

Міністерство освіти і науки України  
Запорізький національний технічний університет

**МЕТОДИЧНІ ВКАЗІВКИ**

до виконання розрахунково-графічного завдання  
з дисципліни

**“Архітектура та проектування  
програмного забезпечення”**

для студентів  
напряму підготовки 6.050103  
“Програмна інженерія”  
(всіх форм навчання)

2014

Методичні вказівки до виконання розрахунково-графічного завдання з дисципліни “Архітектура та проектування програмного забезпечення” для студентів напряму підготовки 6.050103 “Програмна інженерія” (всіх форм навчання) / А. О. Олійник, Т. О. Колпакова, В. М. Льовкін. – Запоріжжя : ЗНТУ, 2014. – 27 с.

Автори: А. О. Олійник, к.т.н., доцент  
Т. О. Колпакова, асистент  
В. М. Льовкін, к.т.н., доцент

Рецензент: С. О. Субботін, д.т.н., професор

Відповідальний за випуск: В. І. Дубровін, к.т.н., професор

Затверджено  
на засіданні кафедри  
програмних засобів

Протокол № 1  
від “20” серпня 2014 р.

## ЗМІСТ

<b>Вступ .....</b>	<b>4</b>
<b>1 Основні теоретичні відомості .....</b>	<b>5</b>
1.1 Збирання та аналіз вимог .....	5
1.2 Розроблення технічного завдання .....	6
1.3 Проектування з використанням UML .....	6
1.4 Аналіз архітектури веб-додатків з використанням CMS .....	9
1.5 Проектування інтерфейсу веб-додатків .....	9
1.6 Аналіз структури веб-додатків з використанням CMS .....	9
1.7 Розгортання веб-додатків з використанням CMS .....	10
1.8 Розширення функціональності веб-додатків за допомогою модулів .....	11
1.9 Аналіз якості та оцінка веб-додатків .....	12
<b>2 Порядок виконання розрахунково-графічного завдання .....</b>	<b>14</b>
<b>3 Вимоги до змісту пояснювальної записки .....</b>	<b>17</b>
3.1 Реферат .....	17
3.2 Зміст .....	17
3.3 Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	18
3.4 Вступ .....	18
3.5 Основна частина .....	18
3.6 Висновки .....	22
3.7 Перелік посилань .....	23
3.8 Додатки .....	24
<b>Література .....</b>	<b>27</b>

## ВСТУП

Дане видання призначене для вивчення та практичного освоєння студентами усіх форм навчання основ архітектури та проектування програмного забезпечення.

Відповідно до графіка студенти перед виконанням розрахунково-графічного завдання повинні виконати лабораторні роботи та ознайомитися з конспектом лекцій та рекомендованою літературою. Звичайно, в дані методичні вказівки неможливо було внести весь матеріал, необхідний для виконання та захисту розрахунково-графічного завдання. Тому тут містяться основні, базові теоретичні відомості, необхідні для виконання розрахунково-графічного завдання. Таким чином для виконання розрахунково-графічного завдання та при підготовці до його захисту необхідно ознайомитись з конспектом лекцій та проробити весь матеріал, наведений в переліку рекомендованої літератури. При цьому не варто обмежуватись лише наведеним списком.

Для одержання заліку з роботи студент здає викладачу цілком оформлений звіт, а також демонструє на екрані комп'ютера результати виконання роботи.

Звіт має містити:

- титульний аркуш;
- тему та мету роботи;
- завдання до роботи;
- лаконічний опис теоретичних відомостей;
- результати виконання лабораторної роботи;
- змістовний аналіз отриманих результатів та висновки.

Звіт виконують на білому папері формату А4 (210 × 297 мм). Текст розміщують тільки з однієї сторони листа. Поля сторінки з усіх боків – 20 мм. Аркуші скріплюють за допомогою канцелярських скріпок або вміщують у канцелярський файл.

Під час співбесіди при захисті розрахунково-графічного завдання студент повинен виявити знання про мету роботи, про методи виконання кожного етапу роботи, по змісту основних розділів оформленого звіту з демонстрацією результатів на конкретних прикладах. Студент повинний вміти правильно аналізувати отримані результати. Для самоперевірки при підготовці до виконання і захисту роботи студент повинен відповісти на контрольні запитання.

# 1 ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

## 1.1 Збирання та аналіз вимог

Аналіз вимог полягає в визначенні потреб та умов які висуваються щодо нового, чи зміненого продукту, враховуючи можливо конфліктні вимоги різних замовників.

Аналіз вимог є критичним для успішної розробки проекту. Вимоги мають бути задокументованими, вимірними, тестовними, пов'язаними з бізнес-потребами, і описаними з рівнем деталізації достатнім для конструювання системи.

### 1.1.1 Класифікація вимог

**Вимоги споживача** представляють собою вирази фактів та припущень які описують очікування до системи в термінах цілей, середовища, обмежень, та міри ефективності й придатності.

