

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ
УНІВЕРСИТЕТ
ХЕРСОНСЬКИЙ ПОЛІТЕХНІЧНИЙ КОЛЕДЖ**

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ З
ДИСЦИПЛІНИ
«WEB-ПРОГРАМУВАННЯ»**

**для студентів спеціальності
122 «Комп'ютерні науки та інформаційні
технології»
Галузь знань: 12 «Інформаційні технології»**

Затверджено методичною радою ХПТК ОНПУ
Протокол №__ від _____20__ р.

Херсон — 2017

Методичні вказівки до виконання курсового проекту з дисципліни
«Web-програмування» / Уклад.: В.М. Левицький — Херсон: ХПТК ОН-
ПУ, 2017. — ____с.

Навчальне видання
МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ
ДИСЦИПЛІНИ «WEB-ПРОГРАМУВАННЯ»
для спеціальності
122 «Комп'ютерні науки та інформаційні технології»
Галузь знань: 12 «Інформаційні технології»

Укладач

Левицький Віктор Миколайович, викладач

Відповідальний

редактор

Технічний

редактор

Коректор

Рецензент

За редакцією укладачів

Надруковано з оригінал-макета замовника

ЗМІСТ

Стр.

1	Вступ	4
2	Загальні відомості до виконання та захисту курсової роботи	5
3	Оформлення пояснювальної записки	8
4	Зміст розділів пояснювальної записки	11

1 Вступ

Курсова робота з дисципліни «Web-програмування» виконується студентами третього курсу спеціальності 122 «Комп'ютерні науки та інформаційні технології» після закінчення вивчення вище зазначеного курсу в V семестрі.

При виконанні курсової роботи студенти повинні досконало дослідити предметну область однієї з запропонованих тем. На основі аналізу предметної області.

В результаті виконання всіх етапів проектування повинен бути розроблений інформаційний web-портал, або програмний додаток та технічна документація до неї.

Темами курсових робіт є задачі різноманітних галузей народного господарства та розрахункові задачі, для вирішення яких широко використовуються принципи об'єктно-орієнтованого програмування. Об'єктний підхід розвиває у студентів вміння та навички будувати зв'язки між різними модулями, працювати та приймати рішення.

2 Загальні відомості до виконання та захисту курсової роботи

Тематика курсових робіт розглядається і схвалюється на засіданні циклової комісії, погоджується завідувачем відділення, та затверджується заступником директора. Перед початком роботи над курсовою роботою студент повинен отримати заповнений, підписаний, ухвалений та затверджений у встановленому порядку аркуш «ЗАВДАННЯ» та індивідуальний план, розроблений керівником проекту. В бланку «ЗАВДАННЯ» наведено:

- тему курсової роботи;
- дані до виконання роботи;
- вказівки по змісту пояснювальної записки;
- дата видачі завдання до курсової роботи, термін закінчення роботи студента над курсовою роботою;
- список джерел, що рекомендуються;
- індивідуальний план виконання курсової роботи з нормованою кількістю балів.

Бланк завдання на курсову роботу підшивається студентом до пояснювальної записки після листа затвердження.

Курсова робота виконується студентом самостійно при консультуванні його керівником проекту у відповідності з графіком. Кожен студент повинен відвідувати консультації згідно з графіком. При відсутності студента на двох або більше консультаціях, керівник подає доповідну записку завідувачу відділення.

На всьому протязі курсового проектування керівник оцінює якість виконання етапів КР у балах рейтингової системи, проставляє їх у відповідний графах індивідуального плану.

Попередній захист курсової роботи відбувається в установленій термін. Мета попереднього захисту – визначення рівня готовності студента до захисту проекту. На попередній захист студент надає програмний додаток, пояснювальну записку (не переплетену). На попередньому захисті

програмний додаток повинен знаходитись на етапі тестування, виконувати всі функції згідно поставленої задачі. Після перевірки працездатності програмного додатку та готовності пояснювальної записки керівник повертає курсову роботу (програму та пояснювальну записку) для ознайомлення із зауваженнями та вказівками щодо виправлення помилок.

