

Лабораторна робота №1

Тема: «Основи скриптової мови PHP»

Мета: Ознайомлення з основами синтаксису PHP, одержання навичок у створюванні простих форм за допомогою PHP-редактора, дослідження операторів математичних операцій, введення даних у Web-форму

Теоретичні відомості

Скриптова мова (мова сценаріїв) програмування PHP, як правило, використовується суто для програмування додатків, пов'язаних з Інтернетом.

PHP-сценарії вбудовуються у код HTML-сторінки інтерпретуються і виконуються на сервері.

Розглянемо приклад: браузер звертається до WEB-сервера (наприклад, my_server) із запитом файла example.php з таким кодом (наприклад, http://my_server/example.php):

```
<html>
  <head>
<title> example</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    Приклад простої PHP-сторінки <br>
    Ці стрічки не інтерпретуються, як код PHP    <br> <br>
    <?php
      echo "Привіт, Я PHP-сценарій!";
    ?>
    <?
      echo("<p>Це теж PHP-сценарій!<p>");
    ?>
  </body>
</html>
```

Розширення файлу (.php) для сервера є ознакою, що цей файл – скрипт і його треба виконати як програму за допомогою інтерпретатора PHP-коду. Виконавши за допомогою інтерпретатора цей PHP-скрипт, сервер одержить результат і надішле нашому браузеру таку WEB-сторінку:

```
Приклад простої PHP-сторінки
Ці стрічки не інтерпретуються, як код PHP
```

```
Привіт, Я PHP-сценарій!
```

Це теж PHP-сценарій!

Розглянемо процес виконання php-сценарію при зверненні браузера до сервера. Отже, спочатку браузер запрошує у сервера сторінку з розширенням .php. Одержавши такий запит, WEB-сервер виконує цей файл через інтерпретатор PHP і далі генерує результат у вигляді html-коду (згенерованої WEB-сторінки), який і надсилає у відповідь WEB-браузеру.

Для того, щоб включити в файл команди PHP, необхідно використовувати спеціальні теги. В залежності від розширення файлу, їх розрізняють 4 види (вони еквівалентні і можна використовувати будь-які):

Інструкція обробки XML:

```
<?php  
...  
?>
```

Інструкція обробки SGML:

```
<?  
...  
?>
```

Інструкція обробки сценаріїв HTML:

```
<script language = "php">  
...  
</script>
```

Інструкція в стилі ASP:

```
<%  
...  
%>
```

Ми дотримуватимемося стилю XML або SGML.

Команда **echo** в PHP застосовується для виводу фактично всього, що зустрічається на web-сторінках (текст, розмітку HTML, числа). Значення її дії, зрозуміле з наведеного прикладу.

Крапка з комою

Необхідно пам'ятати, що в кінці кожної стрічки команди PHP-коду слід розмістити крапку з комою. Наприклад:

```
echo ("Тут крапка з комою ставиться після другої  
фізичної стрічки, вказуючи на завершення команди. \n");
```

Але існує декілька виключень. Крапка з комою не ставиться, якщо в кінці стрічки присутні:

- двокрапка (:),
- відкриваючі ({) та закриваючі (}) фігурні дужки,
- відкриваючі (<?) та закриваючі (?>) теги PHP.

Коментарі

Коментар – це фрагмент програмного коду, який не виконується і призначений лише для поточних пояснень і приміток. PHP надає декілька методів для вставки коментарів.

Перший спосіб пов'язаний з використанням подвійного слешу (//). Після цієї позначки PHP-інтерпретатор ігнорує все, що буде розташоване до кінця рядка.

Аналогічно можна користуватись і символом (#) - типовим способом коментування у скриптових мовах UNIX.

Для випадків, коли потрібно “закоментувати” одразу багато рядків вельми зручним стає в нагоді стиль мови програмування C (/*...*/).

```
<php
    echo("<p>Hello</p>"); // коментар
    echo("<p>Hello</p>"); # і це також коментар
    $a = "Hello, world";
    echo strstr($a, "H");
    // цю функцію ми розглянемо пізніше
    /*
        і це теж коментарі
    */
?>
```

Слід пам'ятати про те, що стилі коментарів PHP діють тільки усередині обмежувачів PHP. Якщо PHP зустрине ці символи коментарів поза обмежувачами, то вони, як і будь-який текст, з'являться в на html-сторінці. Наприклад:

```
<php
    echo("<p>Hello</p>"); // правильно написаний коментар
?>
// а ось цей коментар буде відтворено браузером.
```

До речі, існує ще один коментар – HTML. Він не має безпосереднього відношення до PHP, але може бути корисним в HTML-сторінці. Відмітимо, що за своєю природою він є багаторядковим:

```
<-- Це - коментар HTML. Його можна побачити лише в
джерельному коді HTML, але ж не в вікні WEB-браузера -->
```

Змінні

У PHP змінні починаються із знаку долара (\$). За цим знаком може слідувати будь-яка кількість буквено-цифрових символів і символів підкреслення. Слід також пам'ятати, що назви змінних в PHP чутливі до регістру.