**Архітектурні вимоги** пояснюють що має бути зроблено ідентифікацією необхідної системної архітектури.

**Структурні вимоги** пояснюють що має бути зроблено ідентифікацією необхідної структури системи.

**Поведінкові вимоги** пояснюють що має бути зроблено ідентифікацією необхідної поведінки системи.

**Функціональні вимоги** пояснюють що має бути зроблено ідентифікацією необхідної задачі, дії, чи діяльності які мають виконуватись. Аналіз функціональних вимог буде використаний в функціях верхніх рівнів для функціонального аналізу.

**Нефункціональні вимоги** задають критерій для оцінки операцій системи, замість її поведінки.

**Вимоги продуктивності** пояснюють, до якої міри місії чи функції повинні бути виконані; зазвичай вимірюється в термінах кількості, якості, охопленні, своєчасності чи готовності. Протягом аналізу вимог, вимоги продуктивності будуть інтерактивно розроблятися вздовж всіх виявлених функції що базуються на факторах життєвого циклу системи, і характеризуються в термінах ступеня визначеності в їх оцінках, ступеня критичності успіху системи, і їх відношення до інших вимог.

**Вимоги дизайну.**

**Успадковані вимоги**, тобто вимоги які маютья наувазі вимогами вищого рівня, чи перетворені з них.

**Розподілені вимоги** визначені поділом, чи іншим перерозміщенням високорівневих вимог в кілька низькорівневих вимог.

## **1.2 Розроблення технічного завдання**

Технічне завдання (ТЗ) – вихідний документ для розробки автоматизованої системи або створення програмного продукту, відповідно до якого проводиться виготовлення, приймання при введенні в дію та експлуатація відповідного об'єкта. ТЗ є основним документом, що визначає вимоги і порядок створення (розвитку або модернізації) інформаційної системи.

### **1.2.1 Етапи складання списку вимог ТЗ**

Робота над ТЗ включає виконання низки етапів, а невизначеність, властива цій роботі, викликає проходження їх по кілька разів, ітераційно, від більш загальної постановки завдання до детального опрацювання (проектування носить ітераційний характер і те, що не враховано на початку, може бути враховано на наступних етапах).

Основні етапи розробки ТЗ:

- аналіз завдання заказчика;
- конкретизація цілей проектування;
- обробка зібраної інформації:
  - 1) узагальнення та абстрагування;
  - 2) перевірка на суперечливість;
  - 3) розмежування вимог на умови, обмеження та показники якості;
  - 4) параметризація;
  - 5) зменшення списку вимог;
  - 6) зведення вимог та затвердження замовником.

## **1.3 Проектування з використанням UML**

Робота над ТЗ включає виконання низки етапів, а невизначеність, властива цій роботі, викликає проходження їх по кілька разів, ітераційно, від більш загальної постановки завдання до детального

опрацювання (проектування носить ітераційний характер і те, що не враховано на початку, може бути враховано на наступних етапах).

### **1.3.1 Діаграма класів**

Діаграму класів (class diagram) використовують для подання статистичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування.

### **1.3.2 Діаграма пакетів**

Діаграма пакетів є різновидом діаграми класів, на якій відображаються тільки пакети і залежності.

Між двома пакетами існує деяка залежність, якщо існує яка-небудь залежність між будь-якими двома класами в пакетах.

Хоча пакети не дають відповіді на питання, як зменшити кількість залежностей у вашій системі, проте вони допомагають виділити ці залежності. Як тільки вони опиняться на виду, вам залишиться лише попрацювати над їх скороченням.

### **1.3.3 Діаграма компонентів**

Діаграма компонентів (рис. 3.3) описує особливості фізичної уяви системи. Діаграма компонентів відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись, тобто вона дозволяє визначити архітектуру системи, що розробляється, встановивши залежності між програмними компонентами.

### **1.3.4 Діаграма прецедентів**

Діаграма прецедентів – діаграма, на якій зображено відношення між акторами та прецедентами в системі.

Проектована система представляється у вигляді множини сутностей чи акторів, взаємодіючих із системою за допомогою варіантів використання (use case), що служать для опису сервісів, що система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій системи при діалозі з актором.

### **1.3.5 Діаграма станів**

Діаграми станів зображають всі можливі стани, в яких може знаходитися конкретний об'єкт, а також зміни стану об'єкту, які відбуваються в результаті впливу деяких подій на цей об'єкт. У більшості об'єктно-орієнтованих методів діаграми станів будуються для єдиного класу, щоб показати динаміку поведінки єдиного об'єкту.

### **1.3.6 Діаграма діяльності**

Діаграма діяльності показує переходи між видами діяльності. Модель видів діяльності (activity model) може подавати в графічній формі потік подій для прецеденту.

Кожен прецедент можна моделювати за допомогою одного або декілька графів видів діяльності.

Подія, джерелом якої служить суб'єкт прецедент, що ініціює, це та ж сама подія, що запускає виконання графа видів діяльності. Процес виконання послідовно переходить від одного стану виду діяльності до іншого.

### **1.3.7 Діаграма кооперації**

Діаграма кооперації призначена для специфікації структурних аспектів взаємодії. Головна особливість діаграми кооперації полягає в можливості графічно уявити не тільки послідовність взаємодії, але і всі структурні відносини між об'єктами, що беруть участь в цій взаємодії.

