якою він засяє (0–100)

4) Фільтр Wave робить об'єкт хвилястим.

Синтаксис: `filter: Wave(Freq=freq, Add=add, LightStrength=strength, Phase=phase, Strength=strength)`, де

Freq – кількість хвиль;

Add – накладати чи ні оригінал об'єкта;

LightStrength – сила хвилі;

Phase – кут хвилі;

Strength – інтенсивність хвилі.

Наприклад: Створити скрипт для освітлення малюнка при наведенні мишки.

```

<head>
<script language="JavaScript">
function makevisible(cur,which)
{if (which==0)
cur.filters.alpha.opacity=100
else cur.filters.alpha.opacity=20}
</script>
</head>
<body>
<a href="my.htm">  </a> </body>

```

### 3.3.3. Використання метаданих

#### **Створення мета-тегів**

Пошукові сервери під час реєстрації web-сервера, сайта або сторінки використовують інформацію, що зберігається в спеціальних тегах <META>.

Рекомендації зі складання мета-тегів.

Від правильності складання мета-тегів залежить легкість знаходження сайта за допомогою пошукових систем. Для цього необхідно дотримуватись таких правил:

1) довжина вмісту мета-тегів «descriptions» не повинна перевищувати 200 символів, а «keywords» – 1000 символів. Це пов'язано з тим, що пошукові сервери, як правило, використовують саме такі величини при індексуванні інформації про сайт. Пошукові машини відкинуть частину опису, що перевищує встановлені норми, у результаті чого до каталогу потрапить не вся бажана інформація, або сервер просто відкине реєстрацію сайта. Пошукові машини, як правило, працюють з мета-тегами, а у випадку їхньої відсутності уже із вмістом сторінки;

2) мета-теги мають два можливих атрибути:

```
<META HTTP-EQUIV="ім'я" CONTENT="вміст">
```

```
<META NAME="ім'я" CONTENT="вміст">;
```

3) мета-теги мають знаходитися в заголовку HTML-документа між <HEAD> і </HEAD> (особливо це важливо для документів, що використовують фрейми).

#### **Складання мета-тегів**

1) Для того, щоб при завантаженні документа браузер брав свіжу версію документа, а не версію з кеша, використовується

```
<META HTTP-EQUIV="Expires" Content="Mon, 13 Nov 2004 00:00:01 GMT">
```

2) Для визначення браузером мови, якою написана сторінка, використовується

```
<META HTTP-EQUIV="Content-Type" Content="text/html; charset=Windows-1251">
```

3) Для зазначення імені автора і додаткової інформації про нього використовується

```
<META Name="autor" Content="Ваші_П.І.П., e-mail...">
```

- 4) Для зазначення авторських прав використовується  
<META Name="copyright" Content="Ваша\_фірма...">
- 5) Для зазначення ключових слів і термінів використовується  
<META Name="keywords" Content="Ваші\_ключові\_слова">
- 6) Для зазначення короткого опису використовується  
<META Name="description" Content="Опис\_Вашої\_сторінки">

### **Тег <LINK>**

Тег <LINK> надає документу незалежний від середовища метод визначення відношення даного документа до інших документів і ресурсів мережі. Використовується з аргументами REL і REV. За допомогою тега <LINK> можна:

- створювати в документі спеціальні навігаційні кнопки або меню;
- прив'язувати такі асоційовані ресурси, як таблиці стилів і скрипти;
- надавати альтернативні форми для даного документа.

Наприклад: <LINK rel=help href="http://www.name.com/help.html">, де http://www.name.com/help.html – сторінка допомоги для даного документа.

Атрибути REL і REV вказують на тип зв'язку – зв'язок «уперед» або зв'язок «назад» і можуть також використовуватися з тегом <A>. Теги LINK можуть використовуватися тільки в заголовку документа (у середині тега <HEAD>).

## **4. Порядок виконання роботи**

1. Перед початком виконання роботи необхідно скласти вихідний код створеного документа відповідно до отриманого від викладача індивідуального завдання. **Можна використовувати варіанти індивідуальних завдань з першої роботи.**

2. Налаштувати текст документа в браузерях MS IE і Netscape Navigator.
3. Додати до створеного проекту фільтри IE за додатковим завданнями з п. 3.
4. Підготувати докладний звіт про виконану роботу.

## **5 Додаткові завдання**

Реалізувати фільтри:

- 1) ліхтар маяка (Світіння);
- 2) миготливий текст (Альфа);
- 3) календарик, що переливається (Альфа на двох малюнках);
- 4) «хвилястий» малюнок (Хвиля);
- 5) мультфільм (птаха, що летить) (Альфа на масиві малюнків);
- 6) презентація (ефект розкриття чи проявлення слайдів *revealtrans*);
- 7) градієнтне розфарбовування (Альфа для окремих частин градієнтно замальованої фігури).

## **6. Контрольні запитання**

1. Як скласти і підключити таблицю стилів?
2. Які особливості використання мета-тегів?
3. Які способи використання стилів у документі web?
4. Як використовувати вбудовані фільтри DHTML?
5. Які є способи обробки шрифтів?

## **7. Рекомендована література**

1. Основы WEB-технологий / П. Б. Храмцов, С. А. Брик, А. М. Русак, А. И. Сурин. — М. : Интернет-университет информационных технологий — ИНТУИТ.ру, 2003. — 512 с.
2. Янг М. Л. Internet. Полное руководство : пер. с англ. / Маргарет Левин Янг. — К. : Издательская группа BHV, 2001. — 864 с.
3. Айзекс С. Dynamic HTML / С. Айзекс. — СПб. : BHV, 1999. — 496 с.
4. Спейнауэр С. Справочник Web-мастера / Стивен Спейнауэр, Валери Куэрсиса. — К. : BHV, 1997. — 368 с.
5. Соколов С. А. HTML и CSS : в примерах, типовых решениях и задачах / С. А. Соколов. — М. : Вильямс, 2007. — 416 с. — Профессиональная работа.
6. Фролов А. В. Сервер Web своими руками / Александр Фролов, Григорий Фролов. — М. : Диалог-МИФИ, 1998. — 288 с. — (Библиотека системного программиста ; т. 29).
7. HTML, XHTML и CSS / Брайан Пфаффенбергер, Стивен Шафер, Чак Уайт, Билл Кароу. — 3-е изд. — М. : Вильямс, 2009. — 752 с. — Библия пользователя.

## **ПРАКТИЧНЕ ЗАНЯТТЯ № 4** **Створення сценаріїв на JavaScript**

### **1. Мета роботи**

Вивчення способів створення сценаріїв у web-документах, мови JavaScript.

### **2. Підготовка до роботи**

Перед роботою слід повторити матеріал за курсом лекцій та за рекомендованою літературою.

### **3. Методичні вказівки до виконання роботи**

#### 3.1. Елемент script

Сценарії на сторінці можуть бути зв'язані з документом через використання одного з трьох методів. Найбільш загальний метод полягає в розміщенні програми в елемент script. (Два інших методи розміщують програму в окремий файл і посилаються на нього за допомогою тега <script> або розміщують програму в атрибут події в іншому тегові).

Елемент script є контейнером для скрипкової програми. Він може містити програму, що знаходиться в документі, або посилатися на зовнішній файл. Сценарії, що знаходяться усередині елемента script, можуть бути зв'язані з елементом за допомогою програми, спеціальних атрибутів елемента script або за допомогою залежного від мови механізму. Індивідуальні елементи можуть мати сценарії, що зв'язані з ними безпосередньо атрибутами подій, представлених у самому елементі.

Синтаксис елемента script:

```
<SCRIPT LANGUAGE="language JavaScript">
```

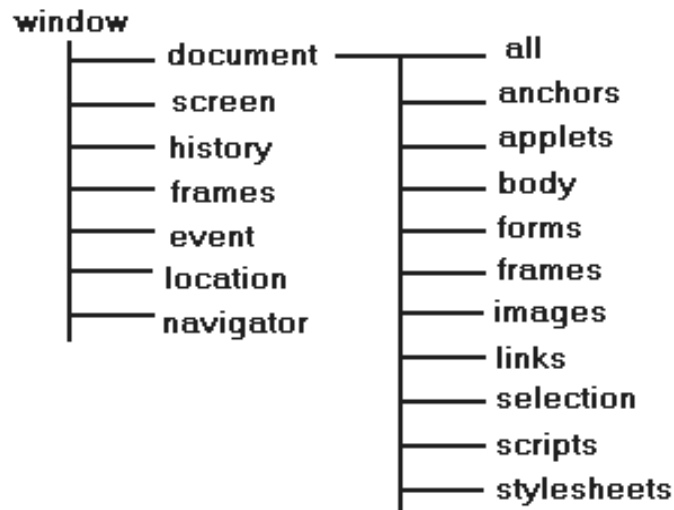
тіло скрипта

```
</SCRIPT>
```

### 3.2. Модель об'єктів JavaScript

Усі операції, які можна виконувати в програмі на JavaScript, описують дії над добре відомими і зрозумілими об'єктами, якими є елементи робочої області браузера і контейнери мови HTML. Є об'єкти з набором властивостей і набір функцій над об'єктами. Останні називаються методами. У JavaScript є події аналогічні до програмних переривань. Ці події також орієнтовані на роботу в World Wide Web, наприклад, завантаження сторінки в робочу область браузера або вибір гіпертекстового посилання.

Усі вбудовані об'єкти браузера Internet Explorer можна подати у вигляді такої ієрархії:



Для Netscape об'єктна модель має такий вигляд:

Більшість програм використовують цю систему класів і не створюють нових.

### 3.3. Вбудовані об'єкти і функції JavaScript

Елементи мови:

break, continue, for, for... in, function, if... else, new, return, var, while, with.

**Елемент function** використовується для створення власних функцій і об'єктів. Функції являють собою підпрограми, що можуть викликатися з коду на JavaScript. Наприклад, можна створити функцію для перевірки правильності заповнення полів форми перед її відсиланням на сервер.

```
<script language="JavaScript">
function SendForm()
{
if(Check())window.document.forms[0].submit;
```

```

    }
function Check()
    {
    var doc = window.document;
    if (doc.forms[0].elements[0].value == "" ||
    doc.forms[0].elements[1].value == "" ||
    doc.forms[0].elements[2].value == "")
    {
    alert('Поля не можуть бути порожніми');
    return false;
    }
else
return true;
    }
</script>

```

У прикладі функція Check виконує перевірку перших трьох полів форми і, якщо хоча б одне з них не містить інформації, виводить попереджувальне повідомлення. Функція SendForm (у якій викликається функція перевірки Check) «прив'язується» до форми в такий спосіб:

```

<form name="rqform" action="request.dll/doform" method="post">
<b> введіть інформацію:</b>
<input type="Submit" value="Надіслати"
onClick="return SendForm()">
<input type="Reset" value="Відмінити"
onClick="history.back(-1) "
</form>

```

В оброблювачі натискання кнопки «Надіслати» вказано, що має викликатися функція SendForm:

```
onClick="return SendForm()"
```

На відміну від об'єктів, створених користувачами, і об'єктів, що складають об'єктну модель браузера, вбудовані об'єкти забезпечуються самим інтерпретатором мови: чи то інтерпретатор, що входить до складу браузера Інтернет Explorer, чи то інтерпретатор, що входить до складу Netscape Navigator.

До основних понять мови відносяться: **об'єкт, властивість, метод, подія і колекція.**

Об'єкт – це сукупність властивостей, методів, подій і колекцій, надана браузером у рамках об'єктної моделі. Усі об'єкти створюються самим браузером, і доступ до них здійснюється через екземпляр за схемою:

**об'єкт.метод | подія | колекція | властивість**

Властивість – змінна в рамках об'єкта, що може використовуватися для одержання якихось значень або встановлення нових. Ряд властивостей може бути доступний тільки для читання.

Метод – це процедура або функція, надана об'єктом для виконання яких-небудь дій або управління властивостями об'єкта.



Подія – це будь-яка дія користувача або момент роботи браузера. Для реакції на події створюються оброблювачі подій.

Колекція – упорядкований набір властивостей, схожий на масив, доступ до якого здійснюється спеціальними засобами.

### 3.4. Базові події

onBlur	onMouseOver
onChange	onSelect
onClick	onSubmit
onFocus	onUnload
onLoad	

Приклад роботи з подіями.

При наведенні мишкою на текст «Завершити роботу браузера», браузер закриється.

```
<a href="url" onMouseOver="window.close()">
```

Завершити роботу браузера

```
</a>
```

### 3.5. Робота з об'єктом *document*

Наведемо приклад скрипта, що виводить дату поновлення документа при кожній його зміні.

```
<SCRIPT LANGUAGE="JavaScript">
```

```
document.writeln(document.lastModified)
```

```
</SCRIPT>
```

### 3.6. Робота з об'єктом *Date*

Об'єкт *Date* і його методи використовуються для роботи з датою і часом у скриптових програмах. Цей об'єкт має великий набір методів для встановлення дати, одержання її значення і виконання різних перетворень. Об'єкт *Date* не має властивостей. Дата в мові JavaScript являє собою число мілісекунд, що пройшли з 1 січня 1970 року.

Для створення екземпляра об'єкта *Date* використовується конструктор *new*:

```
MyDate = new Date([параметри])
```

Можливі такі параметри:

- ніяких параметрів — екземпляр міститиме поточну дату і час. Наприклад, `today = new Date();`

- рядок, що являє собою дату в такому форматі:

«Місяць, день, рік».

Наприклад, `someDate = new Date("May, 15, 1999")`.

Якщо число годин, хвилин або секунд не зазначено, їхні значення дорівнюють 0;

- набір цілочислових значень для року, місяця і дня.

Наприклад, `otherDay = new Date(99, 5, 15);`

- набір цілочислових значень для року, місяця, дня, годин, хвилин і секунд.

Наприклад, `someDay = new Date (96, 4, 15, 15, 30, 0)`.

## Методи об'єкта Date

Метод	Опис
getDate	Повертає день місяця як ціле число від 1 до 31
getDay	Повертає день тижня як ціле число від 0 (неділя) до 6 (субота)
getHours	Повертає число годин як ціле від 0 до 23
getMinutes	Повертає число хвилин як ціле від 0 до 59
getMonth	Повертає номер місяця як ціле від 0 (січень) до 11 (грудень)
getSeconds	Повертає число секунд як ціле від 0 до 59
getTime	Повертає число мілісекунд між 1 січня 1970 року (00 00 00) і датою, заданою об'єктом Date
getTimeZoneOffset	Повертає число хвилин, що складають різницю між локальним часом і часом за Грінвичем
getYear	Повертає дві останні цифри року
Parse	Повертає число мілісекунд між 1 січня 1970 року (00 00 00) і датою, заданою у вигляді рядка
setDate	Встановлює день місяця
setHours	Встановлює число годин
setMinutes	Встановлює число хвилин
setMonth	Встановлює номер місяця
setSeconds	Встановлює число секунд
setTime	Встановлює час
setYear	Встановлює рік
toGMTString	Перетворить локальний час за Грінвичем і повертає його у вигляді рядка
toLocaleString	Перетворить час за Грінвичем у локальний час і повертає його у вигляді рядка
UTC	Повертає число мілісекунд між 1 січня 1970 року, (00 00 00) і датою, заданою у вигляді параметра

Усі методи використовують такий формат даних:

Значення	Діапазон
Число секунд і хвилин	0...59
Число годин	0...23
День тижня	0...6
Дата	1...31
Місяць	0...11 (Січень...Грудень)
Рік	Число років з 1990

#### **4. Завдання для виконання**

Для створення JavaScript-програм використовується один з редакторів HTML.

Для налагодження необхідна наявність кількох типів браузерів.

Перед початком виконання роботи необхідно скласти вихідний код створюваного документа відповідно до отриманого від викладача індивідуального завдання.

**Можна використовувати варіанти індивідуальних завдань з першої роботи.**

Необхідно розробити і помістити в проект такі функції на JavaScript:

- 1) функцію для підтвердження переходу за гіперпосиланням;
- 2) функції для динамічного керування фреймами;
- 3) функцію для керування завантаженням документа в залежності від властивостей об'єкта screen;
- 4) функцію для роботи з об'єктом Date (на вибір: годинник, дата, календар, лічильник часу тощо).

#### **5. Порядок виконання роботи**

1. Перед початком виконання роботи необхідно скласти вихідний текст програми на JavaScript відповідно до отриманого від викладача індивідуального завдання. **Можна використовувати варіанти індивідуальних завдань з першої роботи.**

2. Перевірити роботу документа зі скриптами у кількох браузерах.
3. Скласти докладний звіт про виконану роботу.

#### **6. Контрольні запитання**

1. У чому суть об'єктної моделі браузера?
2. Які особливості реалізації динамічного HTML?
3. Які способи запуску програм на JavaScript?
4. Які вбудовані об'єкти характерні для JavaScript?
5. Які особливості роботи скриптових програм?
6. Які базові події підтримуються в JavaScript?
7. Які існують способи створення оброблювачів подій?

#### **7. Рекомендована література**

1. Основы WEB-технологий / П. Б. Храмцов, С. А. Брик, А. М. Русак, А. И. Сурин. — М. : Интернет-университет информационных технологий — ИНТУИТ.ру, 2003. — 512 с.

2. Фролов А. В. Сервер Web своими руками / Александр Фролов, Григорий Фролов. — М. : Диалог-МИФИ, 1998. — 288 с. — (Библиотека системного программиста ; т. 29).

3. Баррет Д. JavaScript : web-профессионалам : пер. с англ. / Дэн Баррет, Дэн Ливингстон, Мика Браун. — К. : ВНУ, 2001. — 240 с.

4. Рейсиг Д. Инструменты отладки и тестирования // Рейсиг Д. JavaScript. Профессиональные приемы программирования / Джон Рейсиг ; пер. Н. Вильчинский. — СПб. : Питер, 2008. — С. 76—90. — (Библиотека программиста).

5. Флэнаган Д. JavaScript. Подробное руководство / Дэвид Флэнаган ; пер. А. Киселева. — 5-е изд. — СПб. : Символ-Плюс, 2008. — 992 с.

6. Хефлин Д. Разработка Web-скриптов / Д. Хефлин, Т. Ней. — СПб. : Питер, 2001. — 496 с. : ил. — (Библиотека программиста).

7. Кингсли-Хью Э. JavaScript 1.5 : учебный курс / Э. Кингсли-Хью, К. Кингсли-Хью. — СПб. : Питер, 2002. — 272 с. : ил.

## **ПРАКТИЧНЕ ЗАНЯТТЯ № 5**

### **Створення і використання бібліотек сценаріїв**

#### **1. Мета роботи**

Вивчення способів створення сценаріїв у web-документах, мови JavaScript, способів використання бібліотек скриптів.

#### **2. Підготовка до роботи**

Перед роботою слід повторити матеріал за курсом лекцій та за рекомендованою літературою.

#### **3. Методичні вказівки до виконання роботи**

##### 3.1. Бібліотеки JavaScript

Сценарії можуть знаходитися в зовнішньому файлі і бути зв'язані з будь-якою кількістю HTML-документів.

Залежно від браузера сторінки можуть бути кешовані, що збільшує продуктивність, оскільки загальні функції не потрібно записувати на кожну сторінку. Посилання на зовнішній сценарій здійснюються за допомогою атрибута SRC:

```
<SCRIPT LANGUAGE="JavaScript" SRC="File-name.js">
</SCRIPT>
```

##### 3.2. Створення банера

Банер – це «висяча» у сторінці реклама у вигляді графічного об'єкта, що є маршрутизатором гіперпосилання. Наведемо приклад скрипта, що змінює картинку банера при наведенні на неї мишею. Скрипт складається з двох частин. Перша частина вставляється в документ тільки один раз. А другу необхідно вставити стільки разів, скільки на сторінці буде малюнків, що змінюються.

Перша частина:

```
<SCRIPT LANGUAGE="JavaScript">
browser_name = navigator.appName;
browser_version = parseFloat(navigator.appVersion);
if (browser_name == "Netscape" && browser_version >= 3.0) { roll = 'true'; }
else if (browser_name == "Microsoft Internet Explorer" && browser_version >=
3.0) { roll = 'true'; }
else { roll = 'false'; }
function over(img,ref) { if (roll == 'true')
{ document.images[img].src = ref; } }
function out(img,ref) { if (roll == 'true') { document.images[img].src = ref; } }
if (roll == 'true')
{
a1=new Image;a1.src="image1.gif";
a2=new Image;a2.src="image2.gif";
...
aX=new Image;aX.src="imageX.gif";
}
</SCRIPT>
```

Наприкінці першої частини скрипта здійснюється підвантаження зображень натиснутих кнопок. Необхідно довантажити картинку натиснутої і не натиснутої кнопок.

...

```
a=new Image;a.src="image.gif";
```

Ці два рядки вказують те, що необхідно прописати підвантаження усіх малюнків, замість X має бути число.