При оголошенні змінних в PHP не вимагається явно вказувати тип змінної. При цьому одна і та ж змінна може впродовж програми набувати різних типів.

Змінна ініціалізується у момент привласнення їй значення і існує до тих пір, поки виконується програма (тобто скрипт). Це означає, що змінні існують до тих пір, поки не буде завершений запит.

```
<?php
$string="Математичний приклад";
$a=2;
$b=3;
$num= $a+$b;
echo "$string <br> Сума дорівнює $num";
?>
```

Оператори

Арифметичні оператори

+	Додавання
-	Віднімання
*	Множення
/	Ділення
%	Обчислення залишку по модулю (наприклад: 5 % 2=1)

Логічні оператори

or або	АБО
Xor	виключне АБО
and або I	І
!	НІ

+

Робота з формами

Як відомо, HTML-форми використовуються для введення інформації за допомогою WEB-браузера. В залежності від обставин, вони можуть мати різну кількість полів з різними їх типами. PHP чудово взаємодіє з цими об'єктами і дозволяє конструювати потужні і зручні користувацькі інтерфейси. Більш того, власне синтаксис PHP дозволяє легко зчитувати дані з полів форми.

Але спочатку зазначимо, що будь-яка форма повинна вміщувати такі дескриптори й атрибути:

```
<FORM NAME = «НАЗВА ФОРМИ»  
ACTION = «ШЛЯХ ДО ПРОГРАМИ, ЩО ОБРОБЛЯЄ ДАНІ»  
METHOD = «МЕТОД ПЕРЕДАЧІ ДАНИХ»>  
ПОЛЯ ДЛЯ ВВЕДЕННЯ ІНФОРМАЦІЇ ...  
</FORM>
```

В атрибуті ACTION будемо задавати назву .php-файла, який має обробляти дані з нашої форми.

Кожне поле введення інформації має атрибут NAME (наприклад, <INPUT TYPE="text" NAME="first" SIZE="4" MAXLENGTH="4">), значення якого передається до програми, що має обробляти дані.

Для передачі даних із полів форми до програми використовується два методи: GET і POST. Під час використання методу GET після натискання кнопки на формі "Надіслати" значення полів автоматично приєднується до URL, вказаного в атрибуті ACTION:

<http://site.domain/action.php?назв1=знач1&...назв2=знач2>

У випадку використання методу POST значення цих полів передаються в заголовку запиту до сервера, тобто непомітно для користувача.

Хід виконання:

Завдання №1. Створення WEB-калькулятора.

1. Запустити Krusader за допомогою ярлика на робочому столі.
2. Створити новий файл за допомогою піктограми на панелі інструментів (на малюнку обведено червоним колом, рис. 1),

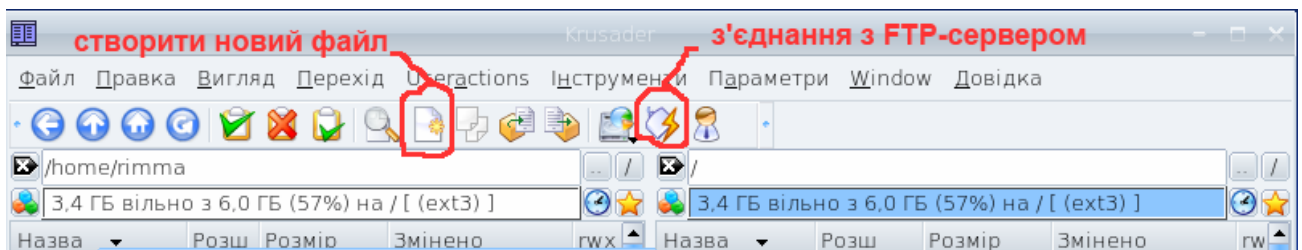
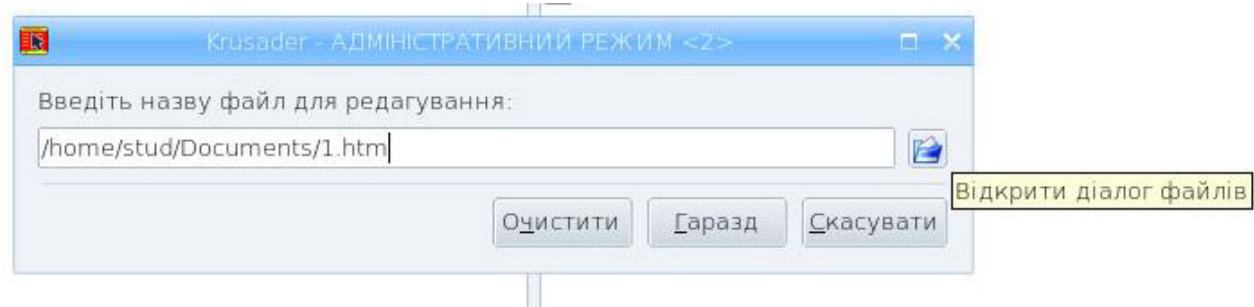


