

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**Методичні рекомендації
до виконання лабораторних робіт
з навчальної дисципліни**

"СКРИПТОВІ МОВИ"

**для студентів напряму підготовки
6.051501 "Видавничо-поліграфічна справа"
всіх форм навчання**

Харків. Вид. ХНЕУ ім. С. Кузнеця, 2014

Затверджено на засіданні кафедри комп'ютерних систем і технологій.
Протокол № 11 від 08.04.2014 р.

Укладач Пандорін О. К.

М54 Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни "Скриптові мови" для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання / укл. О. К. Пандорін. – Х. : Вид. ХНЕУ ім. С. Кузнеця, 2014. – 28 с. (Укр. мов.)

Подано методичні рекомендації до виконання лабораторних робіт за модулями й темами для закріплення студентами необхідних теоретичних основ, надання практичних навичок щодо використання скриптових мов, об'єктно-орієнтованого та компонентного підходу, використання та створення компонентів у сучасних середовищах клієнтської та серверної частин сучасних інтернет-додатків.

Рекомендовано для студентів напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання.

Вступ

Навчальна дисципліна "Скриптові мови" вивчається студентами напряму підготовки 6.051501 "Видавничо-поліграфічна справа" всіх форм навчання протягом п'ятого семестру і є методологічною і методичною основою для опанування студентами предметної області створення мультимедійних продуктів. Навчальна дисципліна належить до групи вибіркових дисциплін і забезпечує підготовку студентів до майбутньої професійної діяльності в умовах сучасного розвитку засобів мультимедіа. Подальший розвиток ідеї дисципліни набувають у дисциплінах "Тримірне моделювання", "Принципи проектування баз даних та баз знань", "Розробка Web-додатків", "Програмування засобів мультимедіа", "Комп'ютерні редакційно-видавничі системи", "Технології підготовки та виробництва мультимедійних видань", "Мультимедійні технології" тощо. Навчальна дисципліна забезпечує підготовку студентів до майбутньої професійної діяльності в умовах розвитку сучасних засобів мультимедіа. Вона готує студентів до вивчення взаємопов'язаних професійно-орієнтованих і спеціальних питань щодо розробки мультимедійних продуктів, а також надає можливість використання отриманих знань при підготовці курсових, кваліфікаційних робіт, дипломних проектів, при розв'язанні практичних задач. Для індивідуалізації навчання студентам видаються диференційовані індивідуальні завдання до лабораторних занять, а також завдання для самостійної роботи. Навчальна дисципліна "Скриптові мови" є забезпечуючою для багатьох дисциплін рівнів "бакалавр", "магістр", а також для курсових і дипломних проектів, адже процес програмування є невід'ємною частиною розробки будь-якого сучасного мультимедійного продукту.

Особливість даного лабораторного курсу програми – її орієнтація на різноманітне програмні технології, вивчення яких допомагає сформуванню різнобічно підготовленого фахівця. У даній дисципліні розглядаються на базі реалізації функціонально однакових інтернет-додатків за варіантом теми індивідуального завдання засобами використання різних програмних середовищ, які подані різноманітними інформаційними технологіями.

У результаті освоєння курсу студенти повинні здобути навички проектування, розробки, впровадження та експлуатації програмних систем, що базуються на компонентному та об'єктно-орієнтованому підході.

Тексти програм визначаються за допомогою літер однакової ширини. Перехід, до іншої строки у фрагментах коду, там, де він вимовлений лише можливостями цього видання, позначається символом "↵". При використанні середовища Visual Studio значна частина коду має будуватися автоматично.

Змістовий модуль 1. Базові концепції використання скриптових мов та їх реалізація у найпоширеніших документних моделях на клієнтській частині

Лабораторна робота № 1. Використання інструментальних засобів для створення та відлагоджування серверної та клієнтської частин додатку (IDE MS VS, IDE NetBeans, FIREBUG, Chrome Developers Tools для навігації простого циклічного додатку)

Мета роботи. Ознайомити студентів з основними технологічними прийомами створення програм на JavaScript. Програми хоч скільки-небудь помітного розміру, незалежно від того, якою мовою вони написані, одразу ж після створення вкрай рідко, виконують всі потрібні від них функції. Особливо гостро ця проблема вирішується в програмах, від яких вимагається функціонування в різних апаратно-програмних середовищах. Прикладом таких програм є броузерные програми, наприклад, частини динамічних html-сторінок. Для них потрібно функціонування під управлінням різних браузерів і різних версій однойменних браузерів (можуть відрізнятися як номер версії браузера, так і специфіка мобільної платформи, склад встановлених браузерних розширень...).

Робота складається з таких частин:

- Налаштування в середовищі MS Visual Studio / Internet Explorer. Містить найбільш комфортні засоби розробника.
- Налаштування в середовищах поширених браузерів – mozilla firefox, google chrome, Opera (поза MS Visual Studio). Іноді програми в різних браузерних середовищах виконуються по-різному й емулятор браузерів MS Visual Studio не в змозі відобразити ці відмінності.

Налаштування в середовищі MS Visual Studio / Internet Explorer Інсталюйте MS Visual Studio.

Придатна будь-яка версія, яку випущено після 2007 року з підтримкою веб-розробки, наприклад, мінімальна безкоштовна MS Visual Studio Express for Web.

Встановити Internet Explorer 9 або 10 (попередні версії не підтримували малювання – тег Canvas HTML 5).

Запустити програму MS Visual Studio.

Необхідно вибрати шлях до місця розташування сайту, який не містить російських букв та короткий – деякі браузери (наприклад, Опера) іншого не розуміють (втім, як і сервери, наприклад Апач).

Створити папку з файлами сайту, з ім'ям вашого проекту, наприклад, snake, у корені диска D.

