

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ**

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

**Методичні рекомендації
до виконання дипломного проекту
для студентів спеціальностей
"Інформаційні управляючі системи
і технології",
"Комп'ютерний еколого-економічний моніторинг"
усіх форм навчання**

Харків. Вид. ХНЕУ, 2011

Затверджено на засіданні кафедри інформаційних систем.
Протокол № 10 від 24.03.2011 р.

Укладачі: Золотарьова І. О.

Ушакова І. О.

Мінухін С. В.

Знахур С. В.

Павленко Л. А.

Тарасов О. В.

Парфьонов Ю. Е.

Поляков А. О.

М54 Методичні рекомендації до виконання дипломного проекту для студентів спеціальностей "Інформаційні управляючі системи і технології", "Комп'ютерний еколого-економічний моніторинг" усіх форм навчання / укл. І. О. Золотарьова, І. О. Ушакова, С. В. Мінухін та ін. – Х. : Вид. ХНЕУ, 2011. – 94 с. (Укр. мов.)

Наведено основні вимоги щодо організації, тематики, змісту та оформлення дипломних проектів. Розроблено з урахуванням галузевого стандарту вищої освіти України з напрямку підготовки "Комп'ютерні науки" і досвіду кафедри інформаційних систем з підготовки фахівців.

Рекомендовано для студентів спеціальностей "Інформаційні управляючі системи і технології" та "Комп'ютерний еколого-економічний моніторинг".

Вступ

Дипломне проектування є одним з найважливіших видів самостійної роботи, яка завершує підготовку фахівця на етапі навчання студентів за програмою спеціаліста ступеневої базової вищої освіти і є основою для проведення державної атестації спеціалістів.

У методичних рекомендаціях викладені загальні вимоги до організації та проведення дипломного проектування, тематики, змісту та обсягу дипломних проектів, планування роботи проекту, оформлення та захисту дипломного проекту.

Методичні рекомендації ознайомлять студентів з тематикою дипломних проектів, з вимогами до його змісту й оформлення, з організацією роботи над проектом.

Зміст, обсяг та структура дипломного проекту, які наведені у методичних рекомендаціях, є типовими і у окремих випадках за письмовим дозволом випускаючої кафедри можуть бути змінені.

Методичні рекомендації призначені для студентів, що навчаються за спеціальністю 7.05010101 "Інформаційні управляючі системи і технології", 7.05010105 "Комп'ютерний еколого-економічний моніторинг".

Вимоги до дипломного проектування повністю відповідають освітньо-кваліфікаційній характеристиці і освітньо-професійній програмі спеціальності 7.05010101 "Інформаційні управляючі системи і технології", 7.05010105 "Комп'ютерний еколого-економічний моніторинг".

1. Мета і завдання дипломного проектування

Дипломне проектування – завершальний етап підготовки спеціалістів. Нормативною формою державної атестації спеціаліста є кваліфікаційний дипломний проект.

Дипломний проект – кваліфікаційна робота, покликана здійснювати об'єктивний контроль за ступенем сформованості умінь у випускників вирішувати типові завдання діяльності, які в ОКХ належать до проектувальної (проектно-конструкторської) та виконавської (технологічної, технічної) виробничих функцій [3].

Дипломний проект передбачає проведення аналізу та теоретичної розробки (моделювання, дослідження процесів і об'єктів), а також повинен мати характер прикладного наукового дослідження об'єкта діяльності.

Мета дипломного проектування – узагальнити та систематизувати знання та практичні навички студентів, одержані ними під час вивчення навчальних дисциплін гуманітарної та соціально-економічної підготовки; математичної та природничо-наукової підготовки; професійної та практичної підготовки. У процесі роботи над дипломним проектом студенти набувають досвіду роботи з науково-технічною, нормативною та довідковою літературою, вчаться використовувати державні стандарти, складати пояснювальну записку до проекту, практично засвоюють знання під час прийняття конкретних проектних рішень.

Завданнями дипломного проектування є:

- систематизація, закріплення та розширення теоретичних знань і практичних навичок за спеціальністю, застосування цих знань при виконанні конкретних завдань дипломного проекту;
- розвиток та закріплення навичок самостійної роботи, оволодіння методикою досліджень та експериментування, моделювання, використання сучасних інформаційних технологій у процесі розв'язання задач, які передбачені завданням на дипломне проектування;
- визначення відповідності рівня підготовки випускника вимогам освітньо-кваліфікаційної характеристики спеціаліста, його готовності та спроможності до самостійної роботи в умовах ринкової економіки, сучасного виробництва, прогресу науки, техніки та культури, а саме, показати вміння користуватись сучасними системами програмування, самостійно ставити та вирішувати інженерні задачі з проектування інформаційних управляючих систем та їх елементів, використовуючи сучасні методології, інформаційні технології, проводити комп'ютерне моделювання, а також вміння обробляти і систематизувати результати досліджень, використовуючи комп'ютерну техніку та інструментальні засоби моделювання та розробки.

Виконуючи дипломний проект, студент повинен повною мірою використовувати набуті знання з інформаційних технологій та комп'ютерної техніки, інтелектуальних систем та баз знань, існуючі пакети, методи та засоби математичної обробки інформації; поєднувати теоретичні знання з виробничим досвідом, отриманим при проходженні практик; використовувати досягнення вітчизняної та світової науки і техніки; враховувати техніко-економічні показники функціонування створюваних програмно-інформаційних систем та комплексів; на високому теоретичному і професійному рівні виконувати проектування

обраних технічних рішень; грамотно, повно і разом з тим лаконічно викладати свої рішення в пояснювальній записці.

Під час захисту дипломної роботи стисло передати її основний зміст, акцентуючи увагу на актуальності та новизні роботи, можливості її практичного застосування, аргументовано подати прийняті в ній технічні рішення та обґрунтувати отримані результати.

Дипломний проект є самостійною роботою студента. За всі прийняті в дипломному проекті рішення, а також правильність і обґрунтованість розрахунків і графічних робіт несе відповідальність студент – автор дипломного проекту.

До дипломного проектування допускається студент, який пройшов повний курс навчання та склав усі передбачені навчальним планом заліки та екзамени, тобто виконав усі вимоги навчального плану з напряму підготовки або спеціальності.

2. Тематика дипломного проектування

Тематика дипломних проектів відповідає вимогам освітньо-кваліфікаційної характеристики і навчальному плану спеціальностей 7.05010101 "Інформаційні управляючі системи і технології", 7.05010105 "Комп'ютерний еколого-економічний моніторинг". Теми дипломних проектів визначаються і розглядаються кафедрою інформаційних систем (ІС), проректором з науково-методичної роботи та затверджуються наказом ректора університету.

Тематика дипломних робіт повинна відповідати таким вимогам:

бути актуальною;

мати практичне значення;

відповідати сучасному стану і тенденціям розвитку інформаційних систем і технологій;

бути спрямованою на вирішення проблем бізнесу, які стоять перед організаціями, що є базою дипломного проектування;

стимулювати студентів на творчий пошук нових науково-технічних, проектних та інших рішень;

викликати у студентів необхідність опрацювання спеціальної науково-технічної літератури;

передбачати вибір оптимального вирішення поставленого завдання на основі використання ефективних математичних методів та сучасних інструментальних засобів проектування.

Практична значимість теми дипломного роботи має визначатися за такими показниками:

тема роботи актуальна, становить інтерес для конкретних підприємств, установ, організацій тощо;

принципові рішення роботи можуть бути використанні в реальних проектах, та запроваджені на практиці;

у роботі наявні елементи новизни або оригінальні рішення;

робота рекомендована ДЕК до впровадження;

комплексність роботи.

Найбільш значимим є розроблення дипломних проектів, що мають практичне значення для підприємств, організацій, закладів, установ та виконані за їхнім завданням.

Теми дипломних проектів кафедра розробляє з урахуванням специфіки спеціальності, за якою здійснюється підготовка фахівців, вимог галузевих стандартів вищої освіти (ОКХ, ОПП, засобів діагностики) для освітньо-кваліфікаційного рівня спеціаліст, власного досвіду керівництва дипломним проектуванням, наукових досліджень та професійних інтересів професорсько-викладацького складу кафедри, замовлень і рекомендацій підприємств, науково-дослідних інститутів, галузевих міністерств і відомств тощо.

Кожна з тем повинна передбачати взаємопов'язаний розгляд питань аналізу та проектування, впровадження та користування створеною інформаційною системою; розроблення питань щодо охорони праці під час використання створеної системи. У дипломному проекті повинен бути чітко відображений системний підхід до вирішуваних завдань.

Дозволяється розробка комплексних тем. У цьому випадку вказується загальна тема проекту і теми кожної частини. Кожен студент зобов'язаний сформулювати та оформити окрему пояснювальну записку (додаток А). Комплексні проекти можуть бути кафедральними і міжкафедральними.

При виборі тем дипломного проекту потрібно ураховувати уподобання та можливості студентів, можливість максимального використання матеріалів курсового проектування, результатів науково-дослідної роботи студентів, практичний досвід роботи студента за фахом. При цьому переважна більшість тем повинна присвячуватися розробці реальних дипломних проектів.

3. Організація дипломного проектування

3.1. Загальні положення

Організація дипломного проектування покладається на випускаючу кафедру, якою є кафедра інформаційних систем (ІС). Для керівництва дипломними проектами на початку навчального року призначаються викладачі кафедри, узгоджуються консультанти з дипломного проектування з інших кафедр.

Кафедра ІС щорічно, не пізніше січня, затверджує тематику дипломного проектування на поточний навчальний рік, що зазначається в протоколі засідання кафедри. Теми дипломних робіт остаточно затверджуються наказом ректора університету. У разі необхідності зміни теми чи заміни керівника дипломної роботи, оформлюється лист на ім'я завідувача кафедрою. Згідно з листом готується наказ про зміну.

3.2. Переддипломна практика

Дипломне проектування фактично починається з переддипломної практики. Кафедра не менш як за місяць до переддипломної практики ознайомлює студентів з тематикою дипломного проектування, проводить розподіл тем серед студентів. Для затвердження теми дипломного проекту студент повинен подати заяву на ім'я завідувача кафедри (додаток Б).

За 10 днів до початку практики оформляється завдання на дипломний проект та календарний план-графік виконання дипломної роботи. Завдання розробляється керівником проекту за встановленою формою (додаток В). Завдання підписується керівником дипломного проекту, а також студентом, який своїм підписом засвідчує отримання завдання для виконання, та затверджується завідувачем кафедри. Оригінал завдання видається дипломнику і є необхідною складовою пояснювальної записки. Ксерокопія оригіналу повинна бути у керівника дипломного проекту. Студент складає календарний план-графік виконання дипломної роботи за встановленою формою, який узгоджується та затверджується керівником роботи.

Згідно з темою роботи студенти отримують індивідуальні завдання від керівника дипломної роботи щодо питань, які необхідно вирішити під час переддипломної практики – ознайомлення зі станом проблеми за темою дипломного проекту, збирання фактичних матеріалів, проведення

необхідних досліджень тощо.

Робота над дипломним проектом починається в період переддипломної практики.

Відповідно до теми дипломного проекту під час проходження практики студенти повинні:

- ознайомитися з напрямками діяльності, структурою управління, інформаційною інфраструктурою підприємства (організації);

- проаналізувати предметну область щодо теми дипломного проекту;

- проаналізувати існуючі варіанти розв'язання завдань, поставлених в дипломному проекті;

- розробити специфікацію вимог до модуля (системи);

- здійснити планування витрат на створення проекту.

Усі зібрані матеріали мають бути відображені в звіті з переддипломної практики і подані для перегляду і їх оцінки керівнику дипломного проектування.

Практика завершується складанням та захистом звіту про проходження практики.

3.3. Дипломне проектування

Процес розроблення і захисту дипломного проекту представляється кількома етапами:

- постановка завдань дослідження та визначення вимог до системи; проектування;

- оформлення пояснювальної записки та графічної частини дипломного проекту;

- підготовка доповіді й матеріалу для презентації проекту;

- попередній захист дипломного проекту;

- рецензування;

- захист дипломного проекту в ДЕК.

Робота над дипломним проектом організовується таким чином.

Кафедра ІС розробляє графік консультацій керівників та консультантів. Студент виконує роботи з підготовки дипломного проекту згідно з планом-графіком. Керівник та консультанти проводять консультації, здійснюють контроль, організовують перегляди дипломного проекту згідно з розкладом консультацій та планом-графіком виконання дипломного проекту. Виконання окремих етапів дипломного проектування помічається підписом керівника у план-графіку, а на

завершальному етапі – підписом керівника і консультантів з відповідних розділів на титульному аркуші пояснювальної записки.

Керівник і консультанти повинні надавати студенту методичну допомогу у вирішенні тих чи інших питань, застерігати від прийняття некваліфікованих хибних рішень, вказати студенту напрями пошуку та ін. Але це не звільняє студента від повної відповідальності за обґрунтованість прийнятих ним рішень, дотримання вимог нормативних документів і виконання календарного плану роботи.

Повідомлення керівників дипломних проектів про хід виконання календарних планів студентами-дипломниками мають регулярно заслуховуватися на засіданнях кафедри.

Студент, який не виконує графік дипломного проектування або значно відстав в його виконанні, повинний бути запрошеним для пояснення цього на засіданні кафедри, яка приймає відповідне рішення.

3.4. Попередній захист

З метою виявлення готовності студента до захисту виконується попередній захист дипломного проекту. Попередній захист складається з двох частин:

- доповідь з презентацією за виконаним повністю проектом та перевіркою пояснювальної записки;

- демонстрація роботи програмного продукту.

Мета попереднього захисту – перевірка готовності студента до захисту відповідно вимог випускаючої кафедри, оцінка обсягу поданого проекту і якості його виконання й оформлення. Попередній захист виконується не пізніше, ніж за два тижні до захисту. Незалежно від ступеня готовності проекту, студент повинний з'явитися на попередній захист.

Для проведення попереднього захисту кафедра ІС визначає склад комісій та складає графік попереднього захисту. Комісія формується з провідних викладачів випускаючої кафедри.

На попередній захист подаються:

- оформлене і своєчасно затверджене завдання;

- повністю оформлена, але не переплетена пояснювальна записка з підписами студента, керівника і консультантів завдання;

- готовий програмний продукт та відеоролик з презентацією його роботи;

- план доповіді та презентація, узгоджені з керівником.

На підставі заслуханої доповіді студента, його відповідей на питання, переглянутої пояснювальної записки, презентації комісія визначає рекомендації студенту:

щодо доповіді (скоротити або збільшити обсяг, на що звернути увагу, на чому зупинитися детальніше, про що взагалі не говорити тощо);

щодо відповідей на запитання (відповідати коротко, тільки по суті, без великих вступів тощо);

щодо змісту пояснювальної записки (додати відсутній пункт, схему, перевірити посилання на використану літературу тощо.);

щодо оформлення пояснювальної записки (перелік конкретних невідповідностей до вимог оформлення, поставити відсутні підписи тощо);

щодо презентації (розташувати слайди у відповідному порядку, які плакати вилучити, які змінити, які додати, що виділити на слайді, як його краще оформити тощо);

щодо демонстрації програмного додатка (вибрати більш наглядний тестовий приклад, ввімкнути звуковий супровід, підготувати образ ОС з програмним забезпеченням для віртуальної машини).

За кожним видом попереднього захисту комісією ставиться оцінка відповідно до критеріїв оцінювання. Допуск до захисту можливий за умови позитивного оцінювання за обома видами попереднього захисту (пояснювальна записка; програмна частка). Студенти, що не пройшли попередній захист, не допускаються до захисту; питання розглядається на засіданні кафедри за участю керівника і студента. Протокол засідання кафедри надається декану для вирішення питання. Недопуск студента до захисту перед ДЕК оформлюється наказом факультету з вказівкою причин. Допуск до захисту оформлюється екзаменаційною відомістю і передається в деканат. Після попереднього захисту дипломний проект підписує керівник проекту.

На останньому тижні дипломного проектування студент проходить нормоконтроль, після якого диплом підписує завідувач кафедрою і направляє на рецензування.

3.5. Захист дипломної роботи

На етапі захисту дипломної роботи студент отримує відгук керівника та рецензію від рецензента, подає роботу секретарю ДЕК (не пізніше, ніж за добу до захисту) та захищає роботу на засіданні ДЕК.

Форма відгуку керівника наведена у додатку Д, а рецензії – у додатку Е.

До ДЕК можуть подаватися й інші матеріали, що характеризують освітню та професійну компетентність випускника, наукову та практичну цінність виконаної ним дипломної роботи: друковані статті, заяви на патент, патенти, акти про впровадження результатів.

Захист дипломного проекту проводиться в ДЕК згідно з графіком захисту. Захист, як правило, проводиться у такій послідовності:

- оголошення секретарем ДЕК прізвища, імені та по батькові студента, теми його дипломного проекту – до 1 хвилини;

- доповідь студента (не більше 7 хвилин) у довільній формі про сутність роботи, основні проектні технічні рішення, отримані результати та ступінь виконання завдання до дипломного проектування. При цьому можуть використовуватися різні форми візуалізації доповіді, слайди, мультимедійні проектори, аудіо-, відеоапаратура тощо. Обов'язковим є роздатковий матеріал роботи, визначений завданням на дипломне проектування;

- демонстрація відеоролика програмного продукту (1 – 2 хвилини);

- відповіді на запитання членів комісії (до 15 хвилин);

- оголошення секретарем ДЕК відгуку керівника або виступ керівника зі стислою характеристикою роботи випускника в процесі дипломного проектування, ступеня його самостійності у вирішенні питань завдання до дипломного проектування, сильні та слабкі сторони як майбутнього фахівця, можливість присвоєння кваліфікації, особиста думка щодо подальшого використання, оголошення секретарем ДЕК рецензії на дипломну роботу (до 2-х хвилин);

- відповіді студента на зауваження керівника роботи та рецензента (до 2-х хвилин);

- оголошення голови ДЕК про закінчення захисту (до 1 хвилини).