Рис.1

або за допомогою меню **Файл- New Text File**, або за допомогою клавіш **<SHIFT> + <F4>**.

Тепер треба прописати шлях до файла, та надати йому назву



3. Створити просту форму для проведення математичних дій над двома числами. Спочатку вставляємо саму форму і налаштовуємо її таким чином:

```
<FORM METHOD=GET ACTION="2.php">
</FORM>
```

Як було означено вище, ACTION = «ШЛЯХ ДО ПРОГРАМИ, ЩО ОБРОБЛЯЄ ДАНІ». Це означає, що всі PHP-скрипти виконуються WEB-сервером. Таким чином, будь-який скрипт треба розміщувати на WEB-сервері. В нашому випадку – це буде www.local.vsau. Як це зробити буде описано нижче.

Тепер вставляємо поля у форму, в які будуть вводиться наші дані, а також кнопку **type="submit"**, за допомогою якої будуть надсилатись дані до WEB-сервера і дали оброблятись скриптом [2.php](#)

```
<html>
<head>
  <title> Математичний приклад</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>

<body>
<form action="2.php" method="get">

Ввести значення змінних: <br>
a: <input name="a" type="text" value=""><br>

b: <input name="b" type="text" value=""> <br>

  <input type="submit" value="Розрахувати">

</form>
</body>
</html>
```

4. Робимо нашу форму більш привабливою за допомогою вже знайомих вам HTML-тегів.



Ввести значення змінних:

a:

b:

Рис. 2.

5. Зберігаємо файл під назвою **1.htm**.
6. А тепер створимо PHP-скрипт, який буде отримувати наші данні (змінні **a** і **b**) з форми і виконувати над ними прості математичні дії:

Додавання
Віднімання
Множення
Ділення

Для цього створюємо новий файл за допомогою кнопки на панелі інструментів або за допомогою меню **Файл- New Text File**.

Між PHP-тегами вставимо такий текст програми:

```
<?php
$string="<font color=#FF0000><b>Математичний
приклад</b></font>";

$sum= $a+$b;
$riz= $a-$b;
$dob= $a*$b;
$dil= $a/$b;
echo "$string<br><br> ";
echo "a=$a, b=$b<br><br> ";
echo "<b>Сума</b> $sum<br> ";
echo "<b>Різниця </b> $riz<br> ";
echo "<b>Добуток</b> $dob<br> ";
echo "<b>Частка</b> $dil<br> ";

?>
```

7. Збережіть цей скрипт під назвою **2.php**.

8. Тепер залишилось скопіювати ці два файли (1.htm та 2.php) у свій каталог на сервер www.local.vsau.

Для цього треба скористуватись ftp-з'єднанням (протокол передачі файлів). Файловий менеджер Krusader дозволяє це легко робити. Але при цьому треба мати таку інформацію:

- назву сервера або його IP-адресу (в нашому випадку – це www.local.vsau)
- назву користувача ftp (в нашому випадку це будуть ek1, ek2 і т.д. Надасть викладач)
- пароль користувача (Надасть викладач).

Для цього за допомогою ярлика на робочому столі запустіть програму Krusader, оберіть піктограму на панелі інструментів **з'єднання з FTP-сервером** (рис. 1), або виберіть пункт меню **Інструменти- New Net Connection**, або комбінацію клавіш **<Ctrl>+<N>** та створіть нове з'єднання.