Створити HTML сторінку.

Замінити її код текстом вашої сторінки. Список варіантів індивідуального завдання за номером у журналі наведено в додатку в кінці глави. Там же наведено і критерії оцінки завдання залежно від функціональності вашої сторінки.

Обрати у ролі браузера Інтернет Експлорер.

Дозволити налагодження в Інтернет Експлорері (на деяких ВЦ адміністратори не дозволяють змінювати параметри налагодження, якщо на вашій машині – перейдіть до стадії налагодження під браузерами).

Незважаючи на те, що налагодження дозволяється для всіх браузерів, ні для жодного, крім ІЕ, Visual Studio не вміє організовувати налагодження.

Встановити точку переривання клацанням миші по панелі переривань.

Запустити сайт з налагодженням.

Дозволити налагодження в файлі конфігурації.

Програма зупинилася в точці, що було позначено.

Тепер можна за операторами переміщатися по програмі і переглядати значення локальних змінних, зокрема властивості завантаженого документа this.

Натиснути на клавішу стрілки в Експлорері.

За значенням key=39 зрозуміло, що була натиснута клавіша зі стрілкою вправо.

Таким чином можливо аналізувати, в якому порядку відбувається виконання операторів програми залежно від значень змінних.

Налагодження за допомогою Засобів розробки Експлорера.

Відкрити сайт в адресному рядку Експлорера.

Адреса d:\sites\snake\HtmlPage.html (зрозуміло, ви повинні набрати адресу сторінки на вашому комп'ютері).

Відкрити засоби розробника – F12.

Бачимо набір налагоджувальних інструментів, який дуже схожий на той, що реалізовано у середовищі Visual Studio.

Відлагодження в середовищах поширених браузерів – mozilla firefox, google chrome, Opera.

Прочитати інформацію про налагоджувальних засобах поширених браузерів у теках /Firefox /Chrome /Mozilla.

Продемонструвати їх використання на прикладі програми з вашого варіанту індивідуального завдання.

Вимоги до звіту.

Звіт повинен містити, крім інформації про виконавця, набір знімків екрану, забезпечених підписами про використання налагоджувальних засобів у VS і кожному з браузерів.

Ви повинні мотивувати доцільність використання кожного з налагоджувальних засобів у кожному конкретному прикладі.

При здачі роботи студенти повинні бути готові продемонструвати використання кожного з інструментів у кожному середовищі і пояснити способи використання.

Варіанти індивідуального завдання – за номером у журналі.

Створити текст сторінки, на якій при натисканні на кнопку на поверхні форми виконувалися б обчислення наступної функції від аргументу вектора $(0, 3, -7, 0, 4, -6, 10, 3)$ і результат відображався б на веб-сторінці.

Список функцій:

1. Максимальний елемент вектора.
2. Мінімальний елемент вектора.
3. Сума позитивних елементів вектора.
4. Сума від'ємних елементів вектора.
5. Кількість позитивних елементів вектора.
6. Кількість від'ємних елементів вектора.
7. Середня позитивних елементів вектора.
8. Середня від'ємних елементів вектора.
9. Перший позитивний елемент вектора.
10. Перший негативний елемент вектора.
11. Останній позитивний елемент вектора.
12. Останній негативний елемент вектора.
13. Номер останнього позитивного елемента вектора.
14. Номер останнього від'ємного елемента вектора.

Додаткова функціональність.

Щоб розраховувати на відмінну оцінку необхідно надати користувачеві можливість змінювати вихідний вектор зручним способом.

Наприклад, вводити компоненти вектора по черзі, або в редагованому поле через кому, або вибором зі списку [1 – 5].

Лабораторна робота № 2. Забезпечення контролю цілісності та юзабіліті засобами клієнтської частини (проходом елементами DOM та перевірок коректності введення значень)

Мета роботи. Ознайомити з різними способами перевірки правильності введення в середовищі браузера з метою забезпечення логічної цілісності об'єктів.

На клієнтській стороні для побудови додатків можуть використовуватися самі різні скриптові мови. Але в середовищі браузера гарантовано є тільки один – JavaScript. Забезпечення правильності введення надзвичайно важливо з точки зору безпеки додатків. У роботі ви створите форму введення даних для певного бізнес-об'єкта на поверхні HTML сторінки (згадайте другу лабораторну за ТООП) і забезпечите правильність введення як засобами HTML5, так і засобами JavaScript з використанням операцій з рядками і списками.

Вимоги до форми.

Форма повинна містити всі введені HTML5 типи об'єктів input:

- color;
- date;
- datetime;
- datetime-local;
- email;
- month;
- number с атрибутами max, min, step, value;
- range с атрибутами max, min, step, value;
- search;
- tel;
- time;
- url;
- week.

Всі типи HTML4-Button, Checkbox, FileUpload, Hidden, Password, Radio, Input, Reset, Submit, Text. Має бути продемонстровано використання атрибутів pattern, placeholder, required, title.

Виконання роботи.

Згадати поведінку бізнес-об'єкта, для якого ви створювали компонент у курсі ТООП, а тепер створити форму.

Описати бізнес-логіку в тексті звіту.

Сформулювати обмеження цілісності у вигляді передумов, марнослів й інваріантів, повідомлення про їх порушення.

Для кожного з обмежень вибрати спосіб реалізації (типи і атрибути HTML, рядки, регулярні вирази, скрипти), описати і мотивувати у звіті.

Вмістити в звіт результати випробувань форми.

Рекомендації по виконанню.

Варто переглянути рекомендації щодо валідації введення в папках /HTML5 та /RegExp.

Веб-сторінку зручніше створювати на тому примірнику редактора, який підтримує підсвічування синтаксису HTML5, список помилок у коді, верстки і налагодження JavaScript (на деяких Visual Studio не встановлені оновлення).