Виправлена пояснювальна записка із підписами студента та керівника проекту подається на нормоконтроль, який здійснюється за відповідним графіком в два етапи:

- на першому етапі — проводиться перевірка текстової документації із розстановкою олівцем позначок нормоконтролера щодо порушень вимог Єдиної системи програмної документації (ЄСПД) і повертаються студентів для усунення порушень,
- на другому етапі — здійснюється перевірка виправлених помилок, допущених студентом у пояснювальній записці, оцінювання виконання курсового проекту на відповідність вимог стандартів ЄСПД та СОУ 2:2008, усунення позначок нормоконтролера. При наявності не виправлених порушень нормоконтролер проставляє позначки червоною пастою і підписує документ. Максимальний бал, який студент може отримати за сумарним рейтингом складає 60 балів. Такий бал, за бажанням студента, може бути зарахованим як оцінка знань без захисту проекту. Сума балів від 36 до 60 є допуском до захисту.

До захисту курсової роботи студент готує доповідь (обсягом до 10 хвилин), в якій чітко формулює постановку задачі курсової роботи, пояснює послідовність її виконання. Доповідає про результати роботи, вказує на можливості реального використання курсової роботи. Захист курсової роботи здійснюється у присутності комісії у складі двох-трьох кваліфікованих викладачів за участю керівника курсового проекту. При оцінюванні результатів захисту КР студент може отримати максимально 40 балів, при

цьому враховується :

- оригінальність, самостійність розробки функціональної моделі та методів реалізації;
- використання у програмному продукті CASE-технологій;
- навички використання UML-мови моделювання;
- використання технологій проектування ПЗ;
- грамотність написання та оформлення пояснювальної записки відповідно до вимог ЄСПД;
- ступінь використання довідкової та технічної літератури, ДСТУ, ГОСТ;
- вміння грамотно захищати розроблену роботу.

Студент, який при захисті курсової роботи отримав незадовільну оцінку, допускається, як виняток, до повторного захисту у новий термін, але після позитивного рішення циклової комісії та дозволу завідувача відділенням.

Студентам, які не захищали курсову роботу з поважної причини, яка документально підтверджена, завідувачем відділення може бути продовжено строк виконання та захисту.

3 Оформлення пояснювальної записки

Пояснювальна записка до курсової роботи виконується як комп'ютерний набір тексту або рукописним способом на одній стороні аркуша формату А4 (210 x 297 мм) відповідно до ГОСТ 2.301-68.

При комп'ютерному наборі необхідно виконувати оформлення курсової роботи на друкуючих та графічних пристроях виведення персональних комп'ютерів /ГОСТ 2.004-88/ з додержуванням правил Єдиної системи конструкторської документації та Єдиної системи програмної документації.

Пояснювальна записка виконується державною мовою. Рекомендований обсяг пояснювальної записки повинен складати 25-30 сторінок друкованого тексту (комп'ютерний набір тексту виконується шрифтом гарнітури “Arial”, кегль - 14 пт, курсивом, з полуторним інтервалом). В дану кількість сторінок не включають сторінки, на яких розміщено додатки.

З використанням комп'ютерного способу виконання на сторінці має розташовуватися не більше 40 рядків (відповідно інтервал між рядками – не більше 8 мм).

Помилки, описки та графічні неточності допускається виправляти підчищенням або зафарбовуванням білою фарбою і нанесенням на тому ж місці виправленого зображення від руки. Виправлене повинно бути чорного кольору.

Заголовки структурних елементів і розділів слід розташовувати з абзацного відступу і друкувати малими літерами, починаючи з першої великої літери без крапки в кінці, не підкреслюючи їх. Кожен структурний елемент та розділ слід починати з нової сторінки.

Заголовки підрозділів, пунктів, підпунктів слід починати з абзацного відступу і друкувати маленькими літерами, окрім першої великої, без крапки і не підкреслюючи їх. Абзацний відступ повинен бути однако-

вим впродовж усього тексту звіту і дорівнювати 5 знакам (15-17 мм). Перенесення слів у заголовок не допускається.

Відстань між заголовком і подальшим чи попереднім текстом за комп'ютерного способу виконання – не менше, ніж 2 рядки. Відстань між основами рядків заголовку, а також між двома заголовками приймають такою, як у тексті.

Не допускається розміщувати назви розділу, підрозділу, пункту, підпункту в нижній частині сторінки, якщо після неї розміщено тільки один рядок тексту.