Захист комплексного дипломної роботи, як правило, планується і проводиться на одному засіданні ДЕК. Студенту, який захищається першим, комісія доручає доповісти як про загальну частину роботи, так і про індивідуальну частину зі збільшенням (за необхідності) часу на доповідь. Усі студенти, які виконували комплексну роботу, повинні бути повною мірою обізнані із загальною частиною роботи і готові до запитань членів комісії не тільки з індивідуальної, а й із загальної частини роботи.

Після закінчення захисту всіх заявлених студентів комісія проводить закрите обговорення кожного захисту і оцінює його відповідно

до критеріїв оцінювання. Результати оцінювання захисту проекту доводяться до відома студентів після завершення роботи ДЕК.

Студентам, які не були допущені до захисту, може бути перенесений захист дипломного проекту на термін, визначений відповідними нормативними документами.

4. Структура, зміст і обсяг дипломного проекту

4.1. Обсяг роботи

Дипломний проект складається з пояснювальної записки (ПЗ) та обов'язкового графічного матеріалу (схеми, діаграми, графіки залежностей, таблиці, малюнки тощо), підготовленого у вигляді презентації, який демонструється при захисті. При захисті може використовуватись додатково демонстраційний матеріал у вигляді відеоролика. Крім того студент готує роздавальний матеріал для членів ДЕК, який містить роздруківку слайдів презентації.

Орієнтовний обсяг роботи:

пояснювальна записка у друкованому вигляді на аркушах А4 – 70 – 80 сторінок;

роздатковий матеріал для членів ДЕК зі стислим викладом положень дипломної роботи 15 – 20 сторінок;

презентація 15 – 20 слайдів.

Основні розділи пояснювальної записки проекту наведені у табл. 4.1.

Таблиця 4.1

Структура дипломного проекту

Найменування розділу	Кількість сторінок
Титульний аркуш	1
Завдання на дипломне проектування	2
Реферат	1
Зміст	1 – 2
Вступ	2 – 3
Розділ 1. Постановка завдань дослідження та визначення вимог до системи (модуля) <назва>	20 – 25
Розділ 2. Проектні та технічні рішення	25 – 30
Розділ 3. Розроблення заходів щодо ергономіки, забезпечення охорони праці та техніки безпеки на об'єкті	7 – 10
Висновки	2 – 3
Використана література	3 – 4
Додатки	без обмежень

Графічний матеріал, підготовлений у вигляді презентації, повинен включати:

- титульний слайд з вихідними даними щодо дипломного проекту;
- зміст презентації (з посиланнями на відповідні слайди);
- актуальність теми та мету дипломного проекту;
- модель організаційної структури підприємства, підрозділу підприємства;
- модель дерева функцій;
- модель управління бізнес-процесом;
- діаграму варіантів використання;
- розкадрування, що реалізують основну функціональність;
- оцінку трудовитрат за варіантами використання;
- діаграму компонент системи;
- математичну модель комплексу задач;
- заповнені форми вихідних та вхідних документів, діаграми, карти;
- логічну та фізичну моделі бази даних;
- діаграму класів, що реалізують основну бізнес-логіку програмної системи;
- діаграму станів основних варіантів використання;
- результати функціонального тестування програмного забезпечення;
- використані інструментальні засоби та технології;
- висновки по результатам дипломного проекту;
- апробацію результатів дипломного проекту;
- заключний слайд.

Рекомендується така структура пояснювальної записки дипломного проекту:

Вступ

Розділ 1. Постановка завдань дослідження та визначення вимог до системи (модуля) <назва>

1.1. Змістовний опис і аналіз предметної області

1.2. Огляд і аналіз існуючих варіантів розв'язання задач системи (модуля)

1.3. Специфікація вимог до системи (модуля)

1.3.1. Глосарій

1.3.2. Бачення

1.3.3. Розроблення варіантів використання

1.3.3.1. Діаграма варіантів використання

1.3.3.2. Специфікація варіантів використання

1.3.3.3. Розкадровка варіантів використання

1.3.3.4. Специфікація функціональних та нефункціональних вимог

1.4. Планування витрат на створення проекту

Розділ 2. Проектні та технічні рішення

2.1. Розроблення архітектури системи (модуля)

2.1.1. Опис архітектури системи (модуля)

2.1.2. Опис комплексу технічних засобів

2.1.3. Захист інформації

2.2. Математична постановка комплексу задач системи (модуля)

2.3. Опис вихідних і вхідних документів

2.4. Проектування БД

2.4.1. Концептуальне інфологічне проектування

2.4.1.1. Словник даних

2.4.1.2. Проектування інфологічної моделі даних

2.4.2. Проектування даталогічної моделі даних

2.4.2.1. Проектування даталогічної моделі даних в нотації IDEF1X

2.4.2.2. Обґрунтування властивостей моделі бази даних, що

спроектована

2.4.3. Проектування фізичної моделі даних

2.4.4. Програмна реалізація бази даних

2.5 Розроблення архітектури програмної системи

2.5.1. Розроблення архітектури програмної системи

2.5.2. Розроблення діаграми класів, які реалізують основну

бізнес-логіку програмної системи

2.5.3. Розроблення діаграми станів елементів графічного інтерфейсу користувача

2.5.4. Розроблення інтерактивної довідкової системи

2.5.5. Тестування програмного забезпечення

2.5.5.1. Специфікації тестів

2.5.5.2. Результати тестування програмного забезпечення

2.5.5.3 Оцінка якості розробки програмного забезпечення

Розділ 3. Розроблення заходів щодо ергономіки, забезпечення охорони праці та техніки безпеки на об'єкті

Висновки

Використана література

Додатки

Зміст пояснювальної записки формується відповідно до вимог, наведених в методичних рекомендаціях, і включає назви розділів, підрозділів та пунктів [48].

При комп'ютерному наборі тексту використовувати шрифт "Times New Roman", кегль – 14, друк через 1,2 інтервалу. Границі тексту – 30 мм від лівого краю аркуша, 10 – від правого, по 20 мм від верхнього та нижнього. Вимоги до оформлення пояснювальної записки наведені в методичних рекомендаціях [48].

4.2. Вимоги до змісту пояснювальної записки

4.2.1. Титульний аркуш

Титульний аркуш містить у собі назву дипломного проекту та підписи студента-дипломника, керівника, консультантів і завідуючого кафедрою. Зразок титульного аркуша наведений у додатку А.

4.2.2. Завдання до дипломного проекту

Завдання до дипломного проекту містить у собі тему дипломного проекту, термін здачі студентом закінченого проекту, вхідні дані до проекту, короткий зміст пояснювальної записки, перелік графічного матеріалу та календарний план виконання дипломного проекту. Зразок завдання до дипломного проекту наведений у додатку В.

4.3.3. Реферат

Реферат – це короткий виклад змісту пояснювальної записки, що включає основні фактичні відомості та висновки, необхідні для початкового ознайомлення з пояснювальною запискою до дипломного проекту. Реферат має бути розміщений після завдання на дипломне проектування.

Реферат містить:

дані про обсяг пояснювальної записки, кількість ілюстрацій, таблиць, додатків, джерел у переліку бібліографічних посилань;

текст, який відображує інформацію, подану у пояснювальній записці, в такій послідовності:

об'єкт проектування;

предметна область проектування;

мета проектування;

методи дослідження й розробки;

результати та їх новизна;

основні технологічні характеристики та показники;

рекомендації щодо використання результатів дипломного проекту;

область застосування;

висновок та прогноз стосовно розвитку розробок проекту;
перелік ключових слів.

Обсяг реферату повинен бути не більше 500 слів.

Текст реферату на пункти не поділяють. Частини тексту реферату, стосовно яких відсутні відомості, пропускають.

Ключові слова існують для розкриття суті проекту та для розповсюдження інформації про розробку. Ключове слово – це слово або словосполучення з тексту документу, яке з точки зору інформаційного пошуку несе смислове навантаження. Перелік ключових слів повинен включати від 5 до 15 слів (словосполучень) в називному відмінку. Перелік подається по ряду через кому великими буквами. Перший рядок – з абзацного відступу, вирівнювання "за шириною".

Приклад реферату наведено у додатку Ж.

4.2.4. Зміст

Зміст складається з назв усіх розділів, підрозділів, пунктів та назв додатків пояснювальної записки із зазначенням відповідних сторінок [48].

4.2.5. Перелік скорочень та умовних позначень

У випадку, коли в дипломному проекті вжито специфічну термінологію, маловідомі скорочення, нові символи, позначення або вони використовуються більше трьох разів, подається сторінка "Перелік скорочень та умовних позначень", яка розташовується перед вступом. При використанні скорочень, термінів менше трьох разів, вони до переліку не включаються, бо розшифровку можна навести в тексті при першому застосуванні. Перелік друкується двома колонками: ліворуч за абеткою наводяться скорочення, праворуч – детальна розшифровка (приклад наведено у додатку З).

4.2.6. Вступ

У вступі необхідно обґрунтувати вибір теми дипломного проекту, лаконічно розкрити її актуальність і значущість для підприємства, показати, якою мірою ця проблема висвітлена в літературі, розроблена теоретично й вирішена на практиці. Конкретизувати мету й завдання проекту, визначити об'єкт і предметну область проектування.

Таким чином, структура вступу повинна включати:

а) актуальність теми дипломного проекту;

- б) мету й завдання проектування;
- в) об'єкт і предмет проектування;
- д) засоби проектування, які використовувались у дипломному проєкті;
- е) можливі галузі застосування отриманих результатів дипломного проєкту.

Розглядаючи актуальність теми, дипломник має стисло розкрити сутність і стан проблеми (завдання), її значення, підставу й вихідні дані для розробки, обґрунтувати необхідність проведення даного проектування. У вступі обов'язково повинне бути переконливе обґрунтування, чому саме розроблені вітчизняні і зарубіжні проєктні рішення в даній області мають потребу в подальшому розробленні в дипломній роботі.

Мета проектування – створення проєкту системи (модуля), який становить технічна документація з докладним описом усіх проєктних рішень.

Досягнення мети проєкту здійснюється шляхом вирішення в процесі проектування таких завдань:

- аналіз предметної області стосовно проблеми, що стоїть перед бізнесом організації;
- огляд і аналіз існуючих варіантів розв'язання завдань системи (модуля);
- специфікація вимог до системи (модуля);
- планування витрат на створення проєкту;
- опис архітектури системи (модуля);
- математична постановка комплексу задач системи (модуля);
- опис вихідних і вхідних документів;
- проектування БД;
- розроблення програмного забезпечення;
- розроблення заходів щодо ергономіки, забезпечення охорони праці та техніки безпеки на об'єкті.

Об'єктами проектування ІС є окремі елементи або комплекси їх функціональних і забезпечувальних частин. Так, функціональними елементами відповідно до традиційної декомпозиції є завдання, комплекси завдань і функції управління. У складі забезпечувальних частин ІС об'єктами проектування слугують елементи і комплекси інформаційного, програмного та технічного забезпечення системи.

Предметною областю проектування ІС є система організаційно-економічного управління діяльністю підприємства.

Засобами проектування ІС – це комплекс інструментальних засобів комплекс, які забезпечують х в рамках обраної методології проектування підтримку життєвого циклу ІС.

Далі у вступі наводяться відомості про можливі галузі практичного

застосування результатів дипломного проекту.

Таким чином, вступ містить усі необхідні кваліфікаційні характеристики дипломної роботи і є дуже відповідальною її частиною.

4.2.7. Розділ 1. Постановка завдань дослідження та визначення вимог до системи (модуля) <назва>

Метою розділу 1 є проведення детального аналізу проблеми, яка виникла на об'єкті управління (підприємстві) при веденні бізнесу, вибір шляхів її вирішення та розроблення специфікації вимог до проектного модуля.

У підрозділі 1.1. *Змістовний опис і аналіз предметної області* необхідно: коротко описати напрями діяльності об'єкту управління (підприємства, організації);

розробити схему організаційної структури управління підприємством (додаток К, рис. К.1);

визначити проблему бізнесу, яку слід вирішити, та бізнес-процеси, які пов'язані з вирішенням даної проблеми, вказати, які підрозділи організаційної структури їх виконують;

розробити схему організаційної структури підрозділу (підрозділів), пов'язаних з визначеними бізнес-процесами (додаток К, рис. К.2);

визначити склад функцій, що входять до бізнес-процесу, розробити діаграму дерева функцій (додаток К, рис. К.3);

розробити схему управління бізнес-процесом (додаток К, рис. К.4) та описати його (табл. 4.2).

Таблиця 4.2

Характеристика бізнес процесу <Назва>

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	
Основні учасники*	
Вхідна подія	
Вхідні документи**	
Вихідна подія	
Вихідні документи**	
Клієнт бізнес-процесу***	

*для кожного учасника указати структурний підрозділ, посаду, його роль у бізнес-процесі

** навести перелік документів

*** процес, що використовує інформацію процесу

Для моделювання предметної області використовують CASE-

інструменти.

У процесі моделювання необхідно виділити, як мінімум, **три бізнес-завдання** в модулі, які повинні бути інформаційно зв'язані та реалізовувати єдину технологію процесу управління об'єктом. У складі задач модуля одна з них повинна бути **аналітичною**.

У процесі моделювання необхідно виділити транзакційну складову бізнес-процесу, яка забезпечує збір, накопичення та обробку кількісних даних про поточний стан об'єкта управління, а також аналітичну складову, яка забезпечує аналіз кількісних показників, сформованих у транзакційні складові.

Аналітична складова бізнес-процесу повинна забезпечити дослідження кількісних показників у різних розрізах та вимірах: за періодами часу, за товарами (продукцією), за клієнтами, підрозділами.

Проведення такого багатоаспектного аналізу забезпечить інформаційну підтримку прийняття рішень, спрямованих на вирішення виявленої проблеми.

У підрозділі 1.2. *Огляд і аналіз існуючих варіантів розв'язання завдань модуля* треба:

виконати аналіз функціональності й інтерфейсу двох або більше програмних продуктів, призначених для автоматизації бізнес-процесів розроблюваного модуля (системи):

представити в таблиці порівняльну характеристику програмних продуктів за такими характеристиками (табл. 4.3):

фірма-розробник;
назва програмного продукту;
версії продукту;
функціональність;
інтерфейс користувача;
допомога користувачу;
тощо.

Таблиця 4.3

Порівняльна характеристика програмних продуктів

Фірма-розробник	<Назва фірми-розробника>	<Назва фірми-розробника >	...
Назва програмного продукту			
Версії продукту			
...			

Для кожного з програмних продуктів навести та коротко описати екранні форми, що характеризують основні варіанти використання продукту.

Зробити висновок щодо можливості використання досвіду ведучих фірм-розробників програмних продуктів, використання їх рішень при розробленні проектних рішень проекту.

У підрозділі 1.3. Специфікація вимог до модуля необхідно розробити глосарій проекту, бачення, варіанти використання з їх розкадровкою, навести специфікацію функціональних та не функціональних вимог.

Пункт 1.3.1. Глосарій

Глосарій – це словник основних використовуваних термінів. Цей документ є найпершим результатом концептуального аналізу предметної області. Глосарій можна розглядати як документ, що засвідчує спільне розуміння основної термінології Замовником і Розробником.

Крім того, глосарій є відправною точкою для побудови більш розгорнутих моделей предметної області, які на стадії реалізації інформаційної системи покладено в основу об'єктної моделі (для об'єктно-орієнтованих застосувань) і моделі даних (для генерації схеми бази даних).

Глосарій проекту представити у вигляді табл. 4.4.

Таблиця 4.4

Глосарій проекту

Термін	Опис терміну
1. Основні поняття та категорії предметної області	
2. Користувачі системи	
3. Вхідні та вихідні документи	

Пункт 1.3.2. Бачення

Бачення (Vision) – це короткий опис суті майбутнього продукту.

У ньому коротко описується, що є продуктом, які цілі і завдання його створення, хто його користувачі і які основні можливості майбутньої системи.

У баченні потрібно визначити:

1) позиціонування виробу:

- короткий опис ділових переваг, що досягаються проектом;
- визначення проблем (табл. 4.5).

Таблиця 4.5

Визначення проблеми

Проблема	(опис проблеми)
Зачіпає	(співвласники, який зачіпає проблема)
Її наслідком є	(який вплив проблеми)
Успішне рішення	(список деяких ключових переваг від успішного рішення)

2) ідентифікувати співвласників і користувачів:

- опис співвласників і користувачів;
- користувальницьке середовище;
- профілі співвласників та профілі користувачів.

Ідентифікація співвласників і користувачів передбачає пошук і фіксацію зацікавлених осіб проекту – **стейкхолдерів** (від англ. stakeholder) – представників організації Замовника і організації Розробника, інвесторів, зовнішніх експертів та ін.

Відомості про співвласників представляє підсумковий список усіх ідентифікованих співвласників (табл. 4.6)

Таблиця 4.6

Відомості про співвласників

Назва	Представляє	Роль

Відомості про користувачі представляють підсумковий список усіх ідентифікованих користувачів (табл. 4.7).

Таблиця 4.7

Відомості про користувачів

Назва	Опис	Співвласник

Користувальницьке середовище включає опис робочого середовища цільового користувача: кількість користувачів, час та періодичність вирішення завдань, використовувані програмно-технічні платформи тощо.