Друга частина:

```
<A HREF="page.htm"
onMouseOver="over('image_name','image2.gif');"
onMouseOut="out('image_name','image1.gif');">

</A>
```

### 3.3. Забезпечення динамічної сумісності сторінки з браузером

Наведений нижче скрипт здійснює пересилання сторінки на іншу залежно від типу браузера, що використовує користувач:

```
<script language="JavaScript">
if (navigator.appName == "Netscape")
window.location.href = "index1.htm";
else
if (navigator.appName == "Microsoft Internet Explorer")
window.location.href = "index2.htm"; else window.location.href = "index3.htm";
</script>
```

### 3.4. Перевірка правильності заповнення формуляра

Задача: перевірити, чи правильно користувач вказав у полі вводу свою електронну адресу. У найпростішому випадку це вирішується перевіркою рядка на наявність символу @.

```
<script language="JavaScript">
function check()
{
if(document.myform.mytext.value.indexOf('@')== -1)
{
alert ("Неправильний формат e-mail адреси!");
return;
}}
</script>
```

Для роботи скрипта створюємо форму *myform*:

```
<form name=myform>
<input type=text size=20 name=mytext>
<input type=button value='Перевірити!'
onClick='check();'>
</form>
```

Якщо необхідно перевіряти кілька форм, то можна написати функцію один раз і викликати її з бібліотеки.

```

<script language="JavaScript">
function check(form_name, i, checked_text, error_message)
{
var myf=window.document;
if(myf.myform.elements[i].value.indexOf(checked_text)==-1)
{
alert (error_message);
return;
}
}
}
</script>

```

Приклад виклику у формі:

```

<FORM bgcolor="green" name="myform">
<INPUT TYPE="text" size=20 name="mytext">
<INPUT TYPE="text" NAME="mytext2">
<INPUT TYPE="text" NAME="mytext3">
<INPUT TYPE="text" NAME="mytext4">
<input type=button value="Перевірити!"
onClick="check('myform1', '0','@','Неправильна адреса!');check('myform1',
'1','***','Неправильні дані!'); ">
</FORM>

```

#### **4. Завдання для виконання**

1. Для створення JavaScript-програм використовується один з редакторів HTML. Для налагодження необхідна наявність кількох типів браузерів. Перед початком виконання роботи необхідно скласти вихідний код створюваного документа відповідно до отриманого від викладача індивідуального завдання.

**Можна використовувати варіанти індивідуальних завдань з першої роботи.**

2. Налагодити текст документа в браузерах MS IE і Netscape Navigator.
3. Необхідно розробити і помістити в проект такі сценарії на JavaScript:
  - 1) сценарій для перевірки типу браузера і динамічного завантаження сторінки;
  - 2) сценарій для перевірки правильності заповнення форми;
  - 3) сценарій для динамічного керування фреймами;
  - 4) сценарії для використання стандартних подій;
  - 5) сценарій для динамічного керування графічними об'єктами.
4. Виконати додаткові завдання з п. 3. При реалізації завдання для збереження параметрів налагодження використовувати Cookie, параметри передавати з іншої HTML-сторінки.

#### **5. Додаткові завдання**

- 1) реалізувати випадне меню (з налагодженням колірної гами);
- 2) створити тест (анкету) з реєстрацією та аналізом результатів;
- 3) меню, що розгортається (дерево), з налагодженням колірної гами; хрестики-нулики з реєстрацією та веденням рахунку гри;
- 4) снігопад (налагодження швидкості сніжинок, інтенсивність появи);

5) графічний годинник (стрілки мають рухатись за курсором або реагувати на прокрутку – прив'язка);

6) власна графічна прокрутка у вікна браузера;

7) адаптивний сайт – для кожного користувача запам'ятовується набір із 3 найбільш відвідуваних сторінок і виводиться у вікно «Швидкий перехід» для спрощення навігації);

8) гра «Життя» (задається розмір поля і клітинки) – 8 сусідів;

9) гра «Життя» (задається розмір поля і клітинки) – 6 сусідів;

10) шашки (реєстрація і ведення рахунку);

11) гра «П'ятнадцять» (реєстрація і ведення рахунку).

## **6. Порядок виконання роботи**

1. Перед початком виконання роботи необхідно скласти вихідний текст програми на JavaScript відповідно до отриманого від викладача індивідуального завдання. **Можна використовувати варіанти індивідуальних завдань з першої лабораторної роботи.**

2. Перевірити роботу документа зі скриптами на кількох браузерах.

3. Скласти докладний звіт про виконану роботу.

## **7. Контрольні запитання**

1. Як підключити бібліотеку скриптів?

2. Що таке банер і як його створювати?

3. Які є способи запуску програм на JavaScript?

4. Як забезпечується сумісність з різними типами браузера?

5. Які особливості роботи скриптових програм?

6. Які базові події підтримуються в JavaScript?

7. Які способи створення оброблювачів подій?

## **8. Рекомендована література**

1. Основы WEB-технологий / П. Б. Храмцов, С. А. Брик, А. М. Русак, А. И. Сурин. — М. : Интернет-университет информационных технологий — ИНТУИТ.ру, 2003. — 512 с.

2. Баррет Д. JavaScript : web-профессионалам : пер. с англ. / Дэн Баррет, Дэн Ливингстон, Мика Браун. — К. : ВНУ, 2001. — 240 с.

3. Рейсиг Д. Инструменты отладки и тестирования // Рейсиг Д. JavaScript. Профессиональные приемы программирования / Джон Рейсиг ; пер. Н. Вильчинский. — СПб. : Питер, 2008. — С. 76—90. — (Библиотека программиста).

4. Флэнаган Д. JavaScript. Подробное руководство / Дэвид Флэнаган ; пер. А. Киселева. — 5-е изд. — СПб. : Символ-Плюс, 2008. — 992 с.

5. Хефлин Д. Разработка Web-скриптов / Д. Хефлин, Т. Ней. — СПб. : Питер, 2001. — 496 с. : ил. — (Библиотека программиста).

6. Кингсли-Хью Э. JavaScript 1.5 : учебный курс / Э. Кингсли-Хью, К. Кингсли-Хью. — СПб. : Питер, 2002. — 272 с. : ил.

## **ПРАКТИЧНЕ ЗАНЯТТЯ № 6**

### **Створення XML-документів**

#### **1. Мета роботи**

Вивчення стандарту XML, засобів виконання базових операцій та особливостей створення XML-документів.

#### **2. Підготовка до роботи**

Перед лабораторною роботою слід повторити матеріал за курсом лекцій та за рекомендованою літературою.

#### **3. Методичні вказівки до виконання роботи**

##### **3.1. Синтаксис XML**

Розширювана мова розмітки (eXtensible Markup Language – XML) – це мова опису документів, схожа на мову розмітки гіпертексту (HyperText Markup Language – HTML). HTML – це мова, яка використовується для створення веб-сторінок і заснована на визначеному наборі тегів, що показують програмному забезпеченню («браузеру»), як представляти вміст сторінки. Подібно HTML, XML являє собою систему тегів, які описують компоненти документа. Насправді XML і HTML є підмножинами стандартної узагальненої мови розмітки (Standard Generalized Markup Language – SGML).

На відміну від HTML, XML дає користувачам можливість визначати теги, надаючи величезні можливості для опису структури і природи інформації, представленої документом. Це означає, що стандартні браузери не зможуть нічого зробити з цими розширеннями. Тому створення програмного забезпечення для XML є набагато більш складною справою.

У XML при відповідному визначенні правил можливі і такі дескриптори, як <ГРУПА> ПОАС-2003 </ГРУПА>. При цьому браузер одержує інформацію про те, що ПОАС-2003 є ГРУПА. Що таке ГРУПА, браузер довідається з файла правил.

За допомогою розширеної мови XML можна дуже просто використовувати інформацію для вибору визначеного стилю. Відкриваються нові перспективи, коли користувач не застосовує web-браузери як оброблювачі даних.

Другий приклад: електронна таблиця повинна за допомогою XML обмінюватися даними з базою даних. У цьому випадку таблиця передавалася б у такому вигляді:

```
<ПЕРСОНАЛ>  
<СПІВРОБІТНИКИ id="Студенти">  
<ПРИЗВИЩЕ> Барабанов </ПРИЗВИЩЕ>  
<ОСОБИСТИЙ НОМЕР> ПЗ03-1-23 </ОСОБИСТИЙ НОМЕР>  
<СТИПЕНДІЯ> 33 </СТИПЕНДІЯ>  
<ХОББИ> Музика </ХОББИ>  
<IMG SRC="1.jpg"> </СПІВРОБІТНИКИ >  
<ПРИЗВИЩЕ > Довгаль </ПРИЗВИЩЕ >  
<ОСОБИСТИЙ НОМЕР> ПЗ03-1-24 </ОСОБИСТИЙ НОМЕР>  
<СТИПЕНДІЯ> 33 </СТИПЕНДІЯ>
```



```
<IMG SRC="2.jpg">
</СПІВРОБІТНИКИ>
</ПЕРСОНАЛ>
```

Поля бази даних стають дескрипторами, які, як правило, «підігнані» до бази даних. У базі даних «Персонал» є поля «Співробітники», що містять нові дані, наприклад «Ім'я», «Особистий номер», «Стипендія», а також необов'язкові поля типу «Хобі». Є й фотографія співробітника — у вигляді гіперзв'язку. Щоб працювати з подібною структурою даних, має існувати файл правил, що визначає порядок роботи. Для цього на початок документа стандарту XML додаються рядки

```
<?XML VERSION="1.0"?>
<!DOCTYPE publisher SYSTEM "hnure.dtd">
```

Як і в HTML, кожен тег береться в кутові дужки (< >), і за ним зазвичай іде текст. За текстом має бути тег </...>. Тег може не мати змісту, і тоді за ним має бути кінцевий тег (<студент> </студент>, або тег може сам закінчуватися прямим слешем (<студент/>). Однак, на відміну від HTML, кінцевий тег потрібен завжди.

Коментарі задаються у формі <!-- ... -->. XML дозволяє використовувати коментар для оточення рядків коду, які бажано зробити недіючими.

### 3.2. DTD

Зміст тегу визначається в оголошенні типу документа (document type declaration – DTD). Це тіло коду, що визначає теги через набір елементів, наприклад:

```
<!DOCTYPE product [
  <!ELEMENT PRODUCT (product_id,
product_name, unit_of_measure?, specification*)>
  <!ELEMENT product_id (#PCDATA)>
  <!ELEMENT product_name (#PCDATA)>
  <!ELEMENT unit_of-measure (#PCDATA)>
  <!ELEMENT specification (variable, value)>
  <!ELEMENT variable (#PCDATA)>
  <!ELEMENT value (#PCDATA)>
]
```

DTD для XML-документа може бути або частиною документа, або зовнішнім файлом. Якщо це зовнішній файл, то все одно оператор DOCTYPE має бути присутнім у документі з аргументом SYSTEM -filename-, де filename є іменем файла, що утримує DTD. Наприклад, якби наведений вище DTD містився в зовнішньому файлі з ім'ям xxx.dtd, то оператор DOCTYPE повинен був би мати вид

```
<!DOCTYPE product SYSTEM xxx.dtd>
```

Такий самий рядок має бути першим рядком файла *xxx.dtd*.

Визначення елемента *product* включає список інших елементів, у даному випадку, *product\_id*, *product\_name*, *unit\_of\_measure* і *specification*. Знак ! після *unit\_of\_measure* означає, що може бути чи не бути одне входження; це необов'язковий елемент. Знак \* після *specification* означає, що елемент є не-

обов'язковим, але допускається одне чи більше число входжень. Якщо після якогось елемента списку присутній знак +, то це означає обов'язковість елемента, а також те, що допускається більше одного входження.

У свою чергу, кожний з елементів списку має бути визначений в одному з наступних рядків. #PCDATA означає, що тег міститиме тег, який може бути розібраний браузером.

### **Регістр**

У XML розрізняються символи верхнього і нижнього регістрів. Усі ключові слова XML складаються із символів верхнього регістра. Символи в іменах тегів мають набиратися на тому самому регістрі, що й у визначенні DTD. У попередньому прикладі імена сутностей/таблиць складаються із символів верхнього регістра, а імена атрибутів/стовпців із символів нижнього регістра. Можна було б зробити і по-іншому.

### **Атрибути**

У тегів можуть бути атрибути. Наприклад, замість перерахування відповідних тегів у визначенні `<!ELEMENT specification (variable, value)>` можна було б додати в DTD такі рядки:

```
<!ATTLIST specification variable CDATA #required> <!ATTLIST specification value CDATA #required>
```

Ці рядки створюють `variable` і `value` як два атрибути `specification`, так що вони не обов'язково з'являться у вигляді окремих елементів.

### 3.3. XML-документ існує в трьох рівнях коректності:

1) Правильно побудований XML-документ – це такий, у якому елементи правильно структуровані у вигляді дерева з коректно розставленими відкриваючими і закриваючими тегами.

2) Діючий XML-документ правильно побудований і містить теги, що відповідають оголошенню типу документа. Він містить тільки елементи і значення атрибутів, які відповідають DTD. Хоча XML-документ може готуватися і читатися без DTD, DTD важливий для встановлення дієвості.

3) Синтаксично коректний XML-документ знаходиться поза контролем XML. Розробник такого документа відповідає за його логічну структурування.

## **4. Порядок виконання роботи**

1. Створити XML-документ на тему «Кадровий облік університету».

2. Перевірити обробку створеного документа в браузері, що підтримує XML.

3. Створити XML-проект на одну з тем:

- навігаційне меню: перелік книжок для магазину/бібліотеки за розділами (пошук за тематикою, автором, ключовими словами);

- навігаційне меню: перелік товарів у магазині за відділами – вибір кошика покупця;

- розклад руху поїздів/літаків/автобусів на вокзалі/аеропорті/автовокзалі – пошук маршруту;

- дошка оголошень (у кожного учасника свій стиль) – пошук оголошення за необхідною тематикою;

- меню в кафе (формування замовлення);
- електронний словник (Simple Lingvo);
- тематичний класифікатор тексту (словник ключових слів за тематикою).

## **5. Контрольні запитання**

1. Назвіть властивості мови XML.
2. У чому відмінності XML від інших мов розмітки?
3. Опишіть структуру XML-документа.
4. Які рівні коректності XML-документа існують?

## **6. Рекомендована література**

1. HTML, XHTML и CSS / Брайан Пфаффенбергер, Стивен Шафер, Чак Уайт, Билл Кароу. — 3-е изд. — М. : Вильямс, 2009. — 752 с. — Библия пользователя.
2. Валентайн Ч. XHTML / Челси Валентайн, Крис Минник. — М. : Вильямс, 2001. — 480 с.

## **ПРАКТИЧНЕ ЗАНЯТТЯ № 7**

### ***Ознайомлення з інтерфейсом редактора Flash MX. Робота з кольором у Flash MX***

#### **1. Мета роботи**

Вивчення основних компонентів інтерфейсу Flash MX, способів його налаштування. Вивчення процесу вибору і налаштування кольору; засобів роботи з імпортованою графікою.

#### **2. Підготовка до роботи**

Перед роботою слід повторити матеріал за курсом лекцій та за рекомендованою літературою.

#### **3. Методичні вказівки до виконання роботи**

Інтерфейс редактора Flash MX відповідає загальнопоширеним стандартам GUI і тому схожий на безліч відомих програмних продуктів. На рис. 1 показаний його зовнішній вигляд. Інтерфейс редактора Flash MX складається з таких елементів:

- основного вікна, з якого створюється графіка. Воно також містить меню з усіма командами редактора Flash MX;
- стандартної панелі, де розташовані кнопки, що забезпечують швидке виконання команд, доступних через меню;
- панелі керування є для програвання майбутнього фільму прямо в редакторі;
- панелі інструментів, що містить кнопки інструментів редагування графіки;
- панелі хронометричної лінійки, що призначена для компоювання, монтажу фільму;
- панелі інспекторів, де редагуються параметри об'єктів фільму;
- вікна бібліотеки, у якому знаходяться створені користувачем для подальшого використання елементи фільму.

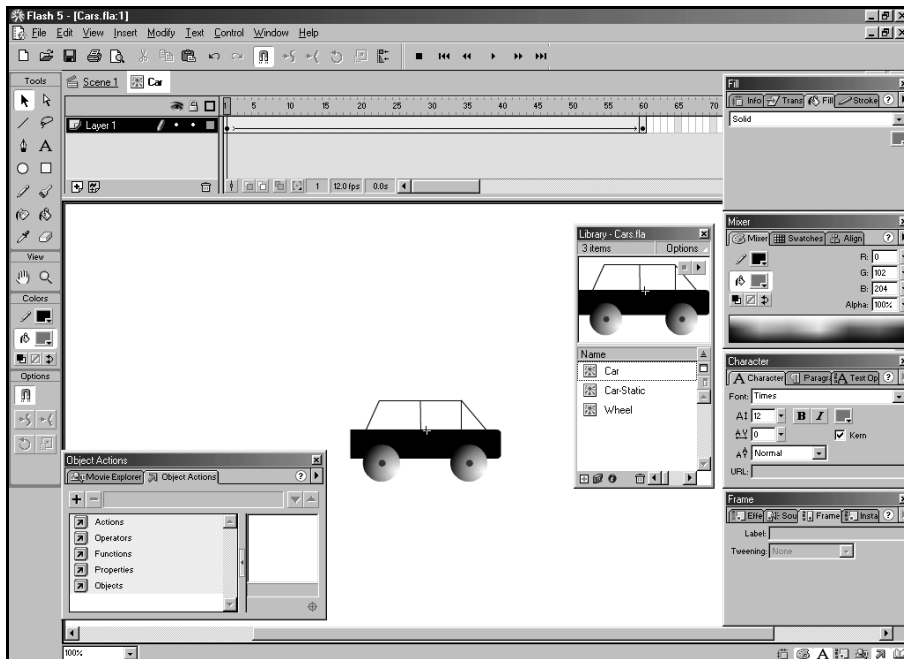


Рис. 1. Елементи інтерфейсу редактора Macromedia Flash MX

Ліворуч знаходяться панелі інструментів. У них можна вибирати інструменти, а також керувати робочою областю, модифікувати об'єкти і вибирати прості кольори. Праворуч знаходяться діалоги настроювання інструментів, кольору, тексту, властивості кадрів і об'єктів. Посередині – середовище програмування ActionScript, і бібліотека об'єктів фільму. На задньому фоні – робоча область, де ви будете діяти, а над нею шкала часу (Timeline) і слої, що дозволяють створювати складне багаторівневе зображення.

Для відображення усіх вторинних діалогів і редакторів використовується наступна панель, що знаходиться в правому нижньому куті інтерфейсу редактора Macromedia Flash MX (рис. 1):




Настроювання фільму, вікна редагування і вікна перегляду фільму можна змінювати за допомогою команд основного меню.

### 3.1. Форми у Flash MX

У Flash існує два типи графічних примітивів – Line, Curves (Лінії, Криві) і Shapes, Fills (Форми, Заливання). Найпростіший варіант лінії – відрізок прямої – характеризується координатами початкової і кінцевої точок, товщиною, кольором і стилем. Вигнута лінія на додаток до переліченого має такий параметр, як вигин. Друга пряма і вигнута лінії являють собою криву Без'є, вигин лінії задається коефіцієнтами формули, що описує криву. Усі математичні розрахунки приховані від очей дизайнера: він займається перетаскуванням кінців і «тіла» кривої. Точної роботи з кривою не передбачено, тобто не можна вручну задати значення коефіцієнтів (винятком є відрізки прямих, координати яких можна вказати за допомогою Інспектора об'єктів); усе доводиться робити «на око», відповідно до технології WYSIWYG (What You See Is What You Get – що бачимо, те й одержуємо).

Форму можна порівняти з мазком фарби. Контур форми описується такими ж кривими Без'є, що й лінія, і має той же набір атрибутів плюс колір заливання. Можна вважати форму замкнутою лінією, що має заповнення, заливання. Тоді різниця між цими двома примітивами стирається: вона полягає тільки в параметрах, атрибутах. При промальовуванні таких форм, як коло, отримуємо два мальовані об'єкти: лінія, що має форму кола, і заливання, що має форму кола і розташоване всередині кола. При цьому об'єкти зовсім самостійні, і ви можете видалити один з них, не побоюючись за долю іншого.

Лінії – найпростіші графічні елементи, які можна малювати в Flash. Лінії створюються двома інструментами: Line (Лінія) і Pencil (Олівець).

 – ця кнопка або відповідна їй «гаряча» клавіша N активізує інструмент Line. Залежно від налаштувань Assistant... курсор буде або рухатися плавно, або «скакати» при наближенні до ліній сітки. Якщо під час переміщення миші ви натиснете на клавішу Shift, то курсор, а разом з ним і кінець лінії, рухатиметься тільки по прямій, що проходить через початкову точку малювання і має нахил, кратний 45°. За допомогою інструмента Arrow (Стрілка) можна пересувати кінцеві точки лінії, змінювати вигин кривої, робити рівнобіжний перенос лінії.

Стиль лінії – це метод промальовування, яким Flash накреслить її. Визначених стилів небагато: суцільна лінія, пунктир, хвиляста лінія та ін. (Усі їх можна побачити на допоміжній панелі Stroke (Штрих)).