Сторінки пояснювальної записки нумеруються арабськими цифрами, додержуючись наскрізної нумерації і впродовж усього тексту. Номер сторінки проставляється у графі 7 основного напису відповідно до ДСТУ ГОСТ 2.104:2006.

Оформлювальний аркуш титульного листа з темою і підписами, лист завдання включають до загальної нумерації сторінок звіту, але номер не проставляють. Елементи дати проставляють повно (наприклад, 21.11.2017). Зразок оформлення титульного аркуша затвердження наводиться у додатку А.

Структурні елементи «РЕФЕРАТ», «ЗМІСТ», «ВСТУП», «ВИСНОВКИ», «РЕКОМЕНДАЦІЇ», «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ» не нумеруються, а їх назви правлять за назви структурних елементів. Обсяг пояснювальної записки курсової роботи не повинен перебільшувати 25 – 30 аркушів без врахування лістингу програми. Структура пояснювальної записки:

- лист затвердження;
- завдання на курсову роботу;
- зміст;
- розділи пояснювальної записки;
- висновки;

- список використаних джерел;
- додатки.

Пояснювальна записка повинна бути виконана у відповідності до стандартів ЄСКД та ЄСПД, діючих на даний час:

- ДСТУ 3008-2015 Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення.
- ГОСТ 2.004-88 ЕСКД. Общие требования к выполнению конструкторских и технологических документов на печатающих и графических устройствах вывода ЭВМ. ЗАЧЕМ???? СТП 17.2-2002 МНОЖИТ НА 0 ЭТОТ ГОСТ!!!
- ГОСТ 2.301-68 ЕСКД. Форматы.
- ГОСТ 19.106-78 ЕСПД. Требования к программным документам, выполненным печатным способом.
- ГОСТ 19.701-90 ИСО 5807-85. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения
- ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам.
- СТП 17.2-2002 Правила оформлення пояснювальної записки до дипломних та курсових робіт. Загальні відомості.

4 Зміст розділів пояснювальної записки

У змісті послідовно перераховують заголовки структурних елементів: розділів, підрозділів, список використаних джерел, додатків і вказують номери сторінок, з яких вони починаються. Зміст оформлюється відповідно до встановленого зразка (приклад оформлення див. додаток В).

Зміст пояснювальної записки курсової роботи : Вступ

1 Дослідження предметної області

1.1 Аналіз предметної області

1.2 Огляд і аналіз існуючих аналогів подібних систем

2. Технічне завдання на розробку web-сайту

2.1 Характеристика web-сайту 2.2 2.3 Технічні вимоги 2.1 Проектування ER-моделі предметної області

3. Проектування інтерфейсу користувача

3 Специфікація вимог до програмного забезпечення 3.1 Глосарій. 3.2 Текстовий формат специфікації варіантів використання. 3.3 Розробка пакету UML-діаграм концептуального рівня 4 Проектні та технічні рішення 4.1 Проектування логічної моделі даних 4.2 Проектування фізичної моделі даних 4.3 Розробка архітектури програмного забезпечення Висновки Список використаних джерел Додаток А Лістинг програмного коду

У розділі «1 Вступ» вміщено огляд змісту пояснювальної записки, який розташовується як окремих розділ.

Вступ повинен містити загальну характеристику курсової роботи, оцінку сучасного стану розв'язуваної науково-технічної проблеми, її актуальність.

У вступі зазначається практичне використання моделі системи, що буде розроблятися, характеристика об'єкту проектування, мета роботи. Слід стисло проаналізувати стан проблеми на сучасному етапі, задачі, питання, які розглядаються у проекті. У вступі доцільно обґрунтувати необхідність виконання роботи, вказати область можливого використання

розробленої системи, надати загальну характеристику роботи, що буде виконуватися. Орієнтований обсяг 1–2 с.

