**ВСТУП**

В даний час бурхливого розвитку комп'ютерних технологій усе більш актуальним завданням є розробка і використання в навчальному процесі електронних навчальних систем, що розробляються з застосуванням гіпертекстових і мультимедійних технологій. Такі системи називаються інтерактивними навчальними Web-матеріалами та можуть використовуватися не тільки для денної, заочної та самостійної форм навчання, але і знайти широке застосування під час самостійної роботи студентів.

Мета виконання дипломного проекту полягає у проведенні досліджень та створенні інформаційно-документальної бази даних для розробки електронного посібника з дисципліни «Комп’ютерна логіка».

Актуальність проведення досліджень по створенню інформаційно-документальної бази електронного посібника з дисципліни «Комп’ютерна логіка» обумовлена необхідністю впровадження сучасних мультимедійних електронних посібників з метою удосконалення навчального процесу для різних форм навчання.

Завдання дипломного проекту – створення електронного посібника з дисципліни «Комп’ютерна логіка», як універсальної методичної розробки для студентів спеціальності 5.091504 «Обслуговування комп’ютерних та інтелектуальних систем та мереж», яка може використовуватися при вивченні дисципліни «Комп’ютерна логіка» в фахових навчальних курсах з комп’ютерного забезпечення.

Об’єктом дослідження є навчально-методичний комплекс спеціальності «Обслуговування комп’ютерних та інтелектуальних систем та мереж». Під час дипломного дослідження був здійсненний аналіз існуючих електронних та мультимедійних розробок, які використовуються в освітній діяльності та можливості застосування сучасних інформаційних технологій при створенні інформаційно-документальної бази для спеціалізованого електронного посібника.

На сучасному етапі розвитку інформаційних технологій все більш актуальним стає використання мультимедійних та Інтернет-технологій, які усе більше проникають в різні сфери нашого життя. Більшість користувачів персональних комп’ютерів починають освоєння мультимедіа з ігрових програм, але цією областю не вичерпуються численні можливості його використання.

Серед основних областей застосування засобів мультимедіа виділяють: ділову сферу; сферу розваги й освіти. В даний час значна частина освітніх, розважальних, інформаційно-довідкових програм на споживчому ринку відноситься до категорії мультимедіа.

В дусі мультимедіа створюється і малотиражна продукція рекламно-інформаційного характеру – каталоги, електронні посібники, довідники, презентації. Приклади використання мультимедіа можна найти в мультимедіа-енциклопедіях «Eyewitness History of the World», «Cinemania» «Encarta», «Большая энциклопедия Кирилла и Мефодия» та багатьох інших.  
Енциклопедії, альманахи, збори довідників, діалогові ігри, освітні програми і кінофільми із супровідним сценарієм, біографіями акторів, примітками режисера й оглядами роблять мультимедіа однієї з найбільш цікавих і творчих областей комп'ютерного світу.

Однак, використання мультимедіа, і персональних комп'ютерів узагалі, не обмежується сферою ділового застосування. Комп'ютер усе ширше входить й у сферу розваг і персональної освіти. Створюються продукти мультимедіа, розраховані на широке коло споживачів. Споживчою сферою застосування мультимедіа є інтерактивні CD-ROM, інтерактивні ігри та он-лайнові послуги.

Розвиток глобальної комп'ютерної мережі показав перспективність і необхідність віддаленого навчання студентів за допомогою інтерактивних електронних посібників, встановлюваних на серверах, підключених до локальної комп'ютерної мережі чи мережі Інтернет. Більш ширше розповсюдження таких інформаційних технологій в освіті дозволяє не тільки підвищити інтенсивність і ефективність процесу навчання, але й істотно розширити аудиторію потенційних слухачів провідних навчальних закладів країни.

Електронний посібник є однією з нових інформаційних технологій, які використовують в навчальному процесі. Тому дослідження та розробка електронного посібника з дисципліни «Комп’ютерна логіка», в даному дипломному проекті, дозволить удосконалити якість навчання студентів ТК ТНТУ ім. Івана Пулюя та дозволить знаходити необхідну їм інформацію серед величезної кількості інформації, яка вивчається студентами комп’ютерних спеціальностей протягом всього навчального процесу.

Даний дипломний проект містить вступ, 3 основних розділи, 2 додаткових розділи висновки, перелік використаної літератури та додатки.

У 1-му розділі розглядається характеристика сучасних мультимедійних та електронних продуктів, зокрема, електронних посібників, які використовуються в навчальних закладах, у 2-му розділі формулюються вимоги до розробки електронних посібників та розглядаються інструментальні засоби їх створення, у 3-му розділі наведено опис інформаційно-документального забезпечення електронного посібника, а також наведені деякі екранні форми та довідкові матеріали для електронного посібника.

**1. ЗАГАЛЬНИЙ РОЗДІЛ**

* 1. **Аналітичний огляд існуючих рішень**

Електронні посібники створюються з використання гіпертекстових технологій та технологій мультимедіа. Дамо стислу характеристику цих технологій.

Від SGML до HTML та XHTML

Стандартна узагальнена мова розмітки SGML — нащадок розробленої в 1960 році в IBM мови GML (від англ. Generalized Markup Language).

Спочатку SGML була розроблена для можливості спільного використання документів, що мають читатися машинами, у великих урядових та аерокосмічних проектах. Також вона широко використовується в печатній та видавничій сферах, але складність документа ускладнила його широке розповсюдження для повсякденного використання.

SGML надає множину варіантів синтаксичної розмітки для використання різними прикладними програмами. Змінюючи декларацію SGML можна навіть відмовитись від використання кутових дужок, хоча, цей синтаксис вважається стандартним.

В 1980 році фізик Тім Бернерс-Лі, який був підрядником у ЦЕРНі, запропонував і розробив прототип ENQUIRE, систему для вчених та дослідників ЦЕРНу для використання документів та обміну даними. В кінці 1990 року Бернерс-Лі узагальнив та написав програмне забезпечення HTML для браузерів та серверів.

Першою публічною згадкою про HTML був документ «HTML Tags» вперше викладений в Інтернеті Бернерсом-Лі наприкінці 1991 року. В ньому описувалось 20 елементів включаючи початковий, відносно просте оформлення. За винятком гіперпосилання будова HTML була дуже схожа з SGML. Тринадцять з тих елементів до сих пір використовуються в HTML 4.

HTML – це мова розмітки, яку браузери використовують для відображення та опису тексту, зображень та даних іншого типу. Базова характеристика роботи з HTML описана в самому браузері, але також може бути змінена чи розширена за допомогою CSS – каскадних таблиць стилів.

Документи HTML складаються повністю з елементів HTML, що зазвичай складаються з трьох елементів: відкриваючого та закриваючого тега, атрибутів всередині відкриваючого тега а також текстової чи графічної інформації між відкриваючим та закриваючим тегом, нерідко інших елементів HTML. Елементом HTML вважається все, що знаходиться всередині тега, включаючи сам тег. Кожен тег поміщується в трикутні дужки.

HTML підтримує декілька типів даних, наприклад: сценарії та стилі, а також безліч типів даних для атрибутів елементів, таких як: унікальний номер, ім’я, посилання, цифри, мови, кольори, кодування дата та час та ін.

Розширювана мова розмітки гіпертексту XHTML  — мова розмітки, що має таку саму виразну силу як і HTML але відповідає синтаксичним правилам XML.

XHTML 1.0 є «реформулюванням трьох типів документів стандарту HTML 4 засобами XML 1.0». W3C також продовжує підтримку Рекомендації HTML 4.01 та активну роботу над специфікаціями стандартів [HTML 5](http://uk.wikipedia.org/wiki/HTML_5) і XHTML 5. В поточному документі Рекомендацій XHTML 1.0, який було опубліковано та переглянуто до серпня 2002 року, W3C зазначив, що, «Сімейство XHTML є наступним кроком в еволюції Інтернету». Шляхом переходу на XHTML сьогодні, розробники контенту можуть увійти в світ XML з усіма його перевагами, залишаючись впевненими в зворотній та майбутній сумісності їхнього контенту.

Проте в 2004 році, незалежно від W3C було створено [Робочу групу з web-додатків та гіпертекстових технологій](http://uk.wikipedia.org/w/index.php?title=%D0%A0%D0%BE%D0%B1%D0%BE%D1%87%D1%83_%D0%B3%D1%80%D1%83%D0%BF%D1%83_%D0%B7_%D0%B2%D0%B5%D0%B1-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BA%D1%96%D0%B2_%D1%82%D0%B0_%D0%B3%D1%96%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D0%B8%D1%85_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D0%B9&action=edit&redlink=1) ([WHATWG](http://uk.wikipedia.org/w/index.php?title=WHATWG&action=edit&redlink=1)), для роботи по вдосконаленню звичайного HTML не заснованого на XHTML. Більшість великих виробників [браузерів](http://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) не бажали реалізовувати функції з нових проектів стандартів W3C XHTML оскільки вважали, що вони не відповідають сучасним потребам розвитку Інтернету, а W3C захопився [формалізмом](http://uk.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%96%D0%B7%D0%BC) XML і не реагував на реальні вимоги виробників. Apple, Mozilla та Opera сформували робочу групу [WHATWG](http://uk.wikipedia.org/w/index.php?title=WHATWG&action=edit&redlink=1), яка почала працювати над стандартом [HTML 5](http://uk.wikipedia.org/wiki/HTML5), який допускав, але не вимагав застосування XML. У 2007 році, Робоча група W3C HTML проголосувала за офіційне визнання HTML 5 і роботу над ним як наступне покоління стандарту HTML.

Завдяки динамічному HTML перед web-дизайнерами відкриваються нові можливості для підготовки змісту, щоб передати його в Internet. Якщо до HTML 3.2 web-сторінки були статистичними, то за новим стандартом з'явилися необмежені можливості у оформленні документів. Завдяки Java-аплетам, Java- та VB-сценаріям, а також Stylesheets, web-сторінки стали не тільки візуально привабливими, але й інтерактивними. Так, наприклад, HTML-документ може безпосередньо реагувати на команди користувача і самостійно оновлюватися.

Щоправда, використання динамічного HTML в web-сторінках потребує принципового планування і чіткої концепції. Наприклад, web-автор повинен заздалегідь визначитися, в якій послідовності елементи сторінки будуть представлені її відвідувачу. Додатково повинен бути розроблений план діалогу, який вкаже користувачу на різні інтерактивні ефекти (анімацію, випливаючі логотипи чи прихований зміст).

Крім того, web-автор повинен знати, що з використанням динамічного HTML, його потенційна клієнтура може зменшитися. Тому, що не кожен мандрівник мережі Internet зможе побачити в своєму web-браузері результати цієї технології. Для цього йому потрібен або Netscape Communicator, або Internet Explorer, і вони повинні бути не нижче 4-ої версії. Використання старішої версії призведе до помилки чи зміщення символів на екрані монітора.

Для запобігання цього на практиці, рекомендується розробити паралельно декілька версій, орієнтованих на конкретний web-браузер. Як правило, буває недостатньо простої версії, розробленої у відповідності зі старим стандартом 3.2.

HTML-редактори

Можна створювати найскладніший HTML-документ, користуючись лише найпростішим текстовим редактором. Проте застосування засобів, спеціально призначених для цієї цілі, дозволить заощадити чимало часу і зусиль. Для створення нових web-документів найбільш ефективними є HTML-редактори. У зв'язку з поширенням Internet в оформленні web-документів зростає кількість дизайнерів, які не хочуть навчатися мові HTML. Тому і створені були такі HTML-редактори, у яких web-дизайнеру більше не треба працювати з HTML-кодами. Цілком достатньо вибрати відповідну команду меню.

Практично для всіх платформ створені окремі незалежні програми, а також програмні надбудови до поширених додатків для роботи з HTML. В основному це умовно-безкоштовне програмне забезпечення. У наступній частині будуть описані найбільше поширені HTML-редактори. Це допоможе зробити обґрунтований вибір того або іншого редактора.

Для створення будь-якого HTML-документа достатньо простого текстового редактора, проте більшість розробників вважають, що краще використовувати спеціалізований HTML-редактор, який забезпечує контроль помилок і виконує інші корисні функції. Стандартні HTML-редактори є самостійними прикладними програмами, призначеними для підготовки HTML-документів, що не потребують додаткових програмних засобів. Існує два основних типи редакторів: текстові та графічні.

При роботі з текстовими редакторами HTML-код виводиться у вигляді звичайного тексту, а HTML-дескриптори виділяються іншим кольором.  
Графічні редактори дозволяють створити макет сторінки за допомогою миші. При цьому на екрані буде чудовий результат і такий, яким побачать створену сторінку відвідувачі.

У багатьох редакторах використовуються два типи інтерфейсу, між якими можна легко переключатися (яскравим прикладом служить HotDog, FrontPage98).

Графічні редактори забезпечують більш естетичний підхід до створення документів, дозволяючи зосередитися на їхньому макетуванні і форматуванні.  
Редактори HTML, що поставляються у вигляді надбудов, є додатковими програмами, що функціонують у якості розширень існуючих додатків (як правило, текстових процесорів). У будь-якому випадку використання надбудов має як сильні, так і слабкі сторони.

Коли стандарт HTML уперше став використовуватися в Internet, існували тільки HTML-редактори, які підтримували лише стандартні HTML-коди (їх називали тестовими або ASCII-редакторами). Текстові редактори і в даний час залишаються самими популярними інструментами створення web-документів. Як правило, такі редактори можуть оперувати тільки деяким набором регулярних кодів. Список цих кодів для більшості редакторів приведений у меню дескрипторів. Як приклади текстових редакторів можна навести Блокнот, Notepad++ і WordPad (Windows), vi та paco (UNIX) і SimpleText(Macintosh).

Одна із причин, чому розробники віддають перевагу текстовим редакторам, полягає в тому, що за їх допомогою набагато простіше вносити в документ будь-які зміни, що доводиться робити практично постійно. Ті, хто використовують текстові редактори, дозволяють набагато легше маніпулювати текстом документа і забезпечують більш гнучкий і творчий підхід при розміщенні в документ програмного тексту.

Розширювана мова розмітки XML

Розширювана мова XML – запропонований консорціумом World Wide Web (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними додатками, зокрема, через Інтернет. Є спрощеною підмножиною мови розмітки SGML. XML документ складається із текстових знаків, і придатний до читання людиною.

Стандарт XML визначає набір базових лексичних та синтаксичних правил для побудови мови описання інформації шляхом застосування простих тегів. Цей формат достатньо гнучкий для того, аби бути придатним для застосування в різних галузях. Іншими словами, запропонований стандарт визначає метамову, на основі якої, шляхом запровадження обмежень на структуру та зміст документів визначаються специфічні, предметно-орієнтовані мови розмітки даних.

Залишивши назви, дозволену ієрархію, та значення елементів і атрибутів відкритою та можливою бути визначеною в спеціалізованих схемах або визначеннях типу документа (DTD), XML утворює синтаксичну основу для створення спеціалізованих, основаних на XML мовах розмітки даних. Загальний синтаксис таких документів стабільний і наперед визначений – документи мають відповідати базовим вимогам XML, гарантуючи те, що довільне програмне забезпечення з підтримкою XML буде здатне щонайменше зчитати і відтворити відносну структуру інформації що міститься в них. Схема лише доповнює синтаксичні правила множиною обмежень. Зазвичай, схеми обмежують назви елементів та атрибутів, дозволені типи значень і допустиму ієрархію елементів, наприклад, дозволяючи лише елементу з назвою «народження» містити під-елемент з назвою «місяць» та з назвою «день», і кожен із них мусить містити лише літери. Обмеження, вказані в схемі, можуть також включати присвоєння певних типів даних для впливу на те, як обробляється інформація; наприклад, дані елемента «місяць» можна визначити як такі, що містять лише місяць, як це визначено відповідно до використаної мови схем.