### ***Інструмент Pencil (Олівець)***

Pencil – це дуже зручний інструмент редактора Flash, що дозволяє малювати від руки. Налаштування інструмента Олівець Pencil Mode (Режим олівця) – вибір правила згладжування ліній при малюванні інструментом Олівець:

- кнопка Straighten (Рівні лінії). У цьому режимі лінії, з яких складається ескіз, сприймаються Flash як відрізки прямих, і редактор самостійно робить їхнє випрямлення;
- кнопка Smooth (Округлі лінії). У даному режимі малюнок буде згладжений, усі лінії сприймаються в першу чергу як криві;
- кнопка Ink (Чорнило). У даному режимі редактор Flash перетворює криву на векторну форму, максимально дотримуючись її вигину.

### ***Інструмент Oval (Овал)***

Oval призначений для малювання кіл і овалів. У редакторі Flash овал малюється не з центру. Точки, у яких ви почнете і закінчите малювання, є кутами прямокутника, описаного біля вашого овала. При малюванні овала створюється два графічних елементи: замкнута крива, що має форму овала, і заливання тієї ж форми. Разом вони являють обкреслений і залитий кольором овал. При визначенні кольору або заливання лінії овала у вікні палітри, крім звичайного набору кольорів, можна вибрати порожній квадратик (лівий верхній кут) – «порожній» колір. «Порожній» колір недоступний при виборі кольору для інструментів Лінія, Олівець та інших, він є тільки в інструментів Oval і Rectangle (Прямокутник).

### ***Інструмент Rectangle (Прямокутник)***

Інструмент Rectangle призначений для малювання прямокутників і квадратів. Створюються вони так, як і овал. Якщо під час малювання натиснути на клавішу Shift, отримаємо прямокутник з однаковими сторонами, тобто квадрат.

Настроювання інструмента Rectangle повторюють настроювання Oval – ті ж стиль і колір лінії, той же колір заливання. Для Прямокутника також доступний вибір «порожнього» кольору. Крім цього, у самому низі блока настроювань прямокутника розташована кнопка настроювання заокругленості кутів: кнопка Round Rectangle Radius (Радіус округлених кутів) відкриває діалогове вікно, де в поле введення вказується радіус кута в пунктах. Нульове значення відповідає відсутності заокругленості кутів. Максимальне значення радіуса – 999 пунктів.

Форму можна створити кількома способами, наприклад, намалювати її інструментами Прямокутник і Овал разом з лінією, що окантовує, або без такої. Форма також може бути створена інструментом Brush (Пензель). Нарешті, інструмент Paint Bucket (Заливання) призначений для заливання області, обмеженої кривими (область не обов'язково повинна бути замкнутою).

### ***Інструмент Brush (Пензель)***

Інструмент Brush багато в чому схожий на однойменний засіб у редакторах растрової графіки. Також спостерігається аналогія з інструментом Pencil, з тією різницею, що малюється не лінія, а форма. Під час малювання Пензлем слід за ним залишається всюди, де він пройшов (як у растровому редакторі). При відпусканні лівої кнопки миші, Flash визначить, яка частина форми повинна бути залишена (залежно від параметра Brush Mode).



– кнопка модифікатора Brush Mode (Режим пензля). При натисканні на неї з'являється випадне кнопочкове меню з такими пунктами:

- кнопка Paint Normal – нормальний режим. У ньому пензель малює поверх усіх графічних зображень. Якщо провести пензлем по порожньому простору, створюється нова форма;

- кнопка Paint Fills – заливання форм. Пензель зафарбовує тільки вже існуючі форми, але не лінії. Якщо провести ним по порожньому простору, створюється нова форма;

- кнопка Paint Behind – «зафарбування за», зовні. Не існуючі лінії та форми не зафарбовуються, нова форма виникає тільки на вільному від графіки місці;

- кнопка Paint Selection – зафарбування виділеного фрагмента. Пензель малюватиме тільки поверх виділених форм; лінії не зафарбовуються ні у вибраній області, ні за її межами. При проведенні пензлем по порожньому простору в цьому режимі сліду не залишається, оскільки в Flash не можна виділити порожній простір;

- кнопка Paint Inside – зафарбування форм зсередини. Зафарбування відбувається тільки в одній формі – у тій, де його почали. У цьому режимі пензель не залишає сліду на вільному просторі.

## 3.2. Робота з кольором


### ***Інструмент Ink Bottle (Чорнильниця)***




– ця кнопка на панелі інструментів або відповідна їй «гаряча» клавіша I активізує інструмент Ink Bottle, призначений для зміни параметрів уже намальованих ліній. Для настроювання нових параметрів лінії виберіть інструмент Ink Bottle, підведіть курсор до тієї лінії, настроювання якої бажаєте змінити, або форми, що ця лінія окреслює, і натисніть на ліву кнопку миші. Після натис-

нення лінія промальовується з новими параметрами – товщиною, кольором і стилем. Отже, лінія, що окантовує форму, завжди існує, просто при видаленні цієї лінії привласнюється деякий параметр «неіснуюча», після чого вона не відображається і начебто відсутня.

### **Інструмент Paint Bucket (Заливання)**

Натискання на кнопку  панелі інструментів або на відповідну «гарячу» клавішу U активізує інструмент Paint Bucket, що робить заливання області, обмеженої формами або кривими, а також створених заливань. Перший параметр – це колір заливання. Нижче розташоване випадне меню настроювання GapSize (Величина розривів). У цьому меню ви вказуєте, яких розмірів можуть бути зазори між елементами – формами і кривими, що обмежують область заливання. Зазначений параметр може набувати таких значень:

- Don't Close Gaps – зазори між обмежуючими елементами не допускаються. Це значення, звичайно, використовується для зафарбування вже намальованих форм або областей усередині замкнених груп ліній;
- Close Small Gaps – допускаються невеликі зазори;
- Close Medium Gaps – допускаються більш великі зазори;
- Close Large Gaps – допускаються великі зазори.

Далі ідуть дві кнопки, що відповідають режимам Lock Fill (Зафіксувати заливання) і Transform Fill (Модифікувати заливання).  – кнопка режиму Transform Fill, призначеного для редагування градієнтних і растрових заливань. Градієнтне заливання, звичайно, застосовується у флеш-фільмах для зображення тривимірних об'єктів на зразок освітлених куль і т. п. У математиці градієнтом називається вектор найбільшого зростання функції. У комп'ютерній графіці градієнтним називається колір, що плавно переходить в інший. Ви задаєте тільки крайні значення кольорів, наприклад, колір правої границі і колір лівої. Після заливання градієнтом об'єкт буде зафарбований відтінками, що плавно змінюють один одний. Градієнтний колір має напрямок зміни кольору і граничних значень кольорів, простір між якими заповнюється інтерпольованими, плавно мінливими відтінками. Коли редагують градієнтне заливання, роблять зміну напрямку градієнта і відстані між його границями, порогами. Режим Transform Fill призначений саме для цих операцій.

При використанні градієнтного заливання ви можете відразу вказати напрямок градієнта. Якщо звичайне зафарбування здійснюється простим натисканням за допомогою миші всередині зафарбованої області, то при заливанні градієнтним кольором лінійного типу варто натиснути на ліву кнопку миші і, не відпускаючи її, перемістити мишу. Від точки натиску до курсору потягнеться лінія, що показує напрямок градієнта. Якщо під час руху миші ви натиснете на клавішу Shift, нахил лінії напрямку градієнта буде кратним 45°. Зазначений режим заливання застосовується тільки до лінійних градієнтів.

### **3.3. Імпортування в Flash**

Редактор Flash MX підтримує безліч форматів даних, починаючи з растрових зображень і закінчуючи фільмами QuickTime. Щоб імпортувати готове растрове зображення (або інший формат) у редактор Flash MX, варто вибрати команду File → Import.

Існує кілька методів використання такого зображення. Відразу після імпорту воно являє собою зображення, що існує саме по собі. Його можна лише переміщати, обертати і масштабувати. Редагувати об'єкт можна тільки як об'єкт бібліотеки ролика. Можливість виділення або редагування частини об'єкта відсутня. Якщо ж виділити імпортований об'єкт і натиснути сполучення клавіш CTRL+B або дати команду Modify → Break Apart, то він стане доступний для редагування. Відтепер в імпортованому об'єкті можна виділити частину растрового зображення, перемістити її, видалити її або змінити її форму. У цьому випадку растрове зображення поводитиметься як растрове заливання.

Для редагування окремих частин растрового зображення можна застосовувати інструменти виділення: Lasso (Ласо) і Magic Wand (Чарівна паличка) для виділення частини зображення за кольором.

Для того, щоб редагувати растрове зображення звичними для Flash MX методами, його потрібно перетворити на векторне. Цей процес називається трасуванням. Дії виконуються над виділеним об'єктом, не розбитим на частини командою Break Apart. Застосовується команда Modify → Trace Bitmap. У діалоговому вікні варто настроїти параметри трасування:

- Color Threshold (Колірний поріг) – визначає, наскільки сусідні відтінки повинні відрізнятися один від одного, щоб вважатися різними кольорами;
- Minimum Area (Мінімальна площа) – задає розмір мінімальної області, у яку ще можна помістити кілька флеш-символів;
- Curve Fit (Підбір кривих) – визначає згладжування;
- Corner Threshold (Кутовий поріг) – ступінь «незграбності» контурів, що виходять при трасуванні.

Чим менше значення Corner Threshold і Minimum Area, тим точніше результат відповідає вихідному зображенню, при занижених значеннях цих параметрів розмір векторного зображення сильно збільшується за рахунок появи великої кількості контурів.

#### **4. Порядок виконання роботи**

1. Настроїти інтерфейс користувача в редакторі Flash MX.
2. Створити примітиви за допомогою редактора Flash MX.
3. Вивчити методи заливання і вибору кольору в Flash MX.
4. Вивчити засоби для роботи з імпортованою графікою. Зробити трасування імпортованого растрового зображення з різними параметрами.
5. Вивчити способи растрового заливання зображень у Flash MX.
6. Установити параметри сітки і зобразити її в робочій області. Установити одиниці виміру лінійки.
7. Намалювати об'єкт і залити його за допомогою всіх чотирьох видів заливання, змінюючи параметри заливання інструмента Paint Bucket.
8. Створити новий градієнт і імпортувати його у файл.
9. Створити градієнтне заливання і скопіювати його атрибути для іншого заливання.
10. Імпортувати растрове зображення у Flash і відредагувати його.
11. Імпортувати у Flash растрове зображення, оптимізувати його, застосувати його для заливання векторного зображення, створеного у Flash.



## **5. Контрольні запитання**

1. Назвіть види комп'ютерної графіки й опишіть їхні основні характеристики.
2. Які два типи градієнта ви знаєте?
3. Що таке публікація фільму?
4. Опишіть процес імпорту зовнішніх файлів у Flash MX і назвіть основні формати даних, які можна імпортувати в Flash MX.
5. Опишіть процес трасування растрових зображень.

## **6. Рекомендована література**

1. Бурлаков М. В. Adobe Flash CS3 : самоучитель / М. В. Бурлаков. — М. : Диалектика, 2007. — 624 с.
2. Мильберн К. Flash 5 : для дизайнерів / Кен Мильберн, Джон Крото. — К. : Диасофт, 2000. — 496 с. — (Внутренний мир).
3. Пакнелл Ш. Macromedia Flash 8 для професіоналів / Шон Пакнелл, Брайан Хогг, Крейг Суонн ; [пер. с англ., ред. Чайки И. В.]. — М. : Вільямс, 2006. — 672 с.
4. Франклин Д. Flash 4. Анімація в Інтернеті / Д. Франклин, Б. Паттон. — СПб. : Символ Плюс, 2000. — 464 с.
5. Adobe Flash CS4 / Елистратов Ф. М. — М. : Эксмо, 2009. — 400 с. — (Официальный учебный курс).

## **ПРАКТИЧНЕ ЗАНЯТТЯ № 8**

### **Створення графічних зображень на основі різних методів малювання у Flash**

#### **1. Мета роботи**

Вивчення інструментів малювання Flash MX. Навчитися створювати примітиви, робити вирівнювання декількох примітивів. Вивчити види текстових полів і засобу їх створення. Вивчити роботу із шарами. Навчитися створювати символи та екземпляри символів.

#### **2. Підготовка до роботи**

Перед роботою слід повторити матеріал за курсом лекцій та за рекомендованою літературою.

#### **3. Методичні вказівки до виконання роботи**

##### ***Інструмент Eraser (Ластик)***

Eraser – інструмент редагування форм. Ним вичищаються області форм і ліній, намальованих іншими засобами. При подвійному натисненні на кнопку ластика буде вилучене усе, що намальовано на сцені. Настроювання інструмента Eraser: кнопка Eraser Mode (Режим ластика) визначає дію ластика і багато в чому дублює опцію Brush Mode (Режим пензля) інструмента Brush. Кнопка відкриває випадне меню, що містить такі значення:

- кнопка Erase Normal – нормальний режим. У ньому відбувається видалення всього, що попадає під ластик – як форм, так і кривих;

- кнопка Erase Fills – стирання форм, заливань. У цьому режимі ластик стирає тільки форми, ігноруючи лінії;
- кнопка Erase Lines – стирання ліній. Даний режим, на відміну від попереднього, дозволяє видаляти тільки лінії, не змінюючи форми;
- кнопка Erase Selected Fills – стирання виділених форм. У цьому режимі ластик стирає ті частини форм, що були попередньо виділені яким-небудь чином;
- кнопка Erase Inside – стирання всередині. У даному режимі ластик стирає тільки ті форми, всередині яких процес стирання був початий;

Наступна опція ластика – Eraser Shape (Форма ластика). Ластик модифікує форми, а це може призвести до збільшення кількості складених кривих, використаних в обмежуючу форму контурної лінії.


У редакторі Flash є спеціальний засіб, що дозволяє перетворити лінію на форму. Тому що лінія не може мати градієнтне зафарбування, після цієї операції завжди виходить форма суцільного кольору, і потім можна змінити колір отриманих форм на градієнтний за допомогою інструмента To Fill. Для того, щоб перетворити криву на форму, необхідно виділити її цілком або частково, а потім виконати команду меню Modify → Curves → Lines To Fill. Після цього лінія перетвориться на форму, яку можна редагувати, як всі інші форми.

Редагування форм і звичайних ліній багато в чому схожі. Границя форми, власне кажучи, є замкнутою кривою, складеною з тих же кривих Без'є. Тому пряме редагування форми здійснюється так само, як і редагування лінії, тобто полягає в переміщенні точок і зміні вигину кривих, що складають границю форми.


### ***Інструмент Arrow (Стрілка)***

Інструмент Arrow використовується не тільки для прямого редагування ліній і форм, але й для виділення об'єктів і їхніх частин, а також для переміщення тих і інших по сцені. Однократне натискання на ліву кнопку миші за формою дозволяє виділити всю її цілком, подвійне натискання відзначає форму і всі лінії, що стикаються з її границею. Для того, щоб додати ще один об'єкт до вже позначеного або зняти виділення тільки з одного об'єкта, одночасно з виділенням за допомогою миші натисніть на клавішу Shift. Щоб зняти виділення з усіх зображень, натисніть мишею по вільному простору або виконайте команду меню Edit → Deselect All, що відповідає комбінації клавіш Ctrl+Shift+A. Відзначити всі присутні на сцені об'єкти можна командою меню Edit → Select All або натисканням на Ctrl+A.

### ***Інструмент Lasso (Ласо)***

 – ця кнопка активізує Lasso, ще один інструмент для виділення об'єктів. З його допомогою можна маркувати фрагменти довільної форми. Для цього необхідно підвести курсор до потрібного місця, натиснути на ліву клавішу миші і намалювати лінію так само, як і у випадку використання інструмента Pencil (Олівець); при відпусканні кнопки миші відбувається виділення всіх об'єктів і їхніх частин, що потрапили всередину кривої, що утворилася.

Інструмент Ласо має кілька кнопок налаштування режимів:

 – ця кнопка включає спеціальний інструмент виділення Magic Wand (Чарівна паличка), украй зручний при роботі з растровими заливаннями. Аналог даного інструмента присутній майже в усіх сучасних растрових редакто-

рах. Натиснувши мишею по растровому заливанню, ви виділяєте частину форми, що містить точки близького кольору. Якщо в растровому заливанні присутній який-небудь об'єкт, пофарбований рівномірно, можна виділити ту частину форми, де він зображений. Потім цей фрагмент можна зрушити вбік або скопіювати, і вийде форма, границі якої відповідають характерному растровому зображенню заливання.



– натиснувши на цю кнопку, ви викликаєте вікно Magic Wand Settings (Настроювання чарівної палички), де можна визначити параметри цього інструмента. Параметр Threshold (Поріг) відповідає за різницю між кольорами точки, до якої ви застосували Magic Wand, і тих точок, що виділяти не треба. Усі області, де колір відрізняється від базового на величину, меншу за введений поріг, будуть марковані. Параметр Smoothing (Згладжування) визначає плавність контуру областей, які виділяють. Він може набувати таких значень:

- Pixels – попиксельне виділення. Відзначені області форми будуть точно відповідати параметру Threshold – жодного зайвого пікселя виділено не буде. Це значення забезпечує найбільш точне виділення, однак форми, які одержують при цьому, мають дуже складні обриси і для їхньої обробки потрібно більше ресурсів комп'ютера;

- Rough – строге виділення. Марковані області будуть дуже близькі до попиксельно виділених, але границя одержаних форм дещо згладиться;

- Normal – нормальне виділення. Границі форм, що маркуються, згладжуються, але в них можуть потрапити точки, що не відповідають параметру Threshold, або, навпаки, точки що відповідають заданому розкиду, можуть бути проігноровані. Утім, це малопомітно при великих розмірах форм, які виділяють, і цілком годиться для великих зображень;

- Smooth – гладке виділення. Відзначені форми згладжуються. Якщо подібним чином обробити невеликі області виділення, буде багато браку; однак це найекономічніший режим щодо використання машинних ресурсів. Не застосуйте дану опцію для виділення невеликих форм.

Помітьте, що Magic Wand застосовується тільки до форм із растровим заливанням. На форми зі звичайним заливанням, суцільним або градієнтним, дія цього режиму не поширюється, і виділення робиться за допомогою інструмента Lasso.



– ця кнопка містить Polygon Mode (Полігональний режим), що відрізняється від звичайного Ласо тільки тим, що при виділенні курсор малює за собою не криву, як інструмент Pencil (Олівець), а відрізки прямих, як Line (Лінія). Виділення інструментом Lasso у звичайному режимі здійснюється кривими лініями, але допускаються й короткочасні переключення в полігональний режим. Для цього треба, не припиняючи виділення, натиснути на клавішу Alt. Поки ця клавіша натиснута, виділення відбувається так, як у режимі Polygon Mode, коли відпущена – як при використанні звичайного Ласо.

### **Інструмент Magnifier (Луна)**



– ця кнопка містить засіб Magnifer, що використовують для збільшення заданих областей фільму. Подвійне натискання на зазначену кнопку рівносиль-

не установці стовідсоткового збільшення. У цього допоміжного засобу є два режими, що включаються кнопками:

- кнопка **Enlarge** (Збільшити). У цьому режимі Лупа наближає зображення. Максимальне збільшення, можливе в Flash 4, складає 2000 %;

- кнопка режиму **Reduce** (Зменшити). У цьому режимі масштаб сцени зменшується вдвічі. Максимальне зменшення, можливе в Flash, дорівнює 7 %.

Існує також кілька визначених режимів збільшення, доступних через меню або основну панель, або випадне меню **Zoom Control** (Керування збільшенням) основної панелі.

### ***Інструмент Hand (Рука)***



– ця кнопка активізує режим **Hand**, що так само, як і **Magnifier**, не має відношення до редагування графіки, а є допоміжним засобом, що забезпечує зручність роботи. Режим допомагає під час роботи зі збільшеними зображеннями і, по суті, відповідає смугам прокручування основного вікна.

### ***Згладжування***



– кнопка модифікатора **Smooth** (Згладжування) знаходиться серед опцій інструмента **Arrow** (Стрілка) і застосовується для згладжування кривих ліній. Виберіть інструмент **Стрілка**, виділіть криву і натисніть на кнопку **Smooth** або виконайте команду меню **Modify** → **Curves** → **Smooth**. Інструмент **Smooth** не виконує ніякої дії з відрізками прямих, однак їх можна попередньо зігнути прямим редагуванням. Згладжувати можна також і форми: при цьому відбувається оптимізація їх контурів.

### ***Випрямлення***



– це кнопка модифікатора **Straighten** (Випрямлення). Вона знаходиться в опціях інструмента **Arrow** і за призначенням протилежна інструменту **Smooth**. Крива, до якої неодноразово застосовували дану операцію, перетворюється на відрізок прямої, що з'єднує кінці вихідної кривої, у випадку, якщо відключені налаштування **Recognize lines** (Розпізнавати лінії) і **Recognize shapes** (Розпізнавати форми) вікна **Assistant**. Якщо ж ці налаштування активізовані, результатом випрямлення може стати геометрична фігура – прямокутник або трикутник, коло або овал, сектор кола, кут. Команда **Straighten** доступна також через меню **Modify** → **Curves** → **Straighten**. На прямих лініях вона ніяк не впливає. Дія даної опції також поширюється на форми; при цьому виробляється випрямлення або розпізнавання границі форми.