Наприклад:

Останнім часом спостерігається стрімкий розвиток інформаційних технологій і впровадження їх в багатьох сферах життєдіяльності людини. Неможливість управляти технічними і економічними процесами, що все більш ускладнюються, зумовила в першу чергу впровадження у сферу промисловості, бізнесу, комерції інформаційних систем, які повинні виконувати збір, управління, корегування та розповсюдження інформації усередині організації. Дані є цінним ресурсом, і можуть забезпечувати як повсякденні потреби організації, так і підтримку ухвалення стратегічних рішень. У зв'язку з інтенсивним розвитком мережі Інтернет й переходом до електронного бізнесу та торгівлі, виникає необхідність у якісних, швидких, надійних та функціональних програмних рішеннях для умов, що складаються. При чому дані рішення повинні забезпечувати як належний рівень прозорості взаємодії учасників, так і необхідний ступінь безпечності інформації. Для того, щоб отримати такі рішення, перш за все необхідно правильно спроектувати програмне забезпечення. Методологія проектування інформаційних систем описує процес створення і супроводу систем у вигляді життєвого циклу інформаційної системи, представляючи його як деяку послідовність стадій і процесів, що на них виконуються. Для кожного етапу визначаються склад і послідовність виконуваних робіт, отримані результати, методи і засоби, необхідні для виконання робіт, ролі та відповідальність учасників тощо. Таке формальне описання життєвого циклу інформаційної системи дозволяє спланувати та організувати процес колективної розробки й забезпечити управління цим процесом. Метою даної курсової роботи є проектування та розробка моделі програмного забезпечення за методологією RUP для автоматизації обліку продажів магазином книг комп'ютерної тематики. Досягнення

мети роботи здійснюється шляхом вирішення таких задач: виділення предметної області управління, яка підлягає автоматизації; моделювання виділеної предметної області з визначенням етапів, підетапів, процесів і функцій; аналіз інтерфейсу та функціональності готового програмного забезпечення, призначеного для автоматизації виділених процесів; опрацювання специфікацій вимог до програмного забезпечення, що розробляється; розроблення програмного забезпечення для автоматизації вирішення комплексу задач. Під час розробки програмного забезпечення були використані такі технології: StarUml та Microsoft Visio. Програмний продукт може бути використаний в організаціях, які займаються роздрібною торгівлею комп'ютерних книг для автоматизації продажу та формування облікової звітності.

У розділі «2 Аналіз предметної області» необхідно описати підрозділи «2.1 Опис предметної області», «2.2 Огляд і аналіз існуючих аналогів, що реалізують функції предметної області » і «2.3 Проектування ER-моделі предметної області».

У підрозділі «2.1 Опис предметної області» наводять аналіз та опис предметної області, визначають вимоги та функції користувачів, що спрямовані на досягнення встановлених цілей. Рекомендовано розкрити наступні питання: - аналіз предметної області, тобто надається характеристика організації (підприємства, підрозділу, закладу тощо), для якої буде розроблена дана підсистема та сфери її діяльності; - визначення сфери застосування інформаційної підсистеми як у теперішньому часі, так і в майбутньому; - опис організаційної структури підприємства; - опис процесів, які мають бути автоматизовані; - опис вимог, яким повинна задовольняти система; - опис функціональних обов'язків працівників, які будуть користувачами підсистеми (якщо автоматизується робоче місце фахівця); - основні терміни та визначення, притаманні висвітлюваній області; - класифікацію інформації, яка має оброблятися розробленою

інформаційною підсистемою.

У підрозділі «2.2 Огляд і аналіз існуючих аналогів подібних систем» необхідно провести аналіз існуючих систем, вказати їх переваги та недоліки. У підрозділі «2.3 Проектування ER-моделі предметної області» необхідно представити у графічному вигляді ER-діаграму проєктованої системи. Наведемо приклад розділу « 2 Аналіз предметної області ».

Підрозділ 2.1 «Опис предметної області»

У результаті дослідження й аналізу предметної області були визначені такі сутності, атрибути і первинні ключі. Зв'язки сутностей визначаються на основі бізнес правил, які побудовані з урахуванням обмежень організаційної структури та операцій, що виконуються в системі: наприклад, у цеху працює декілька працівників. Працівник працює, але тільки в одному цеху; працівник цеху бере участь в отриманні декількох видів спецодягу. Кожне отримання має відношення тільки до одного працівника; один і той же вид спецодягу поступає кілька разів для отримання. Крім того необхідно врахувати: кожен працівник обов'язково працює у цеху. У кожному цеху обов'язково працюють працівники; працівники деяких посад необов'язково беруть участь в отриманні спецодягу. У кожному отриманні обов'язково бере участь працівник; кожен вид спецодягу обов'язково поступає для отримання. Кожне отримання обов'язково належить до деякого виду спецодягу.