Додавайте ділянки коду валідації в програму невеликими частинами, виправляйте синтаксичні помилки і тестуйте після кожної модифікації [1 – 5].

Лабораторна робота № 3. Графічні засоби відображення стану форми

Мета роботи. Ознайомити із засобами побудови багатого користувача інтерфейсу на клієнтській стороні

Ця робота присвячена таким засобам побудови динамічного інтерфейсу користувача:

1. Використання анімації фреймворка JQUERY і JQUERY UI.
2. Використання CSS transition і animation.
3. Використання SVG SMIL анімації.

Таким чином, необхідно побудувати 3 сторінки.

Функціонально вони збігаються зі сторінками з попередньої лабораторної роботи. Збагачення робіт анімацією проводиться трьома способами.

Необхідно використовувати їх усі, порівняти і мотивувати свій вибір оптимальний для ваших цілей.

Далеко не всі властивості CSS3 реалізовані у всіх браузерах.

Зазвичай, якщо деяка можливість CSS3 не працює, необхідно змінити браузер якщо у використовуваному цю властивість не реалізовано.

Розробку проведемо як розвиток сторінки з формою з лабораторної роботи 2.

У максимально спрощеному вигляді форма має такий код:

```
<!DOCTYPE html>
<html>
<head>
  <title>Anim</title>
  <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/↵
jquery.min.js"></script>
</head>
<body>
  <form action="HTMLPage.htm">
    <input id="in" autofocus="True" type=↵
"number" required="true" max="3" min="1" value="1" />
    <input id="Submit" type="submit" value=↵
"submit" /></form>
</body>
</html>
```

Використання анімації фреймворку JQUERY і JQUERY UI.

Додати анімацію, як реакцію на невірне заповнення числового поля.

Відповідна подія `oninvalidate` збуджується при невірному введенні користувачем значень до елемента:

```
<input id="in" autofocus="True" type=↵
"number" required="true" max="3" min="1" value="1" />
```

після натискання на кнопку

```
<input id="Submit" type="submit" value="submit" /></form>
```

Подія виникає в тому випадку, якщо хоч в одному полі введення невірні.

Проаналізувати правильність введення в окремому полі можна за допомогою функції `isValid ()`.

Кожен браузер повідомляє користувачеві про невірно введені значення полів форми.

Оживити поведінку можна за допомогою анімації чисельних значень.

Наприклад, завантажити текст до коду сторінки.

```
<!DOCTYPE html>
<html>
<head>
```

```

<title>Anim</title>
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/↵
1.5/jquery.min.js"></script>
<script type="text/javascript">
    $(document).ready(function () {
        var input = ↵
        document.querySelector("#in");
        input.oninvalid = function () {
            $("#in").animate({ fontSize: ↵ ↵ "1.3em" },
1000);
            $("#in").animate({ marginLeft:↵ ↵ "30px" },
1000);
            $("#in").animate({ marginTop:↵
"50px" }, 1000);
            $("#in").animate({ fontSize: ↵
"1em" }, 1000);
            $("#in").animate({ marginLeft: ↵ "0px" },
1000);
            $("#in").animate({ marginTop: ↵ "0px" },
1000);
                //          alert("88")
        };
    });
</script>
</head>
<body>
<form action=HTMLPage.htm>
<input id="in" autofocus="True" type=↵
"number" required="true" max="3" min="1"
value="1" />
<input id="Submit" type="submit" value=↵
"submit" /></form>
</body>
</html>

```

Запустити її.

При неправильному введенні після натискання на кнопку включається анімація значень.

Завантажити наступний текст до коду сторінки.

```
<!DOCTYPE html>
<html>
<head>
<title>Anim</title>
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/↵
jquery/1.5/jquery.min.js"></script>
<script type="text/javascript">
    $(document).ready(function () {
        var input = document.querySelector("#in");
        input.oninvalid = function () {
            $("#in").animate({↵
fontSize: "1.3em" }, 1000);
            $("#in").animate({↵
marginLeft: "30px" }, 1000);
            $("#in").animate({↵
marginTop: "50px" }, 1000);
            $("#in").animate({↵
fontSize: "1em" }, 1000);
            $("#in").animate({↵
marginLeft: "0px" }, 1000);
            $("#in").animate({↵
marginTop: "0px" }, 1000);
            //          alert("88")
        };
    });
</script>
</head>
<body>
<form action=HTMLPage.htm>
<input id="in" autofocus="True" type=↵
"number" required="true" max="3" min="1"
```

```
        value="1" />
<input id="Submit" type="submit" value="Submit" /></form>
</body></html>
```

При неправильному введенні і натисканні кнопки Submit запускається анімація відображення.

Переглянути зміст папки / Анімація JQUERY /.

Вибрати для кожного поля вашої форми найбільш підходящий варіант поєднання анімації. Мотивацію вибору для кожного з елементів детально викладіть у звіті.

Використання CSS transition і animation.

Засоби валідації та анімації JQUERY з'явилися не на порожньому місці – їх було активно затребувано. На потреби розробників прореагував і W3C. Значна частина можливостей була включена в стандарти HTML5 – (псевдокласи валідації) та CSS3 (властивості transition і animation).

Показати, як реалізувати анімацію керуючого елемента, для якого проведена валідація з їх допомогою.

Розглянемо простий приклад.

```
<!DOCTYPE html>
<html>
<head>
    <title>Anim</title>
</head>
<body>
    <form action="HTMLPage.htm">
        <input id="in" autofocus="True" type="text"
"number" required="true" max="3" min="1" value="1" />
        <input id="Submit" type="submit" value="Submit" /></form>
</body>
</html>
```

Додати стилів – присвоїти "червоний фон" для невірно введених значень і "зелений фон" для вірно введених.