### ***Оптимізація***

Команда меню **Modify** → **Curves** → **Optimize...** або комбінація клавіш **Ctrl+Alt+Shift+C** робить оптимізацію ліній або границь форми, зменшуючи кількість використаних у ній кривих і згладжуючи ті, що залишилися. Слайдер **Smoothing** у верхній частині вікна визначає ступінь згладжування кривих. При установці прапорця **Use multiple passes (slower)** редактор робить оптимізацію кілька разів – до тих пір, поки новий цикл оптимізації приводить до позитивних результатів. При включеному прапорці **Show totals message** редактор виводить вікно з результатами оптимізації по її закінченні.

### ***Обертання***



– це кнопка модифікатора Rotate (Обертати), що відноситься до групи відносних, тобто існуючих «окремо» від об'єкта. Обертання об'єкта у Flash, так само, як зрушення і масштабування, не змінює його геометрії. Згадані опції можна вважати параметрами промальовування. Модифікатор Rotate можна також викликати командою меню Modify → Transform → Rotate. Команда меню Modify → Transform → Rotate Left робить поворот виділених об'єктів на 90° проти годинникової стрілки (–90°), команда Modify → Transform → Rotate Right – на 90° за годинниковою стрілкою (+90°).

### ***Масштабування***



– це кнопка модифікатора Scale (Масштабувати), що також є відносним. Рамки виділення масштабують зображення по одній осі. Модифікатор Scale можна також викликати командою меню Modify → Transform → Scale. Скасувати всі модифікації, пов'язані з масштабуванням, обертанням і зрушенням, можна командою меню Modify → Transform → Remove Transform або натисканням комбінації клавіш Ctrl+Shift+Z.

### ***Переміщення центру***

Обертання, масштабування і зрушення об'єкта (або групи об'єктів) відбуваються щодо його центру. Коли ви виділяєте той чи інший об'єкт, його центр автоматично міститься в геометричному центрі описаного біля виділення прямокутника. Іноді необхідно, наприклад, обертати об'єкт навколо одного з кутів або навколо точки, що знаходиться далеко від виділених об'єктів. Для цього є команда Modify → Transform → Edit Center. Після її виконання на сцені виникає маленьке перехрестя, яке можна пересунути мишею в потрібну точку. Тепер і обертання, і масштабування, і зрушення здійснюватимуться щодо нового центру виділення. Якщо зняти виділення з об'єкта, настроювання центру втрачають актуальність, і при новому виділенні (навіть якщо маркуються ті ж об'єкти) центр треба переміщати знову.

### ***Редагування через меню***

Редактор Flash дозволяє застосовувати масштабування й обертання через команду меню. При цьому користувач може вказати точні значення згаданих модифікацій, що дуже корисно в тих випадках, коли редагування «на око» недостатньо. Викличте пункт меню Modify → Transform → Scale and Rotate...

### ***Редагування за допомогою інспекторів***

Інспектори дають можливість редагувати положення об'єктів, їхні лінійні розміри, а також застосовувати до них модифікації з вказівкою точних значень. Інспектор об'єктів відображає і дозволяє змінювати координати об'єкта на сцені. Коли прапорець Use Center Point (Використовувати центральну точку) знятий, координати відраховуються від лівого верхнього кута об'єкта або виділення, коли встановлений – від центру. Прапорець Skew параметра Rotate, будучи знятим, дозволяє обертати весь об'єкт, а будучи включеним – нахилити його. Модифікації, зроблені за допомогою інспектора модифікацій, також можуть бути скасовані натисканням на кнопку Reset, якщо об'єкт після застосування до нього цих операцій не піддавався прямому редагуванню.

### ***Дзеркальне відображення***

У меню Modify є дві команди, що дозволяють дзеркально відображати об'єкт щодо однієї з осей. Команда Modify → Transform → Flip Vertical (Відобразити вертикально) дзеркально відображає об'єкт по вертикалі, щодо осі X, команда Modify → Transform → Flip Horizontal (Відобразити горизонтально) – по горизонталі, щодо осі Y.

### ***Очищення кольору***

Команда меню Modify → Transform → Remove Colors (Видалити колір) перетворить колір усіх виділених форм на білий, а всіх ліній на чорний. Один з побічних ефектів застосування даної команди полягає в тому, що дотичні форми різних кольорів зливаються, оскільки тепер мають однаковий колір (білий).

### ***Копіювання і вставка***

Редактор Flash MX має стандартний набір засобів для дублювання зображень, їх копіювання, вирізання і вставки з буфера обміну даними. Зазначені операції можна робити за допомогою кнопок стандартної панелі основного вікна редактора Flash.

### ***Редагування тексту в Flash MX***

Для розміщення тексту всередині ролика в Flash використовуються так називані текстові поля, що бувають трьох типів: статичні, динамічні та поля введення. Текстове поле являє собою прямокутну область, у якій може знаходитися який-небудь текст. Ці поля існують як відособлені об'єкти, редагування тексту всередині яких відбувається аналогічно звичайним текстовим редакторам.

Щоб створити текстове поле, необхідно вибрати інструмент Text (Текст). Видгляд покажчика миші зміниться на перехрестя з невеликою буквою А збоку. Тепер натиснення на ліву кнопку миші у будь-якому місці робочої області створить текстове поле. У текстове поле можна ввести будь-який текст. Для того, щоб вибрати накреслення і розмір шрифту, служить панель Character (Оформлення тексту). Розмір шрифту можна задавати за допомогою движка. На панелі Character також є поле Tracking (Трекінг), у яке можна ввести числове значення додаткової відстані між буквами. Крім того, на панелі Character розташовані кнопки, що дозволяють зробити виділений текст напівжирним або курсивним, а також змінити його колір. Список, що розкривається, Character position (Положення символів), дозволяє вибрати режим верхнього або нижнього індексу.

Крім усього переліченого, на панелі Character є прапорець Kern (З кернігом), який повинен включати режим автоматичного кернінгу. Для форматування абзаців призначена допоміжна панель Paragraph (Абзац). На ній розташовані чотири стандартні кнопки для керування вирівнюванням тексту (по лівому краю, по правому краю, по центру і по ширині), а також чотири поля, у яких можна встановлювати величину лівого і правого полів, величину абзацного відступу й інтерліньяж.

### ***Види текстових полів***

Статичне текстове поле призначається для відображення статичного тексту. Цей текст не може змінюватися. Його можна тільки виділяти для копіювання в буфер обміну. Це можливо тільки в тому випадку, якщо на допоміжній панелі

Text Options (Параметри тексту) встановлений прапорець Selectable (Вибраний). У редакторі Flash текстові поля можуть мати фіксовану ширину або бути розширюваними. Розширюване поле автоматично змінює свою ширину залежно від вмісту, а поле фіксованої ширини не змінює своїх параметрів. При створенні текстового поля натисканням на ліву кнопку миші воно спочатку є розширюваним. Якщо ж «потягнути» мишею за його маніпулятор, щоб вручну розширити або звужити, поле набуває фіксованої ширини. Текстові поля фіксованої ширини мають квадратний маніпулятор, а розширювані – круглий. У статичних полів цей маніпулятор розташований у правому верхньому куті, а в інших – у правому нижньому. Щоб поле фіксованої ширини зробити розширюваним, досить просто двічі натиснути на його маніпуляторі. У Flash є можливість довільного обертання і масштабування текстових полів. Для того щоб включити контури шрифтів, варто натиснути одну з кнопок Embed Fonts (Упровадити шрифти) на допоміжній панелі Text Options (Параметри тексту), інакше похилі текстові поля не відобразяться.

На панелі Character (Оформлення тексту) розташоване ще одне поле – URL. Якщо в це поле ввести яку-небудь адресу URL, то текстовий блок працюватиме як гіперпосилання: якщо користувач вибере його за допомогою миші, то браузер завантажить відповідний ресурс.

Динамічні текстові поля також відображають текст, але цей текст може змінюватися під час програвання ролика. Щоб перетворити текстове поле на динамічне, варто вибрати пункт Dynamic Text (Динамічний текст) у списку випадної панелі Text Options (Параметри тексту). У цьому режимі на панелі з'явилося поле Variable (Змінна). У полі Variable (Змінна) потрібно вказати ім'я змінної, котра відповідає вмісту цього текстового поля.

Крім можливості зміни вмісту, динамічні текстові поля мають ще ряд особливостей. По-перше, їх можна зробити однорядковими або багаторядковими, вибравши з випадного списку на панелі Text Options (Параметри тексту) пункт Single Line (Один рядок) або Multiline (Багато рядків) відповідно. Для багаторядкових полів доступний прапорець Word Wrap (Перенос за словами). По-друге, навколо динамічних полів можна намалювати рамку, що позначає їхню границю. Рамка включається за допомогою перемикача Border/Bg (Границя/Тло) на тій самій панелі Text Options.

У динамічні текстові поля можна включати теги HTML для форматування тексту. Щоб програвач Flash правильно розпізнав їх, потрібно встановити на допоміжній панелі Text Options (Параметри тексту) прапорець HTML. Щоб текстові поля розпізнавалися правильно, потрібно надати їм відповідне значення зв'язаної з ними змінної.

Третій тип текстових полів, що існують у роликах Flash, це поля для введення. Вони призначаються не тільки і не стільки для відображення якої-небудь інформації, скільки для введення цієї інформації користувачем. Щоб перетворити текстове поле на поле введення, необхідно вибрати у випадному списку на панелі Text Options (Параметри тексту), пункт Input Text (Введення тексту). Тепер користувач зможе ввести в це поле який-небудь текст, і усе, що

він ввів, присвоюється відповідній змінній. В усьому іншому поля введення мають ті ж особливості, що й динамічні поля.

У полів введення є дві особливості. По-перше, можна обмежити максимальне число символів, що вводяться в поле. Це обмеження задається полем Max Chars (Усього символів) на панелі Text Options (Параметри тексту). Якщо значення цього поля встановлене в 0, кількість символів, що вводяться, не обмежується. По-друге, при виборі режиму Input Text (Введення тексту) у другому списку, що розкривається на панелі Text Options (Параметри тексту), з'являється ще один пункт – Password (Пароль). При виборі цього пункту в полі введення кожен символ відображається у вигляді зірочки, як це прийнято в більшості систем для введення пароля.

Для анімації тексту необхідно перетворити його на графічні об'єкти командою Modify → Break Apart (Змінити → Розділити) і далі працювати з ним, як із контурами і заливаннями.

### ***Робота із шарами в Flash MX***

Шари – зручний інструмент структурування складних зображень, що використовують у багатьох засобах обробки графіки – як векторної, так і растрової. Панелі шарів у Flash MX знаходяться у вікні хронометричної лінійки. Шари розташовані у визначеному порядку: верхні шари перекривають нижні. У Flash використовуються три типи шарів:

- Normal Layer – звичайний. Призначений для графіки, звуку, інтерактивних елементів, команд скрипта й іншого. Такий шар може прикріплюватися до шару маски або шару шляху;

- Guide Layer – шар шляху, або гайд-шар. Зображення, поміщене в ньому, невидиме при програванні опублікованого фільму і є маршрутом руху анімованого об'єкта в прикріплених до нього шарах, названих Guided Layers – відомими шарами;

- Mask Layer – шар, що маскує, або шар маски. Поміщене в ньому зображення також невидиме в опублікованому фільмі; воно є областю, через яку видні зображення на прикріплених до Mask Layer шарах. Вони називаються Masked Layers – шари, що маскуються.

Шари можна створювати, видаляти, переміщати щодо інших, змінювати їхній тип, перейменовувати, забороняти їхнє редагування і відображення. За допомогою кнопок, розташованих на хронометричній лінійці, і контекстного меню шарів можна виконувати наступні команди: змінювати атрибути видимості та закріпленості окремих і всіх шарів; вставляти і видаляти шари; відкрити вікно властивостей шарів.

### ***Символи й екземпляри символів у Flash MX***

У технології Flash використовуються самостійні об'єкти, які називаються флеш-символи (Symbols). Кожен флеш-символ міститься в бібліотеці і згодом може бути використаний на сцені скільки завгодно разів. Флеш-символи можуть містити все, що завгодно: графіку, анімацію, звук і т. п.

Існують флеш-символи трьох типів. Перший з них – це ролик (Movie clip) чи «мувік». «Мувік» може містити будь-які елементи, що може містити основний ролик. Він являє собою окремий фільм, що живе своїм життям.



Другий тип флеш-символу – це кнопка (Button). Кнопка являє собою спеціальний об'єкт, орієнтований на інтерактивність. Щоб кнопка стала «живою», необхідно запрограмувати ті дії, що відбуваються при її натисканні.

Третій тип флеш-символу називається графічним об'єктом або просто графікою (Graphic). Такі об'єкти містять не просто статичну графіку, але й можуть містити анімацію, проте на відміну від «мувіків», вони завжди синхронізовані з основним роликом.

Щоб перетворити якийсь об'єкт на флеш-символ, треба його виділити і натиснути клавішу F8. При цьому з'явиться діалогове вікно Symbol Properties (Властивості символу), де необхідно вказати ім'я флеш-символу і його тип.

Усі, створені в даному файлі флеш-символи, а також імпортовані растрові зображення і звуки, знаходяться в бібліотеці (CTRL+L). У бібліотеці організована структура папок для флеш-символів, аналогічна Провідникові Windows, є можливість програвання флеш-символів у маленькому вікні бібліотеки і редагування їх.

### **Особливості кнопок**

Основна особливість об'єкта кнопки полягає в тому, що її шкала часу відрізняється від шкали часу роликів і графіки. Вона містить чотири кадри: Up (Не натиснута), Over (Активізована), Down (Натиснута), Hit (Область наведення). Ці чотири кадри призначені для керування виглядом кнопки залежно від дій користувача. Коли покажчик миші користувача не наведений на кнопку, то вона відображається, як визначено в кадрі Up. Коли користувач наводить на кнопку покажчик миші, починається відображення кадру Over. Коли користувач натискає кнопку миші, відображається кадр Down. Кадр Hit не відображається ніколи, але він дуже важливий, тому що він визначає область дії кнопки.

## **4. Порядок виконання роботи**

1. Створити кілька примітивів.
2. Вибрати кілька з них і об'єднати в групу.
3. Виділити частину об'єктів і змінити стиль контуру.
4. Об'єднати два або більше примітивів.
5. Створити екземпляри символів кількох типів.
6. Створити статичне і динамічне текстове поле.
7. Анімувати текстове поле.
8. Намалювати псевдо-тривимірну п'ятикутну зірку і зникаючий напис.
9. Створити хромований текст.
10. Створити текст з ефектом світіння.
11. Створити тінь для примітива.
12. Створити зображення, використовуючи тільки два або більше раніше створених екземплярів символів.

## **5. Контрольні запитання**

1. Які типи накладених об'єктів, що використовують у редакторі Flash MX, ви знаєте?
2. Назвіть два типи текстових полів, що використовують у редакторі Flash.
3. Що таке шари? Які переваги використання шарів при роботі з графічними зображеннями?

4. Що таке символ і екземпляр символу в редакторі Flash?
5. Які типи символів ви знаєте?
6. Назвіть властивості шарів.
7. Які є режими роботи шарів?

## 6. Рекомендована література

1. Бурлаков М. В. Adobe Flash CS3 : самоучитель / М. В. Бурлаков. — М. : Диалектика, 2007. — 624 с.
2. Мильберн К. Flash 5 : для дизайнерів / Кен Мильберн, Джон Крото. — К. : Диасофт, 2000. — 496 с. — (Внутренний мир).
3. Пакнелл Ш. Macromedia Flash 8 для професіоналів / Шон Пакнелл, Брайан Хогг, Крейг Суонн ; [пер. с англ., ред. Чайки И. В.]. — М. : Вільямс, 2006. — 672 с.
4. Франклин Д. Flash 4. Анімація в Інтернеті / Д. Франклин, Б. Паттон. — СПб. : Символ Плюс, 2000. — 464 с.
5. Adobe Flash CS4 / Елистратов Ф. М. — М. : Эксмо, 2009. — 400 с. — (Официальный учебный курс).

## **ПРАКТИЧНЕ ЗАНЯТТЯ № 9** **Анімація графічних зображень у Flash**

### 1. Мета роботи

Вивчити методи анімації у Flash. Навчитися працювати з кадрами і сценами. Вивчити вікно хронометричної лінійки. Навчитися створювати покадрову анімацію, анімацію руху (руху за маршрутом) і анімацію з використанням трансформації форми об'єкта (морфінгу), використовуючи різні типи шарів.

### 2. Підготовка до роботи

Перед роботою слід повторити матеріал за курсом лекцій та за рекомендованою літературою.

### 3. Методичні вказівки до виконання роботи

#### 3.1. Хронометрична лінійка

Хронометрична лінійка – це монтажний пульт створюваного фільму. Флеш-фільм складається з окремих уривків, невеликих частин анімації, з яких згодом збирається весь фільм. Ці уривки, у свою чергу, поділяються на кадри. Хронометрична лінійка являє собою графік, на якому розставляються частини фільму, склеюються, зміщаються при необхідності і т. д. У документації Flash хронометрична лінійка зветься Timeline.

Для всіх об'єктів або фільмів, відкритих у редакторі, передбачене єдине вікно хронометричної лінійки (див. рис. 2), де відображається лінійка активного робочого вікна.

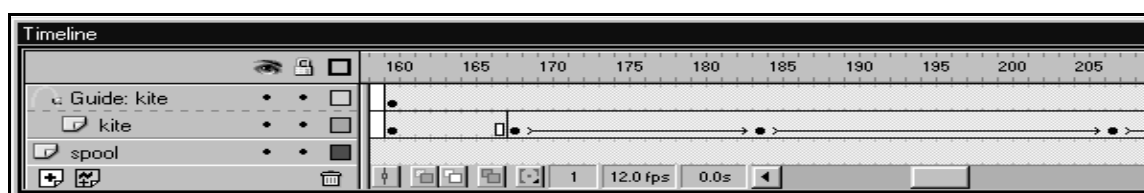



Рис. 2. Хронометрична лінійка

У лівій частині вікна хронометричної лінійки знаходяться шари. Вище розташовані кнопки налаштування видимості шарів, або закріпленості промальовування контурами, нижче – кнопки їхнього створення і видалення. Роздільник між лівою і правою частинами вікна можна пересувати шляхом перетаскування, що дозволяє вам установити бажану ширину панелі шарів. Праву частину вікна хронометричної лінійки займають кадри, де розміщується графіка, код скрипта та ін. Усі картинки або об'єкти розташовуються у визначеному кадрі визначеного шару. Через те що кадри програються послідовно, можна вважати, що номер кадру, у якому знаходиться зображення, є його функція часу. Фільм має чотири виміри: три просторових (X, Y і глибину зображення, обумовлену порядком розташування груп, об'єктів і шарів) і один часовий, що відображається в хронометричній лінійці. Порядок кадрів з їхніми номерами вказується в смузї заголовка правої частини вікна хронометричної лінійки. Поточний кадр позначається на цій смузї червоним маркером, через який проходить лінія, що перетинає всі шари і зазначає кадр у кожному з них. У вікні редагування відображається графіка, що знаходиться в пересічених лінією кадрах усіх шарів. Такий набір кадрів називатимемо зрізом.

Нижче області шарів знаходяться маленькі кнопки, а також три поля статусу, де відображається довідкова інформація. Перше поле показує номер поточного кадру, друге – частоту кадрів – FPS (у розглянутому прикладі – 12 кадрів за секунду) і час у секундах з точністю до десятих часток, що пройшов з початку фільму до вибраного кадру (у даному випадку – 0,0 с). Подвійне натискання на ліву кнопку миші по полю FPS відкриває вікно редагування властивостей фільму.

 – ця кнопка знаходиться у верхньому правому куті вікна хронометричної лінійки. При натисканні лівою кнопкою миші по ній з'являється спливаюче меню, команди якого налаштовують наступні режими зображення кадрів у хронометричній лінійці:

- **Tiny** – усі кадри мають вигляд надзвичайно вузьких, у кілька точок шириною, прямокутників;
- **Small** – кадри мають маленьку ширину;
- **Normal** – кадри мають середню ширину;
- **Medium** – кадри вдвічі ширше, ніж у режимі **Normal**;
- **Large** – ширина кадрів дорівнює висоті;
- **Short** – коли прапорець установлений, малюються кадри половинної висоти. Висота шарів, зрозуміло, також зменшується вдвічі. При цьому налаштування **Layer Height** (Висота шару) визначається щодо висоти кадрів, заданої прапорцем **Short**;
- **Tinted Frames** – прапорець вмикає кольорове штрихування кадрів залежно від змісту (заповнений/порожній, має розрахункову анімацію і т. д.); в іншому випадку використовується крапкове штрихування;
- **Preview** – якщо включений цей режим, у кадрі відображається його вміст (тільки те, що знаходиться в даному шарі); зображення масштабується так, щоб заповнювати весь простір прямокутника кадру. У режимі **Preview** прямокутник кадру має найбільші розміри;

- Preview in Context – у цьому режимі також показується вміст кадру в даному шарі, але він масштабується відповідно до розмірів зображення щодо лінійних розмірів кадру фільму.