Підрозділ 2.2 «Огляд і аналіз існуючих аналогів подібних систем»

Аналізуючи предметну область програмного продукту, було виявлено подібні розробки програми 1С:, яка є програмою-аналогом до розробленої

моделі автоматизації обліку спецодягу працівників підприємства. Проаналізувавши дану систему, було зроблено висновок, що деякі додатки, компоненти та модулі схожі за своєю функціональністю. Головною метою програмного комплексу «1С:» є якісна автоматизація обліку підприємств та установ. У цьому напрямку «1С:» вирішує основні завдання обліку підприємств та установ, а також дозволяє виконувати весь комплекс розрахунків, які здійснюються установою. За допомогою програмного комплексу досягається: Економія часу, завдяки автоматичним розрахункам і можливості масового розповсюдження документів; Комплексний підхід до обліку установи, за допомогою модульної структури програмного комплексу; Ведення обліку у відповідності до чинного законодавства; Можливість оперативної звірки даних обліку в будь-якому часовому розрізі. В одній інформаційній базі можна вести облік діяльності декількох організацій та індивідуальних підприємств. При цьому використовуються загальні довідники контрагентів, співробітників і номенклатури, а звітність формується окремо.

Підрозділ 2.3 «Проектування ER-моделі предметної області»

Сутності, атрибути та первинні ключі предметної області можна представити у вигляді Таблиці 2.1.

Приклад 1

Таблиця 2.1 – Опис елементів моделі предметної області

Сутність Атрибути Первинний ключ

СПЕЦОДЯГ Код спецодягу; Код виробника; Вид спецодягу; Термін придатності; Вартість одиниці. Код спецодягу ЦЕХ Код цеху; Назва цеху; П.І.Б. начальника цеху Код цеху

ПРАЦІВНИК Код працівника; Код цеху; Вид спецодягу; П.І.Б. працівника; Посада; Знижка на спецодяг (ОТРИМАННЯ Код працівника; Код

спецодягу; Дата отримання

Приклад графічного зображення ER-моделі предметної області представлено на Рисунку 2.1. Приклад 2

Рисунок 2.1 - ER-діаграма предметної області У розділі «3 Специфікація вимог до програмного забезпечення» необхідно описати підрозділи «3.1 Глосарій», «3.2 Текстовий формат специфікації варіантів використання» і у пункті «3.3 Розробка пакету UML-діаграм концептуального рівня» описують діаграми варіантів використання (use case diagram), послідовності та класів». Наведемо приклад розділу «3 Специфікація вимог до програмного забезпечення».

Приклад 3

3.1 Глосарій

RUP - (англ. Rational Unified Process - Раціональний Уніфікований Процес) - це сукупність рекомендацій, що стосуються розробки програмного забезпечення. Він описує дисциплінований підхід до призначення задач та відповідальності в організації з розробки програмного забезпечення. Головною метою RUP є надавання певності в керуванні проектом, у високій якості програмного забезпечення, що передається кінцевим користувачам не виходячи за рамки прогнозованого графіку розробки та бюджету. ІТЕРАЦІЯ - це завершений цикл розробки, що виражається у випуску (внутрішньої або зовнішньої) версії. АКТОР - це визначення зв'язаної множини ролей, які може відігравати користувач при спілкуванні з системою ДІАГРАМИ ВИКОРИСТАННЯ - Діаграми використання допомагають зрозуміти що повинна робити система без занурювання в питання як вона це буде робити. UML-ДІАГРАМИ - (Unified Modelling Language або UML - універсальна мова моделювання) — це мова позначень або побудови діаграм, призначена для визначення, візуалізації і

документування моделей зорієнтованих на об'єкти систем програмного забезпечення. **ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ** - це процес вирішення задач та планування для створення програмного рішення. Після того як мета і специфікація програми описані, розробник створить дизайн проекту або скористається послугами дизайнера для розробки плану рішення. У дизайн включаються як описи низькорівневих компонентів, алгоритмів, так і огляд архітектури. **ЖИТТЄВИЙ ЦИКЛ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ** — сукупність окремих етапів робіт, що проводяться у заданому порядку протягом періоду часу, який починається з вирішення питання про розроблення програмного забезпечення і закінчується припиненням використання програмного забезпечення.