Гіпертекстовий препроцесор PHP

PHP – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні web-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері web-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP – проект відкритого програмного забезпечення.

PHP інтерпретується web-сервером в HTML-код, який передається на сторону клієнта.

На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий HTML-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Тим не менш це не забороняє використовувати РНР для генерування JavaScript-кодів які виконаються вже на стороні клієнта.

Історія PHP починається з 1995 року, коли Расмус Лердорф створив простий застосунок мовою Perl, що аналізував відвідування користувачами його резюме на web-сайті. Потім, коли цим застосунком вже користувалися кілька чоловік, а число охочих одержати його постійно збільшувалося, Лердорф назвав своє творіння Personal Home Page Tools версія 1 і виставив для вільного завантаження. З цієї миті почався небувалий зліт популярності PHP.

Як це завжди буває, терміново було потрібно доопрацювання і нові доповнення. Для їхньої реалізації Расмус створює нову версію пакету, тепер уже написану С. Отриманий таким чином інструмент набуває робочої назви PHP/FI (Personal Home Page / Forms Interpreter), надалі він також буде відомий під назвою PHP 2. Ця версія вже більшою мірою схожа на сьогоднішній PHP. Вона мала синтаксис і спосіб іменування змінних в стилі мови Perl, можливість вбудовування PHP операторів в HTML-код сторінки, автоматичну інтерпретацію форм, інтеграцію з базами даних. При цьому все працювало досить швидко, оскільки PHP приєднувалось до web-серверу Apache. До 1997 року PHP використовувався вже на 50,000 доменах (не більше 1% всіх web-серверів).

У тому ж 1997 році до проекту PHP підключилися Зеєв Сураський і Енді Гутманс. Ці студенти Техніону, одного з ізраїльських університетів, намагалися використовувати PHP/FI для одного з комерційних університетських проектів. При цьому їм довелося зіткнутися з багатьма труднощами і обмеженнями цієї технології. Вивчаючи початковий код PHP 2, Зеєв і Енді дійшли висновку про необхідність доопрацювання, а точніше істотної переробки PHP, особливо в плані синтаксису мови. Протягом декількох місяців вони блискуче впоралися з цим завданням.

Закінчивши роботу Зеєв і Енді домовились з Расмусом про співпрацю в галузі розвитку та вдосконалення мови. З цієї миті з'являється PHP Group – група однодумців, що працюють над розвитком технології PHP. Одержаний продукт з'явився на світ у 1998 році під назвою PHP 3.

При цьому головною особливістю PHP 3 була можливість розширення ядра, що привернуло до роботи над PHP безліч сторонніх розробників, що створюють спеціалізовані модулі. Їх наявність дала PHP можливість працювати з величезною кількістю баз даних, протоколів, підтримувати велике число API. До кінця 1998 кількість користувачів PHP перевалила за 100000, а PHP був вже встановлений на не менше ніж 10% серверах Інтернету. У той ж час значному поширенню даної мови сприяли публікації в електронній пресі та вихід книжок для вивчення PHP.

Відразу ж після виходу PHP 3, Енді Гутманс і Зеєв Сураський почали переробку ядра PHP. В першу чергу належало вирішити проблему підвищення продуктивності. Новий продукт, названий Zend Engine, успішно справлявся з поставленим завданням і був реалізований в 1999 році. Основною реалізованою ідеєю була можливість компіляції сценарію у виконуваний модуль, за рахунок чого продуктивність можна було підняти на порядок.

PHP 4, що працює на цьому ядрі, вийшов в 2000 році. На додаток до збільшення продуктивності, PHP 4 мав нові можливості щодо підтримки сесій, буферизацію виводу, безпечні способи обробки інформації, що вводиться користувачем, і нові мовні конструкції. З виходом 4 версії PHP став використовуватися вже на більш ніж 20% доменів Інтернету.

За час з 2000 по 2004 рік продовжувалися активні роботи з покращення 4 версії, але майже відразу PHP Group приступила до продумування можливостей нової версії. В першу чергу було вирішено підсилити об'єктні можливості мови, що дозволяло використовувати його для реалізації масштабних проектів. Роботи по створенню 5-тої версії велися тривалий час, в них брало участь рекордна кількість фахівців, зокрема Стерлінг Хьюз і Маркус Бергера.

У липні 2004 року виходить офіційний реліз PHP 5. В першу чергу, як і планувалося, було перероблено весь механізм роботи з об'єктами. І якщо в попередніх версіях об'єктно-орієнтоване програмування на PHP було можливе в мінімальному ступені, а тому і використовувалося на практиці не часто, то PHP 5 володіє прекрасним потенціалом реалізації об'єктного програмування. Окрім цього, PHP збагатився рядом цінних розширень для роботи з XML, різними джерелами даних, генерації графіки і інше.

Серед інших украй корисних доповнень в PHP 5 слід зазначити нову схему обробки виключень. Конструкція try/catch/throw дозволяє весь код обробки помилок локалізувати в одному місці сценарію.

Всі основні бібліотеки для роботи з XML, запозичені в PHP 4, були піддані серйозній переробці. Такі популярні розширення, як SAX, DOM і XSLT, тепер використовують інструмент libxml2, що робить їх ще ефективнішими.

У PHP 5 також включені два нові модулі для роботи з протоколами — SimpleXML і SOAP. SimpleXML дозволяє значно спростити роботу з XML-даними, представляючи вміст XML-документа у вигляді PHP-об'єкта. Розширення SOAP дозволяє будувати на PHP сценарії, що обмінюються інформацією з іншими застосунками за допомогою XML-повідомлень поверх існуючих web-протоколів, наприклад HTTP. Модуль для роботи з SOAP для PHP 5 надає розробникам засіб для достатньо швидкого створення ефективних SOAP-клієнтів і SOAP-серверів.

Новий модуль PHP 5 MySQLi (MySQL Improved) призначений для роботи з MySQL-сервером версій 4.1.2 та вищих, реалізовуючи не тільки процедурний, але і об'єктно-орієнтований інтерфейс до MySQL. Додаткові можливості цього модуля включають – SSL, контроль транзакцій, підтримка реплікації та ін.

Каскадні таблиці стилів CSS

Каскадні таблиці стилів – спеціальна мова, що використовується для відображення сторінок, написаних мовами розмітки даних. Найбільш часто CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої павутини.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже існуючи функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці web-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їх візуальної презентації.

CSS використовується авторами та відвідувачами web-сторінок для того щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори та ін. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерах та ін.).

Один і той же HTML або XML документ може бути відображений по-різному в залежності від використаного CSS. Стилі для відображення сторінки можуть бути:

* стилі автора (інформація надана автором сторінки):
* зовнішні таблиці стилів частіше за все окремий файл або файли \*.css;
* внутрішні таблиці стилів, включені як частина документу або блоку;
* стилі для окремого елементу;
* стилі користувача;
* локальний .css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера (наприклад Opera);
* стилі браузера;
* стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, те, в якій послідовності і для яких елементів застосовуються стилі. Таким чином використовується принцип каскадності, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена більш загальними стилями.

Звичайно використання CSS стилів має свої переваги та недоліки. До переваг можна віднести наступні якості:

* Інформація про стиль для цілого сайту або його частин може міститися в одному .css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;
* Різна інформація про стилі для різних типів користувачів: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на принтер, стиль для мобільних пристроїв;
* Сторінки зменшуються в об'ємі та стають більш структурованими, за рахунок того що інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з їх урахуванням;

Прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузери здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

Java, Java Applet та сценарії JavaScript

Java– об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Синтаксис мови багато в чому походить від C та C++. У офіційній реалізації, Java програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Sun Microsystems надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель С++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в С/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім, Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

Java вплинула на розвиток J++, що розроблялась компанією Microsoft. Роботу над J++ було зупинено через судовий позов компанії Sun Microsystems, оскільки ця мова програмування була модифікацією Java. Пізніше в новій платформі Microsoft.NET випустило J#, щоб полегшити міграцію програмістів J++ або Java на нову платформу. З часом нова мова програмування С#, стала основною мовою платформи, перейнявши багато чого з Java. J# востаннє включався в версію Microsoft Visual Studio 2005. Мова сценаріїв JavaScript має схожу із Java назву і синтаксис, але не пов'язана із Java.

Розробку проекту започаткував Джеймс Ґослінґ, сам проект мав назву «Green» (Зелений). Створення першої робочої версії, яка мала назву «Oak» зайняло 18 місяців. Оскільки виявилось, що ім'я Oak уже використовувалось іншою фірмою, то в результаті тривалих суперечок навколо назви нової мови з поміж ряду запропонованих було вибрано назву Java, у 1995р. мову було офіційно перейменовано.

Головним мотивом створення Java була потреба в мові програмування, яка б не залежала від платформи (тобто від архітектури) і яку можна було б використовувати для створення програмного забезпечення, яке вбудовується в різноманітні побутові електронні прилади, такі як мобільні засоби зв'язку, пристрої дистанційного керування тощо.

Досить скоро майже всі найпопулярніші тогочасні браузери отримали можливість запускати «безпечні» для системи Java аплети всередині web-сторінок. У грудні 1998 р. Sun Microsystems випустила Java 2 (спершу під назвою J2SE 1.2), де було реалізовано декілька конфігурацій для різних типів платформ. Наприклад, J2EE призначалася для створення корпоративних застосунків, а значно урізана J2ME для приладів з обмеженими ресурсами, таких як мобільні телефони. У 2006 році в маркетингових цілях, Версії J2 було перейменовано у Java EE, Java ME та Java SE, відповідно.

13 листопада 2006 року Sun випустили більшу частину Java в якості вільного та відкритого програмного забезпечення згідно з умовами GNU General Public License (GPL). 8 травня 2007 корпорація закінчила процес, в результаті якого всі початкові коди Java були випущенні під GPL, за винятком невеликої частини коду, на який Sun не мала авторського права.

Період становлення Java збігся у часі з розквітом міжнародної інформаційної служби World Wide Web. Ця обставина відіграла вирішальну роль у майбутньому Java, оскільки Web теж вимагала платформо-незалежних програм. Як наслідок, були зміщені акценти в розробці Sun з побутової електроніки на програмування для Інтернет.

JavaScript– назва реалізації стандарту мови програмування ECMA-Script компанії Netscape, базується на принципах прототипного програмування. Найпоширеніше і найвідоміше застосування мови – написання сценаріїв для web-сторінок, але, також, використовується для впровадження сценаріїв керування об'єктами вбудованими в інші програми.

Незважаючи на схожість назв, JavaScript та мова програмування Java мають дуже мало спільного. Права на назву JavaScript належать компанії Sun Microsystems. Розроблена в компанії Netscape, на даний час підтримується більшістю браузерів. Текст програми включається безпосередньо в HTML-документ і інтерпретується самим браузером. Застосовується в основному для часткової автоматизації обробки і маніпуляції даними, які використовує сторінка.

В 1992 році компанія Nombas розпочала розробку вбудованої скриптової мови Cmm (Сі-мінус-мінус), яка, по замислу розробників, повинна була стати достатньо потужною, щоб замінити макроси, зберігаючи при цьому схожість з Сі, щоб програмістам було легко оволодіти нею. Головною відмінністю від Сі була робота з пам’яттю. В новій мові все управління пам’яттю здійснювалось автоматично: не було необхідності створювати буфери, оголошувати змінні, здійснювати перетворення типів. В іншому мови були доволі схожі між собою: зокрема, Cmm підтримував стандартні функції і оператори Сі. Cmm була перейменована в ScriptEase, оскільки початкова назва звучала надто негативно, а згадка про Сі «відлякувало» людей.

На основі цієї мови було створено CEnvi. В кінці листопада 1995 року Nombas розробили версію CEnvi, впроваджувану в web-сторінки. Сторінки, які можна було міняти за допомогою скриптової мови, отримали назву Espresso Pages – вони демонстрували використання скриптової мови для створення гри, перевірки користувацького вводу в форми і створення анімації.

Станом на 2006 рік, останній стандарт мови – JavaScript 1.7. Якщо спрощено, то ECMA-262 є стандартизованим варіантом мови JavaScript. Стандартом ECMA-357 визначено E4X, розширення мови для роботи з XML.

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам, – функції як об'єкти першого рівня, об'єкти як списки, каррінг – анонімні функції, замикання – що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою Сі має такі корінні відмінності:

* об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів;
* функції як об'єкти першого класу;
* обробка винятків;
* автоматичне приведення типів;
* автоматичне прибирання сміття;
* анонімні функції;

JavaScript містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, JavaScript містить набір вбудованих операцій, які, строго кажучи, не обов'язково є функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто в кінцевому рахунку успадкований від C), але спрощений порівняно з ним, щоб зробити мову сценаріїв легкою для вивчення. Так, приміром, декларація змінної не містить її типу, властивості також не мають типів, а декларація функції може стояти в тексті програми після неї.

Основні архітектурні особливості JavaScript: динамічна типізація, автоматичне керування пам’яттю, прототипне програмування, функції як об’єкти першого класу.

На JavaScript вплинуло багато мов, під час розробки основною ціллю було зробити мову схожою на Java, але при цьому легкою для використання не професійними програмістами. Мовою JavaScript не володіє будь-яка компанія чи організація, що відрізняє її від ряду мов програмування, що використовуються в web-розробках.

На даний час існує безліч інструментальних засобів для розробки HTML-документів. Розглянемо деякі з них.

* HTML Assistant

HTML Assistant призначений для автономного редагування HTML-сторінок. В пакет комерційної версії поставляється посібник користувача та курс навчання у друкованому вигляді. Крім того, HTML Assistant PRO має засоби пошуку файлів та URL, а також фільтри для конвертування HTML-файлів в звичайні документи. Цей редактор дозволяє відкрити вже існуючий HTML-документ та відредагувати його. Така властивість є і в HotMetal Pro, але на відміну від нього HTML Assistant не має засобів перевірки синтаксису HTML. Це зручно, якщо редагується вже існуюча сторінка і людина не турбується точними правилами HTML. Можна вводити будь-які теги однак, якщо ввести неправильні теги, сторінка буде неправильно відображатися. Крім того в HTML Assistant немає режиму попереднього перегляду сторінки, але передбачена можливість запускати будь-який браузер для перегляду створеного HTML-документа. HTML Assistant має довідкову систему. Дуже зручну можливість створювати елементи користувача представляє засіб User Tools. Подібно макромові, вона дозволяє вводити повторно текст натисненням мишки на потрібну кнопку.

Робота в HTML Assistant відрізняється простотою – користувач вибирає потрібний тег на панелі інструментів, а редактор автоматично його вставляє. Текстові стилі теж визначаються нескладно – потрібно виділити текст, який слід відобразити особливим чином (курсив, напівжирний), а потім натиснути мишкою на кнопку Styles. Якщо доводиться часто вводити які-небудь послідовності тегів, HTML Assistant дозволить визначити відповідні елементи інструментальної панелі в діалоговому вікні User Tools.

HTML Assistant представляє собою дуже зручний засіб, якщо необхідно відредагувати HTML-сторінку. Для вводу HTML-тегів потрібно натиснути рекордно малу кількість клавіш. Проте цей же редактор не в змозі допомогти користувачеві перевірити правильність коду та структури HTML-документа.

Найбільшим недоліком HTML Assistant є неспроможність роботи з файлами розміром більше 32 Кб. Це обмеження не має значення при створенні невеликих домашніх сторінок, але при створенні складних HTML-документів стає серйозною проблемою.