### **1.3.8 Діаграма послідовності**

Діаграма послідовності показує учасників взаємодій і послідовність повідомлень, якими вони обмінюються.

На діаграмі послідовності зображуються виключно ті об'єкти, що безпосередньо беруть участь у взаємодії і не показуються можливі статичні асоціації з іншими об'єктами. Для діаграми послідовності ключовим моментом є саме динаміка взаємодії об'єктів у часі.



## 1.4 Аналіз архітектури веб-додатків з використанням CMS

CMS (Content Management System – система управління вмістом) – програмний комплекс, що надає функції створення, редагування, контролю та організації структури веб-додатку (сайту), а також текстової та графічної інформації (вмісту).

Переваги використання CMS при розробці:

- протестований і перевірений багатьма розробниками код;
- універсальність;
- простота використання.

Недоліки використання CMS при розробці:

- обмеження предметної області тими об'єктами які передбачені в CMS;
- визначення основної логіки роботи програми розробниками CMS. Внесення змін в логіку проблематично;
- зайва універсальність і надлишковий функціонал.

## 1.5 Проектування інтерфейсу веб-додатків

Інтерфейс (від англ. Interface – поверхня розділу, перегородка) – сукупність засобів і методів взаємодії між елементами системи. Залежно від контексту, поняття застосовне як до окремого елемента (інтерфейс елемента), так і до зв'язків елементів (інтерфейс сполучення елементів).

Інтерфейс користувача – сукупність засобів, за допомогою яких користувач спілкується з різними пристроями:

- інтерфейс командного рядка – інструкції програмою або пристроєм здійснюється шляхом введення з клавіатури текстових рядків;
- графічний інтерфейс – управління програмними функціями реалізовано графічними елементами екрану.

## 1.6 Аналіз структури веб-додатків з використанням CMS

Базові структурні компоненти CMS:

База даних. CMS, як правило, забезпечують зберігання даних в єдиному репозиторії (сховищі).

Панель управління – окрема частина програми, що має розширений доступ і дозволяє спростити обслуговування ресурсу.

Ядро системи – основна частина CMS, яка управляє показом сторінок користувачеві, всієї навігацією, відповідно реалізує всі додаткові функції, покладені на нього.

Шаблонізатор – частина системи, що дозволяє розробляти і впроваджувати інтерфейс не торкаючись ядра системи.

Додаткові модулі – незалежні частини веб-додатку, що дозволяють розширити його функціональність на базі ядра системи.

## 1.7 Розгортання веб-додатків з використанням CMS

Веб-сервер – це сервер, який приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, і видає їм HTTP-відповіді разом із затребуваними даними: HTML-сторінкою, зображенням, файлом, медіа-потоком і т.д.

На серпень 2011 року найбільш поширеним веб-сервером, що займає більше 65% ринку, є вільний веб-сервер Apache.

Ядро Apache включає в себе основні функціональні можливості, такі як обробка конфігураційних файлів, протокол HTTP і система завантаження модулів. Apache HTTP Server підтримує модульність.

У модулях реалізуються такі можливості, як:

- підтримка мов програмування;
- додавання функціоналу;
- виправлення помилок або модифікація основних функцій;
- посилення безпеки.

LAMP – акронім, що позначає набір (комплекс) серверного програмного забезпечення, який широко використовують в Інтернеті. LAMP названий за першими літерами продуктів, що входять до його складу:

- Linux – операційна система Linux;
- Apache – веб-сервер;
- MySQL – СУБД;
- PHP – мова програмування, що використовується для створення веб-додатків (крім PHP можуть матися на увазі інші мови, такі як Perl і Python).

Аналогічно йому існує WAMP:

- Windows – операційна система від компанії Microsoft;
- Apache;
- MySQL;

– PHP.

Хоча спочатку ці програмні продукти не розроблялися спеціально для роботи один з одним, однак така зв'язка стала вельми популярною через свою гнучкості, продуктивності та низької вартості.

Розгортання CMS в більшості випадків складається з таких етапів:

- створення бази даних в доступній СУБД (MySQL найчастіше);
- завантаження інсталяційного пакета на сервер;
- попередня настройка через систему веб-доступу;
- підключення до бази даних;
- встановлення.

## **1.8 Розширення функціональності веб-додатків за допомогою модулів**

Більшість CMS зроблені за принципом модульності: є базова основа, до якої можливе підключення модулів. Модулі можуть вже входити до складу комплексу, можуть бути встановлені окремо, а також багато систем допускають самостійну розробку та включення додаткових модулів. Таким чином, загальний функціонал сайту залежить від функцій включених в нього модулів.

Фактично зміст тієї чи іншої сторінки залежить від модуля який реалізує функціональний рівень (бізнес-логіку).

Переваги модульних CMS наведено нижче.

### **Простота сприйняття**

Чітко позначені кордони, зв'язність модулів робить їх легкими для розуміння. Вивчення системи невеликими частинами, в кінцевому підсумку призводить до більш глибокого її розуміння.

### **Змінність**

Якщо кожен модуль системи відомий тільки через інтерфейс, це дозволяє легко замінити один модуль на інший, з таким само інтерфейсом.