Відкрити сторінку.

Після чого додати анімацію.

```
<!DOCTYPE html>
<html>
<head>
  <title>Anim</title>
  <style type="text/css">
    input:required:invalid {
      background-color: #f00;
      transition: background 5s ease, color 3s ease;
      transition-property: background;
    }
    input:required:valid {
      background-color: #0f0;
    }
  }
  body {
    animation: spin 5s linear infinite;
  }
</style>
</head>
<body>
  <form action="HTMLPage.htm">
    <input id="in" autofocus="True" type="number"
    "number" required="true" max="3" min="1" value="1" />
    <input id="Submit" type="submit" value="Submit" />
  </form>
</body>
</html>
```

Переглянути зміст теки / CSS transition animation /.

Оберати для кожного поля вашої форми найбільш відповідний варіант під'єднання анімації. Мотивацію вибору для кожного з полів детально викласти у звіті.

Використання SVG SMIL анімації.

Найбільш економним з точки зору апаратних ресурсів є SVG SMIL анімація.

HTML5 дозволяє безпосередньо додати опис векторного малюнка і анімації в текст HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Attribute Animation with SMIL</title>
  </head>
  <body>
    <svg width="300px" height="100px">
      <rect x="0" y="0" width="300" height="
100" stroke="black" stroke-width="1" />
      <circle cx="0" cy="50" r="15" fill="
blue" stroke="black" stroke-width="1">
        <animate attributeName="cx" from="0"
to="100" dur="5s" repeatCount="indefinite" />
      </circle>
    </svg>
  </body>
</html>
```

У браузері відобразиться анімована окружність.

Створити малюнок у форматі SVG на тему вашого індивідуального завдання і вставити його код на поверхню форми. Найбільш близьким до звичних вам інструментам побудови анімації є Adobe Edge. Втім, в його використанні немає абсолютної необхідності. Векторну картинку у форматі SVG можна зберегти практично в будь-якому векторному редакторі, а анімувати її можна безпосередньо в коді HTML.

Основною складовою оцінки даного розділу є креативність анімації [1 – 5].

Лабораторна робота № 4. Створення додатку для мобільної платформи

Мета роботи. Ознайомити із засобами побудови багатого користувача інтерфейсу на клієнтській стороні мобільних пристроїв.

Справжня робота присвячена засобам побудови динамічного користувача інтерфейсу за допомогою мови JavaScript з використанням кроссплатформенної бібліотеки PhoneGap .

PhoneGap дозволяє писати на JavaScript додаток для мобільних пристроїв, iOS пристрій, Android, Symbian, BlackBerry OS, Palm OS або Windows Mobile. Додаток на PhoneGap може використовувати різні функції, характерні скоріше для мобільних пристроїв (телефонів, планшетів, нетбуків): дзвонити, отримувати знімки з камери, координати GPS.

Проілюструвати на прикладі пристроїв на платформі Android.

Установка інструментів.

Необхідно встановити один з інструментів для розробки.

Найзручніше відбувається установка Eclipse adt-bundle та Android Studio.

Створення проекту.

У папці з проектом створити дві нові папки: / libs та / assets / www.

Викачувати PhoneGap. Створити другий activity (так називається додаток Android) у файлі AndroidManifest.xml

```
<activity android:name="com.phonegap.DroidGap" ↵
  android:label="@string/app_name" ↵
  android:configChanges="orientation|keyboardHidden">
  <intent-filter>
  </ intent - filter >
  </ activity >
```

У папці "Android" знайти архів phonegap. Копіювати: phonegap.js в папку / assets / www , phonegap.jar у папку / libs.

Додати шлях доступу до libs PhoneGap. Відкрити файл Java і внести зміни: підключити phonegap і завантажити нашу головну сторінку.

У файлі AndroidManifest.xml редагувати рядки з uses permission, що включають взаємодію з апаратурою пристрою. Всі uses permission у даному файлі будуть показані в android market, якщо ви будете завантажувати туди свій додаток.

Додати головну сторінку index.html.

```
<html>
<head>
<title> PhoneGap </ title >
<script type="text/javascript" charset=↵
"utf-8" src="phonegap.js"> </ script >
<head>
<body>
```

```
<h1> Hello World </ h1 >  
</ body >  
</ html >
```

Зберігти її в / assets / www /.

Запустити.

Вибрати емульований пристрій.

З'явилося вікно емулятора і в ньому наш додаток [18 – 20].

Змістовий модуль 2. Засоби програмування серверної частини додатку

Лабораторна робота № 5. Створення додатку з контролем введення на серверній частині

Мета роботи. Ознайомитись із засобами забезпечення цілісності та безпеки даних у скриптах PHP, роботою з регулярними виразами, фільтрами.

Робота складається з таких етапів:

1. Підготовчий. Створити новий проект у середовищі NetBeans.

Не забути змінити адресу URL згенеровану оболонкою на адресу веб-сервера:

<http://localhost/tst/RegExp/>.

Додати до головного проекту файл index.php.

Заповнити його текстом "приклад використання регулярних виразів" з тексту лекції.

Запустити на виконання.

Перевірити функціональність.

Змінити виведення повідомлень про помилки на більш ергономічний вид, наприклад, вивести повідомлення про некоректне значенні поряд з відповідним полем.

2. Змінити код (текст на PHP, текст регулярних виразів і виведений HTML) таким чином , щоб він відповідав темі вашого варіанту індивідуального завдання .

3. Змінити код (текст на PHP, текст фільтрів і прапорів та HTML, що виводиться) побудувавши веб-додаток з тією ж функціональністю, використовуючи можливості фільтрації.

4. Порівняти можливості і зручність:

- Різних способів перевірки вводу на сервері.
- Різних способів перевірки вводу на клієнті.
- Різних способів перевірки вводу на NET Forms.

Оформити результат, супроводивши звіт скриншотами процесу виконання. [2; 4; 6].

Лабораторна робота № 6. Створення додатків з використанням компонент, що генеруються серверним фреймворком

Мета роботи. Ознайомитись із засобами створення скриптів PHP, роботою з масивами, оператором циклу.

Робота складається з таких етапів:

Підготовчий. Залежно від складу встановленого програмного забезпечення (веб-сервер, версія інтерпретатора PHP, версія IDE) і прав користувача змінюється і порядок проведення підготовчого етапу.

Встановити Denver.

Запустити Денвер Run.exe.

Помістити папки та файли в папку веб-сервера.

Встановити відладчик Xdebug.

Викачати відповідну до версії PHP бібліотеку сервера відгадчика, якщо вона не відповідає версії, що встановлено Денвером – у нашому випадку це PHP Version 5.3.13. Тому нам не довелося качати – підходяща версія Xdebug вже лежить в папці, скопіювалася з даними екземпляром Денвера.

Перевірити настройки php.ini з Z: \ usr \ local \ php5 \ php.ini.

Переглянути його зміст у блокноті.

Замінити на правильний шлях до бібліотеки відгадчика.