3.2 Текстовий формат специфікації варіантів використання

Даний опис може бути представлений як у табличному вигляді, так і у вигляді нумерованого переліку. Нижче наведено приклад нумерованого переліку специфікації варіантів використання.

1. Авторизація в системі: 1.1 Зацікавлення особи прецеденту та їх вимоги: користувач хоче швидко авторизуватися в системі. 1.2 Користувачі ІС, тобто основні особи прецеденту: адміністратор, постачальник та покупець. 1.3 Передумови прецеденту: ІС повинна бути активною. 1.4 Основний успішний сценарій: адміністратор, покупець, постачальник вводить логін та пароль у форму для авторизації, система валідує введені дані, якщо дані правильні то відбувається переадресація адміністратора і постачальник до ІС. 1.5 Розширення основного сценарію або альтернативні потоки: покупець, адміністратор, постачальник ввів некоректні дані: - ІС повідомляє користувачів, що вони ввели некоректний логін або пароль і знову пропонує ввести дані авторизації. - користувачі повторно вводять коректні дані. - Відбувається адресація користувачів на завантажувальну

форму. - Після натиснення кнопки «Розпочати роботу з ІС» користувачі потрапляють на головну форму. 2. Перегляд списку працівників: 2.1 Зацікавлені особи прецеденту та їх вимоги: користувач хоче переглянути список працівників. 2.2 Користувачі ІС, тобто основні особи прецеденту: адміністратор. 2.3 Передумови прецеденту: ІС повинна бути активною користувачами ІС будуть: адміністратор. 2.4 Основний успішний сценарій: ІС відображає повний список працівників. 2.5 Розширення основного сценарію або альтернативні потоки: ІС відображає список працівників за заданими критеріями.

3.3 Розробка пакету UML-діаграм концептуального рівня

Діаграми прецедентів дозволяють створювати список операцій, які виконує система. Як правило, цей вид діаграм називають діаграмами функцій, оскільки набір таких діаграм створює список вимог до системи і визначається з множиною функцій, що виконує система. Кожна діаграма - це опис послідовності дій та взаємодій між акторами та іншими компонентами системи. Діаграми варіантів використання (usecase diagrams) використовуються для відображення сценаріїв використання системи (usecases) та користувачів системи (actors), які використовують її функції. Актори та варіанти використання поєднуються напрямленою асоціацією (unidirectional association) – стрілкою, що спрямована від актора до варіанта використання. Також актори можуть поєднуватися з використанням зв'язків узагальнення. Варіанти використання можуть бути пов'язані між собою трьома видами зв'язків: узагальненням (generalization), розширенням (extend relationship) та включенням (include relationship). Наведемо приклад діаграми usecase на основі системи «Бібліотека».

Приклад 4

Рисунок 3.2 – Діаграма прецедентів

Наступний вид діаграм, який необхідно змодельовати – це діаграма послідовності. Діаграма послідовності (sequence diagram) - діаграма, на якій показані взаємодії об'єктів, впорядковані за часом їх прояви. Особливості взаємодії елементів модельованої системи можуть бути зображені на діаграмах кооперації і послідовності. Діаграми кооперації використовуються для опису функціонування систем, хоча час у явному вигляді в них відсутній. Проте часовий аспект поведінки може мати істотне значення при моделюванні синхронних процесів, що описують взаємодію об'єктів. Саме для цієї мети в мові UML використовуються діаграми послідовності.

Приклад 5

Рисунок 3.3 - Діаграма послідовності

Діаграма класів (Class diagram) визначає склад класів об'єктів і їх зв'язків. Діаграма задається зображенням, на якому класи позначаються поділеними на три частини прямокутниками, а зв'язки — лініями, що з'єднують прямокутники. Це відповідає візуальному зображенню понять і зв'язків між ними. Верхня частина прямокутника — обов'язкова, в ній записується ім'я класу. Друга і третя частини прямокутника визначають відповідно список операцій і атрибутів класу, що можуть мати такі специфікатори доступу: - public (загальний) позначає операцію або атрибут, доступ до яких здійснюється з будь-якої частини програми будь-яким об'єктом системи; - protected (захищений) позначає операцію або атрибут, доступ до яких здійснюється об'єктами класу, в якому вони оголошені, або об'єктами класів-нащадків, - private (приватний) позначає операцію або атрибут, доступ до яких здійснюється тільки об'єктами того класу, в якому вони визначені.