### **Паралельна розробка**

Завдяки своїй незалежності, модулі можуть розроблятися паралельно. Для команди розробників це дає можливість розподілу завдань по межах модулів.

### **Багаторазовість і гнучкість використання**

Кожен модуль, взятий з однієї системи, може бути використаний в іншій. Тут все залежить від його функціоналу і області застосування.

### **Покращення тестування**

Поряд з інтеграційним і модульним тестуванням існує можливість тестування кожного модуля окремо, як єдиного цілого.

## **1.9 Аналіз якості та оцінка веб-додатків**

Тестування програмного забезпечення (Software Testing) – це процес технічного дослідження, який виконується на вимогу замовників, і призначений для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись.

Якість програмного забезпечення – характеристика програмного забезпечення, ступінь відповідності ПЗ до вимог. Частіше за все, використовують визначення ISO 9001, згідно з яким якість – це «ступінь відповідності наявних характеристик вимогам».

Фактори якості – це нефункціональні вимоги до ПЗ, що відносяться до, наприклад, надійності та продуктивності програм.

Деякі з факторів якості:

**Зрозумілість.** Призначення ПЗ повинно бути зрозумілим з самої програми та документації.

**Повнота.** Всі необхідні частини програми повинні бути представлені та реалізовані.

**Стислість.** Відсутність надлишкової інформації та такою, що дублюється. Реалізація принципів DRY.

**Можливість портування.** Легкість в адаптації програми до інших умов: архітектури, платформи, операційній системі тощо.

**Узгодженість.** Вся документація та код повинні виконуватися за єдиними угодами, використовувати єдині формати та позначення

**Покриття тестуванням.**

**Зручність використання.**

**Надійність.**

**Безпечність.**

Модульне тестування тестує мінімальний компонент програми, або модуля. Кожний модуль тестується для перевірки правильності його реалізації.

**Інтеграційне тестування** виявляє дефекти в інтерфейсах та у взаємодії між компонентами (модулями).

**Системне тестування** тестує інтегровану систему для перевірки відповідності всім вимогам.

**Системне інтеграційне тестування** перевіряє, чи система інтегрується в будь-яку зовнішню систему (або системи) відповідно до системних вимог.

**Приймальне тестування** може проводитись кінцевим користувачем, замовником, або клієнтом для перевірки, чи може продукт бути прийнятий до використання:

– **альфа-тестування** – це симульоване або реальне операційне тестування потенційними користувачами/замовником або командою тестувальників на боці розробника.

– **бета-тестування** йде після альфа-тестування. Версії програмного забезпечення, відомі як бета-версії, надаються у користування обмеженій кількості людей поза компанією для того, щоб упевнитись, що програма не містить великої кількості помилок.

## **2 ПОРЯДОК ВИКОНАННЯ РОЗРАХУНКОВО-ГРАФІЧНОГО ЗАВДАННЯ**

2.1 Обрати тему для подальшого проектування системи.

2.2 Розробити загальну схему системи за обраною темою.

2.3 Виконати стислий опис проекту: сформулювати цілі та межі проекту, визначити учасників проекту, розглянути існуючі рішення проблеми.

2.4 Визначити системні сервіси: межі системи, функціональні вимоги, вимоги до даних.

2.5 Сформулювати системні обмеження: вимоги до інтерфейсу, продуктивності, безпеки, експлуатаційні вимоги.

2.6 Визначити проектні питання: представити попередній план-графік виконання основних проектних завдань, попередній бюджет, розглянути питання, які впливають на успіх проекту і не були розглянуті в інших пунктах.

2.7 Сформулювати додатки до документу опису вимог: глосарій, посилання.

2.8 Визначити характеристику області застосування програмного забезпечення, підстави розроблення системи та призначення розробки.

2.9 Сформулювати основні вимоги до програмного забезпечення та до програмної документації. Оформити специфікацію вимог.

2.10 Визначити випробування для контролю та приймання програмного продукту.

2.11 Сформулювати технічне завдання на проект системи за обраною темою.

2.12 Розробити абстрактну модель системи за обраною темою на основі діаграми класів.

2.13 Розробити абстрактну модель системи за обраною темою на основі діаграми пакетів.

2.14 Розробити абстрактну модель системи за обраною темою на основі діаграми компонентів.

2.15 Розробити абстрактну модель системи за обраною темою на основі діаграми прецедентів.

2.16 Розробити діаграми прецедентів для представлення варіантів використання системи, наведених на загальній діаграмі прецедентів.

2.16 Розробити абстрактну модель системи за обраною темою на основі діаграми діяльності.

2.17 Розробити абстрактну модель системи за обраною темою на основі діаграми станів.

2.18 Розробити абстрактну модель системи за обраною темою на основі діаграм кооперації.

2.19 Розробити абстрактну модель системи за обраною темою на основі діаграм послідовності.

2.20 Обґрунтувати вибір способу побудови архітектури веб-додатку: на основі предметної області або CMS.

2.21 Розробити архітектуру веб-додатку на основі обраного підходу (та обґрунтувати вибір CMS у випадку використання даного підходу).

2.22 Визначити особливості роботи кожної групи користувачів з кожним елементом взаємодії користувача з веб-додатком.