Профілі співвласників включають опис кожного типу співвласників системи. Для кожного типу співвласника складається таблиця (табл. 4.8)

Таблиця 4.8

Профіль співвласника <Назва>

Типовий представник	
Опис	
Тип	
Відповідальності	
Критерій успіху	
Участь	
Обов'язки	

Профілі користувачів містять опис кожного унікального користувача системи. Для кожного типу користувача заповнюється таблиця (табл. 4.9).

Таблиця 4.9

Профіль користувача <Назва>

Типовий представник	
Опис	
Тип	
Відповідальності	
Критерій успіху	
Участь	

Пункт 1.3.3. Розроблення варіантів використання включає:

- діаграму варіантів використання;
- специфікацію варіантів використання;
- розкадровку варіантів використання;
- специфікацію функціональних та нефункціональних вимог.

Підпункт 1.3.3.1. Діаграма варіантів використання містить діаграму варіантів використання.

Діаграма варіантів використання відображає функціональність, яка буде реалізована в програмному продукті. Необхідно навести опис потоків подій і діаграми послідовності.

Варіант використання можна розглядати як функцію, що реалізовується системою. Однак будь-яка функція повинна мати цінність і давати можливість отримати кінцевий результат для кінцевого користувача продукту або послуги. Тому при специфікації варіанта використання виділяють серед усього функціоналу системи лише ту функціональність, яка:

- корисна конкретному кінцевому користувачеві;
- дозволяє отримувати користувачеві конкретні закінчені результати.

Перш ніж приступити до специфікації вимог у формі варіанта використання, в RUP звичайно складають реєстр (список) акторів (actors) і варіантів використання.

Актор – це хтось або щось, що володіє активністю відносно програмної системи. Актором звичайно буде користувач системи. Окрім користувача, як актор може розглядатися інша програмна система, апаратний пристрій, у ряді випадків – активна компонента самої системи. Пошук акторів корпоративної інформаційної системи зазвичай зводиться до аналізу ролей різних користувачів. Вибір акторів залежить від їх функціональних обов'язків, розмежування доступу, способів використання інформаційної системи.

Підпункт 1.3.3.2. Специфікація варіантів використання

Кожен варіант використання повинен мати опис. У дипломному проекті навести опис варіантів використання, що реалізують основну функціональність (зазвичай крім ведення довідників) у вигляді таблиці (табл. 4.10).

Таблиця 4.10

Варіант використання <Назва>

Контекст використання	
Дійові особи	
Предумова	
Триггер	
Сценарій	
Постумова	

Назва – коротка фраза в вигляді дієслова в неозначеній формі завершеного виду, яка відображає мету.

Контекст використання містить уточнення мети, а при необхідності – умови її нормального завершення.

Дійові особи:

основна дійова особа – це ім'я ролі основного актора або його опис;
учасники і інтереси – перелік інших акторів-учасників варіанту з вказівкою їх інтересів.

Передумова – варіант використання, який має бути обов'язково виконаний, щоб можна було виконати даний варіант. Передумову описує стан, у якому система повинна перебувати до початку виконання варіанту.

Тригер – подія предметної області, що викликає використання прецеденту (варіанта використання).

Сценарій – перелік кроків сценарію: номер кроку – опис дії.

Постумова – варіант використання, який обов'язково має бути виконаний після виконання даного варіанту; це стан, у якому система повинна перебувати після закінчення виконання варіанту; це те, що гарантується акторам-учасникам, не залежно від успіху виконання даного варіанту. Наприклад, у разі невдалої транзакції всі дані, що були в системі до її початку, зберігаються незмінними.

Події, що описуються передумовами або постумовами, мають бути станами, які користувач може спостерігати.

Підпункт 1.3.3.3. Розкадровка варіантів використання

Розкадровка (storyboard) – це логічний і концептуальний опис функціональних можливостей системи для певного сценарію, включаючи необхідну взаємодію між системою та її користувачами. Метою розкадрування є отримання ранньої реакції користувачів на пропоновану концепцію системи за допомогою некоштовних засобів. У якості інструментальних засобів розкадровки вимог використовуються Microsoft Word, Microsoft Visio, Microsoft PowerPoint, IBM Rational Requirements Composer, Expression Blend Sketch Flow.

Розкадровка – це представлення варіантів використання по кадрах (ескізам екранних форм).

Розкадровка дозволяє зацікавленим особам описати свої потреби аналітикам, які на основі цих потреб визначають вимоги до системи, здійснюють перевірку відповідності вимог поставленим бізнес-завдань і здійснюють з ними зворотний зв'язок.

За результатами створення розкадрування повинні бути сформульовані вимоги (з можливістю відстежити процес "в зворотному напрямі" аж до конкретного елемента розкадрувань), які підлягають узгодженню з зацікавленими особами. Потім ці вимоги повинні бути включені в відповідний набір вимог. Результати розкадрувань використовуються для отримання схвалення зацікавлених осіб, виявлення джерела вимог в рамках розкадрувань.

Підпункт 1.3.3.4. Специфікація функціональних та нефункціональних вимог

Специфікацію функціональних вимог представити в табл. 4.11.

Таблица 4.11

Специфікація функціональних вимог

Ідентифікатор вимоги	Назва вимоги (варіанту використання)	Атрибути вимог		
		Пріоритет	Складність	Контакт

Пріоритет – заповнюється аналітиком, показує пріоритет реалізації вимоги для клієнта. Використовується при управлінні проектом і визначає пріоритет розробки. Можливі значення: обов'язкове, рекомендоване, за вибором (опційне).

Складність – заповнюється менеджером проекту, показує рівень трудовитрат, пов'язаних з реалізацією вимоги. Використовується при управлінні проектом і визначає пріоритет розробки. Складність виконання вимоги може виражатися у вигляді трудомісткості і вказувати кількість людино-днів, потрібних для його реалізації або у вигляді значень шкали: висока, середня, низька.

Контакт – заповнюється аналітиком, ідентифікує людину, яка може надати необхідну інформацію про вимогу. Використовується для гарантії, що розробники можуть отримати інформацію, необхідну їм для реалізації вимоги. Якщо в проекті в якості контакту виступає одна людина – цю колонку можна опустити, але треба ідентифікувати контакт один раз в тексті цього пункту.

Специфікацію нефункціональних вимог представити в таблиці

Специфікація нефункціональних вимог

Ідентифікатор вимоги	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Контакт
1. Застосовність				
2. Надійність				
...				

Нефункціональні вимоги можна поділити на такі групи:

1. Застосовність.

Час, необхідний для навчання звичайних і досвідчених користувачів.

Вимірний час відгуку для типових завдань.

Основні вимоги застосовності нової системи відносно інших систем, які знають користувачі.

Вимоги по відповідності загальним стандартам застосовності, наприклад, стандартам інтерфейсу користувача IBM або стандартам графічного інтерфейсу користувача Microsoft для Windows.

2. Надійність.

Доступність – визначає % доступного часу (xx.xx %), час використання, час, що витрачається на обслуговування, порушення режиму роботи і т. д.

Середній час безвідмовної роботи зазвичай визначається в годинах, але може також визначатися в днях, місяцях або роках.

Середнє напруження до ремонту – як довго системі дозволяють працювати до того, коли повинне бути проведене її обслуговування.

Точність – визначає розрядність (роздільну здатність) і точність (за деяким відомим стандартом), які потрібні у вихідних даних системи.

Максимальна норма помилок або дефектів – зазвичай виражається в термінах кількості помилок на тисячу рядків коду або кількості помилок у функціональній одиниці.

3. Робочі характеристики.

Повинні бути виділені характеристики продуктивності системи. Включіть конкретні характеристики швидкодії. Де можливо, потрібно

зробити посилання на пов'язані прецеденти за іменем.

Швидкодія для транзакції (середнє значення, максимальне).

Продуктивність (наприклад, число транзакцій за секунду).

Місткість (наприклад, число замовників або транзакцій, яке може розміщувати система).

Режими зниженої продуктивності (що є прийнятним режимом роботи, коли система стала деякою мірою гіршою).

Використання ресурсів: пам'яті, дискового простору, комунікацій і т. д.

4. Експлуатаційна придатність.

Указують всі вимоги, які розширюють експлуатаційну придатність або надійність формованої системи, включаючи стандарти кодування, угоди про імена, бібліотеки класів і утиліти підтримки.

5. Проектні обмеження.

Повинні містити всі проектні обмеження до формованої системи. Проектні обмеження становлять рішення, які були сформульовані як обов'язкові і повинні твердо витримуватися. Прикладами можуть бути мови програмування, вимоги до технології програмування, обов'язкове використання інструментальних засобів розробки, архітектурні і конструктивні обмеження купованих компонентів, бібліотек класів і т. д.

6. Вимоги до призначеної для користувача документації і до системи допомоги.

Описують вимоги, якщо вони є, до інтерактивної документації користувача, до системи довідки, до попереджувальних повідомлень і т. д.

7. Куповані компоненти.

Описують всі куповані компоненти, які має використовувати система, всі вживані ліцензії або обмеження щодо використання і всі відомості про сумісність і / або здібність до взаємодії або про стандарти інтерфейсу.

8. Інтерфейси.

Визначають інтерфейси, які повинні бути підтримані додатком. Він має містити адекватну специфіку, протоколи, порти і логічні адреси і т. д. так, щоб програмне забезпечення могло бути розроблене і перевірене на відповідність вимогам інтерфейсів.

8.1. Інтерфейси користувача.

Описуються інтерфейси користувача, які повинні бути реалізовані програмним забезпеченням.

8.2. Апаратні інтерфейси.

Визначаються всі апаратні інтерфейси, які повинні бути підтримані програмним забезпеченням, включаючи логічну структуру, фізичні адреси, очікувану поведінку і т. д.

8.3. Програмні інтерфейси.

Описуються програмні інтерфейси з іншими компонентами програмної системи. Це можуть бути куповані компоненти, багато разів використані компоненти з іншої прикладної програми або компоненти, що розробляються для підсистем поза контекстом цих специфікацій вимог до програмного забезпечення (SRS), але з яким ця прикладна програма повинна взаємодіяти.

8.4. Комунікаційні інтерфейси.

Описуються всі комунікаційні інтерфейси до інших систем або пристроїв типу локальних мереж, видалених послідовних пристроїв і т. д.

9. Вимоги до ліцензування.

Визначаються всі вимоги обов'язкового ліцензування або інші вимоги обмеження використання, які повинні виконуватися програмним забезпеченням.

10. Застереження щодо питань, пов'язаних з авторськими правами.

Описуються всі необхідні юридичні застереження, гарантії, оголошення про авторське право, право спадкоємства, торгівельні марки або емблеми для програмного забезпечення.

11. Вживані стандарти.

Вказуються посилання на всі вживані стандарти і на конкретні розділи таких стандартів, які відносяться до описуваної системи. Наприклад, це можуть бути правові і регулюючі стандарти, стандарти якості, промислові стандарти щодо застосовності, здібності до взаємодії, інтернаціоналізації, відповідності операційній системі і т. д.

У специфікації вказують лише ті нефункціональні вимоги, які є суттєвими для проекту.

Підрозділ 1.4. Планування витрат на створення проекту

У цьому підрозділі необхідно виконати попередню оцінку трудовитрат за кожним варіантом використання та фазами проекту. Результати планування представити у вигляді таблиць (табл. 4.13; 4.14).

Попередня оцінка трудовитрат за варіантами використання

Ідентифікатор вимоги	Назва варіанту використання	Трудовитрати, люд./год

Попередня оцінка трудовитрат за фазами проекту

Фаза проекту	Трудовитрати, люд./год	Відсоток трудовитрат

4.2.8. Розділ 2. Проектні та технічні рішення*Підрозділ 2.1. Розроблення архітектури системи (модуля)*

У підрозділі 2.1.1. *Опис архітектури системи (модуля)* для опису архітектури системи необхідно визначити загальну технологію рішення завдання, тобто обрати:

- трирівневу архітектуру "клієнт-сервер";
- дворівневу архітектуру "клієнт-сервер";
- локальне рішення на одному ПК.

У рамках опису архітектури системи при використанні технології "клієнт-сервер" слід визначити та вказати зв'язки між такими компоненти системи:

презентаційна логіка (користувальницький інтерфейс), яка призначена для роботи з даними користувача;

бізнес-логіка (сервер додатків), яка призначена для перевірки правильності даних, підтримки посилальної цілісності;

логіка доступу до ресурсів (сервер БД), яка призначена для зберігання даних.

Опис архітектури системи необхідно здійснити за допомогою діаграм розміщення та діаграм компонентів в UML.

Діаграма розміщення (deployment diagram) – діаграма, на якій

відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Діаграма розміщення відображає фізичні взаємозв'язки між програмними та апаратними компонентами системи, її основні елементи:

вузол (node) – обчислювальний ресурс (ПК, сервер БД, WEB-сервер, сервер додатків);

з'єднання (connection) – канал взаємодії вузлів.

Діаграма компонент більш детально відображає фізичний рівень системи. На цій діаграмі необхідно визначити компоненти ПЗ і зв'язки між ними. На такій діаграмі звичайно виділяють два типи компонентів: виконувані компоненти та бібліотеки коду. У моделі системи може бути кілька діаграм компонентів, в залежності від числа підсистем та модулів.

На рис. 4.1. наведено приклад діаграми розміщення, на рис. 4.2. наведено приклад діаграми компонентів.

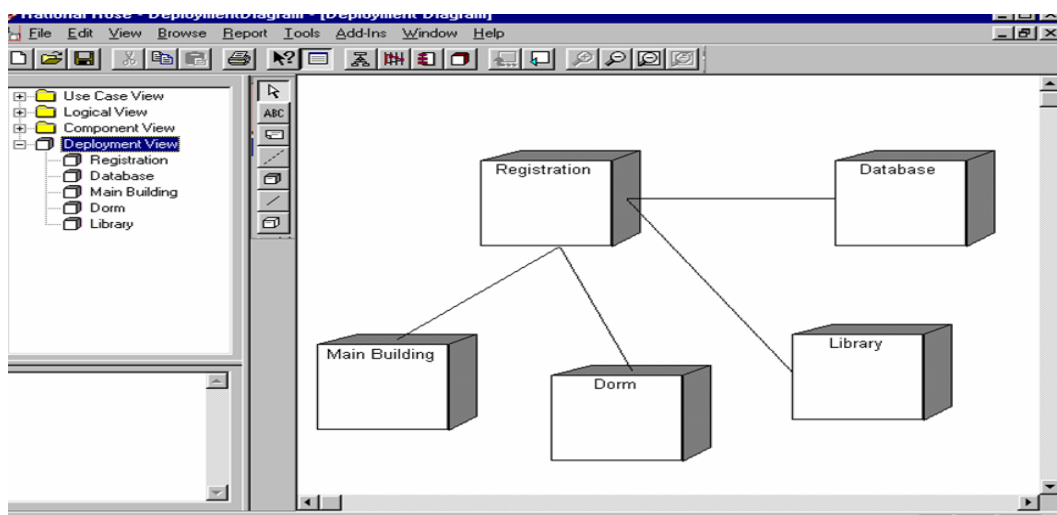


Рис. 4.1. Діаграма розміщення

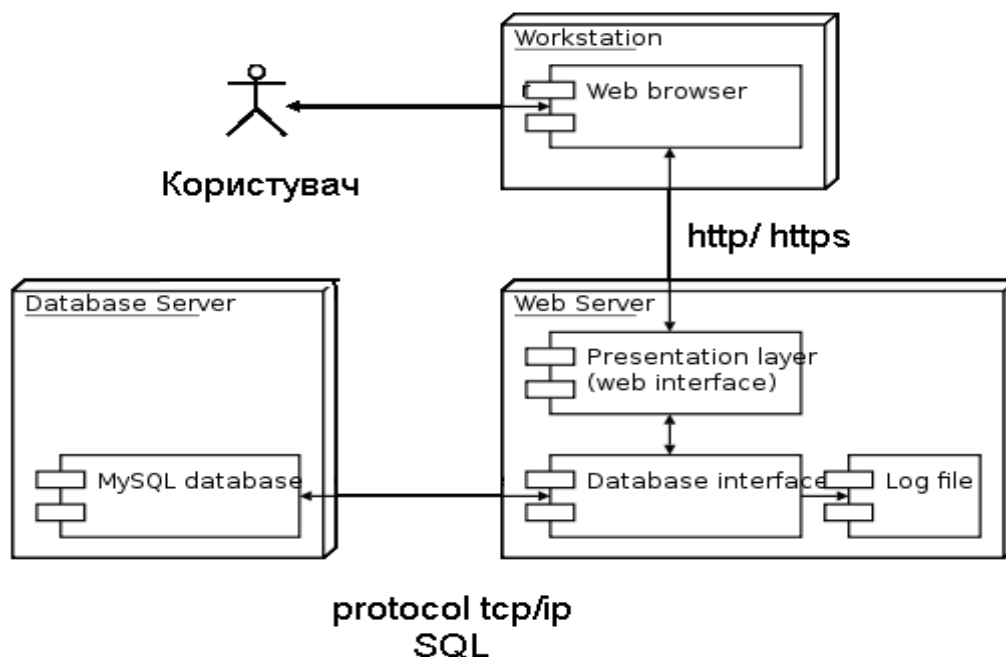


Рис. 4.2. Діаграма компонентів

Підпункт 2.1.2. Опис комплексу технічних засобів (КТЗ)

Опис апаратного забезпечення проектного рішення ґрунтується на нефункціональних вимогах проекту (табл. 4.1), та визначених технічних засобах (WEB-сервери, сервери БД, сервери додатків, ПК, мережні пристрої), які присутні на діаграмах розміщення та діаграмах компонентів. Загальний склад КТЗ може бути наведено у вигляді табл. 4.15.