У вікні, що з'явилось введіть назву ftp-сервера www.local.vsau, ім'я користувача та пароль, які вам видасть викладач. Натиснути кнопку з'єднати (рис. 3)

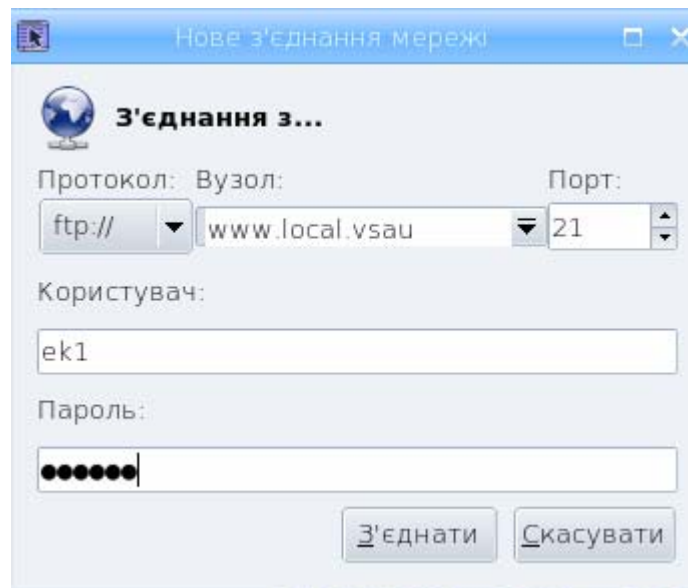


Рис. 3. ftp-клієнт файлового менеджера Krusader

На одній із файлових панелей відкриється ваш каталог на сервері. Після цього скопіюйте: свої файли 1.htm та 2.php на сервер www.local.vsau. Далі програму Krusader не закривайте. Вона може ще знадобитись.

9. Відкрийте браузер за допомогою ярлика на робочому столі. В адресну стрічку наберіть URL до вашого 1.htm файлу. Це має бути щось на кшталт:

www.local.vsau/students/labor/sybd/ek1/1.htm

Зрозуміло, що замість екі ви маєте ввести свою назву ftp-користувача.

10. Якщо ви все правильно зробили, то в браузері відкриється ваша форма. В ній введіть значення змінних **a** і **b** та натисніть кнопку “Розрахувати”. Замість форми має з'явитись WEB-сторінка із результатами розрахунків (рис. 4). Зверніть увагу на URL в браузері. Це вже робота скрипту 2.php