* HTMLed Pro

Основним призначенням HTMLed Pro, як і попереднього редактора, є простота вводу HTML-тегів. HTMLed Pro перевіряє структуру і формат HTML-документів, так що він корисний для швидкого виправлення некоректно написаних web-сторінок. В HTMLed Pro дуже просто добавляти спеціальні символи за допомогою плаваючої інструментальної панелі, яка зручна при редагуванні неангломовних документів. Цей редактор може коректно експортувати файли для UNIX, позначаючи кінці рядків потрібними кодами.

Найбільш привабливими рисами редактора є плаваючі панелі інструментів та швидкий доступ до розширеного набору символів, які використовуються в більшості європейських мовах. Це найпростіший засіб вводу спеціальних символів серед більшості HTML-редакторів. На відміну від HTMLed, версія HTMLed PRO може працювати з файлами більше 32 Кб.

Програма також забезпечується технічною підтримкою, може виділяти кольором елементи синтаксису HTML, імпортувати файли в формат RTF, а також підтримувати файли розташовані на віддаленому сервері, зв'язок з якими здійснюється за протоколом TCP/IP. Інструментальні панелі HTMLed Pro користувач може налагодити за власним смаком та вподобаннями. Ще одна цікава особливість даного HTML-редактора – можливість видалити із виділеного тексту HTML-теги.

HTMLed Pro має довідкову систему, в яку вкладена інформація по синтаксису HTML. Користувач HTMLed Pro забезпечується доступом до он-лайнової служби технічної підтримки фірми Internet Software Technologies . Після реєстрації своєї копії, передбачена можливість отримання документації для користувача та учбовий курс HTML у друкованому вигляді.

* Macromedia DreamWeaver

Перше знайомство із редактором складає не дуже гарне враження. Інтерфейс не найкращий, і здається, що Allaire Homesite перевищує його в декілька разів. Але ж перші враження помилкові і виявляються далекими від дійсності.

DreamWeaver – потужний редактор з дещо нетрадиційним інтерфейсом. Головна зручність даного html-редактора виявляється в тому, що без особливих зусиль можна знайти все те, що потрібно для роботи: Java Applet, ActiveX компоненту, сценарії, область вводу тексту, прокрутки та ін. При створенні не задається жодного питання.

Список властивостей об'єкту автоматично відображається у верхньому віконці, де вони можуть бути легко змінені. Іншим цінним пунктом меню є пункт «Змінити». Там є все те, що можна змінювати. Однією із зручностей даного редактора є підпункт команди «Змінити» - «Властивості сторінки. Інші пункти меню не менш цікаві. Все чітко та зрозуміло. Можна змінювати кількість стовпців і рядків. Дуже гарна підтримка шарів та фреймів, присутній менеджер проектів та підтримка [http://FTP](http://ftp/). Але найкращим є вікно «Поведінка». Якщо потрібно добавити якусь дію при завантаженні сторінки, то натисніть на кнопку, виберіть меню, і ось, – воно вже на вашій сторінці.

* PageMill

Даний редактор, розроблений відомою компанією Adobe, що дуже добре зарекомендувала себе на ринку програмних продуктів. Це один з перших найбільш широко розроблених WYSIWING-HTML-редакторів. Перше, що вражає, це звісно ж відсутність підтримки кирилиці. Хоча потрібно відзначити, що редактор має у своєму розпорядженні чималий набір інструментів. Приємне враження справляють зручні редактори таблиць і кадрів, які цілком конкурують з аналогічними редакторами Visual Page. Новинкою у версії 3.0 є диспетчер сторінок, що має важливе значення, що полегшує керування і завантаження сторінок. Варто відзначити, що в кожний пакет входить графічний редактор Photoshop LE-Edition 3.05.

PageMill дозволяє швидко оформити прості розмітки HTML-Layout кадрами і таблицями. За допомогою відточеного інструмента можна легко і швидко підготувати графічні зображення. На жаль, компанія Adobe проігнорувала стандарт Dynamic HTML, тому динамічні елементи, Stylesheets, рівні, сценарії і навіть аплети Java – все це відсутнє. Жодний редактор цього класу не пропонує такого бідного списку властивостей.

Якщо завантажити в PageMill чужий HTML-документ, то програма без вагань змінить вихідний текст. При цьому програмне забезпечення фірми Adobe також використовує «улюблені» позначки-дескриптори з фірмовою рекламою. У цілому ж непоганий редактор, підтримує практично все, але виглядає на фоні FrontPage значно простіше.

* FrontPage 98

Програма FrontPage 98 від Microsoft орієнтована на фахових розробників web-сторінок, які складають і оформлюють цілі web-вузли. Після запуску програми спочатку з'являється FrontPage Explorer, що керує всіма файлами і каталогами, які відносяться до одного web-вузла. Питання побудови серверу вирішено просто і зразково: сторінки можна з'єднувати одну з одною за допомогою механізму Drag&Drop. Робота оболонки в значній мірі орієнтована на продукти Office 97. З погляду ергономіки в web-редакторі добре вирішене питання переходу між трьома видами представлення: «Звичайний» для опрацювання web-сторінки в режимі WYSUIWYG, «HTML» для роботи з кодами і «Попередній перегляд» у формі моделі роботи браузера. Ці види представлення існують у всіх web-редакторах і активізуються, як правило, через відповідні меню. Microsoft, навпаки, зупинилася на технології вкладок: біля крайнього нижнього краю вікна користувач знайде піктограми з позначенням виду зображення.

Stylesheets підтримується власним редактором, щоправда подає не зовсім те, що хотіли б бачити web-дизайнери. За допомогою діалогових вікон можна змінити всі властивості, що відповідають даному стилю.

З отриманих даних програма складає відповідну команду, яка потім заноситься у редактор подібно записній книжці. Програмний код стилю повинен бути записаний у рядок без прогалин, що дуже ускладнює наступну обробку. Для JavaScript і VBScript існують асистенти, за допомогою можна створювати прості сценарії.

* Notepad++

Повністю безкоштовна програма розроблена спеціально для програмістів всіх рівнів. Працює на базі операційної системи Windows. Підтримує величезну кількість розширень файлів та мов програмування, дозволяє налаштовувати відображення тексту для кожного з них. Забезпечує можливість роботи з декількома файлами одночасно. Зручною є функція запису макросів: програма буде запам'ятовувати кожен написаний рядок, дозволяючи згодом натисненням однієї кнопки вставити в документ цей макрос. Notepad++ підтримує чимало способів кодування тексту, зокрема ANSI, UTF-8 та UTF-8 (без BOM), UCS-2 Little Endian та UCS-2 Big Endian а також кодування багатьох мов, наприклад арабської, балтійської, кирилиці, китайської та ін. А ще дозволяє швидко перекодувати набраний текст у інший. Програма має дуже добре розвинене меню пошуку, в якому можна знайти все – від звичайного пошуку, функцій "перейти до" та ін. до пошуку у файлах, пошуку у попередньо закритих файлах та управління закладками. Це лише частина усіх можливостей програми. Якихось суттєвих недоліків програма не має. Варто також відзначити, що Notepad++ підтримує багато мов, зокрема й українську.

Для розробки електронного посібника з дисципліни «Комп’ютерна логіка» я використав: HTML, каскадні таблиці стилів CSS, JavaScript та Java Applets. Головним інструментом розробки був Notepad++ 5.9.2. Додатково використовував Adobe Photoshop CS5.

* 1. **Аналіз технічного завдання**

Завданням дипломного проекту є розробка та написання електронного посібника з дисципліни «Комп’ютерна логіка». Під час виконання завдання буде використано: HTML, CSS-стилі, JavaScript та Java Applets. Розробка дизайну здійснюватиметься за допомогою Adobe Photoshop CS4, MS Paint та матеріалів із мережі Інтернет. Електронний посібник складатиметься з титульної сторінки, домашньої сторінки, лекцій, лабораторних та практичних робіт та тестів для самоперевірки по темах.

* + 1. **Найменування та область застосування**

Темою дипломного проекту є «Розробка електронного посібника з дисципліни «Комп’ютерна логіка».

Метою даного дипломного проекту є розробка електронного посібника з дисципліни «Комп’ютерна логіка», універсального посібника з викладеним матеріалом, тестами для самоперевірки, лабораторними та практичними роботами. Даний електронний посібник повинен відповідати стандартам існуючих аналогічних електронних посібників в мережі Internet. Сторінки посібника повинні бути невеликими, для швидкого завантаження та містити лише необхідну для користувачів інформацію, бути приємним на вигляд та легким для навігації.

Об’єктом використання даної розробки є будь-який користувач ПК.

* + 1. **Призначення розробки**

Електронний посібник призначений для оптимізації та спрощення процесу навчання, а також з метою впровадження новітніх технологій в процес навчання, підвищення доступності навчального матеріалу для студентів не лише ТК ТНТУ але й інших навчальних закладів.

* + 1. **Вимоги до програмного забезпечення**

Перед створенням електронного посібника, необхідно встановити на нашому комп’ютері відповідне програмне забезпечення, яке буде необхідне нам у майбутньому, а саме:

* Notepad++ 5.9.2;
* Google Chrome 12.0;
* Opera 11.11;
  + 1. **Вимоги до програмної документації**

Для розробки даного дипломного проекту викладач з дисципліни «Комп’ютерна логіка» надала матеріал для лекцій, тестів для самоперевірки та лабораторних і практичних робіт. На основі цього матеріалу було розроблено основне наповнення сайту.

Після закінчення розробки електронного посібника необхідно підготувати наступну документацію:

* опис основних браузерів, які дозволяють переглядати електронний посібник;
* проблеми, що можуть виникнути при завантаженні електронного посібника та їх усунення;
* паспорт електронного посібника із внесеними у нього наступними даними:
  + назва електронного посібника та адреса в Інтернеті (якщо є);
  + мінімальні вимоги до апаратного і програмного забезпечення;
  + роздільна здатність моніторів, під який оптимізовано посібник;
  + оптимальна роздільна здатність монітора для перегляду;
  + основний браузер, за допомогою якого буде переглядатися електронний посібник;
  + середній час завантаження сторінок електронного посібника;
  + час завантаження головної (титульної) сторінки;
  + об’єм електронного посібника (в Мб);
  + число сторінок, файлів, гіперпосилань (зовнішніх та внутрішніх) електронного посібника.
    1. **Стадії та етапи розробки електронного посібника**
* Розробка дипломного проекту відбуватиметься ступенево. Перш за все створюється макет майбутнього посібника: вигляд, фон, кольорова палітра, оптимальний шрифт, його розмір. Розробляється структура посібника.
* Створення статичного оригінального макету. Створюється графічний оригінальний макет, над яким потрібно тривалий час попрацювати.
* Написання програмного коду. HTML-код самого посібника, розробка сценаріїв Java.
* Збір посібника, розробка титульної сторінки та (за бажанням) розміщення його в Інтернеті. Весь графічний, анімаційний, звуковий та програмний матеріал компілюється в готовий продукт: мультимедійний ресурс посібника.
* Тестування та доробка. Готовий електронний посібник тестується на предмет помилок web-програмування і невідповідності технічному завданню.
  + 1. **Порядок контролю та прийому електронного посібника**

Порядок здачі-приймання готового електронного посібника принципово нічим не відрізняється від здачі-прийому звичайного web-сайту. При чому, чим ретельніше обумовлені всі питання на етапі узгодження технічного завдання на створення електронного посібника або його ре-дизайн, тим менша ймовірність того, що при прийманні готового сайту у замовника можуть виникнути які-небудь питання.

Як відомо, вдосконаленню немає меж. Робота над електронним посібником може тривати нескінченно довго. Будь-який web-продукт можна постійно вдосконалювати, і в цьому немає нічого незвичайного.

При прийманні готового сайту замовники повинні найперше звернути увагу на такі параметри:

* обсяг текстової інформації в друкованих знаках з пробілами, кількість;
* орієнтовна кількість сторінок формату А-4;
* обсяг робіт зі створення електронного посібника або його ре-дизайну, які визначаються відповідними договорами;
* внесення змін до договору здійснюється відповідно до законодавства.

Щоб максимально виключити можливість невідповідності готового сайту естетичним очікуванням замовника, останній зобов'язаний завчасно надати в розпорядження web-дизайнера контент сайту (текст сайту, таблиці сайту і графіку сайту).

При цьому слід враховувати наступне: перед здачею сайту Замовникові, виконується його повне тестування, що включає тестування роботи сайту в різних браузерах, тестування посилань (перевірка на «биті посилання»), тестування сайту під навантаженням та інші перевірки.

**2. РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ**

* 1. **Типи електронних посібників та оцінка їх якостей**

Електронний посібник – версія навчального посібника в електронному (цифровому) вигляді. Такі посібники можна читати за допомогою комп'ютерів, мобільних телефонів чи спеціалізованих пристроїв.

В англомовних країнах для позначення цифрових версій [книг](http://uk.wikipedia.org/wiki/%D0%9A%D0%BD%D0%B8%D0%B3%D0%B0) та посібників, для зберігання яких використовують [відкриті](http://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D0%B8%D0%B9_%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82) формати, засновані на простих [текстових файлах](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D0%B8%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB), застосовують також термін «електронний текст». Електронними текстами називають також навчальні електронні посібники, до складу яких, окрім тексту, належать растрові зображення, приклади даних, вправи, спеціально написані коментарі і відповіді на можливі запитання.

Одним із технічних споживачів електронних книг та посібників є Електронна бібліотека, яка надає користувачу (в основному Інтернету) екземпляр даної книги у певному форматі.

Кожна інструментальна розробка має свої недоліки та переваги. Розглянемо переваги розробленого електронного посібника з дисципліни «Комп’ютерна логіка» та його недоліки, які потребують усунення або удосконалення.

Основною перевагою електронного посібника є широкий спектр його застосування.

До переваг відносяться такі його риси:

* зручність при роботі з матеріалом за рахунок можливості застосування розвитої системи гіперпосилань;
* можливість розміщення такого посібника в мережі Інтернет;
* наявність великої кількості програмних засобів для перегляду HTML-документів;
* наявність головного інтерфейсу;
* можливість оновлення новими розділами та вносити в нього необхідні зміни;
* можливість використання при самостійному освоєнні матеріалу;
* простота використання;
* компактність;
* низькі системні вимоги (досить Windows XP);
* наявність гіперпосилань як на внутрішні так і на зовнішні ресурси;
* невеликий обсяг;
* можливість відображати анімаційні малюнки, мультимедійні кліпи і відтворювати аудіо-книжки та mp3-файли.

До недоліків можна віднести:

* екрани комп’ютерів втомлюють очі;
* висока вартість, еквівалентна 50-100 друкованим книгам;
* відсутність деяких мультимедійних компонентів – звуку, відео тощо;
* відсутність внутрішньої пошукової системи та довідки.

Вказані недоліки характерні для першої редакції розробленого електронного посібника, однак в подальшій розробці передбачається використання мультимедійних засобів та удосконалення.

Електронний посібник повинен суміщати в собі функції підручника, викладача, довідниково-інформаційного посібника, консультанта, тренажера а також засобу для самоперевірки по пройденому матеріалу.

Використання системного підходу до розробки електронних навчальних посібників дозволяє зробити серйозний крок на шляху переходу від пізнавальної до прагматичної моделі освіти і сприяє вирішенню проблем створення посібників нового покоління, що дають можливість:

* збільшити кількість користувачів;
* підвищити наочність представлення матеріалу;
* використовувати електронний посібник тривалий час;
* звести до мінімуму витрати на пошук і підбір літератури;
* здійснювати контроль отриманих знань.

Як інструментальний засіб доцільно використовувати електронний посібник, що задовольняє наступним вимогам:

* простота використання електронного посібника:
* вбудовані мультимедійні можливості;
* компактність;
* низькі системні вимоги (досить Windows 95/98/2000/XP/NT).

Для створення інтерактивних електронних посібників необхідні знання, крім знань за курсом, необхідно мати спеціальні навички в області застосування Інтернет-технологій.