Таблиця 4.15

Опис необхідних компонентів КТЗ

Найменування компонента КТЗ (пристрою)	Характеристики (вимоги), які є ключовими для компонентів КТЗ (пристрою)	Кількість пристроїв
Сервер БД		
WEB сервер		
ПК		
Мережний пристрій (Switch)		
Середина передачі даних		

Детальний опис нефункціональних вимог необхідно здійснити для найбільш важливих пристроїв КТЗ. Наприклад, при використанні "клієнт-серверної" технології необхідно навести детальний опис сервера (ів) та

ПК клієнта, при локальному рішенні – ПК користувача.

Приклад детального опису нефункціональних вимог, щодо компонента комплексу технічних засобів (на прикладі вимог до серверу) наведено у табл. 4.16.

Таблица 4.16

Детальний опис нефункціональних вимог до КТЗ

Характеристики	Вимоги
1	2
Основні компоненти сервера	
Процесор	Не менше ніж 2 (два) Intel Pentium IV Xeon 2800 МГц або вище. Системна шина FSB 800 МГц
Підсистема пам'яті	Тип пам'яті – ECC DDR2 SDRAM, PC4300
	Об'єм пам'яті – не менше 4 Гбайт
	Розширюваність – до 16 Гбайт
Відео підсистема	Контролер – AGP або PCI графічний адаптер
Підсистема зберігання	RAID контролер: не менше ніж 2 (два) канали Ultra320 SCSI LVD; не менше ніж 64 Мбайт кеш пам'яті, розширюваної до 128 Мбайт; PCI шина 64-bit; підтримка рівнів RAID 0,1,1+0,5,50; не менше 2 (двох) зовнішніх SCSI LVD роз'єми
	SCSI контролер – не менше ніж 2 (два) канали Ultra320 SCSI
	HDDs: кількість – не менше 8 (восьми); ємність – не менше 146 Гбайт кожен; тип – Hot-swap SCA; інтерфейс – Ultra320 SCSI
	CD-ROM - швидкість не менше ніж 40x
Підсистема вводу-виводу	I/O слоти: не менше ніж 6 (шість) PCI 64біт 33 МГц і 66 МГц слотів; не менше ніж 2 (два) PCI 64 біти 66 МГц слота з підтримкою PCI hot-plug функції
	I/O порти: не менше одного 25-pin рівнобіжного порту (ECP/EPP); не менше двох 9-pin послідовних портів (16550 UART); не менше ніж 2 (два) PS/2 рознімання для клавіатури і миші

1	2
Мережний інтерфейс	Кількість адаптерів – не менше ніж 2 (два) еквівалентних адаптори. Підтримуваний стандарт: 1000BASE-T; 100BASE-TX. Додаткові характеристики: 1000BASE-T/100BASE-TX autosensing; Bus master; Wakeup on LAN. Підтримка Adapter Fault Tolerance technology, Adaptive Load Balancing, Link Aggregation, Fast EtherChannel technologies
Конструкція	Під установку в стійку промислового стандарту 19"
Аксесуари	PS/2 миша; PS/2 анг./рос./укр. клавіатура; Монтажний комплект для установки в стійку 19"
Відмовостійкість	
Джерела живлення	не менше ніж 2 (два) hot-swap джерела живлення з N+1 надмірністю
ПЗ управління, що установлене на сервері	
Операційна система	Microsoft Windows Server 2003
Контроль дефектів	виявлення перегріву процесора; коливання напруги; виявлення помилок диска; виявлення помилок пам'яті; стан системи вентиляції; автоматична реєстрація подій прояву несправностей
Аварійне керування	автоматичне повідомлення про досягнення заданих користувачем граничних значень; автоматичне вимикання сервера при досягненні критичної температури
Віддалене керування	віддалена консоль; дистанційне виконання адміністративних консольних команд; дистанційна установка CMOS
Антивірусне ПЗ	Антивірусне ПЗ для Microsoft Windows Server 2003

Підпункт 2.1.3. Захист інформації

Підрозділ захисту інформації має на увазі розробку елементів політики безпеки на інформаційну систему і програмний продукт, що розробляється. Політика безпеки інформаційної системи повинна включати такі пункти:

1. Перелік ресурсів і циркулюючої інформації (даних), які є критичними (конфіденційними) для цієї інформаційної системи. Кожен ресурс або інформація повинні супроводжуватися коротким обґрунтуванням, чому їх варто відносити до конфіденційних даних. Тут слід виділити інформацію, яка потрапляє під дію законів:

закону України "Про інформацію", в якому дається поняття конфіденційної інформації;

закону України "Про захист персональних даних", який визначає, що таке персональні дані і регламентує правила їх обробки;

закону України "Про банки і банківську діяльність", в якому регламентується правила використання банківської інформації.

Перелік конфіденційної інформації, циркулюючої в ІС, рекомендується оформити у вигляді таблиць (див. табл. 4.17; 4.18). Відносно кожного виду інформації вказати бізнес-процеси, в яких вона використовується, і в колонці "обґрунтування" вказати, чому її варто відносити до конфіденційної інформації.

Таблиця 4.17

Перелік конфіденційної інформації циркулюючої в системі

Назви інформації	Назва бізнес-процесу, в якому використовується	Обґрунтування

Таблиця 4.18

Перелік критично важливих ресурсів інформаційної системи

Назва ресурсу	Назва бізнес-процесу, в якому використовується	Обґрунтування

2. Перелік осіб, що мають доступ до конфіденційної інформації, циркулюючої в інформаційній системі. У цьому пункті слід розписати права доступу до ресурсів бази даних відносно кожного користувача, способи і види доступу до ресурсів мережі інформаційної системи, а також вказати доступність ресурсу з Інтернету. У якості доступності ресурсів в мережі Інтернет потрібно вказати необхідні порти і протоколи, що вимагаються для правильного функціонування ресурсу.

Перелік осіб, які обробляють конфіденційну інформацію, представити у вигляді таблиці (див. табл. 4.19). Для зменшення кількості рядків, можна групувати інформацію, що має однакові права доступу.

Таблиця 4.19

Перелік користувачів, що мають доступ до конфіденційної інформації і критично важливих ресурсів

Користувач ІС	Конфіденційна інформація або ресурс	Права доступу						
		Читання	Запис	Оновлення	Видалення	Архів ація	Відновлення	т.п.
Адміністратор	Паролі до бази даних	+	+	+	–	+	+	

3. Здійснити аналіз інформаційної системи з точки зору криптоаналітика. Виявити слабкі компоненти системи і сформулювати перелік можливих загроз безпеки. Слід класифікувати загрози безпеки відносно основних послуг безпеки, таких як: конфіденційність, цілісність, доступність, причетність, спостережуваність, автентичність та інші. Перелік загроз представити у вигляді табл. 4.20.

Таблиця 4.20

Перелік загроз безпеки інформаційної системи

Назва загрози	Де виникає (бізнес-процес, ресурс)	На який ресурс або інформацію спрямована	Яка послуга безпеки порушується

У табл. 5.19 відносно кожної загрози слід вказати мету загрози і можливе місце виникнення/проникнення загрози. Метою загрози можуть бути певні ресурси системи, робота яких може бути порушена у випадку її проходження. Це проводиться для можливості оцінки завданого збитку, а також для правильного вибору механізмів захисту. Під місцем виникнення/проникнення розуміють ресурс системи або бізнес-процес, в якому може виникнути загроза або через який може проникнути загроза. Виявлення таких тонких місць в ІС дозволить обґрунтовано застосовувати механізми безпеки для відповідних ресурсів системи.

4. Визначити відносно кожного механізму безпеки, якими засобами він може бути реалізований і на перекриття яких загроз спрямований. Під засобами захисту розуміються конкретні програмні продукти або спеціалізовані сервісні функції ОС, Web серверів, баз даних. Перелік механізмів безпеки і засобів захисту представити у вигляді табл. 4.21.

Таблиця 4.21

Перелік механізмів безпеки, що перекривають загрози безпеки ІС

Механізм безпеки	Засіб захисту	Загроза безпеці

5. Дати детальний опис засобів захисту використовуваних в програмному продукті, що розробляється. Тут слід зазначити використані криптографічні методи захисту, методи автентичності користувачів, спосіб зберігання паролів у системі, засоби автентичності при підключенні до бази даних, використання захищених протоколів SSL і т. д.

Для розрахованих на багато користувачів систем, варто, розглянути можливість реалізації послугу безпеки спостережуваності у вигляді логування основних дій користувача при роботі з базою даних і основних подій системи, таких як підключення до бази даних, вхід та вихід користувача і т. п.

Підрозділ 2.2. Математична постановка комплексу завдань системи (модуля) включає математичну модель досліджуваного об'єкта (позначення даних, математичні рівняння досліджуваного об'єкта,

встановлені обмеження на вирішення комплексу задач, а також чітке визначення відомих і невідомих даних). Математична модель – формалізований, тобто представлений математичними співвідношеннями, набір правил або функцій, що описують фактори суттєвого впливу на функціонування об'єкта дослідження в умовах існуючих (визначених) обмежень. Опис математичної моделі виконується термінами кількісних характеристик-показників (змінних, невідомих), значення яких підлягає визначенню в процесі розв'язання комплексу задач та параметрів, величини котрих апріорно відомі.

Для опису математичної постановки комплексу завдань в рамках дипломного проекту необхідно:

1. Визначити комплекс завдань (або предмет дослідження), для якого необхідно використати математичні моделі, методи (підходи, інструменти). Стисло визначити (описати) клас задачі (задач) (наприклад, задачі обліку, прогнозування, кластеризації, регресії, оптимізації, імітаційного моделювання, агрегації, описової статистики та ін.).

2. Визначити результат моделювання (рішення задачі). Описати, що необхідно отримати в якості результату при використанні математичного апарату (наприклад, групування даних, оптимальне значення цільової функції та її параметрів (змінних), результат функціоналу, координати об'єкту та інше).

3. Визначити критерії оцінки адекватності результату (наприклад, статистичні – рівень відносної помилки, рівень коефіцієнту детермінації) або інших критеріїв (наприклад, час отримання результату, точність розрахунку).

4. Визначити умови та фактори (змінні), які необхідно використати при розробці математичної моделі – опису функціоналу, набору правил, обмежень. Стисло описати фактори, дані та умови, які необхідно використати для розрахунків в моделі (задачі).

5. Обрати (визначити, описати) метод або математичний апарат, який необхідно використати для рішення задачі (наприклад, методи математичної статистики, штучного інтелекту, теорії графів, лінійного або нелінійного програмування та інше).

6. Навести опис математичної моделі в термінах, які будуть використані для побудови алгоритму або опису програмної реалізації моделі. При опису моделі необхідно пояснити кожен ідентифікатор (компонент), який використовується в моделі.

У разі необхідності виконання математичної постановки комплексу задач обробки інформації про територіально-розподілені об'єкти управління, або про явища та процеси будь-якого походження (штучні чи природні), моніторинг, опис, аналіз, моделювання стану яких і ухвалення управлінських рішень щодо поліпшення їх характеристик є неповним без просторового аналізу, доцільним є скористатися апаратом геомodelювання в середовищі пакета ArcGIS9. Прикладами таких об'єктів управління є:

- підприємства з територіально розподіленими ресурсами (фінансовими, матеріальними, трудовими, інформаційними, тощо) з необхідністю виконувати моніторинг та оптимізацію розміщення чи пересування ресурсів в просторі та часі;

- підприємства, що надають Web-сервіси для широкого загалу користувачів, з необхідністю координувати використання сервісів в просторі та часі;

- підприємства-оператори мобільного зв'язку з необхідністю оптимального розміщення трансляційних станцій на певній території;

- логістичні структури з необхідністю виконувати моніторинг матеріальних та фінансових потоків товарів чи вантажів в просторі та часі;

- маркетингові фірми з необхідністю обирати та управляти процесом сегментації ринку, розповсюдження та просування товарів чи послуг в просторі та часі;

- автотранспортні фірми з необхідністю оптимізувати управління рухом, забезпеченням паливом та іншими ресурсами;

- екологічні організації з необхідністю виконувати моніторинг та поліпшення стану оточення в просторі та часі;

- будівельні фірми з необхідністю виконувати пошук оптимальних місць для будівництва;

- туристичні фірми з необхідністю обирати та відслідковувати маршрути туристів;

- комунальні підприємства з необхідністю поліпшення управління всіма ресурсами побутового призначення, та багато інших.

Геомodelювання засобами інструменту Spatial Analyst пакета ArcGIS 9 передбачає побудову двох типів моделей: представлення та процесів.

Модель представлення або модель даних (описова модель). Складається з наборів даних, які визначають мету моделювання.

Розробляється у вигляді шарів (тем) просторових растрових або векторних даних. Кожен шар відображає просторові відношення між об'єктами ландшафту та передбачає наявність атрибутивних даних, які в реляційній таблиці надають фактичні дані про об'єкти.

Модель процесів є описом взаємовідносин між об'єктами, які були отримані в моделях даних, та дозволяє оцінити наслідок, якщо буде виконана якась дія. Цей вид моделювання виконується засобами інструментів просторового аналізу. Кожна операція та функція Spatial Analyst є моделлю процесу. Розрізняють такі головні типи моделей процесів:

- моделі придатності (пошук оптимального місцезнаходження чи оптимального пересування матеріальних, трудових, фінансових, інформаційних ресурсів);

- моделі відстаней (обчислення відстані між територіально віддаленими точками з врахуванням рельєфу місцевості, різницею у обліку часу, точками зупинки тощо);

- моделі гідрології (визначення, куди буде направлена гірська лавина чи водотік);

- моделі поверхонь (визначення: кадастрових особливостей, рівня забруднення різних регіонів тощо).

Повна модель складається з послідовності локальних задач, моделей процесів та наборів даних для виконання дій процесів у кожній локальній задачі, дає можливість скласти очікувану карту з локальних наборів та надає уявлення про стан предметної області для прийняття управлінських рішень.

Модель геомodelювання може бути подана у вигляді кортежу:

$$\langle F, f_i, R, C_c, Sh_i, P_m, S_s, W_j, [\text{<допоміжні дані>}] \rangle,$$

де F – мета (або постановка задачі) геомodelювання;

f_i – локальні задачі – складові загальної задачі моделювання з визначенням локальних критеріїв вирішення кожної задачі;

R – глобальний критерій оптимізації – головний (найбільш вагомий) з локальних критеріїв;

C_c – обмеження для прийняття рішення;

Sh_i – вхідні набори просторових даних (векторні теми, атрибутивні таблиці, растрові дані);

P_m – процеси (функції) моделювання;

S_s – ранги параметрів вихідних наборів даних;

W_j – вага вихідного набору даних j .

Параметри, наведені у квадратних дужках, вказуються в разі їх наявності та необхідності використання в обчисленнях або для забезпечення наочності карти.

Далі, в узагальненому вигляді, наведено приклад математичної постановки задачі геомodelювання.

Метою моделювання є пошук оптимального (чи субоптимального або раціонального) плану розміщення нового підрозділу підприємства (відкриття нової філії об'єднання) за критерієм мінімізації вартості будівництва та наявності певних обмежень.

$$\langle F, f_i, R, C_c, Sh_i, P_m, S_s, W_j, [G_g, Q_r] \rangle,$$

де F – оптимізація розміщення нового підрозділу підприємства;

f_i – розміщення підрозділу відносно: споруд існуючих підрозділів; автотранспортних шляхів; пунктів, де не менш ніж 80 % населення є вільною робочою силою; ділянок місцевості з відомою вартістю (критерії формуються, виходячи з того, що: нові споруди не повинні знаходитись далеко від – старих, не повинні бути далеко від автотранспортних шляхів, не повинні бути далеко від населених пунктів з вільною робочою силою, повинні бути розташованими на ділянках місцевості з вартістю, яка на перевищує бажану), $i = 1, \dots, l$;

R – мінімальна вартість будівництва;

C_c – підрозділ не повинен бути розташованим: на землях сільгоспугідь, на території заповідників та заказників;

Sh_i – вхідні набори просторових даних включають векторні теми з відповідними атрибутивними таблицями про: існуючі споруди, автотранспортні шляхи, населені пункти, землі з певною вартістю, землі сільгоспугідь, території заповідників та заказників;

P_m – відбір даних за певною ознакою, вимірювання відстаней між об'єктами, конвертування форматів даних, ранжирування об'єктів, перекласифікація наборів даних, розрахунок оверлейних сум з вагами, тощо;

S_s – ранги параметрів вихідних наборів даних приймаються за відповідною шкалою та за рішенням експертів;

W_j – значення ваги вихідного набору даних j приймається експертами.

Допоміжними даними можуть бути також такі:

G_g – геологічні, гідрологічні особливості місцевості, обраної для розміщення нового підрозділу, $g = 1, \dots, N$;

Q_r – обмеження на специфічні для задачі види ресурсів, $r = 1, \dots, W$.

Геомодельовання дозволяє аналітику прийняти правильне рішення про оптимальне місце розміщення або координати на карті нового підрозділу підприємства за критерієм мінімальної вартості будівництва з врахуванням вказаних обмежень.

У підрозділі 2.3. *Опис вихідних і вхідних документів* необхідно навести перелік вхідних і вихідних документів, промодельовати перехід вхідних документів у вихідні, представити це у вигляді діаграми "Діаграма зв'язку вхідних документів з вихідними для предметної області <...> на базі бізнес класів-сутності" та спроектувати форми вихідних та вхідних документів (додаток Л).