Приклад 6

Рисунок 3.3 – Діаграма класів

Примітка: В якості середовища розробки таких діаграм рекомендо-

вано пакет StarUml версії 5. На вибір студента може бути використано інше середовище, наприклад, Rational Rose або ERWIN. У розділі «4 Проектні та технічні рішення» необхідно описати: Логічне проектування – процес створення моделі використовуваної на підприємстві інформації з урахуванням обраної моделі організації даних, але незалежно від типу моделі життєвого циклу й інших фізичних аспектів реалізації. Ціль етапу – створення логічної моделі даних на рівні моделі записів для досліджуваної частини об'єкта шляхом уточнення і перетворення концептуальної моделі з урахуванням особливостей обраної організації даних у цільову модель. Логічне проектування виконується для певної моделі даних. Для реляційної моделі даних логічне проектування полягає у створенні реляційної схеми, визначенні числа і структури таблиць, формуванні запитів до БД, визначенні типів звітних документів, розробці алгоритмів обробки інформації, створенні форм для введення і редагування даних в БД й рішення цілого ряду інших задач. На цьому етапі проводиться усунення особливостей логічної моделі, які несумісні з реляційною моделлю: вилучення зв'язків M:N; вилучення рекурсивних зв'язків; вилучення складних зв'язків; вилучення зв'язків з атрибутами, вилучення багатозначних атрибутів. Після спрощення в концептуальній моделі можуть мати присутні лише такі елементи: об'єкти і атрибути; зв'язки типу 1:1 і 1:M; зв'язки типу суперклас-підклас. Проектування фізичної моделі даних. На етапі фізичного проектування бази даних, насамперед, необхідно вибрати конкретну цільову СУБД. Тому фізичне проектування нерозривно пов'язане з конкретною СУБД. Між логічним і фізичним проектуванням існує постійний зворотній зв'язок, тому що рішення, прийняті на етапі фізичного проектування з метою підвищення продуктивності системи, здатні вплинути на структуру логічної моделі даних. Основною метою фізичного проектування у випадку реляційної моделі даних є наступне: • створення набору реляційних таблиць і обмежень для них на основі інформації,

представленої в логічній моделі даних; • визначення конкретних структур зберігання даних і методів доступу до них, що забезпечують оптимальну продуктивність системи; • розробка засобів захисту створюваної системи. Фізична модель визначає розміщення даних у зовнішній пам'яті. Вона ще називається внутрішньою моделлю системи і форма її представлення залежить від типу обраної СУБД. Якщо обрана така СУБД, яка підтримує реляційну модель даних, то треба таблиці разом з атрибутами і зв'язки між таблицями перенести в середовище СУБД з урахуванням вимог до відповідних об'єктів БД. Так, ідентифікатори (імена) таблиць і полів мають задовольняти вимогам СУБД, типи даних, розміри полів, обмеження теж мають бути приведені у відповідність до прийнятих у даній СУБД. Далі визначаються стратегії індексування, а також взаємозв'язки між таблицями, первинні та зовнішні ключі на основі визначених у даталогічній моделі та врахуванням способів їх завдання в обраній СУБД. На етапі фізичного проектування слід приділити особливу увагу забезпеченню цілісності БД. Наводиться контрольний приклад заповненої БД з урахуванням встановлених обмежень цілісності.

Приклад 7

Рисунок 4.1 – Схема даних

Таким чином, послідовність робіт при створенні фізичної моделі на основі реляційної моделі даних може бути наступною: • аналіз та вибір СУБД; • розробка структури (фізичної моделі) бази даних засобами вибраної СУБД з урахуванням типів даних та обмежень цілісності атрибутів; • розробка схеми взаємозв'язків бази даних засобами вибраної СУБД з урахуванням обмежень посилальної цілісності; • розробка контрольного приклада заповнення бази даних. Кінцевим результатом роботи даного розділу є представлення таблиць та зв'язків між ними, що описані в даталогічній моделі даних, у середовищі обраної СУБД. При цьому мають бути визначені (запрограмовані мовою опису даних)

обмеження цілісності як для атрибутів, так и посиальна цілісність. На етапі проектування бази даних на фізичному рівні треба, у першу чергу, визначитися з СУБД.