При використанні для розробки навчальних матеріалів кожного з великої кількості наявних інструментальних засобів створення і редагування HTML-документів, наприклад, таких як Netscape Composer чи Microsoft FrontPage, вимагаються базові знання про гіпертекст і визначені навички в області програмування.

Крім того, для надання web-документам інтерактивних властивостей, що дозволяють оперативно реагувати на інформацію, введену користувачем, необхідні будуть ще більш глибокі знання Інтернет-технологій, пов'язані з необхідністю написання програм, для обробки отриманої інформації з використанням Java, JavaScript чи CGI-додатків.

В цьому разі необхідна спільна робота колективу фахівців різного профілю. Ці обставини істотно підвищують трудомісткість створення навчальних web-посібників і, деякою мірою, стримують процес розвитку такої форми освіти.

До сучасних інформаційних технологій, які використовують в навчальному процесі, відносять електронні та гібридні бібліотеки, електронні посібники, довідково-пошукові системи Інтернет та ін.

Як правило, у даний час бурхливого розвитку комп'ютерних технологій електронні навчальні чи довідково-пошукові системи розробляються з використанням гіпертекстових і мультимедійних технологій.

Такі системи називають інтерактивними навчальними web-матеріалами, та можуть знайти і знаходять широке застосування як в звичайних формах навчання, так і в достатньо новій формі навчання, як самоосвіта та під час самостійної роботи студентів.

Тому створення навчальних web-матеріалів пред’являє дуже високі вимоги до кваліфікації розробника в області практичного використання сучасних інформаційних технологій, тому що, окрім знань предмета, необхідно мати спеціальні навички в області застосування Інтернет-технологій, чи вдаватися до допомоги відповідного фахівця чи навіть спільної роботи колективу фахівців різного профілю.

Ці обставини істотно підвищують трудомісткість створення навчальних web-посібників і, деякою мірою, стримують процес розвитку такої форми освіти, як застосування навчальних web-матеріалів.

Електронний посібник– це універсальний методичний посібник, який містить широке коло питань різних навчальних дисциплін, викладених в стислій формі та призначений для використання в процесі навчання.

Аналіз літературних джерел показав, що більшість перших електронних навчальних видань являли собою електронні копії друкованих видань і, здебільшого, не враховували комп’ютерних можливостей подачі матеріалу.

Однак останнім часом значна увага приділяється розробці комп'ютерних посібників, роботу над якими можна розглядати як спробу зробити серйозний крок до вирішення проблеми створення посібників нового покоління.

Для розв'язання цієї проблеми запропоновано використовувати системний підхід при створенні електронних посібників. Системний підхід розглядає об'єкт як систему, що складається з безлічі взаємозалежних елементів, що утворять певну цілісність системних властивостей.

Системний підхід дозволяє не орієнтувати навчальний посібник на конкретну групу користувачів, але створювати його таким чином, щоб ним міг скористатися практично кожен. У залежності від потреб і вже наявних знань, користувач сам обирає матеріал для вивчення, його обсяг, технологію навчання.

Створення різних моделей представлення знань, що в одному випадку представляють об’єкти, характерні для логічного мислення, а в іншому – образи-картини, з якими оперує образне мислення, дає можливість оптимізувати процес навчання. Включення системи тестів на початковому етапі роботи з навчальним посібником дозволяє ідентифікувати особистісні якості студента, а потім оптимізувати налаштування на нього і рекомендувати відповідну методику навчання.

Контроль знань після вивчення кожного розділу може здійснюватися різними способами (за допомогою тестів, контрольних питань і т. д.).  
У залежності від того, наскільки користувач засвоїв матеріал, можна при необхідності повторити вивчений розділ, чи відкоригувати, змінити методику навчання. При низьких результатах можливе з'ясування думки самого користувача про причини поганого засвоєння матеріалу.

Застосування системного підходу при створенні електронного посібника дозволяє розширити область застосування електронних посібників і коло потенційних користувачів. За рахунок включення в посібник блоку нових розробок і проблемно-орієнтованих баз даних, його можна використовувати більш тривалий час і тим самим підвищити його якість, довговічність і точність даних наведених у ньому.

Використання мультимедіа, аудіо та відео-компонентів підвищує наочність представлення матеріалу, а також дає можливість використовувати його людям, що мають різні патології (порушення слуху, зору і т.п.). За рахунок цього можливо різке збільшення кількості користувачів і ефективності використання електронного посібника. Включення перерахованих компонентів в електронний посібник дозволяє перейти від пізнавальної моделі освіти до прагматичної, у якій студент стає активним об'єктом освіти.

Таким чином, використання системного підходу до розробки електронних навчальних посібників дозволяє зробити серйозний крок на шляху переходу від пізнавальної до прагматичної моделі освіти і сприяє рішенню проблем створення посібників нового покоління, що дають можливість збільшити кількість користувачів, підвищити наочність представлення матеріалу, використовувати електронний посібник тривалий час, звести до мінімуму витрати на пошук і підбір літератури, здійснювати контроль отриманих знань та ін.

Розробка навчальних курсів у середовищі мультимедіа є тривалим і дорогим процесом, тому важливо добре уявляти собі всі основні етапи створення комп'ютерного навчального курсу і прийнятні для кожного етапу розробки рішення.

На першому етапі здійснюється вибір навчального курсу для представлення в середовищі мультимедіа. Тут мають бути виявлені вже існуючі курси з даної дисципліни, визначені передбачувані витрати і час, необхідні для створення курсу, а також його можливий тираж і аудиторія, якій адресований даний курс.

На підготовчому етапі передбачається написання тексту курсу, підбір ілюстративного і довідкового матеріалу, створення ескізів інтерфейсу і сценарію навчальної програми, а також сценаріїв окремих блоків (анімаційних фрагментів, відео-фрагментів, програм, що реалізують комп'ютерне моделювання, блоків перевірки знань і т. п.).

При роботі з текстом навчальної дисципліни необхідно виконати його структуризацію з визначенням точного переліку всіх необхідних тем, що повинні бути викладені в даній дисципліні, розподілом на розділи, параграфи і т. д.

Кожен розділ і весь навчальний курс у цілому досягнуть мети, якщо від початку зазначено, які знання і навички студент повинен отримати.

Паралельно з написанням тексту курсу проводиться робота над сценарієм складової мультимедіа-курсу. Він має містити докладний перелік відповідних компонентів і тем дисципліни, що вивчається, а також попередній опис її структури, що буде реалізовуватися. Сюди відносяться: опис анімаційних, аудіо і відео фрагментів, ілюстрацій, і т. п. Написання сценарію проводиться з урахуванням можливостей обраного програмного забезпечення і наявних вихідних матеріалів.

Повний сценарій курсу має на меті використання звичайного тексту і гіпертексту з посиланнями на зв'язані теми, розділи чи поняття, на зображення, звуки, відео фрагменти, використання табличної інформації, ілюстративного матеріалу (графіків, схем, малюнків), анімаційних малюнків, фотоматеріалів, аудіо і відео фрагментів, комп'ютерних моделей.

На основному етапі виконуються роботи з безпосереднього створення мультимедіа-курсу. Зміст при цьому повинен домінувати над формою його викладу.

Форма представлення матеріалу повинна бути якомога стислішою. Сторінка не повинна містити зайвої інформації (графічної чи текстової), що могла б відвернути увагу читача.

Фон повинен бути монотонним, причому необов'язково білим. Не варто використовувати темний фон і світлий шрифт – це буде стомлювати очі читача.

При підборі гарнітури шрифту варто виходити з того, що читабельність тексту, написаного гарнітурою без зарубок, вища, ніж тексту, написаного гарнітурою із зарубками. При цьому варто цілком відмовитися від використання дрібних розмірів шрифтових гарнітур. При включенні в програму графічних зображень потрібно враховувати, що сторінки будуть проглядатися в системах з різним графічним розширенням і глибиною кольору, і орієнтуватися на апаратні засоби, які доступні більшості потенційних користувачів навчальної програми. Використання графічних форматів, що підтримують скомпресоване зображення (GIF, JPEG і т.п.), дозволить скоротити загальний обсяг навчальної програми.

Після проведення завершального етапу відбувається тестування і доробка курсу. Мультимедіа-курс, що пройшов тестування повинен бути зареєстрований як інтелектуальна власність. При цьому необхідно врахувати авторські права колективу розробників, що приймали участь у створенні мультимедіа-курсу.

Корисним є також надання супроводу курсу після його тиражування (зрозуміло, тільки для «легальних» користувачів): оперативне усунення можливих помилок, постачання нових додаткових модулів, поновлення довідкової інформації і т. д.

Необхідно відзначити, що при створенні текстів електронного посібника з дисципліни «Комп’ютерна логіка» необхідно дотримуватися правил розробки електронних сторінок згідно «Закону про електронні документи та електронний документообіг», який встановлює основні організаційно-правові засади електронного документообігу та використання електронних документів.

* 1. **Огляд та структура електронного посібника**

вступне

Будова всіх сторінок окрім титульної є схожою, відрізняється лише вміст. Після завантаження електронного посібника з дисципліни «Комп’ютерна логіка» на екрані з’явиться титульна сторінка, зображена на рис. 2.2.1. Титульна сторінка оснащена гіперпосиланням у вигляді зображення з написом «Комп’ютерна логіка» натиснувши на яке можна перейти на Домашню сторінку.



Рисунок 2.2.1 - Титульна сторінка.

Всі сторінки посібника складаються з декількох частин:

* + Заголовна частина;
  + Ліва панель;
  + Основна частина;
  + Нижня панель.

Заголовна частина (рис. 2.2.2) завжди незмінна і містить назву посібника та годинник. В нижній панелі (рис. 2.2.3) виведена інформація про Автора. Ліва панель (рис. 2.2.4) містить: меню розділів, календар та список гіперпосилань на ресурси Інтернет, які можуть бути корисними.



Рисунок 2.2.2 - Заголовна частина електронного посібника.



Рисунок 2.2.3 - Нижня панель: інформація про Автора.

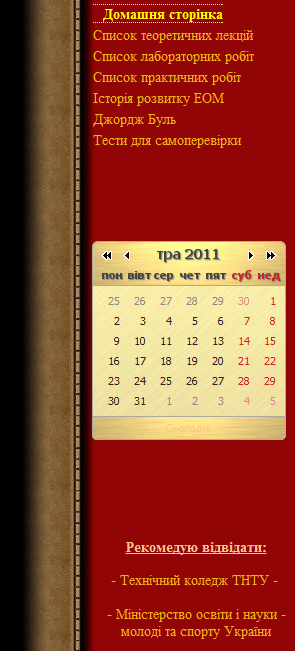


Рисунок 2.2.4 - Ліва панель: меню, календар та список корисних посилань.

В основній частині розташований вміст того чи іншого розділу: чи то список тем, лабораторних, практичних робіт, тестових завдань, чи безпосередньо самі лекції, тести, лабораторні чи практичні роботи.

Лекції представлені у вигляді PDF-вкладок, що надає їм додаткові можливості: можливість збільшити чи зменшити масштаб відображення, зберегти лекцію на комп’ютері та ін. Зразок лекції зображено на рис. 2.2.5.

Окремими розділами виділено біографію Джорджа Буля – основоположника алгебри-логіки та Історію розвитку ЕОМ.

Тести для самоперевірки оцінюються в 4 системах оцінювання: звичайній п’ятибальній, дванадцятибальній, відсотковій та стобальній. Оцінки за один і той самий тест можуть відрізнятись через візницю в самих системах оцінювання. Один із таких тестів зображений на рис. 2.2.6.

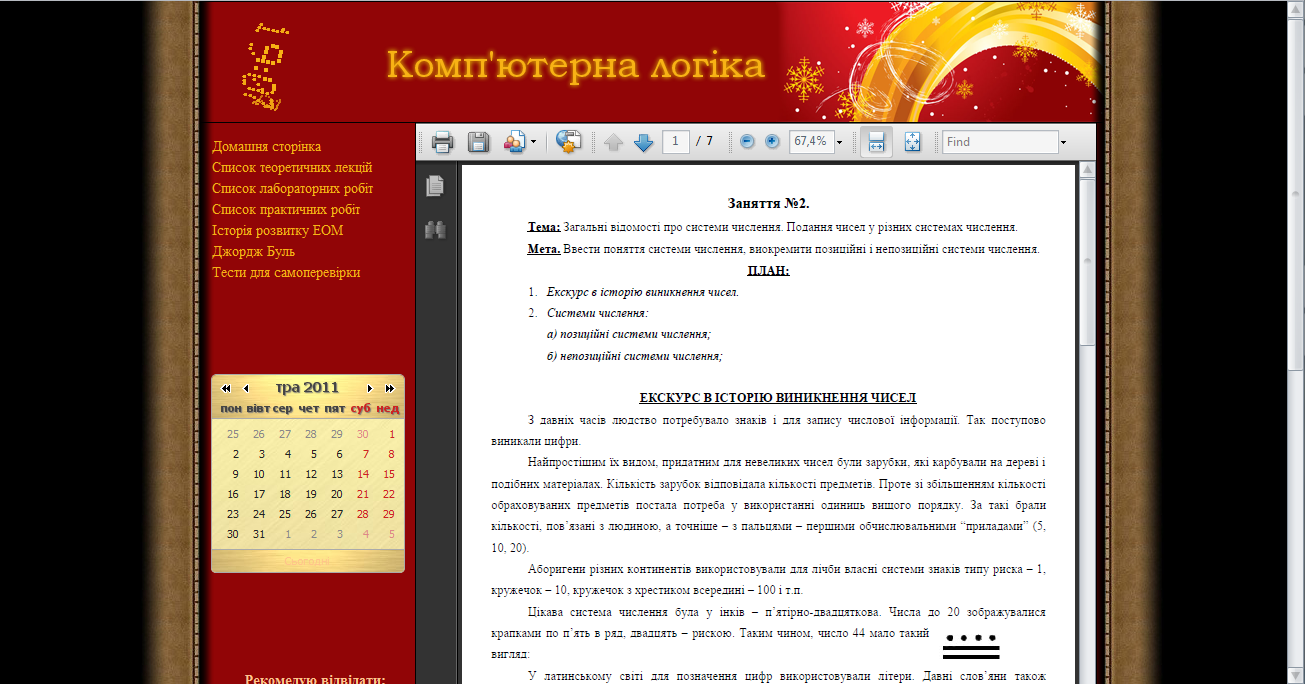


Рисунок 2.2.5 - Зразок лекції.