### 3.2. Сцени і кадри в Flash MX

Сцена – це область, де здійснюється малювання і розміщаються зображення. У документації це поняття позначається терміном Stage. Флеш-фільм складається з послідовності кадрів, у яких знаходяться об'єкти і мальовані зображення. Крім того, цей фільм поділяється на частини, називані Scene (Сцена). Кожна сцена має свою хронометричну лінійку. На відміну від об'єктів, що теж мають цю властивість, сцени не можуть програватися самостійно: вони вибудовані одна за іншою і відтворюються відповідно до порядку проходження. Можна розглядати сцени як єдиний фільм. Коли одна сцена завершується, відразу за нею починає програватися інша. Хронометричні лінійки сцен мають роздільну нумерацію кадрів, але при попередньому перегляді фільму кількість кадрів, що вказується в блоці довідкової інформації при включеному прапорці View → Bandwidth Profiler, дорівнює сумі кадрів усіх сцен. Якщо для фільму призначене циклічне відтворення, по завершенні фінальної сцени починає програватися перша. На початку демонстрації фільму послідовно завантажуються всі сцени, тому що вони є частинами одного файлу.

У сцени є два параметри: її ім'я і порядок проходження серед інших. Редагувати їх можна або через інспектор сцен, або через команди меню. Створення сцени здійснюється командою меню Insert → Scene, у результаті чого нова сцена з ім'ям Scene номер\_сцени додається відразу за поточною. Для видалення призначена команда меню Insert → Remove Scene. Переключення між сценами здійснюється командами меню:

- View → Goto → First – перехід на першу сцену – Home;
- View → Goto → Previous – перехід на попередню сцену (щодо поточної) – PgUp;
- View → Goto → Next – перехід на наступну сцену – PgDn;
- View → Goto → Last – перехід на останню сцену – End.

Можна перейти також на потрібну сцену, вибравши її ім'я в списку, що є в підменю команди View → Goto.

Щоб змінити ім'я сцени з лаконічного Scene номер\_сцени на зручне вам, варто виконати команду меню Modify → Scene... Відкриється вікно, де ви можете ввести бажане ім'я; воно відразу ж буде внесене в підменю команди View → Goto.

Однак зручніше за все керувати сценами за допомогою інспектора сцен, у якому знаходиться список усіх сцен фільму. Інспектор сцен – єдиний інструмент, за допомогою якого можна змінити порядок проходження сцен. Для цього досить перетягнути ім'я сцени мишею на потрібне місце в списку.

### 3.3. Кадри

Flash використовує для формування фільму кілька типів кадрів, названих у документації Frames. Тип кадру визначає його вміст. Залежно від типу кадру і його вмісту змінюються значки в хронометричній лінійці, що дозволяє, дивля-

чись на неї, швидко знаходити потрібні ділянки фільму не за номерами кадрів, а за характерними групами кадрів. Існує два основні різновиди кадрів:

- **Keyframe** – ключовий кадр, яким починається і закінчується будь-яка анімація. У хронометричній лінійці він позначається прямокутником з чорним кружечком всередині. У такому кадрі можуть знаходитися об'єкти, графіка, звук і код скрипту; він може бути і порожній, у такому випадку його називають **Blank Keyframe** – порожній ключовий кадр (виглядає як білий прямокутник з маленьким білим колом всередині). При переміщенні в такий кадр будь-якого об'єкта або графіки він стає звичайним ключовим – **Keyframe**, а кружечок зафарбовується чорним кольором;

- **Frame** – звичайний кадр. Він може бути або порожній, або заповнений тим, що визначено в попередньому ключовому кадрі. У позначенні звичайного кадру на хронометричній лінійці немає кружечка. Якщо кадр порожній, тобто, порожній попередній йому ключовий, то прямокутник білий; якщо ж кадр заповнений, прямокутник залитий сірим кольором.

Ключовий кадр є визначальним; звичайний же являє собою, по суті, продовження ключового. Ключовий кадр і розташовані за ним звичайні можна розглядати як єдине ціле. Якщо сам **Keyframe** – це «голова», у якій визначається заповнення кадру, то наступні звичайні кадри – «хвіст».

Ключовий кадр має параметри, які можна налаштовувати. Щоб викликати відповідне вікно, досить двічі натиснути на ліву кнопку миші по прямокутнику, що позначає потрібний кадр хронометричної лінійки.

### ***Створення кадрів***

Кадри розташовані в хронометричній лінійці, де і виконуються всі операції з ними. Порожня область хронометричної лінійки, що знаходиться з правого боку від останнього кадру, розбита на прямокутники. Вони позначають місце для кадрів – місце, де можуть бути створені нові кадри. Уже існуючі кадри позначаються прямокутниками, перегородки між якими або замінені двома рисками (зверху і знизу), або позначені чіткою чорною лінією. Між звичайними кадрами і ключовим перегородка подана у вигляді рисок, а між останнім звичайним кадром одного ланцюжка і ключовим кадром наступного – у вигляді суцільної чорної.

Створення порожнього ключового кадру здійснюється командою **Inserts Blank Keyframe** або натисканням клавіші **F7**. Якщо з лівого боку від нового ключового кадру є порожній простір, він буде заповнений звичайними кадрами.

За допомогою команди **Insert → Keyframe** або натискання клавіші **F6** можна вставити ключовий кадр, у якому міститься зображення з найближчого ключового кадру ліворуч. Якщо найближчий ключовий кадр ліворуч порожній, то і новий теж буде порожнім; якщо ж в існуючому була графіка або об'єкти, вони автоматично копіюються в новий ключовий кадр зі збереженням їхнього розташування у вікні редагування, параметрів, налаштувань і т. д.

Для створення звичайного кадру в нерозміченій кадровій області необхідно виконати команду **Insert → Frame** або натиснути **F5**. Порожній простір зліва від нього заповниться такими ж кадрами.

### ***Видалення й очищення кадрів***

Для видалення ключового кадру необхідно виконати команду Insert → Delete Frame або натиснути клавіші Shift+F5. Якщо в даного ключового кадру праворуч був «хвіст» зі звичайних кадрів, то один з них видалиться, а вся послідовність зрушиться вліво. Якщо праворуч знаходиться ключовий кадр або порожня кадрова область, то видалиться виділений ключовий кадр. Та ж команда, виконана при виділеному звичайному кадрі, призведе до його видалення і зрушення групи кадрів, що знаходиться з правого боку, вліво. Команда Insert → Clear Keyframe або комбінація клавіш Shift+F6, застосована до ключового кадру, видаляє його і вставляє в дане місце звичайний кадр.

### ***Редагування кадрів***

Кадри можна копіювати, вставляти, пересувати в хронометричній лінійці. Однократне натискання лівою кнопкою миші по кадру робить його виділеним, поточним. Виділяти довільні набори кадрів можна також за допомогою клавіатури, використовуючи стандартні комбінації клавіш, визначених у Windows.

### ***Редагування кількох кадрів***

Flash дозволяє виділяти і редагувати кілька кадрів одночасно. У режимі Onion Skin (Шлейф) ви можете бачити вміст відразу кількох кадрів. Цей режим включається і налаштовується п'ятьма кнопками, розташованими в нижній смузі вікна хронометричної лінійки:

- Center Frame (Центрувати кадр) – пересувається слайдер лінійки так, щоб поточний кадр був видимим.

- Onion Skin (Шлейф) – вмикається і вимикається режим відображення в області малювання кількох кадрів одночасно. Усі кадри, що знаходяться всередині цих дужок, відображаються в області малювання. Поточний кадр показаний чітко, інші виглядають напівпрозорими, причому ступінь прозорості зменшується в напрямку до дужок.

- Onion Skin Outlines (Контурний шлейф) – усі кадри шлейфа, крім поточного, зображуються лініями окреслення, що зменшує навантаження на комп'ютер і збільшує швидкість роботи редактора на слабких машинах.

- Edit Multiple Frames (Редагувати кілька кадрів) – вмикається режим мультикадрового редагування. Якщо ця кнопка натиснута, усі кадри шлейфа відображаються в нормальному, а не напівпрозорому вигляді і доступні для редагування.

- Modify Onion Markers (Редагувати границі шлейфа). Це кнопка випадного меню настроювань шлейфа для пересування дужкових маркерів. Воно має наступні пункти:

- Always Show Markers – показувати маркери шлейфа завжди, навіть коли режим відображення шлейфа вимкнений;

- Anchor Onion – цей прапорець дозволяє закріпити границі шлейфа. Маркер поточного кадру переміщується самостійно, а границі шлейфа (дужки) нерухомі;

- Onion 2 – дужки шлейфа встановлюються на відстані двох кадрів від маркера поточного, завдяки чому в шлейфі показуються п'ять кадрів (два до поточного, два після нього і сам поточний);

- Onion 5 – дужки шлейфа встановлюються на відстані п'яти кадрів від маркера поточного кадру;
- Onion All – дужки маркера встановлюються на початок і кінець ланцюжка кадрів найдовшого шару.

### ***Властивості кадру***

Кожен ключовий кадр має властивості, серед яких можуть бути коментар або мітка, характеристики звукової композиції, вставленої в кадр, код скрипта або настроювання розрахункової анімації. Вікно параметрів кадру, що відкривається подвійним натисканням лівою кнопці миші по кадру, містить чотири вкладки: Label (Мітка) є поле введення для ідентифікатора і перемикач Behavior (Поведінка), для якого передбачені два положення: Label (Мітка) і Comment (Коментар). Вкладка Sound відповідає за настроювання звуку, вміщеного в ключовий кадр. На вкладці Actions програмується процедура флеш-скрипта, що виконується при програванні кадру. Tweening – вкладка установки параметрів розрахункової анімації.

### ***Контекстне меню кадрів***

При натисненні правою кнопкою миші по кадру з'являється контекстне меню. Контекстне меню містить такі команди:

- Create Motion Tween – створює у виділених кадрах анімацію руху;
- Insert Frame – вставляє звичайний кадр;
- Delete Frame – видаляє кадр;
- Insert Keyframe – вставляє ключовий кадр;
- Insert Blank Keyframe – вставляє порожній ключовий кадр;
- Clear Keyframe – очищає ключовий кадр. Видаляє його разом із вмістом і вставляє замість нього звичайний;
- Select All – виділяє всі кадри у всіх шарах;
- Copy Frames – копіює кадри з усім вмістом (включаючи анімацію, звук, код флеш-скрипта тощо) у буфер обміну;
- Paste Frames – вставляє з буфера обміну скопійовані раніше кадри. У разі потреби створює потрібну кількість шарів;
- Reverse Frames – змінює порядок проходження виділених кадрів на зворотний. Використовується для програвання «задом наперед» простих анімацій;
- Synchronize Symbols – синхронізує анімацію об'єктів у виділених кадрах.

### **3.4. Покадрова анімація в Flash MX**

У технології флеш існують два різновиди анімації: покадрова і розрахункова. У документації ці поняття визначені як Frame-by-frame Animation і Tweened Animation відповідно. Якщо вибрана покадрова анімація, дизайнер вручну промальовує всі її кадри і розміщає їх підряд. При їх програванні один за одним така послідовність кадрів виглядає як фільм. Покадровою буде як анімація, цілком намальована за описаною методикою в Flash, так і імпортований анімований gif-файл або фільм Quick Time.

Розрахункова анімація робиться по-іншому: дизайнер задає початкове і кінцеве положення картинки в двох ключових кадрах, а редактор розраховує проміжні положення. Така анімація вже більше відповідає поняттю «комп'ютерна». Її можна розділити на два типи: анімацію руху й анімацію форм, тобто Motion Tweening і Shape Tweening відповідно.

### ***Анімація форм***

На відміну від покадрової анімації, морфінг і анімацію руху називають проміжною анімацією (tweened animation). Замість того, щоб послідовно малювати всі кадри підряд, користувач малює тільки перший і останній кадр анімації, а всі проміжні розраховує програма. На допоміжній панелі Frame (Кадр) і у випадному списку Tweening (Тип анімації), у якому спочатку обране значення None (Нічого), морфінг у програмі Flash позначають терміном shape tweening (проміжна анімація форми), у цьому списку пункт Shape (Форма). Є також можливість зробити поточним кожний з проміжних кадрів і розглянути всі «проміжні» становища між першим і останнім кадрами. Правда, змінити ці проміжні становища програма Flash не дозволить, оскільки малювати можна тільки в ключових кадрах.

У деяких випадках замість стрілки на шкалі часу відображається пунктир. Це означає, що щось зроблено неправильно й анімація не може бути виконана. Якщо виділити початковий ключовий кадр такої «неправильної» анімації, то на допоміжній панелі Frame з'явиться кнопка, натиснувши на яку, можна довідатися імовірну причину невдачі. При морфінгу в більшості випадків ця причина полягає в наявності в ключовому кадрі флеш-символів або згрупованих об'єктів.

Справа в тому, що в ключових кадрах, які беруть участь у морфінгу, не повинно бути ні флеш-символів, ні згрупованих об'єктів, тому що програма не може змінювати форму таких фігур. У цих ключових кадрах (у даному шарі) повинна знаходитися тільки звичайна графіка, причому бажано не більше однієї фігури (інакше результат анімації може бути непередбачуваним). Тому для морфінгу завжди варто використовувати окремий шар. За допомогою анімації форми можна також змінювати колір заливання, масштаб, кут повороту об'єкта і т. п.

### ***Анімація руху***

Ще один вид проміжної анімації – це анімація руху (motion tweening). Анімація руху не змінює форму об'єкта, а тільки пересуває його на інше місце. У початковому і заключному кадрах анімації розташований той самий флеш-символ. Для флеш-символу можна змінити відтінок, прозорість, розмір, його можна повернути і т. і., тому анімація руху здатна поступово змінювати всі ці властивості від першого кадру до заключного. Анімація руху може працювати тільки з флеш-символами або згрупованими об'єктами.

Для створення анімації руху необхідно для нового ключового кадру вибрати на допоміжній панелі Frame (Кадр) з випадного списку Tweening (Тип анімації) пункт Motion (Рух). За допомогою панелі Frame можна також керувати обертанням об'єкта. Для цього призначений випадний список Rotate (Обертання). За умовчанням у ньому звичайно вибраний режим Auto (Автоматично). У цьому режимі об'єкт обертається тільки в тому випадку, якщо це необхідно. Якщо вибрати пункт None, об'єкт не обертається і залишається орієнтований так само, як у першому кадрі анімації. Крім того, можна задати напрямок обертання примусово, вибравши пункт CW (За годинниковою стрілкою) або CCW (Проти годинникової стрілки). У цьому випадку потрібно також вказати в сусідньому полі число повних обертів об'єкта за час анімації.



### ***Рух за маршрутом***

Рух за маршрутом здійснює пересування об'єкта за заданою траєкторією, називаною маршрутом. Маршрут задається у відповідному Guide шарі, а об'єкти, які пересуваються по ньому, у Guided шарах. Усі параметри руху для об'єкта задаються аналогічно параметрам анімації руху – на вкладці Tweening параметрів ключового кадру. Установка прапорця Orient to path direction (Орієнтація за напрямком шляху) змушує об'єкт не тільки рухатися за маршрутом, але й дотримуватися кута нахилу щодо нормалі кожної точки траєкторії. Таким чином, анімується не тільки рух, але й кут нахилу щодо маршруту. Якщо цей прапорець не встановлений, то об'єкт від одного кута нахилу плавно переходить до іншого; в іншому випадку він повертається залежно від вигину кривої маршруту і додатково плавно змінює свій кут нахилу.

Опція Snap to guide (Прив'язка до маршруту) прив'язує об'єкт до маршруту. Якщо прапорець включений, то об'єкт незалежно від використання інструмента Snap (Магніт) рухається відповідно до маршруту.

Опція Easing дозволяє імітувати нерівномірний рух по шляху, Rotate – обертати об'єкт, що рухається, і т. д.

### **4. Порядок виконання роботи**

1. Вивчити вікно хронометричної лінійки.
2. Створити кілька сцен.
3. Створити покадрову анімацію.
4. Створити анімацію об'єкта за допомогою морфінгу.
5. Створити анімацію руху об'єкта.
6. Створити анімацію тексту за допомогою шару-маски.
7. Створити анімацію руху за траєкторією.
8. Виконати ефект розглядання тексту через збільшувальне скло при натисканні на клавішу «Пуск».
9. Створити ефект рентгенівських променів для перегляду якого-небудь об'єкта. Перегляд має починатись, зупинятись і продовжуватись залежно від натиснутої кнопки.
10. Створити ілюзію перегляду сцени через висвітлення прожектора.
11. Трансформувати один об'єкт в інший, змінюючи при цьому колір.
12. Створити Flash-фільм анімованої заставки, де букви трансформуються в розмірі і змінюється їхній колір від прозорого до вихідного.

### **5. Контрольні запитання**

1. Що таке хронометрична лінійка?
2. Які типи кадрів у Flash ви знаєте? Які властивості кадрів ви знаєте?
3. Що таке калькування? Для чого використовується можливість калькування і які його переваги?
4. Які методи анімації, використовуваної в редакторі Flash MX, ви знаєте?
5. Що таке морфінг? Наведіть його основні властивості.
6. Що таке анімація руху?
7. Що таке реверсивна анімація?

## **6. Рекомендована література**

1. Бурлаков М. В. Adobe Flash CS3 : самоучитель / М. В. Бурлаков. — М. : Диалектика, 2007. — 624 с.
2. Мильберн К. Flash 5 : для дизайнеров / Кен Мильберн, Джон Крото. — К. : Диасофт, 2000. — 496 с. — (Внутренний мир).
3. Пакнелл Ш. Macromedia Flash 8 для профессионалов / Шон Пакнелл, Брайан Хогг, Крейг Суонн ; [пер. с англ., ред. Чайки И. В.]. — М. : Вильямс, 2006. — 672 с.
4. Франклин Д. Flash 4. Анимация в Интернете / Д. Франклин, Б. Паттон. — СПб. : Символ Плюс, 2000. — 464 с.
5. Adobe Flash CS4 / Елистратов Ф. М. — М. : Эксмо, 2009. — 400 с. — (Официальный учебный курс).

## **ПРАКТИЧНЕ ЗАНЯТТЯ № 10**

### **Інтерактивні елементи.**

### **Програмування за допомогою Action Script**

#### **1. Мета роботи**

Вивчити основні можливості мови сценаріїв Action Script (AS). Навчитися використовувати засіб AS для створення складної інтерактивної анімації. Навчитися публікувати ролик Flash в Інтернеті, використовуючи різні формати.

#### **2. Підготовка до роботи**

Перед роботою слід повторити матеріал за курсом лекцій та за рекомендованою літературою.

#### **3. Методичні вказівки до виконання роботи**

Для того щоб створювати інтерактивні ролики і зручно керувати анімацією у Flash, існує проста і зрозуміла мова Action Script (AS), якою можна створювати сценарії всередині ролика. Конструкції мови схожі на конструкції Java, JavaScript, C та інших мов програмування. Докладніше про AS можна прочитати у рекомендованій літературі.

#### **4. Порядок виконання роботи**

1. Створити ескіз особистого web-сайта.
2. Створити особисту сторінку за допомогою редактора Flash MX.
3. Додати інтерактивні поля введення в web-сторінку.
4. Додати ще кілька сторінок з персональною інформацією автора і перехресні посилання на них.
5. Додати інтерактивну анімацію в особисту сторінку.
6. Опублікувати ролик Flash у форматі HTML.

#### **5. Контрольні запитання**

1. Що таке сценарій кадру і сценарій символу?
2. Назвіть типи операцій, що використовують у Action Script.
3. Які властивості екземплярів ви знаєте?
4. Як відбувається трасування і налагодження сценарію?
5. Опишіть технологію перетаскування в ролик Flash.

## **6. Рекомендована література**

1. Бурлаков М. В. Adobe Flash CS3 : самоучитель / М. В. Бурлаков. — М. : Диалектика, 2007. — 624 с.
2. Мильберн К. Flash 5 : для дизайнеров / Кен Мильберн, Джон Крото. — К. : Диасофт, 2000. — 496 с. — (Внутренний мир).
3. Пакнелл Ш. Macromedia Flash 8 для профессионалов / Шон Пакнелл, Брайан Хогг, Крейг Суонн ; [пер. с англ., ред. Чайки И. В.]. — М. : Вильямс, 2006. — 672 с.
4. Франклин Д. Flash 4. Анимация в Интернете / Д. Франклин, Б. Паттон. — СПб. : Символ Плюс, 2000. — 464 с.
5. Adobe Flash CS4 / Елистратов Ф. М. — М. : Эксмо, 2009. — 400 с. — (Официальный учебный курс).

## **САМОСТІЙНА РОБОТА**

За традицією основні форми і методи самостійної роботи розроблялися за умов обов'язкового відвідування занять, коли обсяг аудиторних істотно перевищував обсяг позааудиторних занять. Зараз ці види навчальної діяльності, відповідно до діючих навчальних планів, стали формально паритетними, але самостійне навчання останнім часом стає все вагомішим. Поряд з традиційними формами, такими як опрацювання лекційного (теоретичного) матеріалу, підготовка до практичних і лабораторних занять, курсове проектування, з'являється і нова – самостійне вивчення певної частини теоретичного матеріалу дисципліни.