2.23 Розробити інтерфейс для веб-додатку, що реалізує систему.

2.24 Обґрунтувати вибір запропонованих елементів інтерфейсу, ґрунтуючись на вимогах до інтерфейсу та ергономічних показниках інтерфейсу.

2.25 Розробити структуру веб-додатку на основі обраного підходу.

2.26 Виконати конфігурацію сервера та, якщо обрано підхід на основі CMS, інсталювати обрану CMS на сервер відповідно до інструкцій для даної CMS.

2.37 Проаналізувати вимоги до функціональності, що визначені в оформленій документації на програмне забезпечення.

2.28 Визначити необхідні модулі розширення функціональності та застосувати обрані модулі для веб-додатку (або реалізувати функціональність системи у випадку підходу на основі предметної області).

2.29 Внести необхідні зміни у результати попередньо виконаних етапів (зокрема проектування та розроблення архітектури системи), якщо в процесі реалізації їх було відкориговано.

2.30 Розробити тести для аналізу якості веб-додатків.

2.31 Провести тестування сумісності.

2.32 Проаналізувати реалізованість функціональних вимог до програмного забезпечення.

2.33 Перевірити ергономічність веб-додатку.

2.34 Провести тестування безпеки веб-додатку.

2.35 Провести тестування стабільності.

2.36 Розробити специфікацію на програмне забезпечення.

2.37 Розробити опис програми.

2.38 Розробити керівництво програміста.

2.39 Розробити керівництво користувача.



## **3 ВИМОГИ ДО ЗМІСТУ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ**

### **3.1 Реферат**

Реферат призначений для ознайомлення із пояснювальною запискою. Він має бути стислим, інформативним і містити відомості, які дозволяють прийняти рішення про доцільність читання всієї пояснювальної записки. Реферат повинен містити: відомості про обсяг роботи, кількість частин пояснювальної записки, кількість ілюстрацій, таблиць, додатків, кількість джерел згідно з переліком посилань (усі відомості наводять, включаючи дані додатків); текст реферату; перелік ключових слів. Текст реферату повинен відбивати подану у пояснювальній записці інформацію і, як правило, у такій послідовності: об'єкт дослідження або розробки; мета роботи; методи дослідження та апаратура; результати та їх новизна; основні конструктивні, технологічні й техніко-експлуатаційні характеристики та показники; ступінь впровадження; взаємозв'язок з іншими роботами; рекомендації щодо використання результатів роботи; галузь застосування; економічна ефективність; значущість роботи та висновки; прогнозні припущення про розвиток об'єкта дослідження або розроблення. Частини тексту реферату, щодо яких відсутні відомості, випускають. Реферат належить виконувати обсягом не більш як 500 слів, і, бажано, щоб він розміщувався на одній сторінці формату А4. Ключові слова, що є визначальними для розкриття суті роботи, розміщують після тексту реферату. Перелік ключових слів містить від 5 до 15 слів (словосполучень), надрукованих великими літерами в називному відмінку в рядок через коми.

### **3.2 Зміст**

Зміст розташовують безпосередньо після реферату, починаючи з нової сторінки. До змісту включають: перелік умовних позначень, символів, одиниць, скорочень і термінів; передмову; вступ; послідовно перелічені назви всіх розділів, підрозділів, пунктів і підпунктів (якщо вони мають заголовки) суті пояснювальної записки; висновки; рекомендації; перелік посилань; назви додатків і номери сторінок, які міс-

тять початок матеріалу. У змісті можуть бути перелічені номери й назви ілюстрацій та таблиць з зазначенням сторінок, на яких вони вміщені. Зміст складають, якщо пояснювальна записка містить не менш ніж два розділи або один розділ і додаток за загальної кількості сторінок не менше десяти.

### **3.3 Перелік умовних позначень, символів, одиниць, скорочень і термінів**

Усі прийняті у записці малопоширені умовні позначення, символи, одиниці, скорочення і терміни пояснюють у переліку, який вміщують безпосередньо після змісту, починаючи з нової сторінки.

Незалежно від цього за першої появи цих елементів у тексті записки наводять їх розшифровку. Перелік умовних позначень, символів, одиниць, скорочень і термінів повинен розташовуватись стовпцем. Ліворуч в алфавітному порядку наводять умовні позначення, символи, одиниці, скорочення і терміни, праворуч - їх детальну розшифровку.

### **3.4 Вступ**

Вступ розташовують на окремій сторінці. У вступі коротко викладають: оцінку сучасного стану проблеми, відмічаючи практично розв'язані задачі, прогалини знань, що існують у даній галузі, провідні фірми та провідних вчених і фахівців даної галузі; світові тенденції розв'язання поставлених задач; актуальність даної роботи та підставу для її виконання; мету роботи та галузь застосування; взаємозв'язок з іншими роботами.

### **3.5 Основна частина**

Основна частина пояснювальної записки складається з розділів, підрозділів, пунктів і підпунктів і містить наступні розділи:

- вимоги до системи;
- розроблення моделей системи;
- розроблення архітектури та структури системи;
- керівництво програміста;

– керівництво оператора.