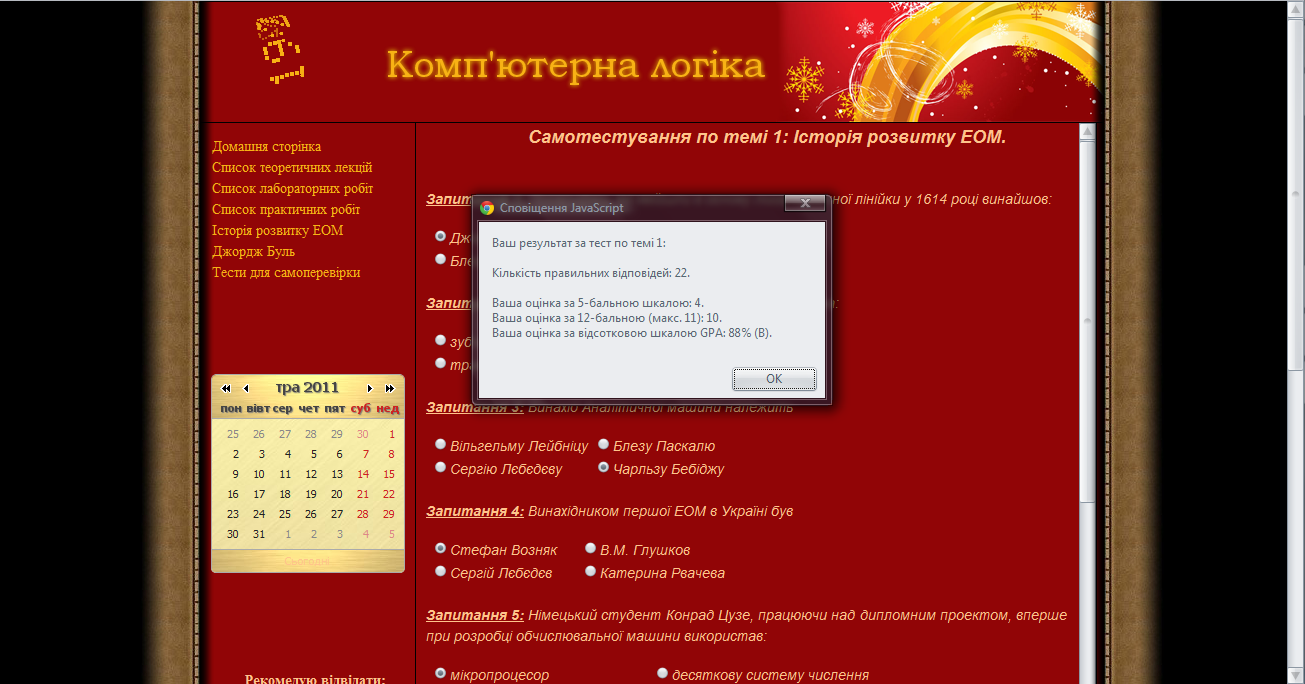


Рисунок 2.2.6 – Зразок тестів.

Перехід між розділами електронного посібника з дисципліни «Комп’ютерна логіка» здійснюється за допомогою меню розділів з лівого боку, зображеного на рис. 2.2.7.

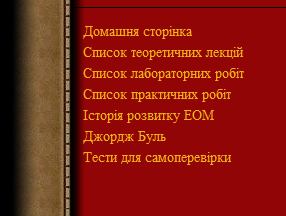


Рисунок 2.2.7 - Меню розділів.

Електронний посібник також оснащено потужним календарем (рис. 2.2.8) та 3D годинним (рис. 2.2.9). Годинник постійно і безперервно обертається по осях X Y та Z. Щоб зупинити/поновити хід годинника необхідно клацнути лівою клавішею миші по ньому.

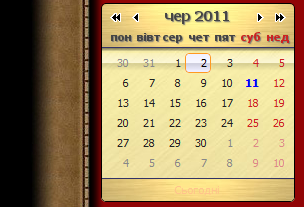


Рисунок 2.2.8 – Календар.



Рисунок 2.2.9 - 3D Годинник.

* 1. **Розробка структури електронного посібника**

Внаслідок загальної комп’ютеризації суспільства, разом з традиційними навчальними посібниками почали використовуватися, так звані, електронні підручники, зручні і для домашнього навчання, і для використання в освітніх установах. Електронні підручники істотно підвищують якість візуальної інформації, вона стає яскравішою, барвистою, а як відомо, чим цікавіше представлений матеріал, тим цікавіше його вивчати. Тому можливості електронних підручників не обмежуються тільки викладом і демонстрацією матеріалу, вони покликані зацікавити учня.

Слідуючи сучасним тенденціям, мною було розроблено електронний підручник з дисципліни «Комп’ютерна логіка».

Мета дипломного проекту **–** створити електронний підручник, який можна було б використовувати на уроках «Комп’ютерної логіки» і не тільки, а також для самостійного навчання.

Робота над підручником проводилася у декілька етапів.

На підготовчому етапі створення електронного підручнику було визначено бібліографію по темі роботи, здійснювався аналіз джерел, збір та узагальнення матеріалу.

Перший етап розробки був присвячений розробці структури підручника, тобто було проведено розбиття матеріалу на розділи, що складаються з модулів, мінімальних за об'ємом, але замкнутих за змістом, а також складено дидактичне наповнення підручника**.**

Другий етап**.** Розробка дизайну, створення основних елементів управління**,** створення анімаційних роликів, компоновка, зв'язок елементів за допомогоювбудованої мови програмування.

Третій етап**.** Верстка НТML-сторінок. В результаті створюється працюючий електронний підручник, який володіє властивостями, що роблять його необхідним для учнів, корисним для самостійних занять і зручним для викладачів.

Четвертий етап**.** Після загальної перевірки усіх HTML-сторінок було розроблено файл автоматичного запуску у форматі \*.bat.

Тема підручника «Комп’ютерна логіка» входить в курс вищих навчальних закладів. Метою профільного курсу «Комп’ютерна логіка» є формування важливої складової інформаційної культури учнів – знань та вмінь, необхідних для повноцінного та ефективного знань в галузі розробки програмних продуктів.

Структура підручника досить проста. Він містить головну сторінку з застосуванням JavaScript для випадкового вибору наповнення та зображення-посилання для переходу безпосередньо до самого посібника; меню, що складається з 7 пунктів, а також годинник та календар.

Сторінка другого ряду містить:

* Лекції, у яких наведено теоретичні матеріали. Такі сторінки мають фреймову структуру і для зручності користування розбиті на дрібні підтеми, відповідно до навчального змісту матеріалу.
* Лабораторні та практичні роботи, призначені для самостійного засвоєння та практичного відпрацювання матеріалу. Вони являють собою набір інструкцій для чіткого і правильного виконання робіт.
* Тестові завдання, пристосовані для самоконтролю. В підручнику розроблено інтерактивні тести на JavaScript. Учню пропонується обрати одну вірну відповідь з кількох запропонованих. Тести такого роду є гарним допоміжним засобом вчителеві для здійснення поточного контролю. Питання розташовані по ступеню зростання складності, кількість різних варіантів відповідей в кожному питанні – 3-5. Відповіді рівноцінні, щоб звести вірогідність вгадування до мінімуму.

Фінальним завершенням основи проекту є переробка створеного мовою HTML підручнику до зручного у застосуванні формату \*.exe чи \*.bat. Відкривши фінальний файл автозапуску, користувач побачить перед собою віконце, за допомогою якого зможе проглянути програму або просто вийти.

Результатом роботи є яскравий електронний підручник, який буде необхідним для самостійної роботи учнів, корисним на практичних заняттях та зручним для викладача.

Процес розробки мультимедійного електронного посібника з дисципліни «Комп’ютерна логіка» передбачає такі етапи (рис. 10):

* етап визначення наповнення електронного посібника.

Цей етап включає відбір і структурування змісту навчального матеріалу й відбір матеріалу, характерного для лекційного курсу дисципліни «Комп’ютерна логіка».

* етап планування та розробки дизайну, написання CSS коду.

На цьому етапі проводиться планування та розробка та написання CSS коду візуального оформлення електронного посібника: окремих елементів та загальний вигляд, підбір додаткового оснащення, як наприклад, годинника чи DHTML календаря.

* верстка HTML-сторінок з лекціями.

Перетворення документів в \*.pdf формат та приєднання до основних сторінок з лекціями, так званої «бази».

* розробка та написання JavaScript.

Цей етап включає: розробку, написання та приєднання JavaScript скриптів до сторінок. В процесі розробки та написання скриптів проводиться одночасно і їх тестування, перевірка на можливі помилки в коді, можливість спрощення коду для економії часу, необхідного на виконання скрипта.

* етап написання кодів тестової системи.

На даному етапі в електронний посібник додається система перевірки набутих знань. Розробляється вигляд тестів, додаються запитання із можливими варіантами відповідей. Весь код адаптується до раніше написаного JavaScript скрипту і тестується на наявність орфографічних чи скрипкових помилок.

* верстка HTML-сторінок лабораторних та практичних робіт.

Перенесення лабораторних та практичних робіт із документа в HTML-код, максимально зберігаючи форматування при цьому.

* загальне тестування роботи електронного посібника.

Кінцевим етапом розробки електронного посібника є його загальне тестування на наявність будь-яких орфографічних помилок, помилок в оформленні, наявність неробочих гіперпосилань, невідповідність заданим параметрам та відстеження середнього часу завантаження сторінок. В процесі перевірки знайдені помилки виправляються.

Візуальна структура етапів розробки електронного посібника з дисципліни «Комп’ютерна логіка» показано на рис. 2.3.1.

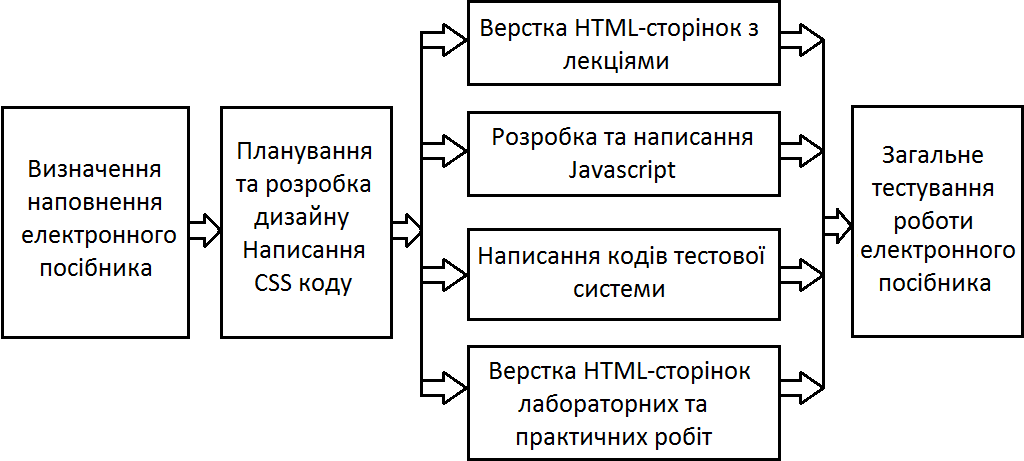


Рисунок 2.3.1 – Схема етапів розробки електронного посібника.

* 1. **Написання функцій**

У розробленому мною електронному підручнику використано декілька скриптів, опис яких наведено нижче.

Функція при завантаженні титульної сторінки

<script language="JavaScript" type="text/javascript">

function justLoaded() {

var rnd = Math.round(Math.random()\*4)+1;

document.getElementById("img").style.background = "url ('src/img/main\_img\_fon"+rnd+".png') 100% center no-repeat #220001";

document.getElementById("text").style.top = innerHeight/2-40;

if (innerWidth > 1024)

document.getElementById("text").style.left = "100px";

}

</script>

…

<body style="margin: 0px;" onLoad="justLoaded()">

Ця функція виконується при завантаженні титульної сторінки посібника і забезпечує випадковий вибір зображення із крилатим(и) висловом(ами). В папці "src/img" знаходиться 5 картинок, які можуть бути використані на титульній сторінці. Всі вони мають схожі імена – main\_img\_fon<number>.png.

Отже: змінна "rnd" і є ключовою в цій функції. Їй присвоюється випадкове число від 1 до 5 яке за допомогою Math.round() заокруглюється до цілого числа. Ця змінна далі і буде визначати, яке саме зображення із 5 наявних буде завантажене цього разу. Далі присвоюємо стиль елементу "img" в якому вказуємо шлях до цього зображення із числом отриманим в "rnd" та іншими параметрами стилю.

Також функція "justLoaded()" виконує вирівнювання зображення-гіперпосилання по висоті, та, якщо ширина робочого поля браузера більша, ніж 1024 пікселів, то і по ширині, для більш естетичного вигляду.

Функція вирівнювання сторінки по центру екрана

Стандартним розміром електронного посібника є 1024х768, але через наявність великої кількості розширень необхідно оптимізувати вигляд електронного посібника. Дана функція відповідає за вирівнювання сторінок посібника по центру екрана.

<script language="JavaScript" type="text/javascript">

function justLoaded() {

//centralization of content

document.getElementById("mainForm").style.margin = "0px "+(innerWidth/2-512);

}

</script>

…

<body onLoad="javascript:justLoaded()">

Функція є простою і складається лише з однієї дії. Весь вміст сторінки електронного посібника поміщено у єдиний елемент <div> “mainForm”. Саме вирівнювання цього елемента і виконує дана функція. Від ширини вікна браузера функція вираховує половину і віднімає 512 одиниць (половина від 1024 – всієї ширини електронного посібника). А далі присвоює отримане число як відступ зліва для даного елемента. Єдиний мінус цієї функії – деякі браузери виконують JavaScript функції після повного завантаження всіх інших елементів, що створює коротку затримку між завантаженням елемента з початковим стилем і присвоєння йому нового параметру.

Функція зчитування відповідей та підрахунку оцінки за тест

Наступні три функції відповідає за зчитування результатів при виборі одного із варіантів відповіді та подальший підрахунок правильних відповідей і оголошення результату.

<script language="JavaScript" type="text/javascript">

var answers = new Array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);

var correct = new Array(4,1,2,2,3,4,4,2,1,3,1,1,2,4,2,3,1,1,3,2,4,1,1,3,1);

function setAnswer(sQuestion,sAnswer) {

answers[sQuestion-1] = parseInt(sAnswer);

}

function testAll() {

var counter = 0;

for (var i=0; i<correct.length; i++) {

if (answers[i] == correct[i])

counter++;

}

return counter;

}

function showResult() {

var counter = testAll();

var mark1 = Math.round(5/correct.length\*counter);

if (mark1 < 2) mark1 = 2;

var mark2 = Math.round(12/correct.length\*counter);

if (mark2 > 1) mark2--;

var mark3 = Math.round(100/correct.length\*counter);

var markA = "";

if (mark3 < 45) markA = "F";

else if (mark3 >= 45 && mark3 < 60) markA = "D";

else if (mark3 >= 60 && mark3 < 75) markA = "C";

else if (mark3 >= 75 && mark3 < 90) markA = "B";

else if (mark3 >= 90) markA = "A";

alert("Ваш результат за тест по темі 1:"+'\n'+'\n'+"Кількість правильних відповідей: "+counter+"."+'\n'+'\n'+"Ваша оцінка за 5-бальною шкалою: "+mark1+"."+'\n'+"Ваша оцінка за 12-бальною (макс. 11): "+mark2+"."+'\n'+"Ваша оцінка за відсотковою шкалою GPA: "+mark3+"% ("+markA+").");

}

…

<input type="radio" onclick=setAnswer(1,1) name="Q1"

…

<input id="button1" type="button" onclick=showResult() value="Перевірити" name="Result">

Функція setAnswer(sQuestion, sAnswer) активується при виборі варіанту відповіді – тобто однієї із декількох RadioButton для кожного запитання. При цьому змінній типу масив answers в комірці з номером, що на одиницю меншим за номер запитання sQuestion присвоюється номер, рівний номеру вибраної відповіді sAnswer. Номер запитання та номер відповіді задається разом із запитанням.

Підрахунок результатів виконується при натиску кнопки «Перевірити». При цьому функція перевіряє на рівність кожну із отриманих відповідей записану в масиві answers із завчасно заданими правильними відповідями в масиві correct. Далі в залежності від отриманих результатів вираховуються та оголошуються бали.

* 1. **Тестування сайту**

Мій дипломний проект виконувався на комп’ютері Intel Core2Duo 1.8 ГГц 1Гб RAM, відео карта Radeon HD 5670 1GB DDR3 під управлінням операційної системи Windows7. Монітор BenQ sesnseye 20”. Встановлено найбільш популярні браузери – Google Chrome 12.0, Opera 11.11, та Mozilla Firefox 4.01.

Оскільки на даний час існує велика кількість різних конфігурацій ПК та встановленого на них програмного забезпечення, виникла необхідність встановити рамки в яких даний web-ресурс буде коректно працювати.

Даний web-сайт було протестовано на середньостатистичному ПК який працює в Інтернет з такою конфігурацією:

* швидкість підключення – 5 Мбіт.