У підрозділі «4.3 Розробка архітектури програмного забезпечення» виконують генерацію програмного коду системи(лістинг), використовуючи певні налаштування та інструкції. Побудову такої архітектури слід виконувати після розробки діаграм прецедентів, послідовностей та діаграми класів, але певні обмеження та урахування особливостей генерації коду слід враховувати із самого початку розробки діаграм. Рекомендації та певні кроки вибору у системі наведені нижче. 1. При створенні моделі Use Case потрібно використовувати тільки латинські символи; 2 Для генерації лістингу вибрати мову програмування C++; 3 Для правильної генерації потрібно виконати такі налаштування: - Переходимо у “Model > Profiles” вибираємо “C++ Profile” та тиснемо кнопку “Include”. Для збереження натиснути “Close”. 4 Для початку генерації лістингу перейти у “Tools > C++ > Generate Code”: - У вікні “Select start package location” вибрати “Use Case Model” та натиснути “Next”. - У вікні “Select the code generation element(s)” вибрати всі елементи натиснути кнопку ”Select All” і потім кнопку ”Next”. - У вікні “Output directory setup” вибрати кінцеву папку для збереження коду та натиснути “Next”. - У вікні “Options Setup” увімкнути “Header file subdirectory” та у полі з права від неї увести назву папки, де вони будуть зберігатись. - Увімкнути “Use Microsoft Visual C++ grammar”. - Перейти у вкладку “File header comment and Default include” та у полі “File header comment” підписати проект. 5 Натиснути ”Next”, “OK” та “Finish” для завершення генерації лістингу програми. Для фізичного подання моделей систем використовуються діаграми реалізації, які включають дві окремі канонічні діаграми: діаграму компонентів і діаграму розгортання. Діаграма компонентів дозволяє визначити архітектуру системи, що розробляється. Основними графічними елементами діаграми

компонентів є компоненти, інтерфейси та залежності між ними.

Приклад 8

Рисунок 4.2 – Діаграма компонентів

У підрозділі «4.4 Тестування програмного забезпечення» наводять дані контрольного прикладу, на яких можна перевірити валідність побудованої схеми даних і підсистеми в цілому. Тестування - це процес перевірки готової програми (або програмного засобу) в статистиці (перегляди, інспекції, налагодження вихідного коду) і в динаміці (прогін на наборі тестових даних) з метою перевірки різних шляхів виконання програми і порівняння отриманих результатів із заздалегідь заданими. Вимоги до даних, які забезпечують тестування системи — їхня показність, що враховує особливості інформації, зазначені в описі предметної області. Такі дані повинні забезпечити налагодження інтерфейсу, реалізацію вимог та підтвердити працездатність реалізованих функцій. Дані контрольного прикладу призначені для тестування, налагодження і демонстрації рішення задачі. У цьому підрозділі наводиться: ? перелік параметрів і стисла характеристика функції із числа тих, що перевірялися у моделі; ? опис вхідних даних для перевірки із наведенням конкретних вхідних даних. Допускається вхідні дані представляти у вигляді таблиць.

Приклад 9

Перевірка функції введення та редагування вхідних даних на основі об'єкту «Особові дані».

Таблиця 4.1 – Дані об'єкту “Особові дані”
Номер карти Прізвище Ім'я
По батькові Дата народження 102 Іванов Іван Петрович 12.12.1960

Висновки

У висновках, що розташовані окремим структурним елементом, тобто як «Висновки», студент повинен навести стислий перелік отриманих результатів, їх оцінку відповідно обраній темі, коротку характеристику розробленої моделі системи, напрямки та можливості вдосконалення

даної роботи. У цьому структурному елементі відображають також відповідність вимогам завдання на курсову роботу. У висновках наводиться стисла викладка показників, отриманих при розробці задачі; вказуються напрями подальшої роботи. У висновках студент вказує, які завдання розв'язані ним у процесі проектування, та які навички та вміння були вдосконалені (отримані) під час виконання курсової роботи. Структурний елемент «Список використаних джерел» містить перелік літератури та інші джерела, які використовувались для виконання курсової роботи. В розділі „Додаток А Лістинг програмного коду” розміщують програмний код, в якому роздруковують модулі(компоненти), які містять опис моделі на обраній мові програмування.