Метою цього розділу є моделювання документів. Для того, щоб це виконати, необхідно класифікувати всі документи, що обробляються в ІС, провести їх декомпозицію, описати атрибути документів, їх типи, значення, правила формування для проектування призначеного користувачеві інтерфейсу системи, проектування бази даних системи і ієрархії класів.

Під вхідними документи слід розуміти документи, що містять інформацію, необхідну для ініціалізації роботи відповідного бізнес-процесу. Будь-який бізнес-процес перед початком роботи повинен отримати необхідну для його правильної роботи інформацію і після чого приступити до виконання, притаманних йому, бізнес-функцій. В результаті виконання бізнес процесу формується вихідна інформація.

Під вихідними документами слід розуміти документи, що містять інформацію, яка отримана в результаті обробки вхідних документів відповідним бізнес-процесом.

Усі документи, що обробляють або формуються у бізнес-процесах відповідної предметної області треба представити у вигляді переліку документів (табл. 4.22). Спроектовані форми документів повинні бути наведені як рисунки в проекті.

Таблица 4.22

Інформаційний список документів

Код документа	Найменування	Вхідний чи вихідний	Вид документа (паперовий чи електронний)

Документами можуть виступати як паперові документи (акти, баланси, заяви, звіти і т. д.), так і електронні (файли, електронні форми, web-сторінки, xml-документи і т. д.). Кожний з документів необхідно проаналізувати і відповідно до цього виконати його декомпозицію. Документ агрегує в собі деяку сукупність даних, які можна розділити на ряд складових, відповідних таким клас-сутностям: заголовок документа, рядків полів даних, підпису документа та ін. Кожна з частин має бути описана відповідним набором полів, в яких міститься їх тип, розмірність. Це дозволить отримати декомпозицію документа для подальшого представлення його в якості сутностей бази даних і класів.

Бізнес клас-сутності (business entity classes) – пасивний клас, містить інформацію, яка повинна зберігатися постійно і не знищується зі знищенням об'єктів цього класу або припиненням роботи модельованої системи, що пов'язані з виключенням системи або завершенням програми. Клас-сутності відображають основні поняття (абстракції) предметної області. Атрибути клас-сутності становлять поля у базі даних і надають первинний опис даних.

Агрегація – це зв'язок між цілим і його часткою, який є тіснішою формою асоціації. Агрегації візуалізують у вигляді лінії з ромбом у класі, що є цілим. Композиція є сильнішим різновидом агрегації, при якому об'єкт-частка може належати тільки єдиному цілому і життєвий цикл часток співпадає з життєвим циклом цілого.

При декомпозиції документів слід враховувати між бізнес-сутностями потужність зв'язку агрегації або композиції, яка описує кількість бізнес клас-сутностей, що беруть участь у зв'язку. Потужність вказується на кожному кінці зв'язку. Потужність означає число зв'язків між однією бізнес-сутністю на початку лінії зв'язку з бізнес-сутностями у кінці лінії зв'язку.

Усі бізнес-сутності описати у вигляді таблиць (табл. 4.23).

Бізнес клас-сутність <НАЗВА>

№ зп	Назва поля	Тип поля	Розмір поля

Кожен документ складається із заголовка, змістовної частини і підпису, що є композитними складовими документу. Частини заголовка і підпис, як правило, містяться в документі один раз, тому повинні мати відношення 1:1. Тоді як змістова частина є набором деяких семантично однакових записів і визначається відношенням n:1. В пакети документи об'єднують, щоб показати, які дані необхідні для початку якого-небудь процесу. Дані, що входять до пакета, позначаються асоціацією агрегації. Бізнес-клас сутності в програмі повинні представлятися як клас.

Для ліпшого розуміння моделі документів представити всі бізнес-клас сутності зі зв'язками агрегації та композиції у вигляді діаграми. Приклад діаграми зв'язку вхідних документів з вихідними для предметної області <...> на базі бізнес класів-сутності наведена в додатку Л.

У підрозділі 2.4. *Проектування бази даних* необхідно розробити інформаційне забезпечення конкретних соціально-економічних систем, пов'язаних з темою дипломного проекту, виконати логічне і фізичне проектування бази даних, проявити творчий підхід до вибору технічного і програмного забезпечення інформаційної системи (ІС), СКБД, CASE-інструментів проектування БД.

Початковими даними для проектування бази даних є:

- опис функціональної схеми організації (підприємства), для якого виконується проект;
- виявлення місця і функцій конкретного підрозділу в структурі усієї організації (підприємства);
- опис технології обробки інформації у рамках даної предметної області (ПрО) конкретного підрозділу, з зазначенням вхідної і вихідної інформації, вирішуваних завдань, функцій і регламенту роботи виконавців обробки інформації, періодичності виконання функцій з обробки інформації;
- аналіз існуючих засобів автоматизації обробки інформації у рамках даної предметної області (апаратне, програмне забезпечення, СКБД);

- обґрунтування необхідності і можливості розробки модуля автоматизації обробки інформації і бази даних цього модуля, як складового елементу загальної інфраструктури даних організації;

- аналіз і опис завдань, що автоматизуються в модулі (підсистемі), що розробляється.

Слід пам'ятати, що сучасна БД складається з двох частин: транзакційної і аналітичної. Характерними рисами транзакційної бази є:

- 1) реляційна структура (переважно);
- 2) можливість накопичення значних об'ємів фактичних даних;
- 3) можливість виконання операцій додавання, видалення, редагування записів;
- 4) відсутні агреговані (обчислені) дані;
- 5) використовується для виконання різноманітних операцій обліку;
- 6) є основою для розробки аналітичної частини БД.

Аналітична частина БД використовується при виконанні оперативного аналізу інформації і розробці моделей для систем підтримки прийняття рішень. Характерними рисами аналітичної БД є:

- 1) багатовимірна структура (підтримка моделей : MOLAP, ROLAP, HOLAP);
- 2) зберігання агрегованих даних;
- 3) відсутність можливості виконання операцій видалення і редагування даних і накопичення їх, як правило, за хронологією.

При розробці транзакційної частини БД можливо використати структурне або об'єктно-орієнтоване моделювання, застосовуючи відповідні CASE-інструменти: ERwin, Rational Rose, ARIS тощо. У разі структурного моделювання необхідно імпортувати словник сутностей і атрибутів, отриманий в BPwin, у середовище ERwin.

При побудові сховища даних необхідно обґрунтувати вибір моделі зберігання (кубічна модель MOLAP або ROLAP: "зірка" або "сніжинка").

У пункті 2.4.1. Концептуальне інфологічне проектування виконується побудова СУБД-незалежної моделі даних.

Підпункт 2.4.1.1. Словник даних. На основі аналізу вхідних та вихідних документів (пункт 2.3) будується модель відображення множини реквізитів вихідних і вхідних документів на множину елементів даних, що підлягають збереженню у базі даних, потім виконується приведення зібраної інформації до зручного для проектування вигляду.

Для цього складають словник даних у вигляді табл. 4.24.

Словник даних

№ з/п	Найменування елемента	Ідентифікатор	Тип і довжина	Призначення елемента

У полі "Призначення елемента" необхідно вказати: елемент збереження є фактичним, чи обчислюваним.

Елементи даних словника наводяться в алфавітному порядку за графою "Найменування елемента даних" з метою подальшого виключення омонімів та дублюючих елементів.

Якщо словник вміщує багато елементів, його виносять у додаток.

Підпункт 2.4.1.2. Проектування інфологічної моделі даних.

Виконується побудова інфологічної моделі даних.

Будується графічне представлення моделі даних у вигляді ERD (нотація IDEF1X) або діаграми класів у середовищі пакетів Rational Rose, ARIS тощо.

Для усіх сутностей моделі слід навести специфікації обмежень, які повинні включати:

- 1) обмеження атрибутів сутностей (табл. 4.25);
- 2) обмеження кортежів (табл. 4.26);
- 3) обмеження унікальності (табл. 4.27);
- 4) динамічні обмеження (табл. 4.28);
- 5) інші обмеження (табл. 4.29);
- 6) операційні правила (табл. 4.30);
- 7) правила посилювальної цілісності (табл. 4.31).

Причому кожна з цих таблиць повинна включати специфікації для усіх сутностей, якщо певний вид обмежень дійсно має місце.

Обмеження атрибутів (приклад)

№ з/п	Ім'я атрибута або агрегату	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням
1	Співробітник. Табельний номер	Від 100 до 1000	9999	NOT NULL	–
2	Співробітник. Телефон	–	(99) 999-99-99		–
3	Виконання робіт. Дата	–	–	–	Поточна дата

Таблиця 4.26

Обмеження кортежів (приклад)

№ з/п	Група атрибутів	Обмеження
1	Дата зарахування, дата звільнення	Дата звільнення > дата зарахування

Таблиця 4.27

Обмеження унікальності (приклад)

№ з/п	Атрибут або група атрибутів	Серед яких примірників, якої сутності або зв'язку має місце унікальність
1	Співробітник. Табельний номер	Для всіх примірників сутності "Співробітник"
2	Поставка товару. Код постачальника + Поставка товару. Код товару + Поставка товару. Дата поставки + Поставка товару. Час поставки	Для всіх примірників сутності "Поставка товару"

Таблиця 4.28

Динамічні обмеження (приклад)

№ з/п	Група атрибутів	Обмеження
1	Студент. Курс	Курс $\leftarrow$ Курс + 1 – значення атрибута курс може лише збільшуватися на одиницю

Таблиця 4.29

Інші обмеження (приклад)

№ з/п	Група атрибутів	Обмеження
1	Кар'єра. Дата звільнення	Дата звільнення, певного співробітника, може бути незаповнена тільки для останнього місця його роботи (поточної)
2	Кар'єра. № п/п, Дата зарахування, дата звільнення	Хронологічна послідовність значень "Дата зарахування", "Дата звільнення" у зв'язках які відповідають одному співробітнику, упорядковані за № з/п

Операційні правила (приклад)

№ з/п	Група атрибутів	Обмеження
1	Атрибути, пов'язані із сутністю "Співробітник": табельний номер, ПІБ і т. д.	При видаленні запису про якого-небудь співробітника всі відомості про нього слід перенести в архівну базу з вказівкою дати та часу, причини видалення та імені користувача, який здійснив видалення. Ці відомості зберігаються в архівній базі не менше 1 року, а потім можуть бути автоматично видалені

Правила оновлення записів посилальної цілісності (приклад)

№ з/п	Батьківська сутність	Дочірня сутність	Правило	Інші правила
1	Підрозділ	Співробітник	Обмежене (restrict)	
2	Співробітник	Кар'єра	Каскадне (cascade)	

Пункт 2.4.2. Проектування даталогічної моделі даних. Виконується опис розробки даталогічної або СУБД-орієнтованої моделі даних.

Підпункт 2.4.2.1. Проектування даталогічної моделі даних в нотації I DEF1X. Розділ включає: обґрунтування вибору СУБД, графічне представлення глобальної даталогічної моделі у вигляді ERD (в нотації IDEF1X) або діаграми класів.

Підпункт 2.4.2.2. Обґрунтування властивостей моделі бази даних, що спроектована. Розділ включає: опис засобів, інструментів, методів, які застосовувалися для забезпечення таких властивостей бази даних: функціональна повнота; мінімальна надмірність; цілісність бази даних (домену, таблична, посилальна цілісність, забезпечувана ключами і тригерами); узгодженість; актуальність; безпека; відновлюваність; логічна та фізична незалежність; ефективність.

Пункт 2.4.3. Проектування фізичної моделі даних. Розділ включає опис розробки моделі даних на фізичному рівні як результат генерації команд SQL-скрипту і створення основних об'єктів зберігання в БД таких як : таблиці, подання, ключі, тощо.

Пункт 2.4.4. Програмна реалізація бази даних. Розділ містить опис розроблених збережених процедур, функцій і тригерів БД, якщо вони були розроблені. Збережені процедури, функції та тригери слід описати за допомогою табл. 4.32 і табл. 4.33 з приведенням вихідного коду в додатку до пояснювальної записки. Для кожної збереженої функції в колонці за номером три вказати тип даних, що повертаються.

Таблиця 4.32

Збережені процедури і функції

№ з/п	Найменування процедури/функції	Опис формальних параметрів	Призначення процедури/функції	Посилання на код процедури/функції у додатку

Таблиця 4.33

Тригери

№ з/п	Найменування тригера	Тип тригера (таблиця, подання, схема БД, СУБД)	Подія	Момент спрацьовування (до або після настання події)	Рівень спрацювання (строковий, операторний, системний)	Призначення тригера	Посилання на вихідний код тригера у додатку

У підрозділі 2.5. Розроблення архітектури програмної системи необхідно:

Розробити структурну схему програмної системи та навести її опис (додаток М).

Розробити UML-діаграму класів, що реалізують основну бізнес-

логіку програмної системи, та навести її опис.

Розробити UML-діаграму станів, у яких можуть знаходитися елементи графічного інтерфейсу користувача, та навести її опис.

Навести посилання на додаток, у якому міститься лістинг програми. Лістинг програми повинен мати коментарі (для кожного класу – призначення класу; для кожного методу (функції) – призначення методу (функції), опис параметрів та значення, що повертається). Якщо програма реалізована з використанням програмної платформи .NET або Java, обов'язковим є використання тегів XML- або JavaDoc для створення документації.

Пункт 2.5.1. Розроблення архітектури програмної системи.

Програмна система означає програмне забезпечення, що розроблюється. Це може бути крупна колекція з безлічі компонентів програмного забезпечення, один додаток або частина додатку.

Найважливішими характеристиками архітектури будь-якої програмної системи є її структура й процес функціонування.

Під структурою програмної системи розуміють стійку в часі сукупність взаємозв'язків між її елементами. Залежно від рівня деталізації структурні елементи можуть бути підсистемами, модулями, процесами, бібліотеками і т. д. Саме структура зв'язує воєдино всі елементи програмної системи й забезпечує її існування як єдиного цілого.

Структурна схема програмної системи – це її графічне зображення у вигляді сукупності складових елементів та інформаційних зв'язків між ними з зазначенням їх напрямку. Приклад структурної схеми програмної системи наведений у додатку К. Опис структурної схеми має містити відомості про призначення елементів програмної системи, їх інформаційні зв'язки та взаємодію.

Процес функціонування програмної системи тісно пов'язаний зі зміною її властивостей або поведження в часі. Тому поряд з визначенням структурних елементів будь-яка архітектура визначає взаємодію між ними, що забезпечує бажане поведження системи.

Одним із принципів побудови моделей складних систем є принцип багатомодельності. Відповідно до цього принципу ніяка єдина модель не може з достатнім ступенем адекватності описувати різні аспекти складної системи.

Стосовно об'єктно-орієнтованого аналізу й проектування

програмних систем це означає, що досить повна модель такої системи повинна містити деяке число взаємозалежних уявлень, кожне з яких адекватно відображає деякий аспект її поведінки або структури.

Найбільш загальними уявленнями складної системи прийнято вважати статичне й динамічне уявлення.

Підпункт 2.5.2. Розроблення діаграми класів, які реалізують основну бізнес-логіку програмної системи

Для уявлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування призначена UML-діаграма класів. Вона може відображати різні взаємозв'язки між окремими сутностями предметної області, такими як об'єкти й підсистеми, а також описує їхню внутрішню структуру й типи відносин.

Клас у мові UML служить для позначення безлічі об'єктів, які мають однакову структуру, поведінку і відносини з об'єктами інших класів. Клас може не мати екземплярів або об'єктів. У цьому випадку він називається абстрактним класом. Графічно клас зображується у вигляді прямокутника, що додатково може бути розділений горизонтальними лініями на секції. У цих секціях можуть вказуватися ім'я класу, атрибути й операції.

Обов'язковим елементом позначення класу є його ім'я. Ім'я класу повинне бути унікальним у межах пакета, що описується деякою сукупністю діаграм класів. Воно вказується в першій верхній секції прямокутника. Рекомендується як імена класів використовувати іменники, записані без пробілів. Необхідно пам'ятати, що саме імена класів утворюють словник предметної області. Прикладами імен класів можуть бути такі іменники, як "Співробітник", "Компанія", "Керівник", "Клієнт", "Продавець", "Менеджер", "Офіс" та ті, що мають безпосереднє відношення до предметної області й функціонального призначення проектованої системи.

У другій зверху секції прямокутника класу записуються його атрибути. Кожному атрибуту класу відповідає окремий рядок тексту, що складається із квантора видимості атрибута, імені атрибута, його кратності, типу значень атрибута й, можливо, його вихідного значення.

Квантор видимості може приймати одне із трьох можливих значень:

1. Загальнодоступний (public). Атрибут із цією областю видимості доступний з будь-якого іншого класу пакета, у якому визначена діаграма.
2. Захищений (protected). Атрибут із цією областю видимості

недоступний для всіх класів, за винятком підкласів даного класу.

3. Закритий (private). Атрибут із цією областю видимості недоступний для всіх інших класів без винятку.

Ім'я атрибута становить собою рядок тексту, що використовується як ідентифікатор відповідного атрибута й тому повинен бути унікальним в межах даного класу. Ім'я атрибута є єдиним обов'язковим елементом синтаксичного позначення атрибута.

Кратність атрибута характеризується загальною кількістю конкретних атрибутів даного типу, що входять до складу окремого класу.

Тип атрибута становить вираз, семантика якого визначається мовою специфікації відповідної моделі. У нотації UML тип атрибута іноді визначається залежно від мови програмування, яку передбачається використовувати для реалізації даної моделі. У найпростішому випадку тип атрибута вказується рядком тексту, що має осмислене значення в межах пакета або моделі, до яких ставиться розглянутий клас.