Інтернет-технології кардинально змінюють світ бізнесу. Кожній компанії, щоб залишитися конкурентоспроможною, необхідно оцінювати значення інформаційна насиченої інфраструктури, що пропонує Інтернет. Системи електронної комерції – це безпосереднє використання інформаційних технологій у бізнесі. До їх задач відносяться підтримка бізнес-процесів і пропозиція товарів і послуг. Датою народження цих систем можна вважати 1993 рік, коли в результаті зниження ціни і підвищення продуктивності апаратних і програмних засобів почалася масова експлуатація Інтернету і систем World Wide Web. Глобальна інформаційна структура стала базою розвитку систем електронної комерції.

Електронна комерція – це будь-яка форма бізнес-процесу, у якому взаємодія між суб'єктами відбувається електронним способом з використанням інтернет-технологій і різноманітних телекомунікаційних технологій. Електронна комерція охоплює питання, що виникають у зв'язку з усіма відносинами комерційного характеру: купівля-продаж, постачання, угода про розподіл продукції, торговельне представництво, факторинг, лізинг, проектування, консалтинг, інжиніринг, інвестиційні контракти, страхування, угода про експлуатацію і концесію, банківські послуги, спільна діяльність та інші форми промислового і ділового співробітництва.

Інтернет-маркетинг – необхідна ринкова діяльність, пов'язана з виробництвом товарів і наданням послуг, що з найбільшою імовірністю будуть реалізовані на ринку. Інтернет-маркетинг – це необхідний комплекс заходів для дослідження мережного ринку Інтернет, з ефективного просування і продажу товарів за допомогою сучасних інтернет-технологій.

Інтернет-економіка, мережна економіка – середовище, у якому будь-яка компанія або індивід, що знаходяться в будь-якій точці економічної системи, можуть контактувати легко і з мінімальними витратами з іншою компанією або індивідом щодо спільної роботи, для обміну ідеями і ноу-хау.

Впровадження у виробництво електронних засобів інтернет-технологій дозволяє перейти від традиційного методу розробки електронної апаратури до використання технологій WWW, що містять: мову гіпертекстової розмітки документів HTML; універсальний спосіб адресації ресурсів у мережі URL; протокол обміну гіпертекстовою інформацією HTTP; універсальний інтерфейс шлюзів CGI. Опанувати ці технології – завдання самостійної роботи студентів.

### **Характеристика підручників і навчальних посібників**

***Карпунин А. В. Интернет-технологии : учеб. пособие / А. В. Карпунин, В. М. Бредихин, Ю. В. Мищеряков. – Х. : СМІТ, 2003. – 308 с.***

У навчальному посібнику викладаються основні концепції, покладені в основу при створенні Інтернет, основні принципи роботи Інтернету, а також правила і способи роботи в глобальній мережі, принципи використання різних послуг, доступних користувачеві в Інтернеті.

Велика увага приділяється розглядові стека протоколів TCP/IP. Докладно розглядаються всі рівні протоколів і їхня взаємодія в процесі передання різних видів повідомлень між клієнтом і сервером. У навчальний посібник включена велика кількість ілюстрацій, що полегшують розуміння викладеного матеріалу.

***Карпова Т. С. Базы данных: модели, разработка, реализация / Т. С. Карпова. – СПб. : Питер, 2001. – 304 с.***

Даний навчальний посібник підготовлений за матеріалами лекційних курсів, присвячених основам теорії баз даних, мові SQL і серверам баз даних, що читалися автором протягом останніх десяти років у Державному Санкт-Петербурзькому університеті аерокосмічного приладобудування й у Державному технічному університеті (Політехнічному інституті).

***Попов В. Практикум по Internet-технологиям : учеб. курс / В. Попов. – СПб. : Питер, 2002. – 480 с.***

Книга являє собою практичний курс з основ сучасних інформаційно-комунікаційних технологій на базі IBM PC-сумісних персональних комп'ютерів. Розглядаються можливості й особливості різних комп'ютерних мереж, програмні компоненти інформаційно-комунікаційних технологій. Центральне місце в книзі відведене вивченню можливостей Інтернету, різних служб, програмних засобів підготовки графічних зображень для web, методик створення мультимедійних презентацій і web-сторінок. У книзі зроблений акцент на детальне відпрацювання практичних навичок користування засобами інтернет-технологій. У кожному розділі присутні дидактичні матеріали: практичні роботи, контрольні запитання і завдання для самостійної роботи.

**Царевич В. В. Электронная коммерция : пособие для вузов / В. В. Царевич, А. А. Кантарович, И. Т. Балабанов. – СПб. : Питер, 2002. – 320 с.**

У підручнику докладно і систематизовано викладені рішення проблем створення ефективних систем електронної комерції, реалізованих засобами мережі Інтернет. На відміну від багатьох відомих видань подібного характеру, дана книга має практичну спрямованість змісту, містить необхідні контрольні запитання і практичний матеріал стосовно роботи з електронними проектами. Видання призначене для викладачів вищих навчальних закладів, студентів, аспірантів і практичних працівників, що займаються обґрунтуванням економічної ефективності і впровадженням систем електронної комерції.

**Дунаев С. INTRANET-технологии. WebDBC. CORBA 2.0. Netscape. Suite. Borlander. Java. JavaScript LiveWire / С. Дунаев. – М. : Диалог – МИФИ, 2005. – 288 с.**

У книзі вперше з достатньою повнотою описується концепція мереж Intranet провідних фірм-виробників програмного забезпечення: Sun Microsystems, Netscape, Microsoft, Borlander.

Крім того, у книзі здійснений опис мови програмування Java і наведені численні приклади програм, сумісність яких перевірена на різних платформах і в різних операційних середовищах: Windows 95, Windows NT, Unix. Видання призначене для студентів і викладачів, адміністраторів мереж і баз даних, системних інтеграторів, розроблювачів інформаційних систем і технологій.

**Ефимова О. В. Практикум по компьютерной технологии. Упражнения, примеры и задачи / О. В. Ефимова, М. В. Мусеева, Ю. А. Шафрин. – М. : НБФ, 2006. – 560 с.**

Книга є практичною допомогою з курсу інформаційних технологій. У книзі міститься великий дидактичний матеріал для практичних занять за всіма розділами комп'ютерної технології. В окремій главі розглядаються сучасні засоби телекомунікацій у співтоваристві Інтернет. Розділ 3 присвячений виготовленню власної web-сторінки. У посібнику вивчаються основи сучасної інформаційної технології на базі комп'ютерів типу IBM PC і об'єктно-орієнтованого середовища Windows корпорації Microsoft. У розділі «Телекомунікації» вивчається ряд програмних продуктів Microsoft, Netscape та інших фірм.

**Якушина Л. Е. Изучаем Интернет, создаем Web-страничку / Л. Е. Якушина. – СПб. : Питер, 2000. – 256 с.**

У книзі в доступній формі розповідається про принципи роботи в Інтернеті, ефективний пошук інформації за допомогою пошукових серверів і каталогів, мережний етикет, електронну пошту й інші інформаційні ресурси. Автор знайомить із основами мови HTML, вчить створювати власні барвисті web-сторінки, розміщати їх у мережі й рекламувати усьому світу. Розповідає про нову мову моделювання віртуальної реальності і її величезні можливості. У додатку подається словник мови HTML і перелік найцікавіших сайтів за тематичними рубриками.

*Стауфер Т. Создание веб-страниц / Т. Стауфер. – СПб. : Питер, 2003. – 448 с.*

Якщо ви хочете навчитися створювати й публікувати сайти в Інтернеті, використовуючи для цього не графічні редактори, що автоматизують процес створення web-сторінок, а засоби мови XHTML, то це саме та книга, що вам необхідна. У ній послідовно, починаючи з азів (термінології, інструментарію, необхідного програмного забезпечення), докладно розповідається про процес створення web-сторінок. Велика увага в книзі приділяється сучасним технологіям побудови сайту, основам web-дизайну, включенню до складу web-сторінок звуку, відео й анімації, CGI-програмуванню й популярній мові написання скриптів JavaScript, графічним редакторам Macromedia Dreamweaver і Microsoft FrontPage. Книга буде корисна всім, хто хотів би навчитися створювати web-сторінки з нуля й ознайомитися із сучасними web-технологіями.

*Джерк Н. Разработка приложений для электронной коммерции. Библиотека программиста / Н. Джерк. – СПб. : Питер, 2001. – 512 с.*

У цій книзі описується застосування інструментальних засобів фірми Microsoft для побудови надійних і масштабних комерційних сайтів. Розглянуто основні питання: від вироблення загальної концепції електронного бізнесу до пророблення інтерфейсу покупця із включенням у нього елементів сучасного маркетингу. Автор ілюструє основні принципи розробки сайту на одному простому, легко досліджуваному прикладі – електронному магазині. Під час його створення задіяні всі ключові принципи й засоби електронної торгівлі, які можуть знадобитися в практичній роботі.

*Леонтьев Б. Web-дизайн: Тонкости, хитрости и секреты / Б. Леонтьев. – М. : Познавательная книга плюс, 1999. – 192 с.*

Книга розрахована як на широке коло користувачів, так і на професіоналів. Web-сторінки можна створювати кількома способами – із нуля або на основі існуючого шаблону. У вас можуть бути проблеми при тестуванні створеного документа, тому що згенерованим посиланням властиво помилятися. Вирішити цю проблему може вже не редактор, а звичайний оглядач WWW. Якщо у вас є колекція картинок, фонів, маркерів і роздільників, то створення web-сторінок – це досить захоплююче заняття. Вам обов'язково знадобиться софт для перетворення звичайних текстових файлів у файли формату Hyper Text Markup Language. Гарний редактор web-сторінок повинен дозволяти редагувати код HTML і переглядати зроблені в такий спосіб зміни в режимі безпосереднього перегляду. Якщо ж ви починаєте з нуля, то ви повинні ввести або імпортувати текст у документ, відформатувати абзаци, розмістити картинки. Усі картини й текстові файли зберігаються в одному каталозі. Потім все це відправляється до вашого постачальника Інтернету на його вінчестер.

### **Методичні вказівки з вивчення дисципліни**

#### **Вступ до інтернет-технологій. Методи доступу до мережі Інтернет**

Експериментальна версія першої комп'ютерної мережі пакетної передачі даних ARPANET (попередниця Інтернету) складалася з 4-х вузлів і була введе-

на в експлуатацію в 1969 році. Сьогодні з Інтернетом працюють понад 150 мільйонів користувачів у 160 країнах. Інтернет об'єднує велику кількість комп'ютерних мереж в єдину супермережу. Основою такого об'єднання є сімейство протоколів TCP/IP (протокол керування переданням/міжнародний протокол). IP відповідає за адресацію в мережі, а TCP забезпечує доставку повідомлень за необхідною адресою.

Для передання даних з одного комп'ютера на інший застосовується метод комутації пакетів, при якому передані дані розбиваються на частини (пакети). До кожного пакета приєднується заголовок, який містить необхідну інформацію щодо маршруту доставки. Комп'ютери у мережі аналізують ці заголовки і передають пакет даних наступному вузлу мережі так, що кожного разу він опиняється ближче до місця свого призначення. Якщо будь-яке мережне з'єднання розривається, комп'ютери-маршрутизатори обирають інші маршрути доставки пакетів, що підвищує надійність мережі.

У мережі Інтернет надаються такі послуги:

1. E-mail (електронна пошта) – це можливість надсилати (та отримувати) електронні повідомлення практично у будь-яку комп'ютерну мережу світу.

2. News (система телеконференцій) – це доступ до своєї бази даних, яка містить понад 200 тисяч різноманітних груп новин (тем дискусій). Тематика телеконференцій охоплює всі аспекти комп'ютерної техніки, науки, бізнесу, розваг. Кожен користувач Інтернету може взяти участь у будь-якій телеконференції, до якої він має доступ (тобто які є на сервері, до якого він підключений).

3. Передання файлів (FTP-архіви) – це можливість отримувати інформацію у вигляді файлів з кількох десятків тисяч комп'ютерів (FTP-серверів), розкиданих по всьому світу. У цих архівах вміщуються різноманітні файли – від комп'ютерних ігор до текстів класичної літератури. Пошук необхідних файлів можна виконувати за допомогою програми Archie.

4. Telnet (доступ до інших комп'ютерів мережі) – це можливість працювати на віддаленому комп'ютері, а також можливість з'єднуватися з великою кількістю відкритих серверів, які містять бібліотечні каталоги, бази даних, зведення фондових ринків, довідники, ігри тощо.

5. WWW (World Wide Web) – це розподілена по всій земній кулі, доступна всім користувачам комп'ютерна енциклопедія, у якій міститься інтегрована інформація, доступна в Інтернеті у вигляді сукупності гіпертекстових документів. Web-сторінка може містити текст, графічні зображення, відео та звук. Web-технологія має можливість доступу до будь-якої з послуг Інтернет. Для підключення до Інтернету потрібно знайти того, хто вже підключений і готовий забезпечити ваше підключення.

Існує 4 види підключення:

1. Off-line. Користувач може читати та писати листи (e-mail) або переглядати матеріали телеконференцій, не займаючи ліній зв'язку. При цьому інформація, що відправляється користувачем, накопичується в комп'ютері користувача, а та, що надходить на його ім'я, – на вузлі Інтернет, до якого він підключений. Користувач запускає спеціальну програму, яка за допомогою модему зв'язується з вузлом, передає та приймає накопичену інформацію, після чого

сеанс зв'язку завершується. У цьому режимі є доступ лише до електронної пошти (e-mail) та телеконференцій (news).

2. On-line. У цьому разі на час сеансу зв'язку комп'ютер користувача перетворюється на екран та клавіатуру, підключені до одного з комп'ютерів вузла. У цьому режимі є доступ до усіх послуг Інтернет, але з деякими обмеженнями.

3. Сеансове IP-з'єднання, IP-з'єднання по комутованій лінії (dial up IP). Використання цього режиму можливе лише під час сеансу зв'язку з вузлом. У цей час користувач має доступ до всіх послуг Інтернет у повному обсязі.

4. Повне підключення до Інтернету (on-line IP-з'єднання). Цей режим потребує наявності постійного (некомутованого) зв'язку між комп'ютером користувача та вузлом. Для цього найчастіше використовуються виділені телефонні лінії.

Розглянемо декілька найбільш відомих пошукових служб:

1. ALTA VISTA (<http://www.altavista.digital.com>). З'явилася у 1995 р., належить компанії Digital. Робота забезпечується оригінальними алгоритмами контекстного пошуку та індексації інформації, знайденої на серверах усього світу.

2. YAHOO! (<http://www.yahoo.com>). Найпопулярніший та найвідоміший у світі сервер, що забезпечує ієрархічний та предметно-орієнтований пошук. Існує з 1994 р. Це найстаріший каталог Інтернет, який уперше дав можливість структурного пошуку згідно з тематикою. Сьогодні практично всі пошукові машини мають такі властивості.

3. Одним з найстаріших пошукових серверів є LUCOS (<http://www.LUCOS.com>). Існує з 1994 р. Зараз частково належить компанії Microsoft.

4. EXCITE (<http://www.excite.com>). За своєю структурою близький до YAHOO! Дані оновлюються щотижня. Можна також переглянути систематизовану рекламу, яка розміщена у телеконференціях за останні 2 тижні.

5. Infoseek (<http://www.infoseek.com>) пропонує не тільки пошук по web-вузлах, але й по директоріях компаній. За невелику платню пропонується пошук у деяких комерційних базах даних.

6. HOTBOT (<http://www.hotbot.com>). З'явилась у 1996 р. Має стандартні логічні оператори пошуку. Виконується сортування за частотою вживання фрази або слова в документі. Можна обмежити пошук ім'ям домена.

7. MAGELLAN (<http://www.mckinley.com>). Існує з 1995 р. Дає результати, старанно проаналізовані та ранжировані.

8. WEBCRAWLER (<http://www.webcrawler.com>). Існує з 1995 р., належить компанії AMERICA-ON-LINE. Як і YAHOO!, класифікований. Має скорочене або нескорочене виведення результатів.

9. METACRAWLER (<http://www.metacrawler.com>). Не має своєї бази даних, але надсилає запит у різні пошукові служби (Excite, Yahoo!, Lycos, Infoseek, Webcrawler, Alta vista). За швидкістю поступається іншим службам, але дає найбільш докладні результати пошуку по всьому WWW.

10. WHOWHERE (<http://www.whowhere.com>). Тут можна знайти адресу e-mail потрібної людини або фірми. Можливе внесення інформації щодо себе, при цьому гарантується конфіденційність.



11. Для пошуку умовно безкоштовного програмного забезпечення існує сервер за адресою <http://www.shareware.com>. Ця пошукова система видає адресу FTP-сервера, у якому можна знайти потрібну програму.

Нижче розглянуто кілька російських пошукових служб.

1. Rambler (<http://www.rambler.ru>). Найбільш потужна система, створена спеціально для пошуку матеріалів на серверах СНД. Працює за всіма кодуваннями.

2. Російська машина пошуку (<http://search.interrussia.com>). Можливі логічні оператори, великі та малі літери розрізняються.

3. Russian Internet Search (<http://www.search.ru>). Пошук у WWW. Є логічні оператори та розрізнення великих та малих літер.

4. Aport! (<http://russia.agama.ru/aport>). Пошук у WWW. Існують логічні оператори та розрізнення великих та малих літер.

Існує українська пошукова служба UAHOO (<http://uahoo.gu.net>).

### **Контрольні запитання та завдання**

1. Що таке пошукова служба?
2. Назвіть найвідоміші пошукові машини.
3. Як треба оформлювати запит до пошукової служби?
4. Як використовують логічні оператори?
5. Обґрунтуйте методику використання пошукових служб.
6. Які вихідні дані потрібні для створення запиту на пошук?
7. Як треба оформлювати запит на пошук?
8. Як треба виконувати аналіз результату пошуку?

**Концепції побудови мереж. Архітектура мереж. Протоколи. Робота в мережі в режимі IP-підключення. IP-адреса і порти, особливості протоколу TCP/IP; IP-телефонія. Засоби комунікації і світові мережі. Електронна пошта, обробка поштових посилок. FTP – засіб передачі файлів у виробництві електронних засобів**

Інтернет складається з величезної кількості мереж, до яких входять комп'ютери, розташовані у різних куточках Землі, вони працюють під керівництвом різних операційних систем, реалізовані на різних платформах (IBM, Sun, Macintosh, VAX). Для ідентифікації кожного комп'ютера в Інтернет є своя унікальна IP-адреса, яка складається з 4-х чисел, відокремлених крапками, наприклад, 194.44.7.6. Загальний адресний простір становить  $2^{32}$  адрес. Число, що стоїть зліва, позначає велику мережу, а числа, які стоять праворуч, вказують на більш дрібні ділянки всередині цієї великої мережі. І так до номера конкретного комп'ютера (за наведеним прикладом це число 6). Для того, щоб полегшити запам'ятовування адрес, комп'ютерам надають спеціальні імена, наприклад, univd.edu.ua.

Таке ім'я зветься доменним. Воно складається з 2-х, 3-х, 4-х або 5-ти позначок, відокремлених крапками. Позначки можуть бути довжиною до 63 символів, а імена – довжиною до 255 символів. Коли користувач вводить ім'я комп'ютера, то маршрутизатори, які обробляють потік даних у мережі, підставляють замість імені відповідні цифри – IP-адреси. Комп'ютер, відповідальний

за якусь ділянку адреси, переводить ім'я в IP-адресу. Такий комп'ютер називається сервер імені, який дає цифрову IP-адресу того комп'ютера, до якого звертається користувач, вказуючи тільки його ім'я. Залежно від місця знаходження цього комп'ютера і географічної відстані від нього до комп'ютера користувача такі запити можуть проходити через декілька серверів імен, перш ніж буде отримана остаточна адреса.

Система доменних імен працює навпаки в порівнянні з IP-адресацією. Якщо в IP-адресі така інформація міститься зліва, то в доменних іменах ця інформація праворуч. Наприклад, комп'ютер з ім'ям test.univd.edu.ua має адресу 194.44.7.6. В імені ua вказується його належність до домену верхнього рівня (усі мережні організації), univd – на належність його до ХНУВС, а test – ім'я комп'ютера.

Операційна система UNIX була розроблена фірмою Bell Telephone Laboratories для ЕОМ фірми DEC. При створенні UNIX була розроблена мова системного програмування C, якою врешті-решт вона й була написана. UNIX архітектурно незалежна від конкретної ЕОМ як за функціональною будовою, так і за мовою реалізації. Це була перша система, яка порівняно легко переноситься з однієї ЕОМ на іншу. Така мобільність є однією з головних причин її великої популярності. Вихідні тексти програм доступні, що дозволяє легко адаптувати її залежно від конкретних вимог. Існує безкоштовна версія цієї програми – Linux. Ця система є багатозадачною, у ній може також працювати багато користувачів. Протоколи Інтернет (TCP/IP) були в першу чергу запроваджені саме в цю систему. Працюючи з програмами telnet або FTP у Windows 95, користувач має справу з таким самим інтерфейсом, як у Linux. Кожен користувач повинен мати ім'я та пароль. Для входу в систему він повинен зареєструватися, тобто ввести своє ім'я та пароль у відповідь на запит системи.

Login: ім'я

Password: пароль

Якщо ім'я та пароль введені правильно, то система виводить запрошення, наприклад Class#.