### **3.5.1 Вимоги до системи**

В першому розділі виконується стислий огляд стану проблеми, де зокрема проводиться аналіз існуючих рішень даної проблеми, та збираються вимоги до програмного забезпечення з метою визначення потреб, які має задовольнити розроблення даної системи.

У межах виконання огляду стану проблеми, зокрема, має бути проведено аналіз аналогічного програмного забезпечення та виконано обґрунтування доцільності власної розробки.

Вимоги до програмного забезпечення мають включати наступні пункти:

- стислий опис проекту: цілі та межі проекту, учасники проекту, існуючі рішення проблеми;
- системні сервіси: межі системи, функціональні вимоги, вимоги до даних;
- системні обмеження: вимоги до інтерфейсу, продуктивності, безпеки, експлуатаційні вимоги;
- проектні питання: попередній план-графік виконання основних проектних завдань, попередній бюджет, нерозглянуті питання, які впливають на успіх проекту;
- глосарій;
- посилання.

### **3.5.2 Розроблення моделей системи**

У другому розділі наводяться результати моделювання системи на основі використання UML (Unified Modeling Language).

У межах даного розділу наводяться наступні схематичні зображення та виконується їх опис:

- загальна схема роботи системи;
- абстрактні моделі системи на основі структурних діаграм: діаграм класів, пакетів та компонентів;
- абстрактні моделі системи на основі діаграм поведінки: діаграм прецедентів, діяльності та станів;
- абстрактні моделі системи на основі діаграм взаємодії: діаграм

кооперації та послідовності

Схеми та моделі необхідно наводити у відкоригованому після реалізації системи вигляді.

### **3.5.3 Розроблення архітектури та структури системи**

У третьому розділі виконується розроблення архітектури та структури системи, що проектується, а також інтерфейсу веб-додатку, що реалізує дану систему.

Архітектура та структура системи проектується на основі використання системи керування вмістом або на основі предметної області. У подальшому архітектура модифікується на основі застосування модулів обраної CMS для розширення функціональності системи або на основі реалізації функціональності за використання підходу на основі предметної області.

Інтерфейс веб-додатку має враховувати вимоги до інтерфейсу програмного забезпечення на основі документу опису вимог та технічного завдання та особливості роботи кожної групи користувачів з кожним елементом взаємодії користувача з веб-додатком.

У даному розділі мають бути наведені наступні схематичні зображення та виконано їх змістовний опис:

- архітектура веб-додатку на основі обраного підходу;
- інтерфейс веб-додатку, що реалізує систему;
- структура веб-додатку на основі використання обраної CMS;
- архітектура веб-додатку за використання модулів;
- структура веб-додатку за використання модулів.

### **3.5.4 Керівництво програміста**

Четвертий розділ пояснювальної записки повинен містити документ “Керівництво програміста” до розробленого програмного продукту, оформлений відповідно до вимог діючих стандартів.

Керівництво програміста оформлюється відповідно до ГОСТу 19.503–79, згідно з яким воно має містити наступні розділи:

- призначення й умови застосування програми;
- характеристики програми;
- звертання до програми;

- вхідні і вихідні дані;
- повідомлення.

У залежності від особливостей документа допускається поєднувати окремі розділи чи вводити нові.

У розділі “Призначення й умови застосування програми” повинні бути зазначені призначення і функції, які виконуються програмою, умови, необхідні для виконання програми (обсяг оперативної пам’яті, вимоги до складу і параметрів периферійних пристроїв, вимоги до програмному забезпечення і т. ін.).

У розділі “Характеристика програми” повинен бути приведений опис основних характеристик і особливостей програми (часові характеристики, режим роботи, засоби контролю правильності виконання і самовідновлення програми і т. ін.).

У розділі “Звертання до програми” повинен бути наведений опис процедур виклику програми (способи передачі керування і параметрів даних та ін.).

У розділі “Початкові та вихідні дані” повинен бути приведений опис організацій початкової і вихідної інформації, що використовується, при необхідності, її кодування.

У розділі “Повідомлення” повинні бути зазначені тексти повідомлень, які видаються програмісту чи оператору в ході виконання програми, опис їхнього змісту і дії, які необхідно виконати при цих повідомленнях.

У додатку до керівництва програміста мають бути наведені алгоритм розгортання обраної CMS на сервері та тести для аналізу якості веб-додатків, а також результати тестування сумісності, безпеки та стабільності програмного забезпечення, результати перевірки ергономічності. Також у додатку можуть бути наведені додаткові матеріали (прикладі, ілюстрації, таблиці, графіки і т. ін.).

### **3.5.5 Керівництво оператора**

Керівництво оператора оформлюється згідно ГОСТу 19.505–78, відповідно до якого воно має містити наступні розділи:

- призначення програми,
- умови виконання програми,
- виконання програми,
- повідомлення оператору.

В залежності від особливостей документа допускається об'єднувати окремі розділи чи вводити нові.

В розділі “Призначення програми” повинні бути зазначені відомості про призначення програми й інформація, достатня для розуміння функцій програми і її експлуатації.

В розділі “Умови виконання програми” повинні бути вказані умови, необхідні для виконання програми (мінімальний та/або максимальний склад апаратних і програмних засобів і т. ін.).