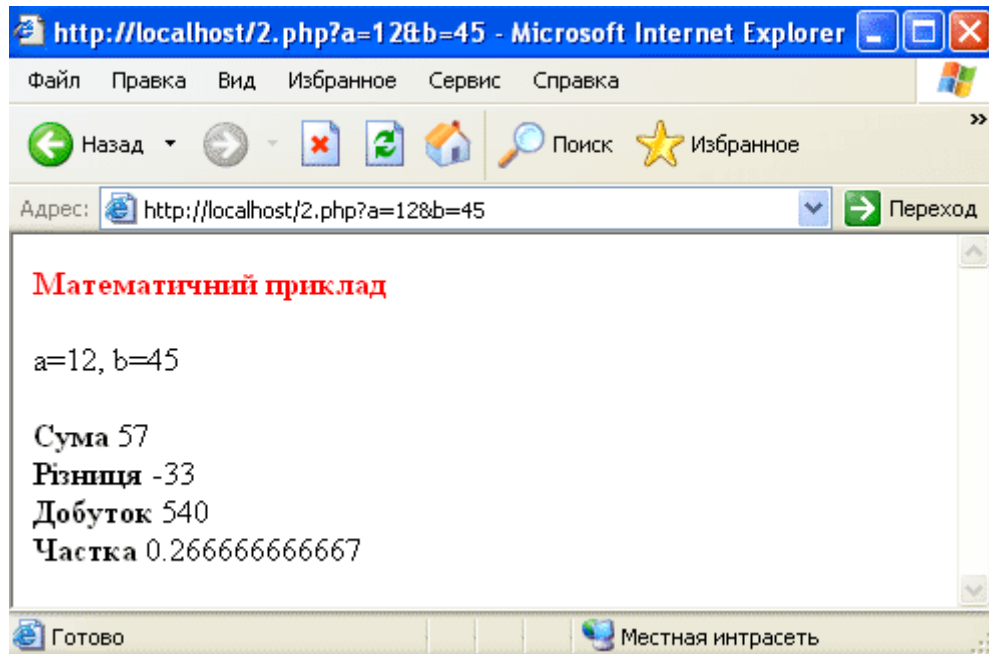


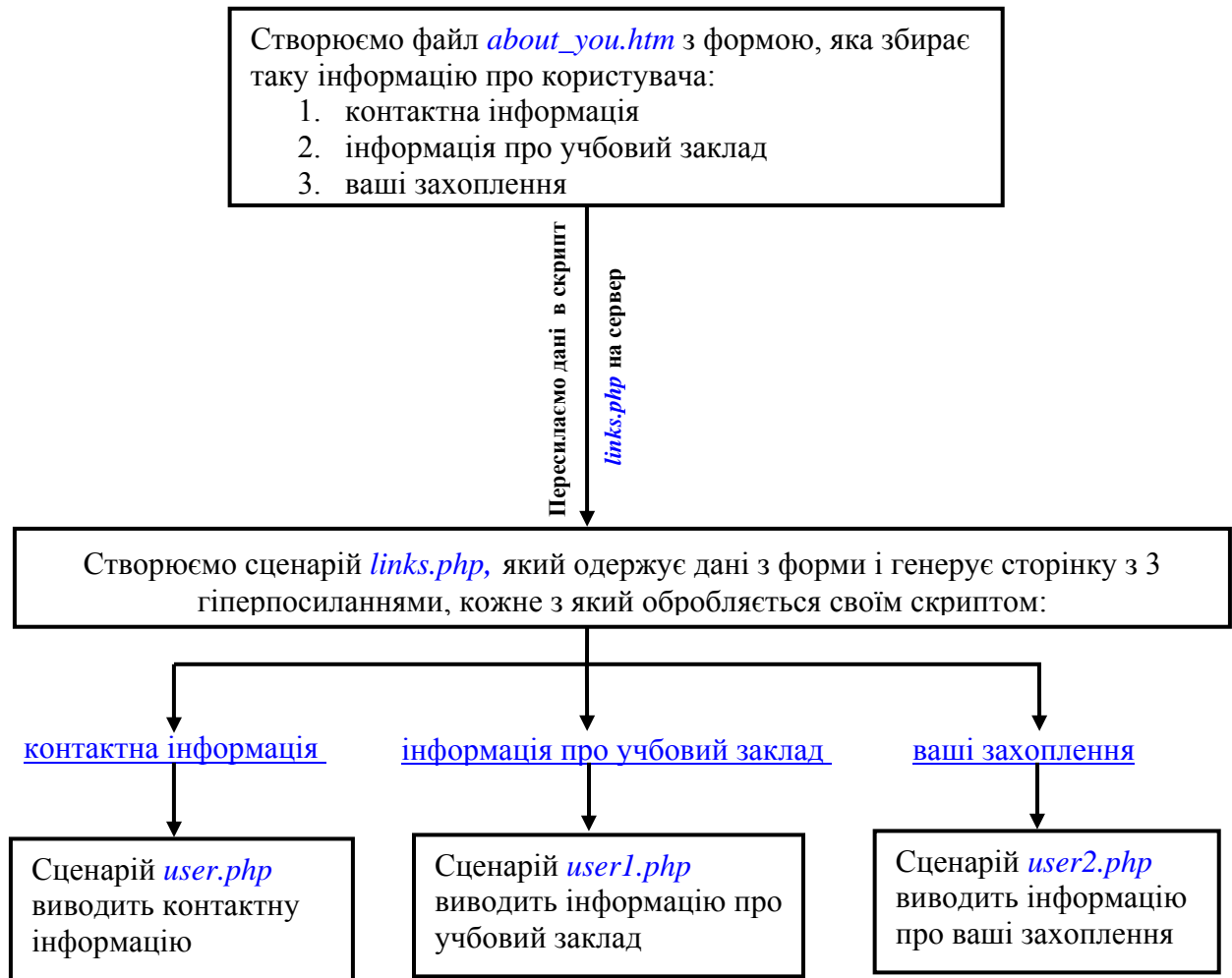
Рис. 4. Результат роботи скрипту 2.php

Завдання № 2. Створення WEB-форми для збирання та надсилення даних про користувача.

Припустимо, що нам потрібно за допомогою Internet проводити опитування користувачів. А результати – надсилати за певною адресою. Використовуючи набуті знання від попереднього завдання, можна уявити, що і в цьому випадку треба також мати якусь WEB-форму, яку можна було би заповнювати з браузера, і яка могла би передавати свої дані скрипту-обробнику інформації.

В найпростішому випадку він міг би просто генерувати WEB-сторінку із одержаними даними як у попередньому випадку. В інших випадках – він міг би надсилати одержані дані по електронній пошті, або заносити їх в базу даних.