Це означає, що вона готова прийняти команди. UNIX була моделлю для каталогів і підкаталогів, які використовуються у MS DOS. В іменах файлів може використовуватися практично будь-який символ. У UNIX розрізняються великі та малі літери. Наприклад, Network.txt та network.txt – два різних файли. Повне ім'я починається з косої риски і містить повний маршрут, який починається з кореневого каталогу, наприклад, /home/user1/mail/letter.txt. Код захисту файла має такий вигляд: gwx gwx gwx. Перший символ означає, що це є файл (у каталозі була б літера d). Наступні 3 символи означають дозвіл читання (r), запису (w) та виконання (x) для власника файла. Те саме значення мають ці літери у таких двох групах: для членів групи власника та для інших користувачів.

### **Контрольні запитання та завдання**

1. Як здійснити вхід та вихід із системи?
2. Назвіть основні особливості операційної системи UNIX.
3. Назвіть команди роботи з файлом.
4. Яку структуру має файлова система UNIX?
5. Які команди служать для сумісності UNIX з DOS?

6. Як визначити маршрут проходження повідомлень до комп'ютера?
7. Як змінити код захисту файла?
8. Що означає код захисту файла?

## **Основні принципи розробки і супроводу web-сторінок**

Основні вимоги до web-сервера:

1. Web-сервер повинен мати цікавий зміст. Інакше він ніколи не стане популярним і відвідуваним.
2. Структура web-сервера повинна бути зрозуміла кожному відвідувачу. Зрозуміла і прозора структура сервера залучає додаткових відвідувачів.
3. Необхідно постійно оновлювати інформацію на сервері.
4. Слід постійно прагнути до створення унікального стилю web-сервера.

### *Зміст web-сервера*

Кінцева мета створення сервера полягає в публікаціях у WWW, тобто в представленні на загальний розгляд документів і матеріалів, а також демонстрації можливостей компанії.

Необхідно поставити дві-три близькі мети, заради яких створюється сервер.

Потрібно керуватися двома принципами: унікальність та авторитетність.

Наявність унікальних ресурсів на сервері збільшить його відвідуваність. Змістом сервера повинен бути не тільки текст, але й графічні та мультимедійні матеріали. Наприклад, на сервері картинної галереї повинні бути ілюстрації, на сервері студії звукозапису – музичні уривки.

Зміст web-сервера повинен цілком відбивати всі напрямки діяльності компанії.

За принципом побудови web-сервери можна розділити на 3 групи:

1. Web-сервери, власники яких не розуміють, що таке WWW. Їм здається, що WWW – нова форма паперової продукції, тому вони будують свої сервери за моделями публікацій, тобто просто розміщують у web уже набридлу всім інформацію, каталоги чи журнали. Такі сервери не стають популярними.

2. Сервери, що містять оригінальну інформацію і багатий набір функцій для спілкування з відвідувачами. Такі сервери можуть привернути увагу динамічністю відновлення матеріалів і можливістю інтерактивної роботи з документами сервера.

3. Сервери, що надають відвідувачам інтерактивну інформацію. Вони повинні миттєво реагувати на запити користувачів і бути зв'язаними з базами даних і базами документів. Важливо забезпечити відвідувачу можливість зробити покупку в режимі реального часу.

### *Структура web-сервера*

Визначення структури web-сервера є одним із ключових етапів у процесі його створення. Від неї залежить зовнішня стрункість, взаємопов'язаність матеріалів, що представляються, і зручність керування даними, розміщеними на сервері.

Під час вибору схеми побудови сервера потрібно виходити з його змісту. Схема залежить від природи web-сервера, специфіки матеріалів, що зберігаються, логічного розподілу його змісту.

Домашня сторінка (Home page) служить точкою входу у вузол web. Зазвичай це вступна сторінка, що містить вітання і посилання на сторінки другого рівня, доступні на даному вузлі. Домашні сторінки формують перше враження, яке справляє на користувача вузол web і та організація, якій він належить. Деякі маленькі вузли web типу персональних сторінок складаються тільки з домашньої сторінки, тоді як на великих вузлах може бути кілька домашніх сторінок.

Діапазон способів організації вузлів web дуже широкий: від вузлів, що мають чітку лінійну структуру, до вузлів, у яких узагалі немає чіткої структури. Звичайно сторінки розташовують в ієрархічному чи лінійному порядку, а також у вигляді павутини. Подорож вузлом повинна бути легкою для користувача, інтуїтивно зрозумілою. Найчастіше використовується деяка комбінація різних способів організації сторінок.

#### *Ієрархічна організація*

Вузли у web, що відповідають ієрархічній чи деревоподібній організації, мають єдину точку входу у вузол, інші сторінки розташовуються на вихідних з неї відгалуженнях. Даний підхід зручний, якщо ви можете легко розбити свою інформацію на категорії і підкатегорії. При ієрархічній організації вузла web до сторінки найнижчого рівня веде один і тільки один шлях.

Така строката структура вузла може викликати проблеми в користувачів. Якщо користувач пройшов на кілька рівнів униз по одному зі шляхів, а потім вирішив потрапити в іншу частину дерева, то йому доведеться повертатися назад.

#### *Лінійна організація*

Якщо ви хочете, щоб користувачі читали вміст вузла, як книгу чи журнал, щоб вони пройшли заданим шляхом від початку і до кінця вузла, потрібно вибрати лінійну організацію web-вузла. Сторінка може мати кілька зв'язків із примітками і доповненнями, але для просування далі користувач повинен знов повернутися до неї.

Для великого вузла web лінійна організація не дуже підходить. Небажано створювати дуже довгі сторінки – це стомлює користувача. Користувачу може не сподобатися необхідність пройти через безліч сторінок, перш ніж потрапити на потрібну.

#### *Організація у вигляді павутини*

Організація інформації у вигляді павутини, імовірно, якнайкраще підходить для більшості випадків. У цій структурі сторінки зв'язані одна з одною загальним контекстом. До однієї сторінки може вести кілька зв'язків, з кожного документа є принаймні два виходи. Зв'язки іноді створюють коло.

Для подорожей по web ця організація вузла вкрай приваблива. Користувачам подобається можливість швидкого проходу по вузлу і свобода переміщення. Головний недолік такої структури в тому, що користувач може що-небудь пропустити, не пройшовши по всіх зв'язках.

При розробці організаційної структури web-сервера краще дотримуватися одного з готових варіантів.

Найбільш розповсюдженими схемами побудови сервера є книга, просторова схема і дерево. Ніколи не слід комбінувати і з'єднувати кілька схем.

#### *Книга (лінійна)*

Книги взагалі споконвічно прекрасно підходять для розміщення на web-сервері. У них є чіткий розподіл на частини і глави, розділи і підрозділи, які можна помістити в окремі документи, створивши на них посилання.

Недоліком є необхідність дроблення вмісту web-документів на маленькі частинки-сторінки. У звичайних книгах є закладки і нумерація сторінок – тут цього немає. Це призводить до деяких складностей при переміщенні по цих сторінках.

#### *Просторова схема (павутина)*

Просторова схема використовується, коли є кілька слабо зв'язаних між собою розділів змісту, що включають матеріали, які не мають контактів з іншими розділами.

Тут теж є недоліки. Дуже непросто грамотно розробити переходи між такими розділами й усередині розділів.

#### *Дерево (ієрархічна)*

Зміст web-сервера розходиться від загального кореня до галузей, що представляють собою великі розділи, потім до гілок, що є рубриками великих розділів і, нарешті, до листків – кінцевих неподільних інформаційних блоків. Така схема зазвичай застосовується на серверах, що зберігають свої матеріали у файлових системах, а не в базах даних і документних базах. Тому будь-яка файлова система побудована за такою схемою з директоріями, каталогами і підкаталогами.

Недоліком є складність переміщення по галузях дерева при великих його розмірах. Складно також переміщати документи, тому що потрібно відстежувати посилання.

Створюючи каркас сервера, потрібно пам'ятати про аудиторію, на яку він розрахований. Кожна психологічна група користувачів має свої методи роботи з документами. Одні хочуть працювати з каталогами, інші – з пошуковими системами сервера. Якщо сервер створюється для роботи з однією групою користувачів, то його структура повинна цілком задовольняти деякий стереотип поведінки. Якщо таких груп небагато, то на сервері повинно бути реалізовано кілька схем, що відповідають кожній групі, причому дані схеми повинні існувати незалежно і не перемішуватися.

### **Розташування графіки. Загальні компоненти інтерфейсу Flash MX**

Робота з комп'ютерною графікою – один з найпопулярніших напрямків використання персонального комп'ютера. Комп'ютерна графіка включає введення, обробку і виведення зображення креслення, малюнків, текстів засобами обчислювальної техніки. Для роботи з комп'ютерною графікою є багато програм. Існує три види комп'ютерної графіки: растрова, векторна і фрактальна, що відрізня-

ються принципами формування зображення при відображенні на екрані монітора або при друкуванні на папері. Растрову графіку застосовують при розробці електронних (мультимедійних) і поліграфічних видань. Основним елементом растрового зображення є крапка. Програмні засоби для роботи з векторною графікою використовуються в першу чергу для створення ілюстрацій і меншою мірою для їх обробки, основним елементом зображення є лінія. Слово «фрактал» означає складений з фрагментів. Здатність фрактальної графіки моделювати образи живої природи обчислювальним способом часто використовують для автоматичної генерації незвичних ілюстрацій, створення візерунків у текстильній промисловості. Програмні засоби для роботи з фронтальною графікою призначені для автоматичної генерації зображень шляхом математичних розрахунків. Створення фронтальної художньої композиції реалізується в програмуванні.

Банер (banner) – рекламний заголовок – графічний елемент, який за домовленістю або за будь-яку плату розташований на певних web-серверах, при натисканні мишею на нього користувач автоматично потрапляє на сайт рекламодавця. Основні вимоги до банерів: банерами повинні бути файли з розширенням gif, swf, jpg; вони повинні мати графічну інформацію або елемент мультиплікації, який несе рекламний або інформаційний характер; у циклі адміністрованого банера є можливість змінювати банер на інший без змін коду сайта. Поради з підвищення ефективності банера: його розмір не повинен перевищувати критичний розмір і не впливати на інформативність web-сторінок; завантаження банера не повинне впливати на час завантаження всієї web-сторінки.

### **Створення документів HTML. Вивчення базових HTML-директив (тегів) формування тексту, вставки ілюстрацій, гіпертекстових посилань та таблиць у web-документи**

HTML (Hyper Text Markup Language – мова розмітки гіпертексту) – це метод представлення зовнішнього вигляду і змісту документів, заснований на використанні стандартного ASCII-файла. HTML складається зі спеціальних розмічувальних директив, чи тегів (tag), змішаних зі звичайним текстом. Документ – це те, з чого складається web-сторінка. Документи можуть створюватися за допомогою простого ASCII-редактора.

HTML потребує від розроблювача структурного мислення. Мета полягає в тому, щоб передати користувачу структуру документа.

HTML так описує документи, що будь-який браузер, незалежно від того, на якому комп'ютері він запущений, зможе прочитати і відобразити ці документи. На відміну від цього, наприклад, документ, створений у Word, може бути переглянутий тільки в Word.

HTML – це засіб опису структурованих документів, тобто документів, що мають загальні елементи, такі як заголовки, підзаголовки, абзаци і списки.

HTML – це формат представлення інформації, під час використання якого не потрібно піклуватися про проблеми форматування тексту.

Основні команди HTML розбиваються на такі категорії:

1. Команди визначення структури. Ці команди визначають, що даний файл є документом HTML і описують дані, що знаходяться в ньому.

2. Команди форматування абзацу. Ці команди визначають розташування абзаців і рівні відступів.

3. Команди форматування символів. Ці команди дозволяють застосовувати різні типи написання символів у документі (напівжирний, курсив, підкреслений).

4. Команди визначення списків. HTML підтримує кілька різних типів списків, включаючи нумерований, список з позначками і список визначень.

5. Команди визначення гіпертекстових зв'язків. За допомогою цих команд можна забезпечити перехід від одного документа до іншого.

6. Команди інтегрування мультимедіа. За допомогою цих команд можна відображати графічні файли, забезпечувати web-сторінки звуковими ефектами і надавати користувачу можливість перегляду відеофрагментів.

HTML-файл – це звичайний ASCII-файл, що має розширення htm (у UNIX розширення html).

Теги – це інструкції браузеру, що вказують, яким чином слід відобразити текст. Тег завжди починається зі знака < і закінчується знаком >.

При написанні тегів заголовні і малі літери не розрізняються. Наприклад, теги <HTML>, <Html>, <html> припустимі.

**Основи мови розмітки гіпертекстових посилань HTML. HTML-код web-сторінки. HTML-редактор SNK-HTMLPad. Додаткові можливості HTML. Створення web-сторінок з допомогою HTML-редактора. DHTML – динамічна мова гіпертекстової розмітки**

Існує три теги, які використовують при описі структури HTML-документа. Вони не впливають на зовнішній вигляд відформатованого документа і використовуються тільки для того, щоб допомогти браузеру і різним пошуковим програмам розібратися в структурі документа.

Перший тег, що повинен знаходитися в будь-якому HTML-документі – <HTML>...</HTML>. Цей тег указує на те, що даний документ дійсно містить HTML-текст. Усе, що міститиметься в документі, повинно знаходитися всередині цього тега.

```
<HTML>
```

Текст документа

```
</HTML>
```

Необхідно розбити документ на дві частини – заголовок і власне текст.

Визначення заголовка повинно міститися всередині тега <HEAD>...</HEAD>:

```
<HTML>
```

```
<HEAD>
```

Опис заголовка

```
</HEAD>
```

Текст документа

```
<HTML>
```

Для того, щоб указати заголовок, використовується тег <TITLE>...</TITLE>. Коли браузер зустрічає цей тег, він відображає все, що

знаходиться всередині нього, як заголовок. Netscape відображає заголовок у рядку заголовка. Весь інший HTML-документ, включаючи весь його текст, міститься всередині тега <BODY>...</BODY>:

```
<HTML>
<HEAD>
<TITLE> Це заголовок документа </TITLE>
</HEAD>
<BODY>
Текст документа
<BODY>
<HTML>
```

Більшість браузерів відобразить документ і без цих тегів, але їх бажано використовувати – так легше відредагувати документ надалі, коли ці теги стануть необхідними.

У HTML-файл можна додавати коментарі, що браузер ігнорує і не відображає на екрані. Коментарі роблять структуру HTML-файла більш зрозумілою. Кожен рядок коментарів повинен знаходитися всередині тега <!-- ... -->

Наприклад, <!-- Це коментар -->

На відміну від коментарів у програмах, написаних традиційними мовами програмування, коментарі в HTML-файлах передаються користувачу. Тому коментарі великого обсягу можуть сповільнити передачу файла.

Основним елементом HTML-файла є абзац. Текст, що вводиться у файл поза тегів <HTML>, <HEAD>, <TITLE>, <BODY>, відобразатиметься у вигляді абзаців. Браузер виводить текст абзаців зліва направо і зверху вниз, використовуючи поточний розмір вікна. Цей процес називається автоматичним вирівнюванням. Таким чином, якщо користувач змінить розмір вікна, текст буде розміщений дещо по-іншому, оскільки браузер по-іншому перенесе слова. Тому те, як розбитий текст на рядки при написанні HTML-файла, не впливає на те, як текст відобразатиметься.

Тільки тег <P> відзначає кінець абзацу і відокремлює абзаци один від іншого.

```
<BODY>
```

```
Перший абзац тексту <P>
```

```
Другий абзац тексту <P>
```

```
</BODY>
```

Для того, щоб указати початок нового рядка, використовується тег <BR>.

Тег <HR> указує на те, що браузер повинен відобразити горизонтальну лінію, що йде через весь екран.

Для опису заголовка використовується тег <Hn>, де n – число від 1 до 6. Чим більше число, тим більший відступ заголовка від лівого краю сторінки (цифра – це рівень заголовка).

Наприклад, <H1> Опис курсу </H1>

Різні рівні заголовка дозволяють організувати документ ієрархічно.

Браузер звичайно видаляє всі зайві пробіли між словами. Єдиним винятком є попередньо відформатований текст. Цей текст відображається шрифтом фіксованої ширини зі збереженням усіх пробілів.



Усі пробіли, розташовані всередині тега <PRE>...</PRE>, будуть відображені на екрані. Це дає можливість розташувати текст саме так, як ви хочете.

Завчасно відформатований текст найкраще використовувати для вихідних текстів ієрархії, а також графіків, діаграм, таблиць, створених за допомогою стандартного набору символів.

Теги форматування символів дозволяють керувати відображенням окремих символів і слів.

<B> ... </B> – для виділення напівжирним.

<I> ... </I> – для виділення курсивом.

<U> ... </U> – для виділення підкресленням.

Для введення деяких спеціальних символів потрібно ввести такі команди:

Команда	Символ
& lt	<
& gt	>
& amp	&
& quot	"
& nbsp	(пробіл)

HTML підтримує кілька типів списків:

- а) неупорядкований (з позначками);
- б) упорядкований (пронумерований);
- в) список без позначок;
- г) меню – кожен елемент займає один рядок;
- д) словник термінів (список визначень).

Спочатку вказується одна з команд (тегів):

Неупорядкований – <UL> ... </UL>.

Упорядкований – <OL> ... </OL>.

Без позначок – <DIR> ... </DIR>.

Меню – <MENU> ... </MENU>.

Словник термінів – <DL> ... </DL>.

Перед кожним елементом списку потрібно поставити тег <LI>.

Списки можуть бути вкладеними.

Приклад:

<P> Види списків <P> Неупорядкованого <UL>

Списку <LI> неупорядкований

<LI> упорядкований

<LI> меню

-----

<UL>

Вид на екрані

Види списків

Неупорядкованого

- неупорядкований
- упорядкований
- меню

Приклад: <P> Види списків <P>  
Упорядкованого <OL>  
Списку <LI> неупорядкований  
<LI> упорядкований  
<LI>меню

-----

<OL>

Вид на екрані

Види списків

Упорядкованого

- 1) неупорядкований
- 2) упорядкований
- 3) меню

Меню і списки без позначок – це варіанти неупорядкованого списку. Вони призначені для відображення списку з елементів невеликого розміру, які можна відобразити в стислому вигляді. Елементи меню дуже часто оформляють як гіпертекстові зв'язки. Меню – це список елементів (чи навіть невеликих абзаців), що позбавлений будь-яких позначок. Списки визначень призначені для відображення списків термінів і їхніх визначень. Кожен елемент списку складається з двох частин: терміна і його визначення. Терміну передує тег <DT>, а визначенню – тег <DD>. Визначення повинно бути всередині тега <DL>.

<P> Визначення: <P>

<DL>

<DT> Браузер

<DD> Прикладна програма для перегляду гіпертекстових документів

<DT> ARPANET

<DD> Мережа агентства провідних дослідницьких проектів

</DL>

Теги визначення гіперзв'язків дозволяють указувати, як документ зв'язаний з іншими документами web і як частини документа зв'язані між собою.

Гіпертекстовий зв'язок – це покажчик на визначене місце в іншому документі. У найпростішому випадку місце, що вказується, знаходиться всередині документа. Але можна також визначити гіперзв'язок до довільних документів, а також до зовнішніх графічних файлів, відео- й аудіофрагментів.

Для того, щоб установити гіперзв'язок, потрібно знати початкову і кінцеву точки. Вони позначаються тегом <A>. Потрібно знати:

- а) ім'я (чи URL) файла, з яким потрібно встановити зв'язок;
- б) текст, що буде виділений у документі.

Щоб одержати більш докладну інформацію про фірму Sony, клацніть <A HREF="http://www.sony.com"> тут </A>. Буде підкреслене слово «тут». Це і є позначення гіперзв'язку. Після кліка мишею користувач переміститься на початкову сторінку, зазначену в тезі <A>. Якщо посилання в тому ж документі, то можна використовувати таку конструкцію:

Клацніть <A HREF="# TargetName"> тут </A>

Текст документа

`<A NAME="TargetName"></A>`

Сюди буде здійснений перехід.

Універсальна форма (комбінація попередніх). Щоб одержати докладну інформацію, клацніть `<A HREF="http://www.domain.name.html#TargetName">` тут `</A>`.

У цільовому файлі повинен знаходитися тег, що описує відповідну точку, куди буде здійснений перехід (наприклад, `<A NAME=...`).

## **Створення динамічних web-сторінок. Модель об'єктів JavaScript. Методи та події об'єктів JavaScript. Мови сценаріїв JavaScript і VB Script. Інтернет-технології і мова програмування Java. Використання мов сценаріїв VB Script або JavaScript для виробництва електронних засобів**

Мультимедіа – це об'єднання тексту, звуку, графічних зображень і оцифрованого відео.

Інтерактивна мультимедіа забезпечує взаємодію з користувачем.

Web дозволяє здійснити передачу оцифрованого звуку на комп'ютер користувача і потім програвати його за допомогою спеціальних програм.

Звукові файли для PC з Windows повинні бути у форматі wav (від wave – хвиля).

Інший звуковий формат – midi (musical instrument digital interfase – цифровий інтерфейс музичних інструментів) – міжнародний стандарт для представлення звучання музичних інструментів, що використовують музиканти під час студійних записів. У форматі mid, крім мелодій, нічого не можна записати (наприклад, шуми чи гавкіт собаки).