У третій зверху секції прямокутника записуються операції класу. Операція становить деякий сервіс, що надає будь-який екземпляр класу на певну вимогу. Сукупність операцій характеризує функціональний аспект поведінки класу.

Кожній операції класу відповідає окремий рядок, що складається із квантора видимості операції, імені операції, вираження типу, що повертається операцією.

Ім'я операції становить рядок тексту, що використовується як ідентифікатор відповідної операції й тому повинен бути унікальним у межах даного класу.

Крім внутрішнього устрою або структури класів на відповідній діаграмі вказуються різні відносини між класами. При цьому сукупність типів таких відносин фіксована в мові UML і визначена семантикою цих типів відносин. Базовими відносинами або зв'язками в мові UML є:

1. Відношення залежності.
2. Відношення асоціації.
3. Відношення узагальнення.
4. Відношення реалізації.

Відношення залежності в загальному випадку вказує на деяке семантичне відношення між двома елементами моделі або двома множинами таких елементів, що не є відношенням асоціації, узагальнення або реалізації. Відношення залежності використовується в такій ситуації, коли деяка зміна одного елемента моделі може

потребувати зміни іншого залежного від нього елемента моделі.

Відношення асоціації відповідає наявності деякого відношення між класами. Найбільш простий випадок даного відношення – бінарна асоціація. Вона зв'язує в точності два класи й, як виключення, може зв'язувати клас із самим собою.

Тернарна асоціація й асоціації більш високого порядку в загальному випадку називаються N-арною асоціацією. Така асоціація зв'язує деяким відношенням три і більше класів, при цьому один клас може брати участь в асоціації більш ніж один раз.

Окремим випадком відношення асоціації є відношення агрегації, яке, в свою чергу, теж має спеціальну форму – відношення композиції.

Відношення агрегації має місце між декількома класами в тому випадку, якщо один із класів становить деяку сутність, що включає в себе як складові частини інші сутності. Дане відношення має фундаментальне значення для опису структури складних систем, оскільки застосовується для уявлення системних взаємозв'язків типу "частина-ціле". Розкриваючи внутрішню структуру системи, відношення агрегації показує, з яких компонентів складається система і як вони зв'язані між собою. Це відношення по своїй суті описує декомпозицію складної системи на більше прості складові частини, які також можуть бути піддані декомпозиції, якщо в цьому виникне необхідність надалі.

Відношення композиції є окремим випадком відношення агрегації. Це відношення слугує для виділення спеціальної форми відносини "частина-ціле", при якій складові частини в деякому змісті перебувають усередині цілого. Специфіка взаємозв'язку між ними полягає в тому, що частини не можуть виступати у відриві від цілого, тобто зі знищенням цілого знищуються й всі його складові частини.

Відношення узагальнення є відношенням між більш загальним елементом (батьком або предком) і більш приватним або спеціальним елементом (дочірнім або нащадком). Стосовно до діаграми класів дане відношення описує ієрархічну будову класів і спадкування їхніх властивостей і поведження. При цьому передбачається, що клас-нащадок має всі властивості й поведження класа-предка, а також має свої власні властивості й поведження, які відсутні у класа-предка.

Підпункт 2.5.3. Розроблення діаграми станів елементів графічного інтерфейсу користувача

Інтерфейс користувача складається з елементів, які можуть

перебувати в певних станах, а також у процесі взаємодії з користувачем програми переходити зі стану в стан. Він є системою, що управляється подіями (СУП).

Поводження таких систем найкраще характеризується їхньою реакцією на зовнішні події. Як правило, СУП перебуває в стані очікування, поки не одержить повідомлення про подію. Після того як СУП відреагує на подію, вона знову переходить у стан очікування наступної події. Для таких систем важливо визначити насамперед стійкі стани, події, що ініціюють переходи з одного стану в інший, і дії, що виконуються при зміні стану.

При формальному описі СУП доцільно використовувати UML-діаграми станів.

Діаграма станів зв'язує події й стани. При виникненні події наступний стан системи залежить як від її поточного стану, так і від події. Зміна стану називається переходом. Діаграма станів – це граф, вузли якого моделюють стани, а спрямовані дуги, позначені іменами відповідних подій, – переходи.

У даному пункті пояснювальної записки необхідно навести UML-діаграми станів, у яких можуть перебувати елементи інтерфейсу користувача при виконанні кожного з бізнес-варіантів використання, а також їхній опис.

При розробленні інтерфейсу користувача необхідно керуватися такими методичними рекомендаціями.

Методичні рекомендації щодо розроблення інтерфейсу користувача

Користувальницький інтерфейс є своєрідним комунікаційним каналом, по якому здійснюється взаємодія користувача й комп'ютера.

Щоб створити ефективний інтерфейс, що здійснював би роботу із програмою приємною, потрібно розуміти, які завдання будуть вирішувати користувачі за допомогою даної програми і які вимоги до інтерфейсу можуть виникнути в користувачів.

Загальні принципи проектування інтерфейсу користувача:

1. Програма повинна допомагати виконувати завдання.

Це значить, що інтерфейс повинен бути легким для освоєння й не створювати перед користувачем перешкоду, яку він повинен буде подолати, щоб приступити до роботи.

2. При роботі із програмою користувач не повинен відчувати дискомфорт.

Для реалізації цього принципу необхідно:

забезпечити перевірку результатів як можна більшого числа "некоректних" дій користувача, але не робити її повсюдно;

вказувати користувачеві, що саме йому робити, і виводити інформаційні повідомлення в ситуаціях, коли це дійсно необхідно;

надати досвідченим користувачам можливість відключення виведення інформаційних повідомлень;

добре продумувати зміст повідомлень, що виводяться користувачеві.

Широко відомі евристичні правила авторитетного американського фахівця в галузі проектування інтерфейсів Якоба Нільсена. Вони визначають мінімальні критерії, яким повинен відповідати інтерфейс будь-якої програми:

1. Наочність стану системи (правило зворотного зв'язку).

Система (у цьому випадку – комп'ютерна програма) повинна завжди інформувати користувача про стан своєї роботи за допомогою засобів зворотного зв'язку в прийнятний час.

При розгляді цього правила потрібно враховувати кілька аспектів:

користувач завжди повинен мати інформацію про поточний стан програми (наприклад, скільки часу пройшло від початку процесу копіювання файлів);

користувач обов'язково повинен бачити, до чого призвела будь-яка його дія, наприклад, уведення даних, натискання кнопки;

вибір конкретного засобу зворотного зв'язку залежить від типу інформації, яку потрібно донести до користувача, а також типу дії, що викликає потребу в зворотному зв'язку.

Уважається, що якщо користувач виконав якусь дію й очікує результат його виконання, то цей результат (або повідомлення про помилку) повинен бути виведений в окремому діалоговому вікні. Якщо ж користувачеві необхідно надати поточну інформацію про процес, що не є прямим наслідком його дій, то можна обмежитися відображенням відповідного повідомлення в панелі стану.

Для організації зворотного зв'язку можуть бути використані й інші засоби. Найпопулярніші з них – звуки. Найбільш часто звукове повідомлення допомагає тоді, коли поява на екрані діалогових вікон є небажаною, а повідомлення в панелі стану можуть бути не помічені

користувачем.

Важливо пам'ятати, що звукове повідомлення не повинне бути основним засобом організації зворотного зв'язку. Звук повинен лише доповнювати текстові повідомлення.

Проміжок часу, протягом якого користувач одержує інформацію про реакцію на його дію або про подію, повинен бути мінімальним. Це особливо важливо, тому що наявність або відсутність у користувача інформації про поточний стан системи визначає його подальші дії.

2. Відповідність між системою й реальним світом.

Система повинна взаємодіяти з користувачем на його мові. У цьому випадку мається на увазі використання понять, образів і цілих концепцій, які знайомі користувачеві з реальної предметної області. У жодному разі не можна використовувати спеціалізовані терміни, які придатні тільки для професійних довідників з програмування.

Найпоширеніший приклад реалізації цього принципу – побудова користувацького інтерфейсу, що імітує об'єкти реального світу. Наприклад, більшість із програм, що реалізують функції годинників, калькуляторів, програвачів компакт-дисків, записних книжок виглядають майже точно так, як їхні матеріальні аналоги. Знаменитий "сміттєвий кошик" на "робочому столі" операційної системи Windows, у яку можна "кинути" непотрібний файл або папку, – класичний приклад побудови інтерфейсу на основі об'єктів реального світу.

3. Управління користувачами та свобода їхніх дій.

Користувачі повинні мати можливість управління системою й зміни її поточного стану. Однак, вони часто дають різні команди помилково (наприклад, випадково натиснувши кнопку або вибравши не той пункт меню). Отже, у користувача повинен бути "аварійний вихід" із цієї ситуації, чітко позначений у програмі. Найчастіше такий "вихід" реалізується у вигляді кнопки "Відміна", розташованої в діалоговому вікні, що дозволяє припинити виконання поточної операції або закрити це діалогове вікно. Крім цього, традиційним і тому звичним для більшості користувачів засобом "аварійного виходу" є натискання на клавіатурі клавіші <Escape>.

Найоптимальнішим вважається сполучення цих способів "аварійного виходу".

Ще один спосіб виходу з помилкової ситуації – команди "Undo" ("Відмінити") і "Redo" ("Повторити"). Вони є настільки зручними й

підтримуються такою великою кількістю програм, що користувачі підсвідомо очікують, що будь-яку їхню дію можливо відмінити, повернувшись до попереднього стану.

Все це зобов'язує розроблювача дружнього користувальницького інтерфейсу комп'ютерної програми підтримувати дані команди. Якщо ж за якимись причинами дію, на виконання якої дав команду користувач, не можна відмінити, то на екран повинно бути виведеним відповідне попередження, а також прохання підтвердити виконання команди.

4. Несуперечність і стандарти.

Цей принцип означає використання тих самих засобів для вираження схожих образів і виконання дій, що мають однакову природу.

По-перше, це означає несуперечність при виборі засобів оповіщення про події та дії. Наприклад, інформація про поточний стан програми, звичайно виводиться в панелі стану, а повідомлення з результатами запитів користувача – в окремих діалогових вікнах.

Повідомлення про критичні помилки при цьому повинні сильно відрізнятися від звичайних інформаційних повідомлень: наприклад, вони можуть супроводжуватися різким звуком.

По-друге, це означає несуперечність при оформленні елементів користувальницького інтерфейсу. Наприклад, якщо він ґрунтується на класичному інтерфейсі Windows-додатків, що характеризується строгою колірною гамою, прямими лініями й кутами, то дуже дивним виглядало б рішення додати до одного з вікон програми овальну форму й розфарбувати його яскравими кольорами.

По-третє, це означає несуперечність при виборі термінів. Користувачів не повинно спантеличувати те, що кілька різних понять, які використовуються в програмі, насправді означають те саме.

Головне – вирішити, що для позначення якоїсь конкретної дії або події буде застосовуватися один конкретний термін, який буде використовуватися певним способом (наприклад, слово "Інтернет" буде починатися із великої букви й не відмінюватися).

Принцип несуперечності – одне з найважливіших правил проектування користувальницьких інтерфейсів. Несуперечливий інтерфейс інтуїтивно зрозумілий і дуже легкий для освоєння, тому що при його вивченні користувач не зіштовхується із сюрпризами, і навіть ті частини інтерфейсу, які він бачить уперше, здаються йому давно знайомими.

5. Запобігання помилок.

Стосовно теми проектування користувацького інтерфейсу комп'ютерних програм, цей принцип означає таке: "Дизайн, що запобігає виникненню помилок, краще, ніж найкраще повідомлення про помилку".

6. Розуміння краще, ніж запам'ятовування.

При розробленні інтерфейсу потрібно робити всі об'єкти, функції, дії легкодоступними користувачеві. Користувач не повинен постійно намагатися утримати в пам'яті інформацію з однієї частини програми, щоб застосувати її в іншій. У будь-який момент часу користувачеві повинно бути зрозуміло, що йому зараз потрібно робити. У хорошому інтерфейсі інструкції з використання системи доступні для виклику в будь-який час, коли це потрібно. Це може бути реалізоване як у вигляді продуманої організації елементів інтерфейсу, так і у вигляді підказок користувачеві.

Це правило відбиває принцип "прозорого" інтерфейсу – інтерфейсу, що зрозумілий і не змушує користувача згадувати, яку кнопку потрібно натиснути або який пункт меню вибрати в даний момент.

7. Гнучкість і ефективність використання.

При проектуванні інтерфейсу користувача перед розроблювачем часто постає така проблема: потрібно, щоб інтерфейс був однаково зручний і для новачків, і для досвідчених користувачів.

Для рішення цієї проблеми використовують простий прийом: функції, які прискорюють роботу, оформлюють так, щоб їх не бачив початківець, але щоб вони були доступні досвідченим користувачам. Найпростіший приклад – це "гарячі клавіші", за допомогою яких можна швидко викликати функції програми, що часто виконуються. Позначення "гарячих клавіш" пишуться поруч із відповідними пунктами меню, тому вони, з одного боку, не заважають новачкам, а з іншого боку – доступні досвідченим користувачам.

Інший приклад реалізації універсального користувацького інтерфейсу – можливість виконати складні функції програми як за допомогою "майстра", що, немов за руку, "проведе" починаючого користувача по всіх етапах процесу, так і вручну, за допомогою настроювання опцій у відповідному діалоговому вікні.

8. Естетичний і мінімалістський дизайн.

Це правило означає: "Нічого зайвого". Не потрібно засмічувати користувацький інтерфейс програми елементами, які є недоречними

й малокорисними. Справа в тому, що кожний елемент, чи то кнопка або текстовий підпис, обов'язково відволікає частину уваги користувача. Це може призвести до того, що видимість і, відповідно, легкість сприйняття користувачем дійсно потрібних і корисних частин інтерфейсу буде сильно зменшена за рахунок елементів, без яких цілком можна було б обійтися.

9. Розпізнавання й виправлення помилок.

"Допомагайте користувачеві розпізнавати й виправляти помилки" – говорить це правило.

Воно визначає проектування повідомлень про помилки. "Гарні" повідомлення про помилки – це повідомлення, які пояснюють, у чому полягає проблема й, найголовніше, як її виправити. Таким чином, "гарне" повідомлення про помилку повинне складатися із двох частин: опису помилки й опису вирішення проблеми.

Опис помилки повинен бути чітким, ясным і зрозумілим, давати користувачеві всю необхідну інформацію про причини й місце виникнення помилки. Найпростіше рішення – створити в довідковій системі програми відповідний розділ, що пояснює зміст проблеми й причини її виникнення. У самому ж діалоговому вікні з повідомленням про помилку може бути присутнім кнопка "Довідка" для виклику цього розділу.

Ще одним прикладом рішення даної проблеми є кнопка "Докладніше", при натисканні на яку діалогове вікно з повідомленням про помилку "розорюється", відображаючи більш докладну інформацію про причину виникнення збою.

Дуже важливо пам'ятати те, що повідомлення про помилку повинне містити її опис, а не числовий код помилки.

Існує багато програм, у яких повідомлення про помилки містять недостовірну інформацію, повідомляючи користувача зовсім не про ті проблеми, які виникли насправді. Тому при складанні описів помилок потрібно не забувати перевіряти коректність повідомлень, що генеруються програмою.

Відомо, що інформація про те, як виправити помилку або вирішити проблему має навіть більше значення, ніж опис помилки або проблеми. При описі шляху вирішення проблеми потрібно уникати складання занадто об'ємних текстів. В іншому разі користувачі будуть просто пробігати їх очима, не доходячи до змісту написаного. Найкраще скласти покрокову інструкцію, кожний крок якої складається з 1 – 2 речень.

10. Довідка й документація.

Це правило полягає в необхідності надання користувачеві довідкової системи й документації до програми.

Проектування форм

Форми – це "будівельні блоки" інтерфейсу користувача.

Щоб створити добре спроектовану форму, необхідно усвідомити її призначення, спосіб і час використання, а також її зв'язки з іншими елементами програми.

Особливий вид форм – форми, призначені для введення даних. При їхній розробці основну увагу варто приділити швидкості їхнього використання. Основне правило – якщо користувач збирається ввести в базу даних велику кількість записів, то він не повинен підтверджувати введення кожної з них.

Щоб прискорити процес уведення даних, необхідно:

1. По можливості використовувати для додавання й редагування даних ту саму форму.
2. Призначати клавіатурні сполучення для команд.
3. Не змушувати користувача "перестрибувати" з однієї частини форми в іншу.
4. Не ставити процес уведення даних у залежність від вмісту окремих елементів управління форми.
5. Використовувати засоби зворотного зв'язку з користувачем.

Робота з декількома формами

Якщо додаток повинен містити кілька форм, то можна використовувати однодокументний (SDI) або багатодокументний (MDI) інтерфейс користувача.

Однак, у кожному разі взаємодія користувача з формами відбувається за допомогою обробки подій, що надходять від елементів управління форми. Тому, якщо в додатку передбачено кілька форм, то програму необхідно спроектувати так, щоб у користувачів не було можливості порушити послідовність її виконання (наприклад, вивести форму, для якої ще не готова інформація).

Ефективні меню

Ще одна важлива частина розробки форм – створення змістовних і ефективних меню. Деякі важливі рекомендації:

1. Дотримуйтеся стандартних угод про розташування пунктів меню, прийнятим в операційній системі.
2. Групуйте пункти меню в логічному порядку й за змістом.
3. Для групування пунктів у меню, що розкриваються, використовуйте розділові лінії.
4. Уникайте надлишкових меню.
5. Уникайте пунктів меню верхнього рівня, які не мають меню, що розкриваються.
6. Не забувайте використовувати символ <...> для позначення пунктів меню, що активізують діалогові вікна.
7. Обов'язково використовуйте клавіатурні еквіваленти команд і "гарячі" клавіші.
8. Розміщуйте на панель інструментів часто використовувані команди меню.