Поки що наш план – це генерація [WEB-сторінки із стрічками-гіперпосиланнями](#) на інший скрипт. Зміст кожного гіперпосилання (URL) буде визначатись інформацією, яка буде занесена в [першу WEB-сторінку](#) з браузера.



1. Створимо форму, за допомогою якої отримуємо анкетну інформацію про користувача:
 - як можна з ним зв'язатись
 - захоплення
 - учбовий заклад в якому навчається (рис. 5).

Важливою умовою для цього сценарію є правильне застосування назв текстових полів (пам'ятаємо, що кожне із них – це об'єкт). Ці назви будуть використовуватись у якості назв змінних, в яких і буде зберігатись отримана інформація з форми.

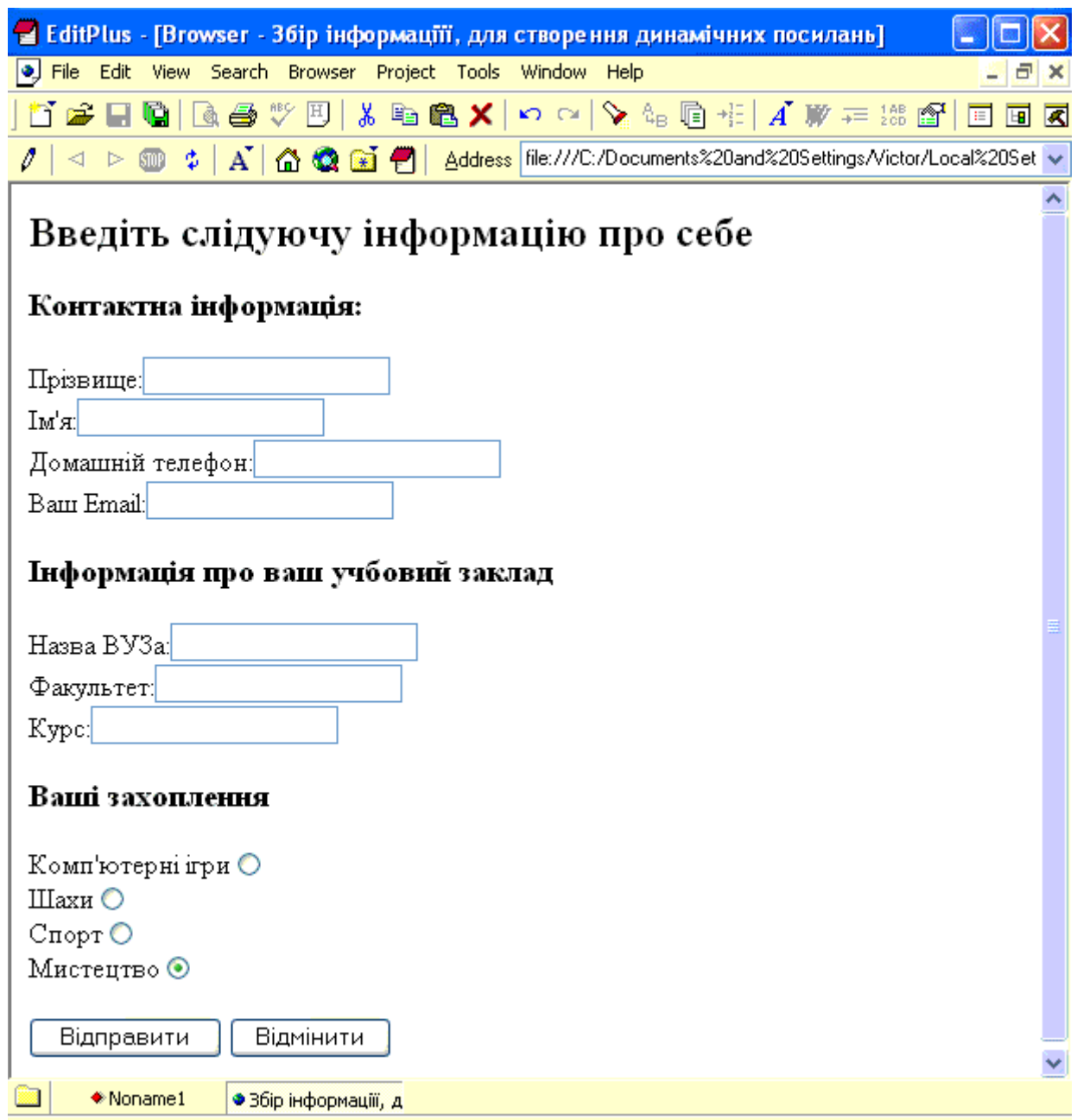


Рис. 5 Результат роботи сценарію about_you.htm

А це зміст HTML-файлу about_you.htm, який містить у собі цю форму

```
<html>
<head>
  <title>Збір інформації, для створення динамічних
посилань</title>
</head>
```

```

<body>
  <h2>Введіть слідуочу інформацію про себе</h2>

<form action="links.php" method="GET">

  <h3> Контактна інформація:</h3>
  Прізвище:<input name="first_name" type="text">
  <br>Ім'я:<input name="last_name" type="text">
  <br>Домашній телефон:<input name="home_phone" type="text">
  <br>Ваш Email:<input name="email" type="text">

  <h3>Інформація про ваш учбовий заклад </h3>
  Назва ВУЗа<input name="vuz_name" type="text">
  <br>Факультет:<input name="fac_name" type="text">
  <br>Курс:<input name="kurs" type="text">

  <h3>Ваші захоплення </h3>