Програмне забезпечення, яке встановлене на ньому було наступне:

* Google Chrome 11.74;
* Opera 10.66;
* Mozilla Firefox 4.01;
* Internet Explorer 8;
* Java Platform.

Під час тестування було перевірено середню швидкість завантаження кожної сторінки посібника, швидкість завантаження зображень, а також якість відображення стилів в браузерах, в яких електронний посібник повинен працювати, також будо перевірено всі наявні гіперпосилання на внутрішні та зовнішні ресурси.

**Розділ 3. СПЕЦІАЛЬНИЙ РОЗДІЛ**

Розробці електронного посібника з дисципліни «Комп’ютерна логіка» передувало вивчення матеріалів спеціалізованих періодичних видань, а також опрацювання інформації в мережі Інтернет.

За результатами пошуку інформації в Інтернеті й спеціалізованих періодичних виданнях, та посібниках було прийнято рішення на основі знайдених матеріалів, розробити електронний посібник з дисципліни «Комп’ютерна логіка».

* 1. **Встановлення і налаштування web-сервера**

Для того щоб розроблений web-сайт працював коректно, необхідно встановити відповідне програмне забезпечення. Перш за все рекомендовано встановити web-сервер. Існує декілька найпопулярніших веб-серверів: IIS, Apache та Denwer. В даному випадку ми зупинимося на останньому.

Щоб встановити сервер Denwer 3 потрібно завантажити з сайту www.denwer.ru дистрибутив Denwer 3 у вигляді інсталятора з ім'ям Denwer3\_Base\_2010-11-07\_a2.2.4\_p5.3.1\_m5.1.40\_pma3.2.3.exe.Після того як файл завантажено потрібно його запустити на виконання. Першим пунктом чи то пак кроком до встановлення є підтвердження своїх намірів становити базовий пакет (рис. 3.1.1)

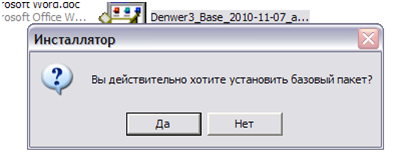


Рисунок 3.1.1 – Вікно запиту інсталяції Denwer.

Далі інсталятор почне розпаковувати файли, необхідні для встановлення сервера на комп’ютер (рис. 3.1.2).

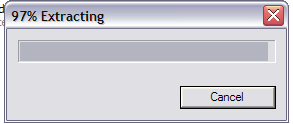


Рисунок 3.1.2 – Розпаковування фалів сервера Denwеr.

Наступні кроки встановлення будуть відбуватись в консольному режимі (рис 3.1.3).

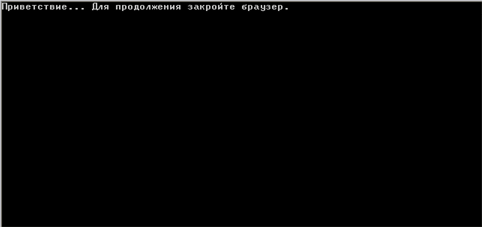


Рисунок 3.1.3 – Консольне вікно устанвки сервера.

Під час того як відкривається консольне вікно, відкривається вікно бразура, в якому відображається, текст з усіма програмами які містить у собі Dеnwer, а саме Apache, MySQL, PHP, Perl та PHPAdmin (рис 3.1.4).

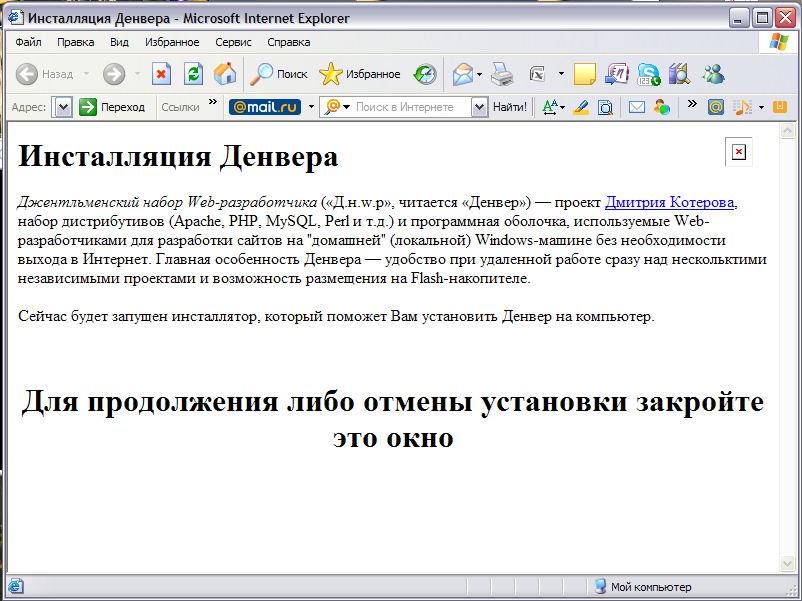


Рисунок 3.1.4 – Перелік програм в пакеті Denwer.

Для продовження встановлення потрібно закрити вікно браузера та натиснути будь-яку кнопку «Enter» у вікні консольного режиму (рис. 3.1.5).

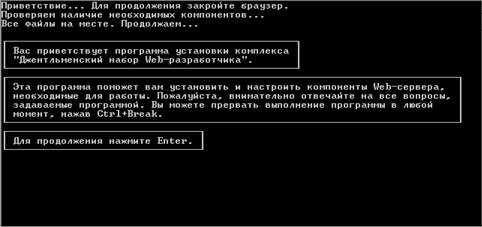


Рисунок 3.1.5 –Початок встановлення.

Після того, як користувач продовжив встановлення сервера, інсталятор шукає всі необхідні для встановлення файли, і пропонує обрати папку, в яку буде встановлено сервер (рис. 3.1.6). Якщо не вкзаувати папку, то програма інсталяції встановить сервер в папку по замовчуванню, а саме в C:\Websrvers.

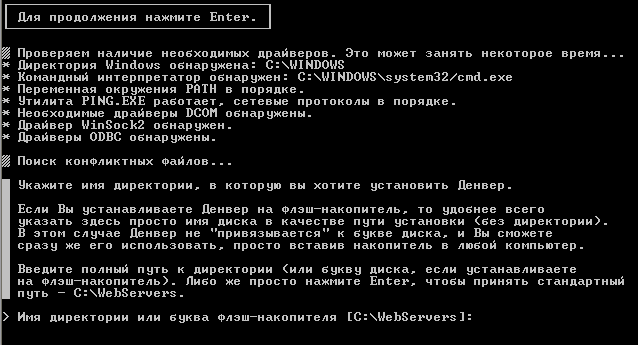


Рисунок 3.1.6 – Вибір папки для встановлення сервера.

Отже після того як вказано папку для встановлення сервера, інсталятор скопіює всі необхідні для роботи сервера файли в цю папку.

Після закінчення встановлення сервера, програма пропонує два варіанти запуску: перший варіант дозволяє встановити віртуальний диск одразу після завантажені ОС (рис 3.1.7). При цьому запуск сервера здійснюється за допомогою ярлика на робочому столі. Коли робота сервера припиняється віртуальний диск не відключається; другий варіант – не встановлювати віртуальний диск, створюються лише ярлики запуску сервера на робочому столі, і віртуальний диск створюється тільки під час роботи сервера, а отже й забирає більше часу.



Рисунок 3.1.7 – Завершення установки: вибір шляху запуску сервера.

У разі якщо, користувач обрав перший варіант запуску, програма пропонує обрати букву для віртуального диску, по-замовчуванню це Z:\, оскільки скоріш за все диска з такою буквою на комп’ютері ще нема.

Після того як консольне вікно закрилось, відкривається вікно браузера, в якому розробник даного сервера, дякує користувачу за те, що той обрав даний продукт. Для того щоб перевірити чи встановлений сервер працює як належне, розробник пропонує пройти за наступною адресою – http://localhost. Якщо сервер встановився без помилок, то користвач побачить повідомлення, як продемонстровано на рис. 3.1.8.

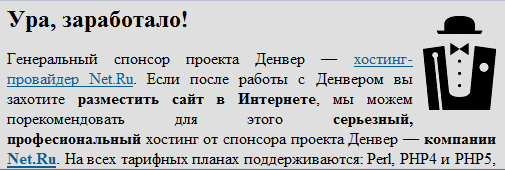


Рисунок 3.1.8 – Домашня сторінка localhost.

Налаштування Apache

Як було зазначено раніше, базовий пакет Denwer містить в собі наступні компоненти: Apache, MySQL, PHP та інші.

Отже основний файл конфігурації сервера Apache – "httpd.conf". Він знаходиться в каталозі "conf" в папці С:\WebServers\user\local\apache\conf. Синтаксис файлу дуже простий: "директива значення", всі рядки не відповідні цьому вигляду можна видалити.

Увага! Деякі директиви можуть виглядати так:

#директива значення

<блок директив />

директива значення

</блок директив>

<блок директив />

# Коментарі

директива значення

# Коментарі

</блок директив>

Ці рядки видаляти не потрібно!

Символ коментаря в "http.conf" – "#" (решітка). Всі символи, що йдуть в рядку після "#" не читаються сервером. Тому ми можемо додавати власні коментарі.

Далі потрібно відредагувати файл "httpd.conf". Нижче приведено список директив та їх значень, які повинні бути присутніми у файлі конфігурації Apache.

Каталог з файлами сервера:

ServerRoot "C:/Server/Apache/Apache2"

Прив'язує Apache до конкретного порту:

Listen 80

Ім'я сервера (на роботу це не впливає):

ServerName localhost:80

Директива “Адміністратор сервера” містить адресу електронної пошти, яка відображатиметься при деяких помилках сервера: ServerAdmin your@email.name.

По замовчуванню Apache встановлює значення "DocumentRoot" – "/ServerRoot/htdocs". Але оскільки ми встановили Apache в теку "C: \Apache", то в "DocumentRoot" записуємо адресу так: "C:/ Apache/Apache2/htdocs"). Змінити значення "DocumentRoot" на "C:/Sites/home/localhost/www".

Наступним етапом буде створити на диску "C:" папку "Sites" із вкладеною папкою "home", вже в ній - "localhost", "neebet", "mysite". У кожній з цих папок створив каталог "www" (для зберігання html документів), "cgi" (для зберігання cgi-скриптів), порожні файли access.log (журнал доступу до сервера) і error.log (журнал помилок сервера).

Далі замінив блок "<Directory "C:/Server/Apache/Apache2/htdocs">" на наступне:

<Directory "C:/Sites">

Options Indexes Includes

AllowOverride All

Order allow,deny

Allow from all

</Directory>

Вказівка індексного файлу для папки (цей файл сервер відображатиме при зверненні до папки без вказівки імені файлу):

DirectoryIndex index.php index.html index.html.var

Зазначення псевдоніма для каталогу з cgi-скриптами "C:\Sites\home\localhost\cgi". При вказівці шляху в вигляді http://localhost/cgi/ або http://localhost/cgi-bin/Apache звертатиметься до папки "C:\Sites\home\localhost\cgi":

ScriptAlias /cgi/ "C:/Sites/home/localhost/cgi/"

ScriptAlias /cgi-bin/ "C:/Sites/home/localhost/cgi/"

Каталог "C:\Sites\home\localhost\cgi" також буде доступний при зверненні такого вигляду "http://імя\_віртуального\_хосту/cgi-bin/cgi-скрипт.bat". Наприклад, якщо ввести в браузері http://neebet/cgi-bin/cgitest.bat, то буде виконаний код, що знаходиться у файлі "C:\Sites\home\localhost\cgi\cgitest.bat", який також доступний за адресою http://localhost/cgi/cgitest.bat. Каталоги "cgi" не доступні для перегляду в браузері, і при прямому зверненні до них ви отримаєте повідомлення про помилку "403". Вказуємо Apache, що файли з розширенням \*,cgi, \*,bat, \*,exe потрібно сприймати як cgi-скрипты:

AddHandler cgi-script .cgi .bat .exe

Блок "<Directory "C:/ Apache2/cgi-bin">" замінив на:

<Directory "C:/Sites/home/localhost/cgi">

AllowOverride None

Options None

Order allow,deny

Allow from all

</Directory>

Решта рядків виду "AddLanguage lang .lang" можна перетворити в коментарі (якщо звичайно не потрібна підтримка цих мов):

AddLanguage en .en

AddLanguage ru .ru

Встановлюємо мовний пріоритет:

LanguagePriority en ru

Настройка кодувань:

AddDefaultCharset WINDOWS-1251

AddCharset WINDOWS-1251 .cp-1251 .win-1251

За визначення кореневої папки відповідає DocumentRoot у файлі С:/Apache2.2/conf/httpd.conf.

DocumentRoot "C:/Apache2.2/htdocs"

Для зручності можна змінити її і зробити інший каталог кореневою директорією сервера. По-замовчуванню, Apache налаштований так, щоб заборонити будь-який доступ до директорій сайтів і дозволи повинні бути встановлені явно для кожної директорії, як це зроблено для директорії C:/Apache2.2/htdocs. Якщо ж Ви перенесете кореневу директорію сервера в інший каталог, не зробивши відповідні дозволи, то у відповідь буде видана помилка 403 Forbidden.

Тому, якщо ви НЕ збираєтеся використовувати Apache для надання послуг хостингу в Інтернеті або в локальній мережі, то слід відразу ж зняти встановлений за замовчуванням захист.

Знайдіть у файлі С:/apache2.2/conf/httpd.conf рядка:

<Directory />

 Options FollowSymLinks

 AllowOverride None

 Order deny,allow

 Deny from all

 Satisfy all

</Directory>

Зверніть увагу на рядок Deny from all. Він забороняє будь-який доступ до папок, у тому числі і до папок віртуальних хостів. При спробі звернення до них Apache відповість помилкою 403 Forbidden. Протилежною їй є директива «Allow from all». Установка цієї директиви відміняє дію директиви «Deny from all» і дозволяє доступ до папок. Можна встановлювати директиву Allow from all для кожної директорії і для кожного віртуального хоста зокрема, але можна зробити це простіше і прибрати (або перетворити в коментар) директиву «Deny from all».

<Directory />

 Options FollowSymLinks

 AllowOverride None

 Order deny,allow

# Deny from all

 Satisfy all

</Directory>

Після редагування файлу конфігурації збережіть його та перезапустіть сервер.

* 1. **Інструкція з експлуатації електронного посібника**

Важко очікувати, що кожен викладач у достатньому обсязі освоїть техніку web-програмування, але значну частину роботи, необхідну для створення сучасного мультимедіа-курсу, орієнтованого на мережеве використання (web-курсу), може виконати практично кожний.

Нижче наводяться рекомендації, що можуть бути корисними для такої підготовчої роботи:

По-перше в цьому випадку необхідно створити електронну версію курсу в текстовому редакторі Word. При цьому можна використовувати інструментальні засоби створення малюнків, формульний редактор, можливості вставки зображень і т. д.

Маючи web-версію курсу, особливу увагу варто приділити структуруванню матеріалу – поділу його на розділи, підрозділи, складанню змістів розділів. Кожна мінімальна логічна частина (підрозділ) повинна бути поміщена в окремий файл, а всі створені файли варто розмістити в одному каталозі, що при необхідності має підкаталоги.

При структуруванні посібника варто враховувати, що розмір кожного файлу повинен бути таким, щоб при його відображенні в Web-браузері він не займав більше 3-4 екранів. Звичайно це відповідає 2-3 сторінкам тексту в doc-файлі при розмірі шрифту 12.

Наступним кроком необхідно створити необхідні гіпертекстові посилання, що встановлюють зв'язки, як між окремими розділами, так і з необхідними ресурсами Інтернет. Файл, з яким установлюється зв'язок, повинен вже існувати в створеному каталозі посібника.