Рекомендації із проектування Web-інтерфейсу користувача:

1. Мінімізуйте зусилля, які необхідно зробити користувачеві для прийняття рішень про навігацію.
2. По можливості використовуйте один екран. Використовуйте багатопанельні елементи управління, спливаючі вікна й майстри, щоб користувачі могли виконувати якомога більшу частину завдань, не використовуючи побічну навігацію.
3. При створенні декількох сторінок поєднуйте їх у розділи. Спростіть переміщення між розділами за допомогою основного елемента управління навігацією та додаткового елемента управління для переміщення між розділами.
4. Переконайтеся, що ключові елементи, такі як меню, заголовки й інша інформація, що відображається на всіх сторінках, оформлені одноманітно з візуальної точки зору.
5. Варто бути дуже обережним відносно навігаційних елементів управління для переміщення користувачів між внутрішніми сторінками різних розділів. Це можливо, але це може дезорієнтувати користувача.
6. Використовуйте додаткові засоби навігації, такі як дерева й карти вузлів, але не покладайтеся на них.

7. Завжди думайте про майбутнє розширення. Якщо в майбутньому очікується включення в продукт нових функцій, спочатку необхідно вирішити, як буде розширюватися навігація для ефективного включення нових екранів.

Пункт 2.5.4. Розроблення інтерактивної довідкової системи.

Необхідно навести короткий опис призначення кожної екранної форми додатку та усіх її інтерактивних елементів.

Пункт 2.5.5. Тестування програмного забезпечення.

Тестування програмного забезпечення – це процес перевірки готової програми у статичному режимі (перегляд, інспекція, налагодження вихідного коду) і динамічному шляхом прогону кінцевого набору тестових даних, що перевіряють різні шляхи виконання програми й порівнянні отриманих результатів із заздалегідь запланованими. Існує дві форми перевірки коду – модульна й інтеграційна. При цьому використовуються стандарти (IEEE 829-1996 й IEEE 1008-1987) перевірки й тестування модулів програмного забезпечення. У процесі різних видів перевірок збираються дані про помилки, дефекти, відмови і т.п. й оформлюється відповідна документація (таблиці типів помилок, частота й час появи відмов). Дані, зібрані при тестуванні, використовуються для оцінки характеристик якості готового програмного забезпечення, наприклад, показника надійності – інтенсивність відмов, кількість помилок та ін.

При тестуванні виявляються недоліки: відмови і дефекти як причини порушення роботи програми, збої як небажані ситуації, помилки, як наслідок збоїв та ін. Базовим поняттям тестування є тест, що виконується в заданих умовах і на заданих наборах даних. Тестування вважається успішним, якщо знайдено дефект або помилка, і вони відразу усуваються. Якість тестування визначається критерієм покриття системи тестами, перевірки всіх можливих шляхів виконання алгоритмів програм.

У межах виконання дипломного проекту необхідно обрати рівні тестування:

тестування окремих елементів, що полягає в перевірці окремих, ізольованих і незалежних частин ПЗ;

інтеграційне тестування, що орієнтоване на перевірку зв'язків і способів взаємодії (інтерфейсів) компонентів один з одним, включаючи компоненти, розташовані на різних архітектурних платформах

розподіленого середовища;

тестування системи призначене для перевірки правильності функціонування системи в цілому, з виявленням відмов і дефектів у системі і їхнє усунення. При цьому контролюється виконання сформульованих нефункціональних вимог (безпека, надійність й ін.) у системі, правильність завдання й виконання зовнішніх інтерфейсів системи із середовищем оточення та ін.

Після вибору рівня тестування необхідно обрати методи тестування:

функціональне тестування, що забезпечує перевірку реалізації функцій, які визначені у вимогах, а також правильність їхнього виконання;

регресійне тестування, орієнтоване на повторне вибіркоче тестування системи або її компонентів після внесення в них змін на тих же тестах, що й до модифікації;

тестування ефективності, що перевіряє продуктивність, пропускну здатність, максимальний обсяг даних і системні обмеження у відповідності зі специфікаціями вимог;

стрес-тестування, що перевіряють поведінку системи при максимально припустимому навантаженні або при її перевищенні;

альфа й бета тестування, що виконують внутрішнє тестування кодів системи й зовнішнє тестування інтерфейсів. Альфа – це внутрішнє тестування (функцій й алгоритмів), а бета – зовнішнє (взаємозв'язків з іншими системами й середовищем);

конфігураційне тестування, що перевіряє структури й ідентифікації системи на різних наборах даних, а також роботу системи на різних конфігураціях апаратур й обладнань.

Обов'язковим у дипломному проекті є функціональне тестування. Метою функціонального тестування є виявлення невідповідностей між існуючою поведінкою реалізованих функцій і їх поведінкою відповідно до специфікацій та вихідним вимогами. Функціональні тести повинні охоплювати всі **основні** реалізовані функції (**не менше п'яти**) з урахуванням найбільш ймовірних типів помилок.

Для функціонального тестування у дипломному проекті необхідно:

визначити найбільш важливі функціональні вимоги;

визначити послідовність функцій відповідно до їх використання у програмному забезпеченні;

визначити критерії оцінки якості отриманих результатів функції або процесу її здійснення;

визначити множини вхідних даних кожної функції і областей їх зміни;

побудувати тестові набори та сценарії тестування функцій;

отримати результати тестування всіх функціональних вимог.

Послідовність процесу тестування:

планування процесу тестування (складання плану тестів, визначення тестів, визначення контрольного набору даних для тестів) і оцінювання показників якості програмного забезпечення;

проведення тестування компонентів програмного забезпечення;

генерація необхідних тестових сценаріїв, що відповідають середовищу виконання програмного забезпечення;

верифікація правильності реалізації системи й валідація правильності реалізації вимог до програмного забезпечення;

збір даних про відмови, помилки й інші непередбачених ситуаціях при виконанні програмного продукту;

підготовка звіту за результатами тестування й оцінка характеристик системи.

Метрики тестування. Для вимірювання результатів тестування програмного забезпечення, а також при проведенні аналізу якості необхідно використовувати метрики. Вимірювання як частина планування й розробки тестів базується на розмірі програм, її структурі й кількості виявлених помилок і дефектів. Метрики тестування забезпечують вимірювання процесу планування, проектування й тестування; а також результатів тестування на основі визначення відмов і дефектів, покриття границь тестування, перевірки потоків даних й ін.

Пункт 2.5.5.1. Специфікації тестів.

Для тестування програмного забезпечення необхідно розробити специфікації тестів для його функцій (функціоналу), що описані у пункті 1.3.3.4. методичних рекомендацій. У дипломному проекті потрібно детально розробити специфікацію тестів для **не менш ніж п'яти функцій** (у тому числі: функцій вводу даних, роботи елементів інтерфейсу, отримання вихідних форм (відеокадрів)).

Приклад опису специфікації наведено у табл. 4.34.

Специфікації тестів для функції <Назва функції >

Код та назва функції (функціоналу)	Специфікація
1	2
1. Специфікації на входи	"Специфікації на входи" регламентують входи, необхідні для проходження всіх тестів для даного рішення. У числі таких входів можуть бути різні дії користувачів. Опис специфікацій поряд з найменуванням входу має містити його характеристики, такі як значення даних, інтерфейси, протоколи, бази даних, файли
2. Специфікації на виходи	"Специфікації на виходи" регламентує виходи, очікувані як результат проходження тестів для даного рішення. Виходи можуть містити дані, інформацію про час вирішення, повідомлення, лог-файли, файли, графіки
3. Середовище тестування	У даному розділі необхідно вказати характеристику і конфігурацію середовища тестування. Такий опис може включати програмне забезпечення тестування, конфігурацію середовища тестування
4. Спеціальні вимоги	У цьому розділі вкажіть спеціальні вимоги до організації тестування
5. Взаємозалежності	У даному розділі приведіть існуючі взаємозв'язки між різними тестами. Наприклад, їх послідовність
6.1. Тест 1 Властивість А	У даному розділі описується тестування деякої частини розробленої функціональності. При цьому формулюються наступні положення: очікувана поведінка; очікувана продуктивність; очікувана достовірність
Властивість В	очікувана поведінка; очікувана продуктивність; очікувана достовірність
6.2. Тест N Властивість А	

Пункт 2.5.5.2. Результати тестування програмного забезпечення

Результати тестування **для всіх функцій, які описані у пункті 1.3.3.4**, необхідно представити у вигляді табл. 4.35.

Опис специфікацій тестів та результатів тестування

Функціонал (функція)	Специфікація входу функціоналу	Специфікація виходу функціонала	Результат тестування згідно обраним метрикам
Ідентифікатор та найменування функції (згідно варіантам використання)	"Специфікації на входи" регламентує входи, необхідні для проходження всіх тестів для даного рішення. У числі таких входів можуть бути різні дії користувачів. Опис специфікацій поряд з найменуванням входу має містити його характеристики, такі як значення даних, інтерфейси, протоколи, бази даних, файли	"Специфікації на виходи" регламентує виходи, очікувані як результат проходження тестів для даного рішення. Виходи можуть містити дані, інформацію про час вирішення, повідомлення, лог-файли, файли, графіки	Метрики: кількість сценаріїв (тестів) для даного функціонала; кількість знайдених помилок та їх типи; кількість помилок виправлених; кількість повторних перевірок

Пункт 2.5.5.3. Оцінка якості розробки програмного забезпечення

На основі даних результатів тестування програмного забезпечення необхідно навести графік порівняного аналізу знайдених та виправлених помилок (приклад графіка наведено на рис. 5.3).

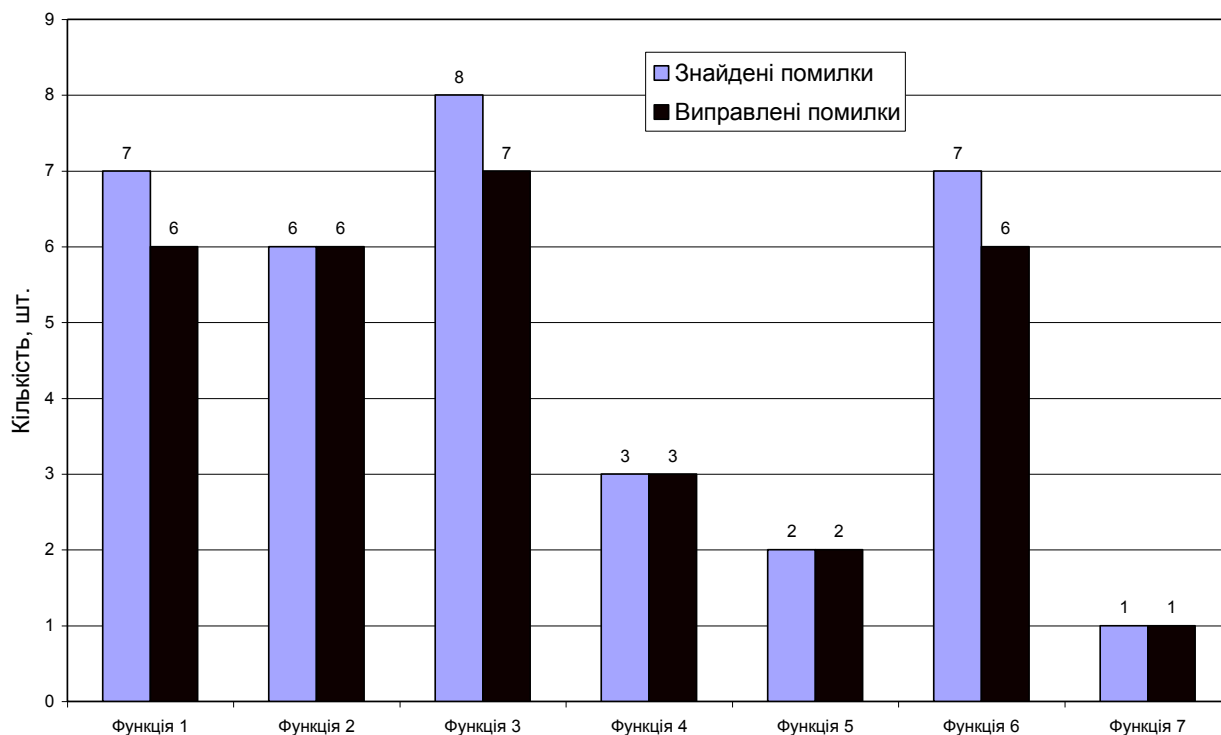


Рис. 4.3. Оцінка якості розробки програмного забезпечення

4.2.9. Розділ 3. Розроблення заходів щодо ергономіки, забезпечення охорони праці та техніки безпеки на об'єкті

Зміст і обсяг розділу конкретизується консультантом відповідно до специфіки теми дипломного проекту.

4.2.10. Висновки

Розділ містить висновки по кожному етапу виконаного проекту та по роботі в цілому. Зроблені виставки необхідно співвіднести з метою і завданням на дипломне проектування.

Необхідно зазначити практичну цінність результатів роботи ступінь впровадження, дати рекомендації для подальшого вдосконалення об'єкту проектування. Зазначаючи практичну цінність одержаних результатів, важливо окреслити ступінь їх готовності до використання, масштабів використання, а також надати стислі відомості щодо впровадження результатів досліджень із зазначенням назв організацій, в яких здійснена реалізація, форм реалізації, реквізитів документів тощо. Якщо дипломний проект впроваджений на підприємстві, то до дипломної роботи додається довідка або акт про впровадження.

Наводять відомості, де й коли було апробовано отримані результати (на яких конференціях, семінарах вони доповідались), перераховуються публікації студента за матеріалами дипломного проекту.

4.2.11. Використана література

Розділ повинен містити відомості про літературні джерела, використані при розробленні проекту. Список використаної літератури рекомендується складати в такому порядку:

1. Основні нормативні документи і матеріали (державні і урядові).
2. Друковані джерела суспільно-політичного, соціального, економічного, природничо-наукового, соціально-культурного характеру.
3. Книги.
4. Статті.
5. Дисертації.
6. Автореферати.

7. Патентні документи.
8. Нормативно-технічні документи.
9. Каталоги промислового устаткування виробів.
10. Депоновані рукописи.

Приклади оформлення бібліографічного опису у списку джерел [52].

Книги

Один автор

1. Коренівський Д. Г. Дестабілізуючий ефект параметричного білого шуму в неперервних та дискретних динамічних системах / Д. Г. Коренівський. – К. : Ін-т математики, 2006. – 111 с.

2. Матюх Н. Д. Що дорожче срібла-золота / Н. Д. Матюх. – К. : Асамблея діл. кіл: Ін-т соц. іміджмейкінгу, 2006. – 311 с. – (Ювеліри України; т. 1).

Два автори

1. Суберляк О. В. Технологія переробки полімерних та композиційних матеріалів : підруч. для студ. вищ. навч. закл. / О. В. Суберляк, П. І. Баштанник. – Львів : Растр-7, 2007. – 375 с.

Три автори

1. Акофф Р. Л. Идеализированное проектирование: как предотвратить завтрашний кризис сегодня. Создание будущего организации / Р. Л. Акофф, Д. Магидсон, Г. Д. Зддисон ; пер. с англ. Ф. П. Тарасенко. – Днепропетровск : Баланс Бизнес Букс, 2007. – 265 с.

Чотири автори

1. Механізація переробної галузі агропромислового комплексу : підруч. для учнів проф.-техн. навч. закл. / О. В. Гвоздєв, Ф. Ю. Ялпачик, Ю. П. Рогач та ін. – К. : Вища освіта, 2006. – 478.

П'ять і більше авторів

1. Психология менеджмента / П. К. Власов, А. В. Липницкий, И. М. Лушихина и др. ; под ред. Г. С. Никифорова. – 3-е изд. – Х. : Гуманитар. центр, 2007. – 510 с.

Без автора

1. Проблеми типологічної та квантитативної лексикології : зб. наук. праць / наук. ред. В. Каліущенко та ін. – Чернівці : Рута, 2007. – 310 с.

Багатотомний документ

1. Межгосударственные стандарты : каталог в 6 т. / сост. И. В. Ковалева, Е. Ю. Рубцова ; ред. В. Л. Иванов. – Львов : НТЦ "Леонорм-

Стандарт", 2005. – Т. 1. – 2005. – 277 с.

2. Бондаренко В. Г. Теорія ймовірностей і математична статистика / В. Г. Бондаренко, І. Ю. Канівська, С. М. Парамонова. – Ч.1 – К. : НТУУ "КПГ", 2006. – 125 с.

Матеріали конференцій, з'їздів

1. Кібернетика в сучасних економічних процесах : зб. текстів виступів на республік, міжвуз. наук.-практ. конф. / Держкомстат України, Ін-т статистики, обліку та аудиту. – К. : ІСОА, 2002. – 147 с.

2. Матеріали ІХ з'їзду Асоціації українських банків, 30 червня 2000 р. інформ. бюл. – К. : Асоц. укр. банків, 2000. – 117 с.

3. Оцінка й обґрунтування продовження ресурсу елементів конструкцій : праці конф., 6 – 9 черв. 2000 р., Київ. Т. 2 / відп. ред. В. Т. Трощенко. – К. : НАН України, Ін-т пробл. міцності, 2000. – С. 559–956.

Препринти

1. Шиляев Б. А. Расчеты параметров радиационного повреждения материалов нейтронами источника ННЦ ХФТИ/ANL USA с подкритической сборкой, управляемой ускорителем электронов / Б. А. Шиляев,

В. Н. Воєводин. – Х. : ННЦ ХФТИ, 2006. – 19 с. – (Препринт / НАН Украины, Нац. науч. центр "Харьк. физ.-техн. ин-т" ; ХФТИ 2006-4).