```
zend_extension_ts = " C: \ WebServers \ php \ ext \ ↵  
  php_xdebug - 2.2.0 - 5.3- vc9.dll "
```

```
zend_extension = " C: \ WebServers \ php \ ext \ ↵  
  php_xdebug - 2.2.0 - 5.3- vc9.dll "
```

```
xdebug.remote_enable = On
```

```
; xdebug.remote_host = localhost
```

```
xdebug.remote_connect_back = 1
```

```
xdebug.remote_port = 9000
```

Номер порту повинен відповідати номеру порту відладчика в меню "Сервіс"> "Параметри"> "PHP" в середовищі IDE NetBeans:

```
xdebug.remote_handler = dbgp
```

Подивитися ім'я отладочного процесу на сервері:

```
xdebug.remote_mode
```

```
xdebug.remote_mode = req
```

Раскомментируем рядка

```
xdebug.remote_handler = " dbgp "
```

```
xdebug.remote_port = 9000
```

dbgп порт 9000 необхідно вказати в імені відладчика NetBeans.

Запустити портативний NetBeans та налаштувати його – набудувати параметри налагодження.

Створити новий проект.

Підказка шляху, яку було запропоновано – невірна. Необхідно змінити на <http://localhost/tst/tictactoe>.

Відкрити існуючий файл ігри в хрестики – нулики.

Встановити точку зупину клацанням по першому стовпцю вікна коду.

Запустити командою "Почати налагодження" в меню "Відлагодження".

Відладчик зупинився на точці зупину. Це позначає , що ми правильно налаштували Денвер, XDEBUG і NetBeans

Тепер створити новий проект, який буде містити ваше індивідуальне завдання.

Папка з проектом повинна знаходитися на віртуальному диску диску Z: (Денвер повинен бути запущено).

Створити текст сторінки, на якій, при завантаженні виконувалися б наступні функції від аргументу – вектора (0, 3, -7, 0, 4, -6, 10, 3) і результат відображався б на веб-сторінці, що сформовано кодом на PHP .

Продемонструвати всі можливості відлагодження на прикладі вашої програми.