В принципі, робота з текстом матеріалу на розглянутому (підготовчому) етапі на цьому може завершитися. Проте, якщо викладач має намір подальшу роботу проводити самостійно чи активно брати участь у роботі разом із програмістом, корисно зробити ще одну операцію – перетворення тексту в HTML- формат.

Для цього в меню «Файл» необхідно вибрати режим «Зберегти як…», у вікні «Збереження документа» у полі «тип файлу» встановити «HTML Document». У результаті цієї операції буде створений файл \*.html, де \* – ім'я doc-файлу. При цьому усі формули, написані у формульному редакторі Word, будуть автоматично перетворені у файли-зображення формату .jpg (імена цих файлів за замовчуванням image\*.jpg, де \* – порядковий номер 1,2,3...).

На сам кінець потрібно скопіювати всі html-файли з дотриманням структури каталогів посібника і передати матеріал програмісту web-майстру сервера, на якому передбачається розміщення посібника.

Отримані файли ще вимагатимуть визначеної редакції фахівцем у web-програмуванні, але пророблена робота дозволить істотно скоротити час остаточної доробки посібника.

Щоб перейти до використання електронного посібника спочатку необхідно скопіювати наступні файли та папки:

* папка html із вкладеними папками та файлами;
* папка src із вкладеними папками та файлами;
* index.html;
* start.bat.

При цьому структура дерева папок не повинна бути зміненою чи пошкодженою, оскільки всі внутрішні гіпертекстові посилання мають строго визначену адресу.

Електронний посібник з дисципліни «Комп’ютерна логіка» можна завантажити декількома способами:

1. Запустити файл start.bat – в цьому випадку посібник буде завантажено в браузері, що встановлений по замовчуванню.
2. Спробувати відкрити файл index.html – в цьому випадку буде завантажено програму (браузер) з яким асоційовані файли з розширенням \*.html.
3. Завантажити браузер та в рядку адреси прописати шлях до файлу index.html.

Рекомендовано використовувати Google Chrome, Opera або Mozilla Firefox для коректного відображення вмісту електронного посібника з дисципліни «Комп’ютерна логіка». Internet Explorer поки що не підтримується. Подальше вдосконалення посібника буде включати в собі підтримку IE версії 6 або вищої.

Після завантаження на екрані з’явиться титульна (головна) сторінка. Щоб перейти безпосередньо до самого посібника, потрібна натиснути на текст «Комп’ютерна логіка».

Після завантаження домашньої сторінки можна обрати один і розділів із меню, що розташоване зліва. Щоб перейти до конкретного розділу, достатньо натиснути на ньому. Деякі із розділів містять список тем, чи перелік пунктів. Кожен із пунктів також є гіперпосиланням, і при натисненні відкриє ту чи іншу сторінку.

Редагування вмісту електронного посібника

Будь-які дані в посібнику можуть бути змінені. В першій версії електронного посібника з дисципліни «Комп’ютерна логіка» не передбачено адміністративну частину, тому будь-які зміни потрібно вносити вручну, редагуючи HTML-код сторінок.

Для редагування коду можна використовувати звичайний Блокнот. Деякі браузери, Opera наприклад, дозволяють редагувати код джерела безпосередньо в самому браузері. Але не дивлячись на це, рекомендовано використовувати спеціальні програми для цього, як от FrostPage98, DreamWeaver чи Notepad++.

**Розділ 4. ЕКОНОМІЧНИЙ РОЗДІЛ**

* 1. **Загальні положення**

Метою економічної частини дипломного проекту є здійснення економічних розрахунків, спрямованих на визначення економічної ефективності електронного посібника з дисципліни «Комп’ютерна логіка» і прийняття рішення про її подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Для розрахунку вартості НДР необхідно виконати наступні етапи:

* Описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
* Визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
* Визначити суму матеріальних витрат;
* Обчислити витрати на електроенергію для науково-виробничих цілей;
* Розрахувати транспортні витрати;
* Нарахувати суму амортизаційних відрахувань;
* Визначити суму накладних витрат;
* Скласти кошторис та визначити собівартість НДР;
* Розрахувати ціну НДР;
* Визначити економічну ефективність та термін окупності продукту.

* 1. **Визначення стадій технологічного процесу та загальної тривалості проведення НДР**

Для визначення загальної тривалості проведення НДР доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю 1.

Таблиця 1

Середній час виконання НДР та стадії (операції) технологічного процесу.

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Назва операції (стадії) | Виконавець | Середній час виконання операції, год. |
| 1. | Підготовка опису задачі | інженер | 3 |
| 2. | Розробка алгоритму рішення задачі | інженер | 9 |
| 3. | Розробка блок-схеми алгоритму | інженер | 4 |
| 4. | Розробка графічного дизайну посібника | інженер | 23 |
| 5. | Написання кодів електронного посібника | інженер | 66 |
| 6. | Тестування готового продукту | інженер | 7 |
| **Разом** | |  | 112 |

* 1. **Визначення витрат на оплату праці та відрахувань на соціальні заходи**

Відповідно до Закону України «Про оплату праці» заробітна плата – це «винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та допоміжної оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов’язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Основна заробітна плата розраховується за формулою:

*Зосн.* = Тс ∙ Кг, де

Тс – тарифна ставка, *грн.*;

Кг – кількість відпрацьованих годин.

Рекомендовані тарифні ставки: інженер – 10-12 грн./год.

*Зосн.* = 12 ∙ 112 = 1344 *грн.*

Додаткова заробітна плата становить 10-15% від основної зарплати.

*Здод. = Зосн. ∙ Кдопл.*, де

Кдопл. – коефіцієнт додаткових виплат працівникам.

*Здод.* = 1344 ∙ 0,10 = 134,4 *грн.*

Звідси загальні витрати на оплату праці (Во.п.):

*Во.п. = Зосн. + Здод.*

*Во.п.* = 1344 + 134,4 = 1478,4 *грн.*

Крім цього, слід визначити відрахування на соціальні заходи (Вс.з.):

* відрахування в фонд страхування на випадок безробіття (1,6%);
* фонд по тимчасовій втраті безробіття (1,4%);
* пенсійний фонд (33,2%);
* внески на страхування від нещасного випадку на виробництві та професійного захворювання (1,4%).

Разом зазначені відрахування становлять 37,6%. Отже отримаємо:

*Вс.з. = ФОП ∙ 0,376*, де

ФОП – фонд оплати праці, грн.

*Вс.з.* = 1478,4 ∙ 0,376 = 555,8 *грн.*

Зведемо розрахунки витрат на оплату праці в таблицю 2.

Таблиця 2

Зведені розрахунки витрат на оплату праці.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № п/п | Категорія працівників | Основна заробітна плата | | | Додаткова заробітна плата, грн. | Нарахуван. до ФОП, грн. | Всього витрати на оплату праці, грн. |
| Тарифна ставка, грн. | К-сть відпрацьов год. | Фактично нарах. з/пл., грн. |
| 1 | Інженер | 12 | 112 | 1344 | 134,4 | 555,8 | 2034,2 |
| **Разом** | | | 112 | 1344 | 134,4 | 555,8 | 2034,2 |

* 1. **Розрахунок витрат на електроенергію**

Затрати на електроенергію 1-ці обладнання визначається за формулою:

*Зе = W ∙ T ∙ S*, де

W – необхідна потужність, *кВт*;

T – кількість годин роботи обладнання;

S – вартість кіловат-годин електроенергії;

Підставивши в формулу значення W = 0,5 кВт, S = 0,77 грн., кількість годин затрачених на розробку – 112.

*Зе* = 0,5 ∙ 112 ∙ 0,77 = 43,12 *грн*.

* 1. **Розрахунок суми амортизаційних відрахувань**

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Комп’ютери та оргтехніка належать до четвертої групи основних фондів. Мінімально допустимі строки їх корисного використання 2 роки.

Для визначення амортизаційних відрахувань застосуємо формулу:

, де

А – амортизаційне відрахування за звітний період, *грн.*;

БВ – балансова вартість групи основних фондів на початок звітного періоду, *грн.*;

НА – норма амортизації, *%*;

Приймемо вартість ПК за БВ = 3840. Підставивши ці дані в формулу отримаємо:

* 1. **Обчислення накладних витрат**

Накладні витрати пов’язані із обслуговуванням виробництва та утриманням апарату управління підприємства (або фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб’єкта, накладні витрати можуть становити 20-60% від суми основної та додаткової заробітної плати працівників.

*Нв = Во.п.* ∙ 0,2…0.6.

За розрахунками проведеними раніше маємо, що Во.п. = 1478,4 грн. Приймемо, що накладні витрати становлять 40% від цієї суми і отримаємо:

*Нв* = 1478,4 ∙ 0,4 = 591,36 *грн.*

Зведемо кошторис витрат на розробку електронного посібника з дисципліни «Комп’ютерна логіка» в таблицю 3.

Таблиця 3

Кошторис витрат на розробку електронного посібника.

|  |  |  |
| --- | --- | --- |
| **Зміст витрат** | **Сума, грн.** | **В % до загальної суми** |
| Витрати на оплату праці | 1478,4 | 53,11 |
| Відрахування на соціальні заходи | 555,8 | 19,97 |
| Витрати на електроенергію | 43,12 | 1,55 |
| Амортизаційні відрахування | 114,69 | 4,12 |
| Накладні витрати | 591,36 | 21,25 |
| **Собівартість** | 2783,37 | 100 |

* 1. **Розрахунок ціни електронного посібника**

Ціну електронного посібника з дисципліни «Комп’ютерна логіка» можна обрахувати за наступною формулою:

, де

Ррен. – рівень рентабельності, *30%*;

К – кількість замовлень, *од.*;

Вн.і. – вартість носія інформації, *грн.*;

ПДВ – ставка податку на додану вартість (20%).

Отже, підставивши значення в формулу отримаємо ціну електронного посібника з дисципліни «Комп’ютерна логіка»:

* 1. **Визначення економічної ефективності і терміну окупності капітальних вкладень**

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві (фірмі) за певний проміжок часу.

Економічна ефективність (Ер) полягає у відношенні результату виробництва до затрачених ресурсів:

, де

П – прибуток;

Св – собівартість.

Підставимо значення в формулу визначення економічної ефективності і отримаємо:

З цього випливає, що термін окупності капіталовкладень становить:

*.*

Підсумовуючи всі проведені вище розрахунки можна сказати, що розробка електронного посібника з дисципліни «Комп’ютерна логіка» є економічно доцільною і має допустимий термін окупності, а затрати на створення такого посібника є порівняно не великими. В таблиці 4 показано результати техніко-економічних показників:

Таблиця 4

Результати техніко-економічних показників.

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Показник** | **Значення** |
| 1. | Собівартість, грн. | 2783,37 |
| 2. | Плановий прибуток, грн. | 1002,02 |
| 3. | Ціна, грн. | 4342,06 |
| 4. | Економічна ефективність | 0,36 |
| 5. | Термін окупності, рік | 2,8 |

**Розділ 5. ОХОРОНА ПРАЦІ**

* 1. **Вимоги до виробничих приміщень**

Облаштування робочих місць, обладнаних відео терміналами, оргтехнікою, повинно забезпечувати:

* належні умови освітлення приміщення і робочого місця, відсутність відблисків;
* оптимальні параметри мікроклімату (температура, відносна вологість, швидкість руху та рівень іонізації повітря);
* належні ергономічні характеристики основних елементів робочого місця.

А також вжити заходи проти таких небезпечних і шкідливих чинників:

* наявність шуму та вібрації;
* м’яке рентгенівське випромінювання;
* електромагнітне випромінювання;
* ультрафіолетове та інфрачервоне випромінювання;
* електростатичне поле між екраном монітору та оператором;
* наявність пилу, озону, оксидів азоту й аероіонізації.

Будівлі, і ті їх частини, в яких розташовуються ЕОМ, повинні мати не нижче ІІ ступеня вогнестійкості. Неприпустимим є розташування приміщень виробництв з мокрими технологічними процесами поряд з приміщеннями, де розташовуються ЕОМ, виконується їх обслуговування, налагодження і ремонт, а також над такими приміщеннями або під ними.

Службові приміщення, в яких розташовані ЕОМ, не повинні межувати з приміщеннями, де рівні шуму та вібрації перевищують норму (механічні цехи, майстерні тощо).

Робочі місця з відеотерміналами або персональними ЕОМ у приміщеннях з джерелами шкідливих виробничих чинників повинні розміщуватися в ізольованих кабінах з обладнаним повітрообміном.

Стіни кабін виготовляються з негорючих матеріалів. Дозволяється виготовляти їх зі скла та металевих! конструкцій. У кабіні мусить бути оглядове вікно (вінка). Висота оглядового вікна має бути не менше 1,5м., а відстань від підлоги не більше 0,8м.

Відповідно до ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин», затверджених МОЗ України 10.12.98, є неприпустимим розташування приміщень для роботи з відеотерміналами та ЕОМ у підвалах та цокольних поверхах.

Площу приміщень, у яких розташовують відеотермінали визначають згідно з чинними нормативними документами, виходячи з розрахунку на одне робоче місце, обладнане відеотерміналом: площа – не менше 6,0 кв. м., об’єм – не менше 20,0 куб. м., з урахуванням максимальної кількості осіб, які одночасно працюють у зміні.

Стіни, стеля, підлога приміщень, де розміщені ЕОМ, повинні виготовлятися з матеріалів, дозволених для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду.

Обслуговування, ремонт та налагодження ЕОМ, вузлів та блоків ЕОМ слід виконувати в окремому приміщенні (майстерні).

Підлога всієї зони обслуговування, ремонту та налагодження ЕОМ, вузлів та блоків ЕОМ має бути вкрита діелектричними килимками, термін використання яких після їх випробування на електричну міцність не закінчився, або викладена ізольованими підстилками (шириною не менше, ніж 0,75-08м.) для ніг.

Приміщення комп’ютерних класів (залів) у яких проводиться навчання на ЕОМ, повинні мати суміжне приміщення (лаборантське) площею не менше 18 км. м. з двома входами: в навчальне приміщення та в коридор (на сходову клітку).

Заземлені конструкції, що знаходяться в приміщеннях (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном тощо), мають бути надійно захищені діелектричними щитками або сітками від випадкового дотику.

У приміщеннях з ЕОМ слід щоденно проводити вологе прибирання, повинні бути медичні аптечки першої допомоги.

Приміщення з ЕОМ, крім приміщень, у яких розміщуються ЕОМ типу ЕС, СМ та інші великі ЕОМ загального призначення, повинні бути оснащені системою автоматичної пожежної сигналізації.

Підходи до засобів пожежогасіння повинні бути вільними.

Умови праці осіб, які працюють з ЕОМ, повинні відповідати І або ІІ класу згідно з Гігієнічною класифікацією праці за показниками шкідливості та небезпечності чинників виробничого середовища, важкості та напруженості трудового процесу № 4137-86, затвердженого МОЗ СРСР 12.08.86.

* 1. **Вимоги до рівнів шуму та вібрації**

Для забезпечення нормованих рівнів шуму у виробничих приміщеннях та на робочих місцях застосовуються шумопоглинальні засоби, вибір яких обґрунтовується спеціальними інженерно-акустичними розрахунками.

Як засоби шумопоглинання повинні застосовуватися негорючі або, важкогорючі спеціальні перфоровані плити, панелі, мінеральна вата з максимальним коефіцієнтом звукопоглинання в межах частот 31,5-8000 Гц, або інші матеріали аналогічного призначення, дозволені для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду. Крім того, необхідно застосовувати підвісні стелі з аналогічними властивостями.

Приміщення з ЕОМ повинні бути обладнані системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією відповідно до СНиП 2.04.05-91 «Отопление вентиляция и кондиционирование».