  Комп'ютерні ігри<input name="hobby" type="radio"
value=" Комп'ютерні ігри" checked>
  <br>Шахи<input name="hobby" type="radio" value="Шахи"
checked>
  <br>Спорт<input name="hobby" type="radio" value="Спорт"
checked>
  <br>Мистецтво<input name="hobby" type="radio"
value=" Мистецтво" checked>

  <p><input type="submit" name="Надіслати" value="Надіслати">
  <input type="reset" name="Відмінити" value="Відмінити"></p>

</form>

</body>

</html>

```

2. Створіть сценарій links.php, який буде одержувати дані з форми і створювати гіперпосилання, які далі можна застосувати для надсилання конкретних даних до іншого сценарію. Таких гіперпосилань у нас буде три. Як наведено на рис. 6, перше з них призначене для отримання контактної інформації, друге – про учбовий заклад в якому навчається, а третє про захоплення користувача.

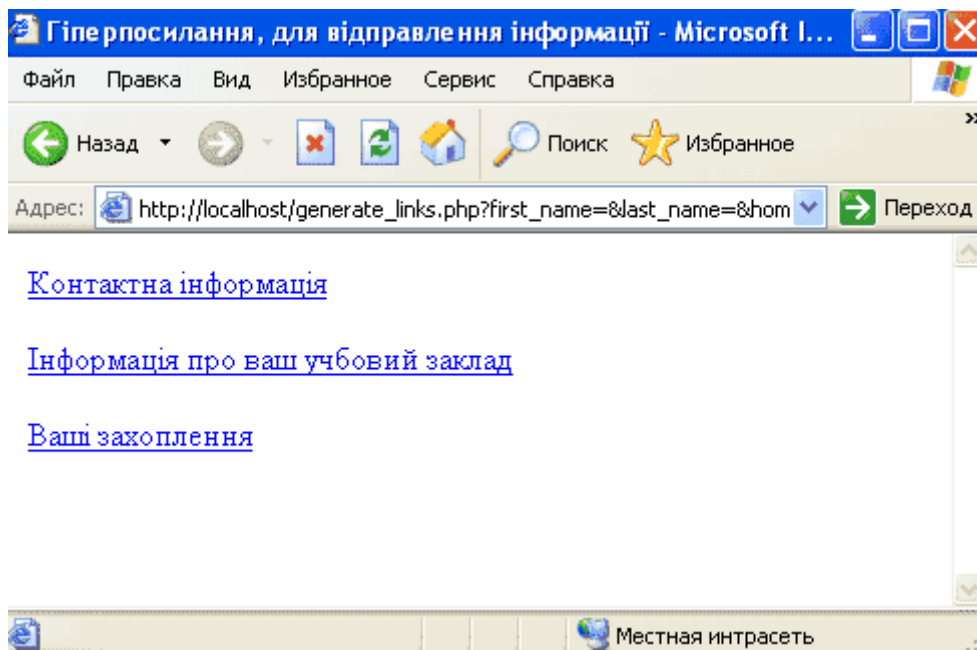


Рис. 6. Результат роботи сценарію `links.php`

А ось зміст сценарію `links.php`

Зверніть увагу, що у стрічках 7-8 створюється гіперпосилання, яке передає контактну інформацію у сценарій `user.php`. Формуючи URL для сценарію `user.php`, потрібно додати пари «назва/значення» після назви сценарію для передачі йому даних.

Всі елементи, які знаходяться за знаком питання (?) в цьому URL, складаються із назви змінної та її значення, яке й передається з WEB-форми до сервера. Кожна із пар «назва/значення» відокремлена від іншої символом амперсанд (&).