Показати, як відбувається створення PHP-сторінки для обчислення суми елементів вектора.

Створити новий проект. Створити папку для проекту , доступну веб-серверу – обов'язково всередині Z: \ home \ localhost \ www \ . Дати їй значуще ім'я. Увага! Шлях, що запропоновано оболонкою, – неправильний , має бути <http://localhost/tst/sum/>.

Зберігти проект.

Додати форму PHP с допомогою контекстного меню папки файл з ім'ям index.php. Набрати код програми обчислення суми (рис. 6.1).

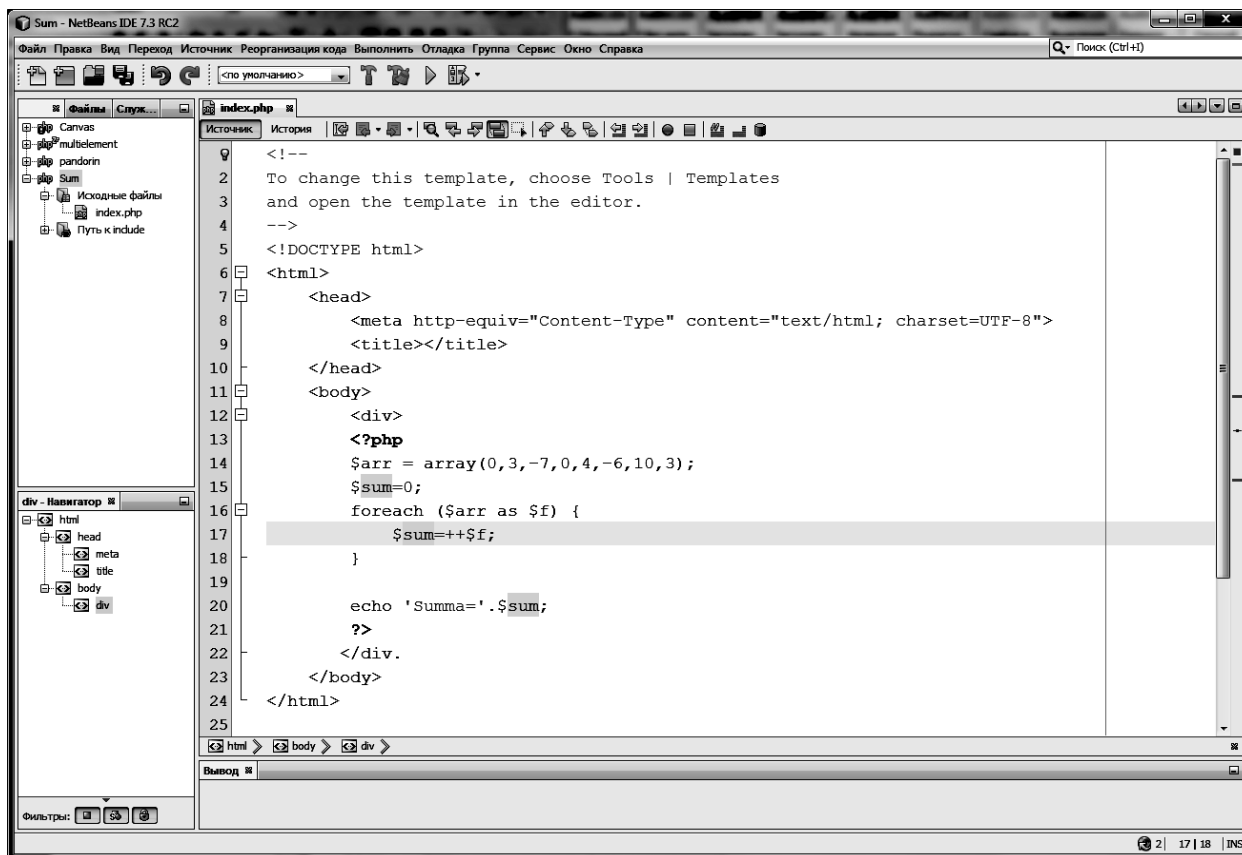


Рис. 6.1. Текст програми обчислення суми елементів вектору

Запустити програму. Результат показано на (рис. 6.2).

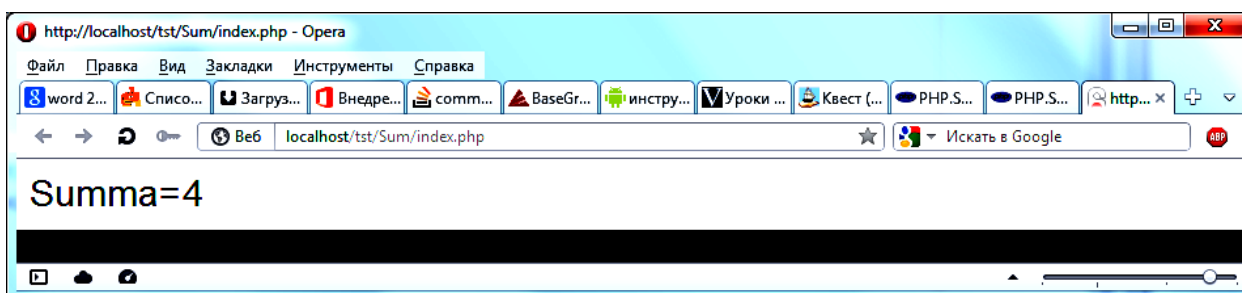


Рис. 6.2. Результат виконання програми обчислення суми елементів вектору

Додати передачу вектора з клієнта на сервер (рис. 6.3).

Запустити програму. Результат показано на рис. 6.4 – 6.5 [2; 4; 6].

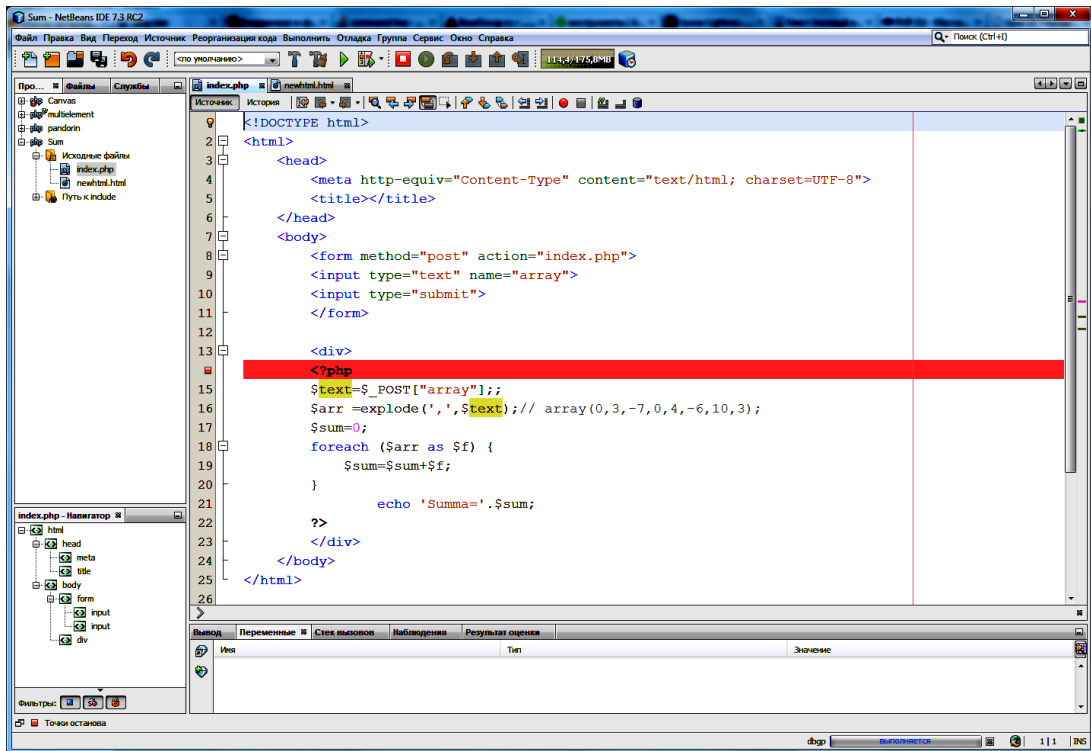


Рис. 6.3. Текст програми обчислення суми елементів вектору з передачею вектору через параметри запиту

Запустити програму. Результат показано на рис. 6.4 – 6.5.



Рис. 6.4. Завантажена програма обчислення суми елементів вектору з передачею вектору через параметри запиту

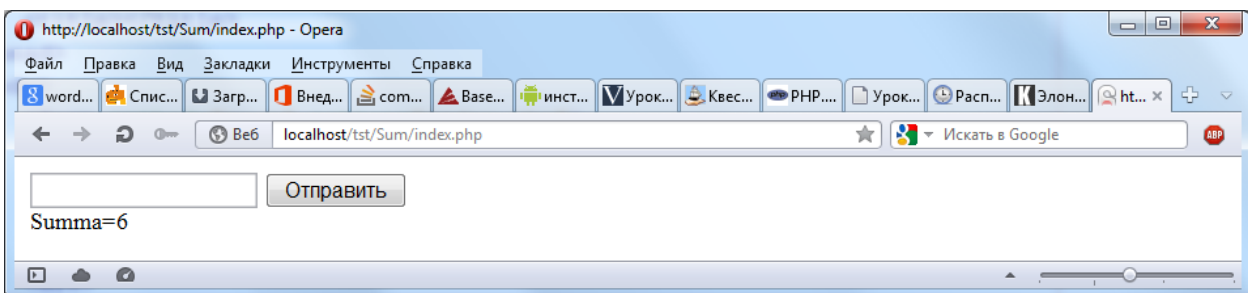


Рис. 6.5. Результат виконання програми обчислення суми елементів вектору

Лабораторна робота № 7. Створення додатків з використанням компонент, що використовують взаємодію серверної та клієнтської частин (ч. 1)

Мета роботи. Познайомиться з форматами передачі значень об'єктів в AJAX на клієнтській частини.

Зазвичай використовується один з форматів двох форматів передачі значень передачі значень JSON або XML.

Для перетворення значень об'єкта JavaScript в JSON існує метод `JSON.stringify ()`, який бере змінну і повертає рядок JSON, що представляє її зміст.

Приклад використання.