Для графіки найчастіше використовуються формати gif і jpg (jpeg).

Для цифрового відео використовуються такі формати:

avi (Audio Video Interleave – суміш звуку і відео) – це стандартний формат файлів Microsoft Video for Windows;

mov – формат для різних платформ, створений Apple Computer;

mpeg – новий формат, що вимагає для перегляду спеціального апаратного забезпечення (відеоадаптер з розпакувальником mpeg).

Перевага mpeg – можна побачити на комп'ютері повноекранне відео в реальному режимі часу.

Картинки можуть зробити сторінку дуже привабливою.

Для вставки графіки використовується тег `<IMG>`.

`<IMG SRC="ім'я файла">`

У HTML підтримуються 2 формати: gif і jpg.

Може бути такий формат: `<IMG SRC="ім'я файла" ALT="опис картинки">`

Наприклад, `<IMG SRC="rain bow.gif" ALT="картинка з зображенням веселки">`

Текстовий опис відобразатиметься в текстових браузерях, а також тоді, коли відключена підтримка графіки в графічних браузерях.

Можна описати картинку всередині визначення гіперзв'язку.

<A HREF="місце призначення"> <IMG SRC="ім'я файла"> </A>

Тепер якщо користувач клацне мишею на будь-якій частині малюнка, він активізує відповідний гіпертекстовий зв'язок.

Для організації відсилання електронної пошти з web-сторінки використовується конструкція <A HREF=mail to:ім'я користувача@домен> Текст посилання </A>.

Коли користувач клацне мишею на словах «текст посилання», то з'явиться діалогове вікно поштової програми Netscape.

Для вставки звукових файлів використовується конструкція:

<A HREF="file://audio.wav"> Клацніть тут </A>, щоб почути звук.

Після кліка на виділеному тексті браузер прийме зазначений файл із сервера і програє його за допомогою програми, здатної працювати з файлами формату wav.

Для перегляду оцифрованого відео використовується конструкція виду

<A HREF="file://video.avi"> Клацніть тут </A>, щоб побачити цифрове відео.

У HTML існують додаткові можливості для створення інтерактивних форм.

Інтерактивні форми дозволяють web-серверу одержувати інформацію від користувача за допомогою зручного інтерфейсу, що складається з керуючих елементів.

Введіть своє ім'я

Введіть по батькові

Введіть прізвище

Введіть ім'я користувача

Введіть пароль

Введіть свою e-mail адресу

Після того, як користувач увів усю необхідну інформацію, браузер передає накопичені дані серверу, що ці дані обробляє. Форма складається з одного чи кількох елементів керування. Елементи керування – це елементи інтерфейсу, що дозволяють користувачу взаємодіяти з HTML-документом.

1. Рядки введення тексту, що використовують для введення алфавітно-цифрових даних.

2. Рядки введення пароля.

3. Области введення тексту, що дозволяють вводити більше одного рядка.

4. Вимикачі – прапорці перевірки, що дозволяють чи забороняють дію тієї чи іншої опції.

5. Перемикачі – радіокнопки, що дозволяють користувачу вибрати одну кнопку зі списку.

6. Спливаючі меню і списки, що розкриваються, – вибір одного елемента зі списку.

7. Кнопки Submit і Reset – передання даних на web-сервер чи очищення введених даних.

Існують спеціальні редактори для створення HTML-документів.

## **Індивідуальні розрахункові завдання, контрольні завдання, теми для рефератів**

Слово «реферат» походить від латинського *refero*, що означає «переказувати, відтворювати, доповідати, повідомляти». Під рефератом розуміють стисле викладення, огляд матеріалу з деякої проблеми, скорочений виклад змісту книги. Рефератом називають стислу доповідь з оглядом існуючих готових матеріалів певної проблеми, але при цьому автор готує письмовий план і конспект доповіді.

Написання реферату передбачає:

- 1) вивчення літератури з визначеного питання;
- 2) вивчення інформації, яка є в літературі та Інтернеті;
- 3) збирання та узагальнення матеріалу;
- 4) складання плану реферату;
- 5) оформлення реферату.

Реферат відрізняється від курсової та дипломної роботи тим, що ступінь творчості в рефераті менша. У рефераті подається тільки первинне осмислення й узагальнення визначеного обсягу інформації, який накопичений вченими та викладений у літературі.

При оцінюванні реферату враховують:

- 1) чи зміг автор підібрати достатній список літератури, який необхідний для осмислення питання, що позначене в якості теми;
- 2) чи склав логічно обґрунтований план, який відповідає сформульованій меті та поставленим задачам;
- 3) чи вдалося зібрати необхідний матеріал і осмислити його правильно;
- 4) чи вміє автор аналізувати інформацію;
- 5) чи відповідає реферат вимогам об'єктивності, коректності, грамотності, логічності, аргументованості, доказовості, ясності стилю та чіткості викладення;
- 6) чи достатньо обґрунтовані висновки, чи відповідають вони задачам;
- 7) які методи в роботі над рефератом використовував автор;
- 8) наскільки самостійно виконана робота;
- 9) чи правильно в цілому оформлений реферат, посилання на використані джерела, список літератури.

### *Складання бібліографії*

Перед тим як приступити до безпосереднього збирання матеріалу, необхідно скласти список літератури, де висвітлене питання, стосовно якого повинен бути складений реферат. Джерелами для формування такого списку можуть бути рекомендації викладача, алфавітні бібліографічні та систематичні предметні каталоги бібліотек. Крім того, можна використати енциклопедії, енциклопедичні словники або тематичні добірки посилань в Інтернеті.

### *Збирання матеріалу*

У процесі вивчення літератури використовують методи конспектування і виписок. Конспекти бувають кількох видів: планові, текстуальні, вільні і тематичні. Планові конспекти – це конспектування джерела за його планом, такий

конспект повністю відображає структуру книги. Текстуальні представляють собою збірку цитат, яка подає основний зміст книги через авторське трактування найважливіших ідей. Вільний конспект – комбінований, складається як із цитат, так і з переказу своїми словами змісту окремих розділів. Тематичний конспект – передбачає підбір цитат з різних джерел або переказ чужих думок, які рознесені по рубриках, пунктах плану та розкривають зміст теми. Конспектування використовують при роботі з однією або двома монографіями.

Якщо літератури багато, використовують виписки. Для конспектування відбирають декілька фундаментальних джерел, а інші переглядають та виписують найбільш цікаві відомості, які роблять з посиланням на джерело.

### *Осмислення та систематизація матеріалу*

У процесі осмислення матеріалу відшукуються відповіді на такі запитання: що? де? коли? чому? навіщо? і з якою метою? Відповіді повинні бути обґрунтованими, необхідно представити аргументи, які доводять справедливість висновків. Відповіді також не можуть суперечити одна одній.

### *Складання плану реферату*

План реферату відображає в концентрованому вигляді його суть. Це схематичне вираження того, що хоче сказати автор.

План може складатися трьома способами:

- 1) взяти за вихідну точку обрану тему і виходячи з неї сформулювати мету та задачі – вони дадуть назви розділів та параграфів реферату;
- 2) виходити із зібраного матеріалу, логіка якого підкаже структуру викладення;
- 3) змішаний, який складається з перших двох.

Структура плану включає:

Вступ, у якому містяться обґрунтування теми та її значимість, пояснення причин, чому обрано саме цю тему, чим обумовлена зацікавленість. Далі йде огляд літератури з обраної теми. У вступі необхідно чітко сформулювати мету, яку автор ставить перед собою, та за допомогою яких задач вона буде реалізована.

Основна частина реферату звичайно складається з двох розділів: а) теоретичного осмислення проблеми; б) викладення емпіричного, фактичного матеріалу, який аргументовано підтверджує поданих у першому розділі теорію. Вона повинна розбиватися на кілька підрозділів та параграфів.

Висновки містять результати осмислення проблеми, висновки, а також оцінку їх значимості.

Написання реферату здійснюється на підставі отриманого матеріалу відповідно до стандартів оформлення документації.

## **Теми для рефератів**

1. Електронний бізнес як економічне явище (поняття електронного бізнесу, інтернет-бізнесу, електронної комерції і комерції в Інтернеті; історія розвитку електронного бізнесу; основні напрямки розвитку і проблеми електронного бізнесу; стан електронного бізнесу в Україні).

2. *Напрямки розвитку електронної комерції* (сутність електронної комерції, її основні учасники; сектори електронної комерції: бізнес-бізнес (B2B), бізнес-споживач (B2C) тощо; переваги і недоліки електронної комерції в порівнянні з традиційним бізнесом).

3. *Концепція ASP – надання послуг доступу* (поняття і типи ASP; переваги використання послуг ASP; проблеми вибору ASP; прогнози і перспективи розвитку ринку ASP).

4. *Он-лайнві аукціони* (визначення і види он-лайнвих аукціонів; принцип роботи он-лайнвих аукціонів; ризики учасників торгів).

5. *Надання послуг за допомогою Інтернету* (поняття послуги, його відмінність від інших об'єктів продажу (товарів і робіт); інтернет-банкінг – надання банківських послуг; інтернет-трейдинг (брокерські послуги); інтернет-страхування; туристичні послуги в Інтернеті; надання послуг доступу (ASP)).

6. *Брокерські послуги в Інтернеті (інтернет-трейдинг)* (сутність брокерської діяльності; способи надання брокерських послуг в Інтернеті: зі сторінок web-сайта компанії, із застосуванням спеціального програмного забезпечення; основні системи інтернет-трейдингу; типи брокерських рахунків).

7. *Підприємство в системі електронної комерції* (умови впровадження систем електронної комерції; можливості підприємства у сфері електронного бізнесу (електронний документообіг, інтернет-постачання, торгівля через Інтернет, здійснення платежів, інтернет-маркетинг); віртуальні підприємства як нове явище в економіці).

8. *Система MRP – планування потреби в матеріалах* (історія і методологія MRP; логіка роботи MRP-модуля: інформація на вході і виході, схема циклу; недоліки MRP і подальший розвиток MRP-модуля).

9. *Система ERP – планування ресурсів корпорації* (історія розвитку ERP: попередні MRP і CRP-модулі; елементи системи ERP і логіка її роботи; переваги і проблеми).

10. *Інтернет-постачання підприємства* (що являють собою системи інтернет-постачання і для чого вони необхідні; корпоративні системи інтернет-постачання; системи управління ресурсами (ERP, MRP, CRP); переваги і недоліки систем інтернет-постачання).

11. *Електронний бізнес для малих підприємств* (визначення й особливості малого бізнесу; основні напрямки використання електронної комерції в малому бізнесі (представництво, комунікації, реклама, використання інформаційних можливостей Інтернет); особливості побудови електронного бізнесу для малих підприємств).

12. *Мережі як основа електронної комерції* (визначення Інтернету, його властивості, переваги і недоліки; структура Інтернету; мережна інфраструктура: адреси, протоколи, ресурси; інші види мереж (екстранет, інтранет тощо)).

13. *Всесвітня павутина WWW та її роль в електронній комерції* (виникнення і розвиток WWW; структура WWW; web-сайт: поняття, структура, класифікація; ефективність web-сайта: як її визначити і від чого вона залежить).

14. *Інтернет-представництво підприємства* (для чого необхідно представляти бізнес в Інтернет (порівняння традиційного й електронного бізнесу); види ін-

тернет-представництв (вітрини, столи замовлень, корпоративні сайти, бізнес-портали); представництва для багатобрендових компаній).

15. *Віртуальна торгівля* (торговий процес і його учасники; системи електронної торгівлі (вітрина, стіл замовлень, магазин, ТІС); типова структура інтернет-магазину; функції електронних магазинів, їхні переваги і недоліки).

16. *Організація інтернет-магазину* (що являє собою інтернет-магазин, його архітектура; вимоги до організації й етапи створення інтернет-магазину, його реєстрація, установка й оформлення; просування інтернет-магазину: відкриття, реклама, супровід).

17. *Платіжні системи в Інтернет* (поняття платіжної інтернет-системи, умови, учасники й інструменти платежів; кредитні схеми платежів; дебетові платіжні системи: електронні чеки, електронні гроші; перспективи і проблеми розвитку).

18. *Основні системи електронних платежів в Україні* (визначення системи електронних платежів, їх законодавча основа; національна система масових електронних платежів; характеристика найбільш поширених в Україні карткових систем: VISA, EUROCARD, MASTERCARD тощо; характеристика дебетових систем: WebMoney, PayPal, PayCash тощо).

19. *Банківські послуги в Інтернеті* (поняття вилученого банкінгу, його види (телефонний, відео-, РС-банкінг, інтернет-банкінг); банківські послуги в Інтернеті, технології інтернет-банкінгу; транзакції «Клієнт-банк»; проблеми безпеки інтернет-банкінгу).

20. *Інтернет-реклама* (особливості інтернет-реклами в порівнянні з традиційною; банерна реклама; альтернативна реклама (текстова, rich-media, міні-сайти, колажі); нав'язлива реклама (байрики, interstitials); оф-лайнова реклама інтернет-проектів).

21. *Сплата за інтернет-рекламу* (інтернет-реклама і критерії сплатити за неї; сплата на основі сrm; цінові моделі розміщення реклами: фіксована плата, срс, сrv, сра, сrs; ефективність застосування різних форм сплати).

22. *Електронні гроші (e-cash)* (що являють собою електронні гроші, історія їх розвитку; функції, які виконують електронні гроші; їх переваги і недоліки в порівнянні з традиційними грошима; як здійснюються платежі за допомогою електронних грошей; найвідоміші емітенти електронних грошей; проблеми і перспективи електронних грошей).

23. *Мобільний Інтернет (m-commerce)* (суть мобільної комерції, її апаратне забезпечення; протокол war; безпроводні мережі передачі даних; мобільні фінансові послуги, мобільний офіс; переваги і недоліки мобільної комерції).

### **Приклади розв'язання типових задач**

Встановіть початкову сторінку пошукового сервера meta-ukraine.com як домашню сторінку своєї програми-браузера MS Internet Explorer. Поверніть попередні налаштування домашньої сторінки браузера.

*Виконання:*

1. Запустіть програму Internet Explorer через пункти меню *Пуск* → *Програми* → *Internet Explorer* або через відповідну піктограму з Робочого стола.



2. Виконайте пункти меню *Вид* → *Властивості оглядача*, оберіть вкладку *Загальні вікна*, введіть до поля *Домашня сторінка* адресу meta-ukraine.com.
3. Закрийте вікно *Властивості оглядача* і саму програму Internet Explorer.
4. Запустіть програму ще раз і впевніться, що під час підключення до Інтернету одразу завантажується початкова сторінка пошукового вузла meta-ukraine.com.
5. Виконайте дії, подібні до дій пунктів 2–4 для встановлення домашньої сторінки e-commerce.com.ua.

Дизайн web-сторінок. Функції web-графіки. Розробка логотипів і баннерів. Розробка проекту web-сайта.

*Виконання:* Налаштувати систему NetMeeting для роботи; налаштування відео- й аудіоз'єднання; проведення розмови в chat-додатку. Опис дій з налаштування відеоконференції й chat-додатки; основних опцій програми; висновки роботи.

Перейдіть до пошукового сервера www.yandex.ru. Оберіть рубрику *Бизнес* і сформуєте в ній запит про електронний маркетинг.

*Виконання:*

1. Запустіть програму Internet Explorer. У вікні програми до поля *Адрес* уведіть назву пошукового сервера www.yandex.ru. Дочекайтеся завантаження початкової сторінки цього вузла.
2. У вікні початкової сторінки цього вузла натисніть на гіперпосиланні назви рубрики *Бизнес* і введіть у лапках до поля *Запрос* фразу «електронний маркетинг». Натисніть у вікні кнопку *Найти* або з клавіатури клавішу ENTER.
3. Через певний час на екрані з'являться результати пошуку, їх можна переглянути, натискаючи на їх посилання. Якщо ви бажаєте, щоб сторінка з результатами пошуку залишалася на екрані, нові сторінки за посиланнями, що вам сподобалися, можна відкривати у нових вікнах через пункт контекстного меню гіперпосилання *Открыть в новом окне*. Контекстне меню будь-якого об'єкта в операційних системах Windows відкривається клацанням правою кнопкою миші на об'єкті.

Перейдіть до пошукових серверів Rambler (www.rambler.ru), META (www.meta-ukraine.com) і виконайте пошук за різними видами сформованих запитів. З'ясуйте, як реагують пошукові сервери на ваші запити, наскільки вони результативні.

Послідовно зайдіть на сайти українських і російських інтернет-крамниць – Azbooka (www.azbooka.com.ua), Bambook (www.bambook.com.ua), Megashop (www.megashop.ru), Dostavka (www.dostavka.ru) і дайте відповіді на такі запитання:

1. Які групи товарів (послуг) можна придбати (отримати) у цих крамницях?
2. Які механізми використовуються для надання цінової інформації на запропоновані товари (послуги)?
3. Як потенційні покупці можуть ознайомитися з товарами (послугами)?

4. Які механізми замовлення товару через Інтернет працюють у цих інтернет-крамницях?

5. Чи існує в цих інтернет-крамницях система знижок на товари? Які вони порівняно з традиційними крамницями?

6. Які системи доставки товарів пропонуються покупцям?

7. Які регіони охоплює система доставки товарів цих інтернет-крамниць?

8. Чи зручний для користувача інтерфейс мають ці сайти? Зробіть їх порівняльний аналіз.

9. Чи є он-лайновий бізнес відображенням оф-лайнового бізнесу?

Зайдіть на сайти он-лайнових аукціонів (наприклад, [www.molotok.ru](http://www.molotok.ru), [www.galamarket.stavka.ru](http://www.galamarket.stavka.ru)) і дайте відповіді на запитання.

1. Чи є на цих аукціонах система рейтингових оцінок покупців і продавців?

2. Які механізми використовуються для надання інформації про лоти?

3. Якими шляхами потенційні покупці можуть ознайомитися з товарами?

4. Зробіть порівняльний аналіз зручності для користувача інтерфейсу, що мають ці сайти.

Послідовно зайдіть на сайти платіжних систем WebMoney Transfer ([www.webmoney.com.ua](http://www.webmoney.com.ua)), CyberPlat ([www.cyberplat.ru](http://www.cyberplat.ru)), PayCash ([www.paycash.kiev.ua](http://www.paycash.kiev.ua)). Проведіть їх порівняльний аналіз за переліком таких питань:

1. Які установи в он-лайн є організаторами створення цих платіжних систем?

2. До якого типу електронних платіжних систем їх можна віднести – дебетових чи кредитних?

3. Які схеми електронних платежів пропонуються користувачам на цих сайтах?

4. Як саме можна стати користувачем цих платіжних систем?

5. Чи надають ці платіжні системи можливість створення і підтримки інтернет-крамниць?

6. Як вирішуються питання захисту комерційної й особистої інформації при її передаванні мережею?

7. Чи можна використовувати у цих платіжних системах електронний цифровий підпис для аутентифікації користувачів?

8. Чи є регіональні обмеження на використання цих платіжних систем?

9. Чи мають сайти цих платіжних систем демонстраційний режим роботи, з допомогою якого пересічний користувач Інтернету може ознайомитися з їх роботою?

10. Чи зручний для користувача інтерфейс мають ці сайти? Зробіть їх порівняльний аналіз.

11. Як можна придбати програмне забезпечення для користування цими електронними платіжними системами і створення інтернет-крамниць, користування послугами інтернет-банкінгу?

12. Знайдіть за допомогою пошукових серверів за запитом «система електронних платежів» або подібним інші українсько- чи російськомовні платіжні системи і проведіть їх аналіз.

Вивчити основні компоненти та зробити налаштування інтерфейсу Flash MX, навчитися створювати примітиви, вирівнювання кількох примітивів.

1. Встановіть параметри сітки Flash MX і зобразіть її у робочій області.
2. Установіть одиниці виміру лінійки.
3. Зобразіть об'єкт Flash MX і залийте його за допомогою усіх чотирьох видів заливання, змінивши параметри заливання інструмента Paint Bucket.
4. Побудуйте новий градієнт Flash MX й імпортуйте його у файл.
5. Побудуйте градієнт заливання Flash MX і скопіюйте атрибути іншого заливання.
6. Імпортуйте растрове зображення у Flash і відредагуйте його.
7. Імпортуйте у Flash растрове зображення, оптимізуйте його, використайте його для заливання векторного зображення, побудованого у Flash.

Навчальне видання

# **МЕТОДИ ТА ЗАСОБИ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**методичні вказівки  
до проведення практичних занять та самостійної роботи**

Розробники:

**Горелов** Юрій Петрович  
**Кобзев** Ігор Володимирович  
**Лановий** Олексій Феліксович

Редактор *Н. С. Леонова*  
Комп'ютерне верстання *С. В. Гончарук*

Формат 60x84/16. Ум.друк.арк. 4,9. Тираж 50 пр. Зам. № 38.

Видавець і виготовлювач –  
Харківський національний університет внутрішніх справ,  
просп. 50-річчя СРСР, 27, м. Харків, 61080  
Свідоцтво суб'єкта видавничої справи ДК № 3087 від 22.01.2008.