В розділі “Виконання програми” повинна бути зазначена послідовність дій оператора, що забезпечують завантаження, запуск, виконання й завершення програми, приведений опис функцій, формату і можливих варіантів команд, за допомогою яких оператор здійснює завантаження і керує виконанням програми, а також відповіді програми на ці команди.

В розділі “Повідомлення оператору” повинні бути наведені тексти повідомлень, що видаються у ході виконання програми, опис їхнього змісту і відповідні дії оператора (дії оператора у випадку збою, можливості повторного запуску програми і т. ін.).

Допускається зміст розділів ілюструвати прикладами, таблицями, схемами, графіками.

В додатки до керівництва оператора допускається включати різні матеріали, що недоцільно включати в розділи керівництва.

### **3.6 Висновки**

Висновки вміщують безпосередньо після викладення суті записки, починаючи з нової сторінки. У висновках наводять найбільш важливі результати, одержані в розрахунково-графічному завданні, виконують оцінку одержаних результатів роботи з урахуванням світових тенденцій вирішення поставленої задачі; можливі галузі використання результатів роботи; народногосподарську, наукову, соціальну значущість роботи.

У висновках необхідно наголосити на якісних та кількісних показниках отриманих результатів, обґрунтувати достовірність результатів, викласти рекомендації щодо їх використання.

Текст висновків може поділятися на пункти. Рекомендований обсяг висновків – 3 – 4 сторінки.

### 3.7 Перелік посилань

Перелік посилань наводять у кінці тексту записки, починаючи з нової сторінки. У відповідних місцях тексту мають бути посилання.

Перелік посилань є складовою частиною роботи і відображує ступінь вивчення даної проблеми автором. Рекомендований обсяг переліку посилань – не менше 25 літературних джерел.

Бібліографічні описи в переліку посилань подають у порядку, за яким вони вперше згадуються в тексті. Порядкові номери описів у переліку є посиланнями в тексті (номерні посилання). За необхідності джерела, на які є посилання тільки в додатку, наводять у окремому переліку посилань, який розташовують у кінці цього додатка.

Бібліографічний опис джерел складають відповідно до чинних стандартів з бібліотечної та видавничої справи. Зокрема, потрібну інформацію можна одержати із таких стандартів: ГОСТ 7.1 – 84 “Библиографическое описание документа. Общие требования и правила составления”, ДСТУ 3582-97 “Інформація та документація. Скорочення слів в українській мові в бібліографічному описі. Загальні вимоги та правила”, ГОСТ 7.12-93 “Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила”.

У якості обов’язкових елементів бібліографічного опису книги виступають: відомості про назву, перелік авторів, місце видання, рік видання, кількість сторінок.

Книга може мати окрім основної назви додаткову. Перед додатковою назвою, що пояснює зміст основної назви, а також перед іншими відомостями, які відносяться до назви (підручник, навчальний посібник та ін.), ставлять двокрапку.

Якщо книга має більше трьох авторів, відомості про авторів (редакторах, укладачах) наводять після даних, що відносяться до області назви. Перед прізвищами авторів, редакторів, укладачів ставиться коса риска. При кількості авторів більше чотирьох після косої риски наводять прізвища та ініціали перших трьох авторів з додаванням скорочення «та ін.».

Найменування місця видання дається повністю у називному відмінку. У скороченому вигляді прийнято вказувати тільки назви наступних міст: Москва (М.), Ленінград (Л.), Санкт-Петербург (СПб.), Київ (К.).

Перед назвою видавництва ставиться двокрапка, а після нього – кома та рік видання.

На складову частину видання (статтю, главу, розділ та ін.) складають бібліографічний опис, який містить відомості про статтю (главу, частину, розділ) та відомості про видання де вона опублікована.

Перед відомостями про видання ставляться дві косі риски. Замість загальної кількості сторінок вказують початкову та кінцеву сторінки частини видання.

### 3.8 Додатки

Додатки можуть містити матеріал, який є необхідним для повноти пояснювальної записки, але включення його до основної частини записки може змінити впорядковане й логічне уявлення про роботу; не може бути послідовно розміщений в основній частині записки через великий обсяг або способи відтворення; може бути вилучений для широкого кола читачів, але є необхідним для фахівців даної галузі.

У додатки можуть бути включені: додаткові ілюстрації або таблиці; матеріали, які через великий обсяг, специфіку викладення або форму подання не можуть бути внесені до основної частини (оригінали фотографій, проміжні математичні докази, формули, розрахунки; протоколи випробувань; копія технічного завдання, інструкції, методики, опис комп'ютерних програм, розроблених у процесі виконання роботи та ін.).

У додатках обов'язково має бути представлена наступна документація:

- технічне завдання;
- специфікація;
- опис програми
- текст програми.

Технічне завдання на проект системи має включати наступні пункти:

- вступ: найменування та коротка характеристика області застосування програмного забезпечення;
- підстави для розроблення: умови для проведення розробки, найменування та умовне позначення теми розробки;
- призначення розробки: функціональне та експлуатаційне при-



значення;

- основні вимоги до програмного забезпечення: вимоги до функціональних характеристик, вимоги до надійності, умови експлуатації, вимоги до складу та параметрів технічних засобів, вимоги до маркування та упакування, вимоги до транспортування та збереження, вимоги до програмної документації;

- стадії та етапи розробки;

- порядок контролю та приймання системи.