```
<script type="text/javascript">
var cart = {
    "orderId": 12345,
    "shopperName": "Ваня Иванов",
    "shopperEmail": "ivanov@example.com",
    "contents": [
        {
            "productID": 34,
            "productName": "Супер товар",
            "quantity": 1
        },
        {
            "productID": 56,
            "productName": "Чудо товар",
            "quantity": 3
        }
    ],
    "orderCompleted": true
};

alert ( JSON.stringify( cart ) );
</script>
```

Для розбору строки JSON в JavaScript, існує вбудований метод `JSON.parse()`. Він отримує рядок JSON і повертає об'єкт або масив JavaScript, який містить дані.

Приклад використання.

```
<?php
$jsonString = '
    {
        "orderID": 12345,
        "shopperName": "Ваня Иванов",
        "shopperEmail": "ivanov@example.com",
        "contents": [
            {
                "productID": 34,
                "productName": "Супер товар",
                "quantity": 1
            },
            {
                "productID": 56,
                "productName": "Чудо товар",
                "quantity": 3
            }
        ],
        "orderCompleted": true
    }
';
$cart = json_decode( $jsonString );
echo $cart->shopperEmail . "<br>";
echo $cart->contents[1]->productName . "<br>";
?>
```

Доповнити вашу програму з модуля 1 засобами запису і читання значень об'єктів у форматі JSON.[2; 4; 6; 18].

Лабораторна робота № 8. Створення додатків з використанням компонент, що використовують взаємодію серверної та клієнтської частин (ч. 2)

Мета роботи. Pozнайомитися з форматами передачі значень об'єктів в AJAX на північній частині.

PHP, як і JavaScript, має вбудовані функції для роботи з JSON рядками.

Створити рядок JSON із змінної PHP. Функція `json_encode()` приймає змінну PHP і повертає рядок JSON, що подає зміст змінної.

```
<?php
$cart = array(
    "orderID" => 12345,
    "shopperName" => "Ваня Иванов",
    "shopperEmail" => "ivanov@example.com",
    "contents" => array(
        array(
            "productID" => 34,
            "productName" => "Супер товар",
            "quantity" => 1
        ),
        array(
            "productID" => 56,
            "productName" => "Чудо товар",
            "quantity" => 3
        )
    ),
    "orderCompleted" => true
);

echo json_encode( $cart );
?>
```

Для перетворення рядка JSON в змінну PHP використовується метод:

```
json_decode ().print.  
<?php  
$jsonString = '  
{  
  "orderID": 12345,  
  "shopperName": "Ваня Иванов",  
  "shopperEmail": "ivanov@example.com",  
  "contents": [  
    {  
      "productID": 34,  
      "productName": "Супер товар",  
      "quantity": 1  
    },  
    {  
      "productID": 56,  
      "productName": "Чудо товар",  
      "quantity": 3  
    }  
  ],  
  "orderCompleted": true  
}  
';  
  