Параметри мікроклімату, іонного складу повітря, вміст шкідливих речовин на робочих місцях, оснащених відеотерміналами, повинні відповідати вимогам пункту 2.4 СН 4088-86 «Санітарні норми мікроклімату виробничих приміщень», затверджених МОЗ СРСР, ГОСТ 12.1.005-88 «ССБТ Общие санитарно-епидемиологические требования к воздуху рабочей зоны» (табл. 1), СН 2152-80 «Санітарно-гігієнічні норми допустимих рівнів іонізації повітря виробничих та громадських приміщень», затверджених МОЗ СРСР (табл. 2).

Таблиця 1

Нормовані параметри мікроклімату для приміщень з ВДТ та ПЕОМ.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Пора року | Категорія робіт згідно з ГОСТ | Температура повітря, °С | Відносна вологість повітря, % | Швидкість руху повітря, м/с |
|  | 005-88 | оптимальна | оптимальна | Оптимальна |
| Холодна | легка-1 а | 22-24 | 40-60 | 0,1 |
|  | легка-1 б | 21-23 | 40-60 | 0,1 |
| Тепла | легка-1 а | 23-25 | 40-60 | 0,1 |
|  | легка-1 б | 22-24 | 40-60 | 0,2 |

Таблиця 2

Рівні іонізації повітря приміщень при роботі на ВДТ та ПЕОМ (відповідно до СН 2152-80).

|  |  |  |
| --- | --- | --- |
| Рівні | Кількість іонів в 1 см. куб. повітря | |
| n | n” |
| Мінімально необхідний | 400 | 600 |
| Оптимальний | 1500-3000 | 3000-5000 |
| Максимально допустимий | 50 000 | 50 000 |

Для підтримки допустимих значень мікроклімату та концентрації позитивних та негативних іонів необхідно передбачити установки або прилади зволоження та/або штучної фонації, кондиціонування повітря.

* 1. **Вимоги до організації робочого місця користувача ЕОМ**

Організація робочого місця користувача відеотерміналу та ЕОМ повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ГОСТ 12.2.032 «ССБТ Рабочее место при выполнении работ сидя. Общие эргономические требования»; характеру та особливостям трудової діяльності.

Площа, виділена для одного робочого місця з відеотерміналом або персональною ЕОМ, повинна складати не менше 6 кв. м., а об’єм – не менше 20 куб. м.

Робочі місця з відеотерміналами відносно світлових прорізів повинні розмішуватися так, щоб природне світло падало збоку, переважно зліва.

При розміщенні робочих місць з відеотерміналами та персональними ЕОМ необхідно дотримуватись таких вимог:

* Робочі місця з відеотерміналами та персональними ЕОМ розміщуються на відстані не менше 1м від стін зі світловими прорізами;
* Відстань між бічними поверхнями відеотерміналів має бути не меншою за 1,2м;
* Відстань між тильною поверхнею одного відеотерміналу та екраном іншого не повинна бути меншою 2,5м;
* Прохід між рядами робочих місць має бути не меншим 1м.

Вимоги щодо відстані між бічними поверхнями відеотерміналів та відстані між тильною поверхнею одного відеотерміналу та екраном іншого враховуються також при розміщенні робочих місць із відеотерміналами та персональними ЕОМ в суміжних кімнатах, з урахуванням конструктивних особливостей стін та перегородок.

Конструкція робочого місця користувача відеотерміналу (при роботі сидячи) має забезпечувати підтримання оптимальної робочої пози з такими ергономічними характеристиками: ступні ніг – на підлозі або на підставці для ніг; стегна – в горизонтальній площині; передпліччя – вертикально; лікті – під кутом 70-90° до вертикальної площини; зап’ястя зігнуті під кутом не більше 20° відносно горизонтальної площини; нахил голови – 15-20° відносно вертикальної площини.

Якщо користування відеотерміналом та персональною ЕОМ є основним видом діяльності, то вказане обладнання розміщується на основному робочому столі, як правило, з лівого боку.

Якщо використання відеотерміналу та персональної ЕОМ є періодичним, то устаткування, як правило, розміщується на приставному столі, переважно з лівого боку від основного робочого столу. Кут між поздовжніми осями основного та приставного столів має бути 90-140°.

При періодичному використанні відеотерміналів та персональних ЕОМ дозволяється обладнувати в приміщенні, що відповідає вимогам даних Правил, окремі робочі місця колективного користування з відеотерміналом та персональною ЕОМ.

Висота робочої поверхні столу для відеотерміналу має бути в межах 680-800мм, а ширина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля.

Рекомендовані розміри столу: висота – 725мм, ширина – 600-1400мм, глибина – 800-1000мм.

Робочий стіл для відеотерміналу повинен мати простір для ніг висотою не менше 600мм., шириною не менше 500мм., глибиною на рівні колін – не менше 650мм.

Робочий стіл для відеотерміналу, як правило, має бути обладнаним підставкою для ніг шириною не менше 300мм. Та глибиною не менше 400мм., з можливістю регулювання по висоті в межах 150мм. Та кута нахилу опорної поверхні – в межах 20°. Підставка повинна мати рифлену поверхню та бортик на передньому краї заввишки 10мм. Робоче сидіння (сидіння, стілець, крісло) користувача відеотерміналу чи персональної ЕОМ повинно мати такі основні елементи: сидіння, спинку та стаціонарні або знімні підлокітники.

У конструкцію сидіння можуть бути введені додаткові елементи, що не є обов’язковими: підголівник та підставка для ніг.

Робоче сидіння користувача відеотерміналу чи персональної ЕОМ повинно бути підйомно-поворотним, таким, що регулюється за висотою, кутом нахилу сидіння та спинки, за відстанню спинки до переднього краю сидіння, висотою підлокітників.

Регулювання кожного параметра має бути незалежним, плавним або ступінчастим, мати надійну фіксацію.

Хід ступінчастого регулювання елементів сидіння має становити для лінійних розмірів – 15-20мм, для кутових – 2-5°.

Зусилля під час регулювання не повинні перевищувати 20 Н.

Ширина та глибина сидіння повинні бути не меншими за 400мм. Висота поверхні сидіння має регулюватися в межах 400-500мм., а кут нахилу поверхні – від 15° вперед до 5° назад.

Поверхня сидіння має бути плоскою, передній край – заокругленим.

Висота спинки сидіння має становити 300х20мм., ширина – не менше 380мм., а радіус кривизни в горизонтальній площині – 400мм.

Кут нахилу спинки повинен регулюватися в межах 0-30° відносно вертикального положення. Відстань від спинки до переднього краю сидіння повинна регулюватись у межах 260-400мм.

Для зниження статичного напруження м’язів рук необхідно застосовувати стаціонарні або знімні підлокітники довжиною не менше 250мм., шириною – 50-70мм., що регулюються по висоті над сидінням у межах 230±30мм. Та по відстані між підлокітниками в межах 350-500мм.

Поверхня сидіння, спинки та підлокітників має бути напівм’якою з неслизьким, ненаелектризовуючим, повітронепроникним покриттям та забезпечувати можливість чищення від бруду.

Екран відеотерміналу та клавіатура мають розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм., з урахуванням розміру алфавітно-цифрових знаків та символів.

Розташування екрану відеотерміналу має забезпечувати зручність зорового спостереження у вертикальній площині під кутом +30° від лінії зору працівника.

Клавіатуру слід розміщувати на поверхні столу або на спеціальний робочій поверхні окремо від столу на відстані 100-300 мм. Від краю, ближнього до працівника. Кут нахилу клавіатури має бути в межах 5-15°.

Робоче місце з відеотерміналом слід оснащувати пюпітром (тримачем) для документів, що легко переміщується.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрану відеотерміналу, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля: по висоті 900-1300мм., по глибині 400-500мм.

* 1. **Розрахунок освітлення**

Проведемо розрахунок освітлення дільниці монтажу та складання плати субблоку автоматики аналогового.

Розрахунок штучного робочого освітлення проведемо за методикою згідно [Л5]. Застосуємо метод коефіцієнта використання світлового потоку, який застосовується для розрахунку загального рівномірного освітлення приміщень при відсутності вагомих затемнень робочої поверхні.

Загальне освітлення розраховуємо по формулі:

*(5.2.2.1)*

де: ЕН – нормативна мінімальна освітленість, яка становить, згідно табл. 12.1 [Л5] для виробництв з високою точністю зорової роботи і середнім контрастом об’єкта розрізнення на фоні середньої світлості, ЕН = 300 лк.;

КЗ – коефіцієнт запасу, приймемо КЗ = 1,5;

S – площа дільниці, згідно (3.2.7) розділу 3 S = 24 м2;

Z – коефіцієнт нерівномірності освітлення, вибираємо згідно Z = 1,3;

h – коефіцієнт використання світлового потоку, h = 0,68.

Для освітлення дільниці приймемо люмінесцентні лампи типу ЛБЦ потужністю 40 Вт, які мають номінальний світловий потік Fл = 3039 лм.

Необхідну кількість ламп розраховуємо за формулою:

*(5.2.2)*

Приймемо кількість ламп за 7.

Розрахуємо кількість світильників по формулі:

*(5.2.3)*

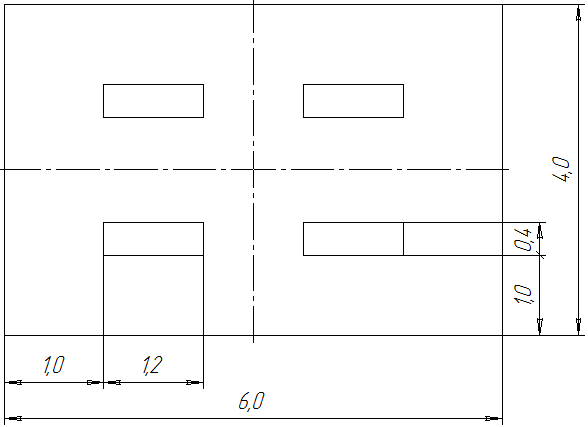
де n – кількість ламп у вибраному типі світильника.

Використаємо на дільниці світильники типу ЛПО-01В-2x40, в яких використовується по дві люмінесцентні лампи.

Кількість світильників:

*шт.*

Приймаємо кількість світильників NС ≈ 4 шт.



* 1. **Заходи з покращення умов праці і техніки безпеки на дільниці**

Одним із основних завдань щодо покращення умов праці є збереження і зміцнення здоров’я працівників, підвищенням працездатності і тривалості життя, а також задоволеність працею.

Виконання цього завдання забезпечується, перш за все, дотриманням правил техніки безпеки на робочих місцях. Організація техніки безпеки на потрібному рівні можлива тільки при умові постійного режимного аналізу виробничої обстановки на основі результатів проведення вимірів параметрів шкідливих виробничих факторів, засобів захисту працівників. Усе це містить Паспорт санітарно-технічного стану цеху, який допомагає отримувати інформацію для повної характеристики виробництва.

Для забезпечення систематичного покращення умов праці на дільниці необхідно намітити комплекс заходів і проводити їх в плановому порядку. Для забезпечення покращення умов праці, для попередження виробничого травматизму і захворюваності, в план робіт по охороні праці мають бути включені такі заходи:

* модернізація виробничого обладнання;
* впровадження автоматичного і дистанційного управління виробничими процесами;
* впровадження систем автоматичного контролю і сигналізації про наявність небезпечних і шкідливих виробничих факторів;
* впровадження блокуючих пристроїв, які забезпечують аварійне відключення виробничого обладнання у випадку його несправності;
* впровадження технічних пристроїв, які забезпечують захист працюючих від враження електричним струмом;
* встановлення запобіжних і захисних пристосувань та технологічному обладнанні;

Важливим заходом також є атестація робочих місць на відповідність до вимог і норм охорони праці, яка має на меті визначення стану технічної безпеки, санітарно-гігієнічних умов. Для покращення умов праці робітників дуже ефективним є проведення ряду заходів, які зменшують втомлюваність, відповідно підвищують продуктивність праці. На підприємствах повинні організовуватись кабінети професійної діагностики, де проводиться професійний відбір на роботу з урахуванням психофізичних, психологічних, фізіологічних особливостей. Одним з найважливіших заходів є раціональне харчування працівників, а також естетичне оформлення виробничого приміщення, робочого місця.

Суттєвим показником виробничого середовища є і сприятливий психологічний клімат, від чого в значній мірі залежить емоційний стан працівників, робочий настрій, при покращенні якого збільшується продуктивність праці, зменшується небезпека травматизму. Необхідним заходом є виробнича гімнастика, фізична пауза, регламентовані перерви, що попереджують втомлюваність.

**ВИСНОВОК**

Використання електронних посібників як сучасних інформаційних технологій в освіті дозволить внести суттєві зміни в процес навчання.

Аналіз питань використання сучасних засобів показує, що розробка електронних посібників в мережі Інтернет розвивається доволі повільно. В дипломному проекті запропонована розробка електронного посібника, як універсального засобу для організації навчання, в ТК ТНТУ та мережі Інтернет.

Електронний посібник відносяться до достатньо складних програмних продуктів і потребують для реалізації не тільки засобів комп’ютерної графіки, але й спеціалізованих інструментальних засобів розробки.

На основі проведеного порівняльного аналізу авторських систем, в якості інструментальних засобів практичної реалізації дипломного проекту, розробку електронного посібника використано HTML та CSS стилі, а також додатково JavaScript та JavaApplet.

Крім цього, в економічній частині дипломного проекту виконано розрахунок економічної ефективності за рік, термін окупності такого електронного посібника, ціну та прибуток, який можна отримати від розробки такого проекту.

Слід відмітити, що електронний посібник може бути корисний не тільки для студентів комп’ютерних спеціальностей, але для інших спеціальностей, які вивчають аналогічні предмети, що дозволяє додатково заощадити кошти.

Результати дипломного проекту в розділі загальні вимоги до розробки електронного посібника можуть використовуватися як база для розробки електронних посібників різних напрямків.

Практична розробка – це електронний посібник для студентів комп’ютерних спеціальностей задуманий як універсальна методична розробка для студентів спеціальності 5.091504 «Обслуговування комп’ютерних та інтелектуальних систем та мереж» при вивченні широкого спектру питань комп’ютерної логіки і не тільки.

Електронний посібник розробляється як відкрита система, яка в подальшому дає змогу доповняти її новими розділами, а також вносити необхідні зміни.

Використання такого посібника в учбовому процесі дозволить підвищити ефективність навчання студентів ТК ТНТУ, а також значно зменшити час використання Інтернет для пошуку інформації з широкого спектру питань при вивченні достатньої кількості питань курсу.

**ПЕРЕЛІК ДЖЕРЕЛ**

1. Посібник користувача ПК. Друге видання. Тхір Л.І., Калушка В.П., Юзьків А.В.
2. Шевченко А.I. Методика розробки електронних підручників для загальноосвітніх середніх шкіл України. — Донецьк, 2001
3. Зарецька І.Т. та інші, Інформатика, ч. 2. — К., Київ — Форум, 2004
4. Гаєвський О.Ю. Інформатика, 7–11 клас. — К: «Видавництво А.С.К.», 2003
5. Глинський Я.М. Інформатика, ч. 2. — Львів, 2004
6. Гуржій А.М., Жук Ю.О., Волинський В.П. Засоби навчання: Навчальний посібник для студентів вищих педагогічних навчальних закладів та слухачів системи післядипломної освіти; АПН України, Інститут педагогіки. — К., 1997.
7. http://www.zippo.net.ua
8. http://www.rusnauka.com
9. http://www.umniki.com.ua
10. http://www.confcontact.com
11. http://www.chpal.edukit.ck.ua
12. http://www.w3schools.org
13. http://www.htmlbook.ru