Специфікація на програмне забезпечення складається з наступних розділів:

- документація;

- комплекси;

- компоненти.

Документ “Опис програми” має вміщувати наступну інформацію:

- загальні відомості;

- функціональне призначення;

- опис логічної структури;

- використані технічні засоби;

- виклик та завантаження;

- вхідні та вихідні дані.

Якщо додатки оформлюють на наступних сторінках записки, кожний такий додаток повинен починатися з нової сторінки. Додаток повинен мати заголовок, надрукований вгорі малими літерами з першої великої симетрично відносно тексту сторінки. Посередині рядка над заголовком малими літерами з першої великої повинно бути надруковано слово "Додаток \_\_\_" і велика літера, що позначає додаток.

Додатки слід позначати послідовно великими літерами української абетки, за винятком літер Г, Є, З, І, Ї, Й, О, Ч, Ь, наприклад, додаток А, додаток Б і т. д. Один додаток позначається як додаток А.

Додатки повинні мати спільну з рештою записки наскрізну нумерацію сторінок.

За необхідності текст додатків може поділятися на розділи, підрозділи, пункти і підпункти, які слід нумерувати в межах кожного додатка відповідно до наведених вище вимог. У цьому разі перед кожним номером ставлять позначення додатка (літеру) і крапку, напри-

клад, А.2 – другий розділ додатка А; Д.4.2 – пункт 4.2 додатка Д; Ж. 1.3.3.4 – підпункт 1.3.3.4 додатка Ж.

Ілюстрації, таблиці, формули та рівняння, що є у тексті додатка, слід нумерувати в межах кожного додатка, наприклад, рисунок А.3 – третій рисунок додатка А; таблиця А.2 – друга таблиця додатка А; формула (А.1) – перша формула додатка А. Якщо в додатку одна ілюстрація, одна таблиця, одна формула, одне рівняння, їх нумерують, наприклад, рисунок А.1, таблиця А.1, формула (В.1). В посиланнях у тексті додатка на ілюстрації, таблиці, формули, рівняння рекомендується писати: "... на рисунку А.2 ...", "... на рисунку А.1 ..." – якщо рисунок єдиний у додатку А; "... в таблиці Б.3 ...", або "... в табл. Б. 3 ..."; "... за формулою (В.1) ...", "... у рівнянні (Г.2) ...".

Джерела, що цитують тільки у додатках, повинні розглядатися незалежно від тих, які цитують в основній частині записки, і повинні бути перелічені наприкінці кожного додатка в переліку посилань.

Форма цитування, правила складання переліку посилань і виносок повинні бути аналогічними прийнятим у основній частині записки. Перед номером цитати і відповідним номером у переліку посилань і виносках ставлять позначення додатка.

Якщо у записці як додаток використовується документ, що має самостійне значення і оформлюється згідно з вимогами до документа даного виду, його копію вміщують у записці без змін в оригіналі. Перед копією документа вміщують аркуш, на якому посередині друкують слово "Додаток \_\_\_" і його назву (за наявності), праворуч у верхньому куті аркуша проставляють порядковий номер сторінки. Сторінки копії документа нумерують, продовжуючи наскрізну нумерацію сторінок записки (не займаючи власної нумерації сторінок документа).

## ЛІТЕРАТУРА

1. Мацяшек, Л.А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML : Пер. с англ. [Текст] / Л.А. Мацяшек. – М. : Издательский дом “Вильямс”, 2002. – 432 с., ил.
2. Вигерс, К.И. Разработка требований к программному обеспечению : Пер. с англ. [Текст] / К.И. Вигерс. – М. : Издательско-торговый дом “Русская Редакция”, 2004. – 576с. : ил.
3. Буч, Г. Язык UML. Руководство пользователя [Текст] / Гради Буч, Джеймс Рамбо, Ивар Якобсон. – М. : ДМК Пресс, 2006. – 496 с.
4. Леоненков, А.В. Самоучитель UML. 2-е издание [Текст] / А.В. Леоненков. – СПб. : “БХВ – Петербург”, 2004. – 432 с.
5. Раскин, Дж. Интерфейс : новые направления в проектировании компьютерных систем : Пер. с англ. [Текст] / Джеф Раскин. – СПб. : Символ-Плюс, 2004. – 272 с.
6. Скотт, Б. Проектирование веб-интерфейсов : Пер. с англ. [Текст] / Б. Скотт, Т. Нейл. – СПб. : Символ-Плюс, 2010. – 352 с., ил.
7. Томлинсон, Т. CMS Drupal 7 : руководство по разработке системы управления веб-сайтом, 3-е издание/ Т. Томлинсон. – М. : “Вильямс”, 2011. – 560 с.
8. Граф, Х. Создание веб-сайтов с помощью Joomla! 1.5 / Х. Граф. – М. : “Вильямс”, 2009. – 304 с.