У стрічках 9-10 аналогічно створюємо гіперпосилання, яке передає інформацію про учбовий заклад у сценарій `user1.php`; і у стрічках 11-12 створюємо посилання про ваші захоплення яке передає інформацію у сценарій `user2.php`.

```

1.     <html>
2.     <head>
3.     <title>Гіперпосилання для надсилання інформації</title>
4.     </head>
5.     <body>
6.
7.     <?php
      echo "<p><a
      href=\"user.php?first_name=$first_name&last_name=$last_name
      &
      home_phone=$home_phone&email=$email\">";

```

8. `echo "Контактна інформація</p>";`
9. `echo "<p><a
href=\"user1.php?vuz_name=$vuz_name&fac_name=$fac_name&kurs
=$kurs\">";`
10. `echo "Інформація про ваш учбовий заклад</p>";`
11. `echo "<p>";`
12. `echo "Ваші захоплення</p>";`
13. `?>`
14. `</body>`
15. `</html>`

3. Створюємо сценарій `user.php`, який має виводити дані про користувача, якщо у сценарії `links.php` було обрано гіперпосилання “Контактна інформація” (Рис. 7).

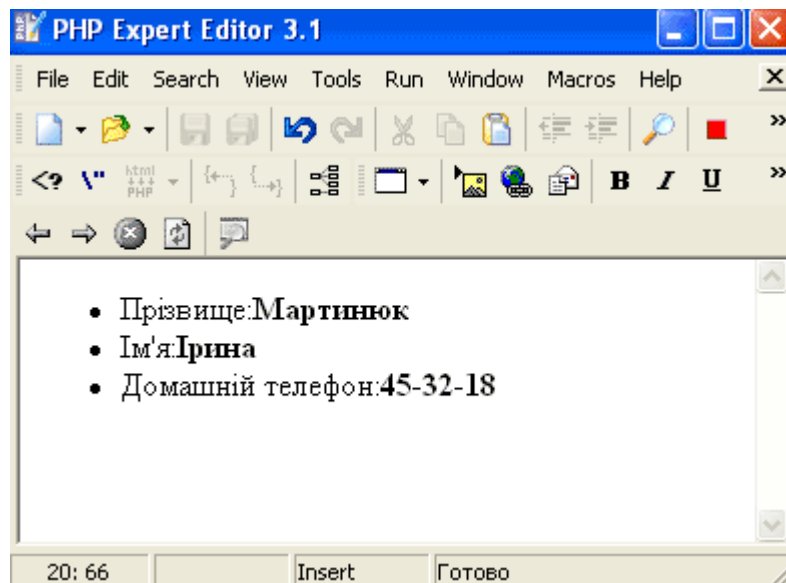


Рис. 7. Результат роботи скрипта `user.php`

А це сам скрипт:

```
<html>
<head>
  <title>Інформація, що надіслана за допомогою
гіперпосилань</title>
</head>

<body>
```

```
<ul>
<li> Прізвище:<b><?php echo $first_name ?></b></li>
<li> Ім'я:<b><?php echo $last_name ?></b></li>
<li> Домашній телефон:<b><?php echo $home_phone ?></b></li>
</ul>

</body>
</html>
```

4. Самостійно створіть сценарій `user1.php` та `user2.php`

Контрольні запитання

1. Де виконуються PHP-сценарії?
2. Яким чином PHP-сценарій вбудовується в HTML-сторінку?
3. Яка команда використовується для виводу інформації на екран?
4. Що ставиться в кінці кожної стрічки команди PHP-коду?
5. Коли не ставиться крапка з комою?
6. Як можна вставити коментарі на сторінку?
7. Як позначаються змінні у PHP-сценарії?
8. Для чого існує атрибут ACTION у формі?
9. Які є методи передачі даних з форми і як вони працюють?
10. Куди передаються дані з форми і як вони обробляються?
11. Яким чином додається назва і значення змінної до URL?
12. Яким чином формується гіперпосилання у скрипті `links.php`?
13. Які додаткові скрипти використовуються у сценарії `links.php` і для чого?
14. Які елементи знаходяться за знаком (?) в URL, коли дані передаються з форми до сервера?
15. Що означає знак & (амперсанд) в URL, коли дані передаються з форми до сервера?
16. Для чого потрібен скрипт `user.php`? Яким чином передається інформація у цей сценарій?
17. Для чого потрібен скрипт `user1.php`? Яким чином передається інформація у цей сценарій?
18. Для чого потрібен скрипт `user2.php`? Яким чином передається інформація у цей сценарій?