$cart = json_decode( $jsonString );  
echo $cart->shopperEmail . "<br>";  
echo $cart->contents[1]->productName . "<br>";  
>
```

Доповнити вашу програму з модуля 2 засобами запису і читання значень об'єктів у форматі JSON. Побудувати зв'язок за допомогою об'єкту `HttpRequest` між клієнтською та серверною частинами додатку.[2; 4; 6; 18].

Рекомендована література

Основна

1. Дунаев В. Сценарии для Web-сайта. PHP и JavaScript / В. Дунаев. – СПб. : БХВ-Петербург ; 2009. – 576 с.
2. Маккоу А. Веб-приложения на JavaScript / А. Маккоу. – СПб. : Питер, 2012.– 288 с.
3. Крокфорд Д. JavaScript. Сильные стороны JavaScript / Д. Крокфорд. – СПб. : Питер ; 2010, 176 с.
4. Кузнецов М. В. PHP. Практика создания Web-сайтов / М. В. Кузнецов. – СПб. : БХВ-Петербург, 2009. – 1 264 с.
5. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript / Р. Никсон. – СПб. : Питер 2013 г. – 496 с.
6. Флэнаган Д. JavaScript. Подробное руководство / Д. Флэнаган. – М. : Символ-Плюс. – 2013. – 1 080 с.

Додаткова

7. Гаврилов В. П. Сборник задач по учебной дисциплине "Основы программирования и алгоритмические языки" для студ. спец. 7.080401, 7.080407 всех форм обуч. / В. П. Гаврилов, В. В. Браткевич. – Х. : ХНЭУ, 2006. – 99 с.
8. Гиковатый В. М. Основы алгоритмизации на базе объектно-ориентированного программирования [Текст] : учебн. пособ. / В. М. Гиковатый. – Х. : ХНЭУ, 2006. – 90 с.
9. Молчанов В. П. Основы проектування Web-видань [Текст] : конспект лекцій / В. П. Молчанов. – Х. : ХНЕУ, 2008. – 166 с.
10. Молчанов В. П. Робоча програма навчальної дисципліни "Опрацювання відео- та аудіоінформації" для студ. спец. 8.092702 "Технологія електронних мультимедійних видань" усіх форм навч. / В. П. Молчанов, О. К. Пандорін. – Х. : ХНЕУ, 2009. – 31 с.
11. Молчанов В. П. Робоча програма навчальної дисципліни "Основы проектування WEB-видань" для студ. спец. "Комп'ютеризовані технології та системи видавничо-поліграфічних виробництв" усіх форм навч. / В. П. Молчанов. – Х. : ХНЕУ, 2008. – 27 с.

12. Молчанов В. П. Технології WEB-дизайну [Текст] : конспект лекцій / Харківський національний економічний університет / В. П. Молчанов. – Х. : ХНЕУ, 2011. – 211 с.

Ресурси мережі Інтернет

13. Болховитинова С. М. и др. Композиция изданий: Особенности проектирования различных типов изданий : учебн. пособ. ; под ред. С. М. Болховитиновой. – М. : Изд-во МГУП, 2000. –166 с. [Электронный ресурс]. – Режим доступа : <http://www.hiedu.ru/ebooks/xbook095/01/index.html>.

14. Обзор инструментальных средств разработки мультимедиа [Электронный ресурс]. – Режим доступа : <http://clubrus.kulichki.net/obzor.html>.

15. Пишем приложения с PhoneGap [Электронный ресурс]. – Режим доступа : <http://freehabr.ru/blog/android/3201.html>.

16. Разработка мобильных приложений на PhoneGap и jQuery Mobile [Электронный ресурс]. – Режим доступа : <http://habrahabr.ru/post/118059/>.

17. Разработка мобильных приложений: Часть 1. PhoneGap и Dojo Mobile на Android [Электронный ресурс]. – Режим доступа : <http://www.ibm.com/developerworks/ru/library/wa-mobappdev1/>.

18. JSON: основы использования [Электронный ресурс]. – Режим доступа : http://ruseller.com/lessons.php?rub_id=28&id=1212.

Зміст

Вступ.....	3
Змістовий модуль 1. Базові концепції використання скриптових мов та їх реалізація у найпоширеніших документних моделях на клієнтській частині.....	4
Лабораторна робота № 1. Використання інструментальних засобів для створення та відлагоджування серверної та клієнтської частин додатку (IDE MS VS, IDE NetBeans, FIREBUG, Chrome Developers Tools для навігації простого циклічного додатку).....	4
Лабораторна робота № 2. Забезпечення контролю цілісності та юзабіліті засобами клієнтської частини (проходом елементами DOM та перевірок коректності введення значень)	7
Лабораторна робота № 3. Графічні засоби відображення стану форми	8
Лабораторна робота № 4. Створення додатку для мобільної платформи	14
Змістовий модуль 2. Засоби програмування серверної частини додатку	16
Лабораторна робота № 5. Створення додатку з контролем введення на серверній частині.....	16
Лабораторна робота № 6. Створення додатків з використанням компонент, що генеруються серверним фреймворком.....	17
Лабораторна робота № 7. Створення додатків з використанням компонент, що використовують взаємодію серверної та клієнтської частин(ч. 1)	21
Лабораторна робота № 8. Створення додатків з використанням компонент, що використовують взаємодію серверної та клієнтської частин (ч. 2).....	23
Рекомендована література.....	25

НАВЧАЛЬНЕ ВИДАННЯ

**Методичні рекомендації
до виконання лабораторних робіт
з навчальної дисципліни
"СКРИПТОВІ МОВИ"
для студентів напряму підготовки
6.051501 "Видавничо-поліграфічна справа"
всіх форм навчання**

Укладач **Пандорін** Олександр Костянтинович

Відповідальний за випуск **Пушкар** О. І.

Редактор **Промський** М. Н.

Коректор **Маркова** Т. А.

План 2014 р. Поз. № 130.

Підп. до друку 03.12.2014 р. Формат 60 x 90 1/16. Папір MultiCopy. Друк Riso.

Ум.-друк. арк. 1,75. Обл.-вид. арк. 2,19. Тираж 50 прим. Зам. № 313.

Видавець і виготівник – видавництво ХНЕУ ім. С. Кузнеця, 61166, м. Харків, пр. Леніна, 9-А

Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи

Дк № 481 від 13.06.2001 р.