Депоновані наукові праці

1. Социологическое исследование малых групп населения / В. И. Иванов [и др.] ; М-во образования Рос. Федерации, Финансовая академия. – М., 2002. – 110 с. – Деп. в ВИНТИ 13.06.02, № 145432.

2. Разумовский В. А. Управление маркетинговыми исследованиями в регионе / В. А. Разумовский, Д. А. Андреев. – М., 2002. – 210 с. – Деп. в ИНИОН Рос. акад. наук 15.02.02, № 139876.

Словники

1. Українсько-німецький тематичний словник / укл. Н. Яцко та ін. – К. : Карпенко, 2007. – 219 с.

2. Європейський Союз : словник-довідник / ред.-упоряд. М. Марченко. – 2-ге вид., оновл. – К. : К.І.С., 2006. – 138 с.

Законодавчі та нормативні документи

1. Кримінально-процесуальний кодекс України : за станом на 1 груд. 2005 р. / Верховна Рада України. – Офіц. вид. – К. : Парлам. вид-во, 2006. – 207 с.

2. Медична статистика : зб. нормат. док. / упоряд. та

голов. ред. В. М. Заболотько. – К. : МНІАЦ мед. статистики : Медінформ, 2006. – 459 с.

3. Експлуатація, порядок і терміни перевірки запобіжних пристроїв посудин, апаратів і трубопроводів теплових електростанцій : СОУ-Н ЕЕ 39.501:2007. – Офіц. вид. – К. : ГРІФРЕ : М-во палива та енергетики України, 2007. – VI, 74 с.

Стандарти

1. Графічні символи, що їх використовують на устаткуванні. Показчик та огляд (ІСО 7000:2004, ЮТ) : ДСТУ ІЗО 7000:2004. – [Чинний від 2006-01-01]. – К. : Держспоживстандарт України, 2006. – IV, 231 с.

2. Якість води. Словник термінів : ДСТУ ІЗО 6107-1:2004 – ДСТУ ІЗО 6107-9:2004. – [Чинний від 2005-04-01]. – К. : Держспоживстандарт України, 2006. – 181 с.

Каталоги

1. Межгосударственные стандарты : каталог : в 6 т. / сост. И. В. Ковалева, В. А. Павлюкова ; ред. В. Л. Иванов. – Львов : НТЦ "Леонорм-стандарт", 2006. – Т. 5. – 2007. – 264 с.

2. Горницкая И. П. Каталог растений для работ по фитодизайну / И. П. Горницкая, Л. П. Ткачук. – Донецк : Лебедь, 2005. – 228 с.

Бібліографічні показники

1. Куц О. С. Бібліографічний показник та анотації кандидатських дисертацій, захищених у спеціалізованій вченій раді Львівського державного університету фізичної культури у 2006 році / О. Куц, О. Вацеба. – Львів : Укр. технології, 2007. – 74 с.

Дисертації

1. Петров П. П. Активність молодих зірок сонячної маси : дис. доктора фіз. мат., наук : 01.03.02 / П. П. Петров. – К., 2005. – 276 с.

Автореферати дисертацій

1. Новосад І. Я. Технологічне забезпечення виготовлення секцій робочих органів гнучких гвинтових конвеєрів : автореф. дис. на здобуття наук, ступеня канд. техн. наук : спец. 05.02.08 "Технологія машинобудування" / І. Я. Новосад. – Тернопіль, 2007. – 20 с.

2. Нгуен Ші Данг. Моделювання і прогнозування макроекономічних показників в системі підтримки прийняття рішень управління державними фінансами : автореф. дис. на здобуття наук, ступеня канд. техн. наук : спец. 05.13.06 "Автоматиз. системи упр. та прогрес, інформ. технології" / Нгуен Ші Данг. – К., 2007. – 20 с.

Авторські свідоцтва

1. А. с. 1007970 СССР, МКИЗ В 25 J 15/00. Устройство для захвата неориентированных деталей типа валов / В. С. Ваулин, В. Г. Кемайкин (СССР). – № 3360585/25-08 ; заявл. 23.11.81 ; опубл. 30.03.83, Бюл. № 12.

Патенти

1. Пат. 2187888 Российская Федерация, МПК Н 04 В 1/38, Н 04 и 13/00. Приемопередающее устройство / В. И. Чугаева; заявитель и патентообладатель Воронеж, науч.-исслед. ин-т связи. – № 2000131736/09 ; заявл. 18.12.00 ; опубл. 20.08.02, Бюл. №23 (II ч.).

Частина книги, періодичного продовжуваного видання

1. Козіна Ж. Л. Теоретичні основи і результати практичного застосування системного аналізу в наукових дослідженнях в області спортивних ігор / Ж. Л. Козіна // Теорія та методика фізичного виховання. – 2007. – № 6. – С. 15–18; 35–38.

2. Гранчак Т. Інформаційно-аналітичні структури бібліотек в умовах демократичних перетворень / Т. Гранчак, В. Горовий // Бібліотечний вісник. – 2006. – № 6. – С. 14–17.

3. Валькман Ю. Р. Моделирование НЕ-факторов – основа интеллектуализации компьютерных технологий / Ю. Р. Валькман, В. С. Быков, А. Ю. Рыхальский // Системні дослідження та інформаційні технології. – 2007. – № 1 – С. 39–61.

Електронні ресурси

1. Бібліотека і доступність інформації у сучасному світі : електронні ресурси в науці, культурі та освіті : (підсумки 10-ї Міжнар. конф. "Крим-2003") [Електронний ресурс] / Л. Й. Костенко, А. О. Чекмарьов, А. Г. Бровкін та ін. // Бібліотечний вісник – 2003. – № 4. – С. 43. – Режим доступу : <http://www.nbuv.ua/articles/2003/03klinko.htm>.

У межах кожної групи список формується в порядку алфавіту або черговості згадування документа в тексті. Про кожне джерело даються наступні зведення: прізвище та ініціали автора, заголовок джерела, місце видання, видавництво і рік видання, обсяг у сторінках. Для статті вказуються найменування журналу, рік його випуску і номера сторінок, на яких поміщена стаття.

4.2.12. Додатки

У додатках розміщують матеріали, що необхідні для повноти дипломного проекту, але долучення їх до основної частини проекту може змінити впорядковане і логічне подання роботи через великий обсяг чи

специфіку форми подання. Ці матеріали не можуть бути внесені в основну частину (наприклад схеми, перелік елементів, специфікації, опис комп'ютерних програм, протоколи випробувань, та ін.). На додатки має бути посилання у відповідних розділах. Додатки можуть містити:

- графічний матеріал;
- таблиці, що доповнюють основний текст;
- розрахунки;
- оригінали фотографій;
- опис програмно-технічних комплексів;
- інструкції і методики, опис алгоритмів і програм вирішення задач на ПК, які розроблені в процесі виконання дипломного проекту;
- ілюстрації допоміжного характеру;
- та інші матеріали.

Використана література

1. Про вищу освіту / Закон України № 2984-III // Відомості Верховної Ради. – 2002. – № 20. – 134 с.
2. Комплекс нормативних документів для розробки складових системи стандартів вищої освіти. Додаток 1 до наказу Міністерства освіти України від 31.07.1998 р. № 285 із змінами та доповненнями, уведеними розпорядженням Міністерства освіти і науки України від 05.03.2001 р. № 28-р. // Інформаційний вісник "Вища освіта". – 2003. – № 10. – 82 с.
3. Галузевий стандарт вищої освіти України з напрямку підготовки 6.050101 "Комп'ютерні науки" : збірник нормативних документів вищої освіти. – К. : Видавнича група BHV, 2011. – 85 с.

Рекомендована література

Основна

4. Автоматизированные информационные системы, базы и банки данных. Вводный курс : учебн. пособ. – М. : Гелиос АРВ, 2002. — 368 с.
5. Анфилов В. С. Системный анализ в управлении. / В. С. Анфилов, А. А. Емельянов, А. А. Кукушкин. – М. : Финансы и статистика, 2002. – 468 с.
6. Астелс Д. Практическое руководство по экстремальному программированию / Д. Астелс, Г. Миллер, М. Новак ; пер. с англ. – М. : Издательский дом "Вильямс", 2002. – 320 с.
7. Баранов О. А. Інформаційне право України: стан, проблеми, перспективи. – К. : Видавничий дім "СофтПрес", 2005. – 316 с.
8. Басс Л. Архитектура программного обеспечения на практике / Л. Басс, П. Клементс, Р. Кацман. – СПб. : Питер, 2006. – 576 с.
9. Бек К. Экстремальное программирование / К. Бек. – СПб. : Питер, 2002. – 224 с.
10. Береза А. М. Основи створення інформаційних систем : навч. пос. / А. М. Береза – К. : КНЕУ, 2001. – 214 с.
11. Блэк Р. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование. – М. : Лори, 2011. – 544 с.
12. Богданов В. В. Управление проектами в Microsoft Project / В. В. Богданов. – СПб. : Питер, 2004. – 604 с.
13. Бондаренко М. Ф. Моделирование и проектирование бизнес-систем: методы, стандарты, технологии : учебн. пос. / М. Ф. Бондаренко,

С. И. Маторин, Е. А. Соловьев. – Х. : Компания СМИТ, 2004. – 272 с.

14. Брауде Э. Технологии разработки программного обеспечения / Э. Брауде – СПб. : Питер, 2004. – 655 с. : ил.

15. Веретенников В. І. / Управління проектами : навч. пос. / В. І. Веретенников, Л. М. Тарасенко, Г. І. Гевлич. – К. : Центр навчальної літератури, 2006. – 280 с.

16. Винниченко И. Автоматизация процессов тестирования / И. Винниченко. – СПб. : Питер, 2005. – 203 с.

17. Вигерс К. Разработка требований к программному обеспечению / К. Вигерс ; перс. англ. – М. : Изд.-торговый дом "Русская редакция", 2004. – 576 с.

18. Гайфуллин Б. Н. Автоматизированные системы управления предприятиями стандарта ERP/MRP II. Производственное издание / Б. Н. Гайфуллин, И. А. Обухов. – М. : "Богородский печатник", 2001. – 104 с.

19. Гейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем / Б. Гейзер. – СПб. : Питер, 2004 – 320 с.

20. Гради Б. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч, А. Роберт, У. Майкл и др. – М. : Издательский дом "Вильямс", 2010. – 720 с.

21. Горбатов В. С. Основы технологии PKI / В. С. Горбатов, О. Ю. Полянская. – М. : Горячая линия – Телеком, 2004. – 248 с. : ил.

22. Годин В. В. Управление информационными ресурсами / В. В. Годин, И. К. Корнеев. – М. : ИНФРА- М, 2000. – 352 с.

23. ГОСТ 19.701-90. Схемы алгоритмов, данных, программ и систем. Условные обозначения и правила выполнения. – М. : Изд. стандартов, 1990. – 16 с.

24. ГОСТ 34.201-89. Виды, комплектность и обозначение документов при создании автоматизированных систем. – М. : Изд. стандартов, 1989. – 16 с.

25. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания. – М. : Изд. стандартов, 1990. – 12 с.

26. ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы. – М. : Изд. стандартов, 1990. – 24 с.

27. Гужва В. М. Інформаційні системи і технології на підприємствах : навч. пос. / В. М. Гужва. – К. : КНЕУ, 2001. – 400 с.

28. ДСТУ 2938-94. Системи оброблення інформації. Основні поняття. Терміни та визначення. – К. : Держстандарт України, 1995. – 32 с.
29. ДСТУ 2940-94. Системи оброблення інформації. Керування процесами оброблення даних. Терміни та визначення. – К. : Держстандарт України, 1995. – 28 с.
30. ДСТУ 2941-94. Системи оброблення інформації. Розробки систем. Терміни та визначення. – К. : Держстандарт України, 1995. – 20 с.
31. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Правила оформлення. – К. : Держкомстат України, 1995. – 28 с.
32. ДСТУ 3918-99 (ISO/IEC 12207:1995). Інформаційні технології. Процеси життєвого циклу програмного забезпечення. – К. : Держкомстат України, 1999. – 48 с.
33. ДСТУ ISO/IEC TR 15504-1-2002 (Частини 1–9). Інформаційні технології. Оцінювання процесів програмування. К. : Держспоживстандарт України, 2002. – 42 с.
34. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт. – 8-е изд. – М. : Вильямс, 2005. – 1328 с.
35. Дэниел О. Л. ERP системы. Современное планирование и управление ресурсами предприятия. Выбор, внедрение, эксплуатация / О. Л. Дэниел. – М. : ООО "Вершина", 2004. – 272 с.
36. Елиферов В. Г. Бизнес-процессы: Регламентация и управление : учебник / В. Г. Елиферов, В. В. Репин. – М. : ИНФРА-М, 2004. – 320 с.
37. Єрьоміна Н. В. Банківські інформаційні системи : навч. посібн. / Н. В. Єрьоміна. – К. : КНЕУ, 2000. – 272 с.
38. Зима В. М. Безопасность глобальных сетевых технологий. – 2-е изд. / В. М. Зима, А. А. Молдовян, Н. А. Молдовян. – СПб. : БХВ-Петербург, 2003. – 368 с.: ил.
39. Золотарьова І. О. Інформаційні системи та технології в банківській сфері : навч. посібн. / І. О. Золотарьова, Р. К. Бутова, А. А. Гаврилова. – Х. : Вид. ХНЕУ, 2009. – 332 с. (укр. мов.).
40. Информатизация бизнеса: концепции, технологии, системы / А. М. Карминский, С. А. Карминский, В. П. Нестеров, Б. В. Чернишов ; под ред. А. М. Карминского. – М. : Финансы и статистика, 2004. – 624 с.
41. Інформаційні системи в економіці / за ред. В. С. Пономаренка. – К. : Академія, 2002. – 542 с.

42. Кальянов Г. Н. CASE-технологии. Консалтинг в автоматизации бизнес-процессов / Г. Н. Кальянов. – М. : Горячая линия – Телеком, 2002. – 320 с.
43. Каменова М. Моделирование бизнеса. Методология ARIS / М. Каменова, А. Громов, М. Ферапонтов и др. – М. : Весть-Мета Технология, 2001. – 328 с.
44. Канер С. Тестирование программного обеспечения / С. Канер, Дж. Фолк, Е. К. Нгуен. – К. : "ДиаСофт", 2001. – 544 с.
45. Карминский А. М. Информатизация бизнеса / А. М. Карминский, А. С. Карминский, В. П. Нестеров и др. – М. : Финансы и статистика, 2004. – 624 с.
46. Карпова Т. Базы данных. Модели, разработка, реализация : учебник / Т. Карпова. – СПб. : 2001. – 302 с.
47. Конноли Т. Базы данных: проектирование, реализация и сопровождение учебн. пособ. / Т. Конноли. // Теория и практика. – 2-е изд. ; пер. с англ. – М. : Издательский дом "Вильямс", 2000. – 1120 с.
48. Коберн А. Быстрая разработка программного обеспечения / А Корбен. – М. : Лори, 2002. – 314 с.
49. Коберн А. Современные методы описания функциональных требований к системам / А. Коберн. – М. : Издательство "Лори", 2002. – 263 с.
50. Коробов П. Н. Математическое программирование и моделирование экономических процессов / П. Н. Коробов. – М. : ДНК, 2010. – 376 с.
51. Кузнецов О. О. Захист інформації в інформаційних системах. Методи традиційної криптографії / О. О. Кузнецов, С. П. Євсєєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2010. – 316 с.
52. Кузнецов О. О. Захист інформації в інформаційних системах / О. О. Кузнецов, С. П. Євсєєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2011. – 504 с.
53. Кузнецов О. О. Захист інформації та економічна безпека підприємства / О. О. Кузнецов, С. П. Євсєєв, С. В. Кавун. – Х. : Вид. ХНЕУ, 2009. – 360 с.
54. Кузнецов О. О. Сигнали і коди. Алгебраїчні методи синтезу / О. О. Кузнецов, С. П. Євсєєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2009. – 384 с.
55. Ленков С. В. Методы и средства защиты информации. В 2-х томах / С. В. Ленков, Д. А. Перегудов, В. А. Хорошко ; под ред. В. А. Хорошко. – К. : Арий, 2008. – Т. I. Несанкционированное получение информации. – 464 с. : ил.

56. Ленков С. В. Методы и средства защиты информации. В 2-х томах / С. В. Ленков, Д. А. Перегудов, В. А. Хорошко ; под ред. В. А. Хорошко. – К. : Арий, 2008. – Т. II. Информационная безопасность. – 344 с. : ил.
57. Леоненков А. В. Самоучитель UML / А. В. Леоненков. – СПб. : БХВ-Петербург, 2004. – 432 с.
58. Леффингуелл Д. Принципы работы с требованиями к программному обеспечению / Д. Леффингуелл, Д. Уидриг. – М. : ИД "Вильямс", 2002. – 448 с.
59. Лодон Дж. Управление информационными системами / Дж. Лодон, К. Лодон ; под. ред. Д. Р. Трутнева ; пер. с англ. – 7-е изд. – СПб. : Питер, 2005. – 912 с.
60. Лямец В. И. Системный анализ. Вводный курс / В. И. Лямец, А. Д. Тевяшев. – Х. : ХНУРЭ, 2004. – 448 с.
61. Маглинец Ю. А. Анализ требований к автоматизированным информационным системам / Ю. А. Маглинец. – М. : БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий – ИНТУИТ. ру, 2008. – 200 с.
62. Макарова Н. В. Информатика : учебник / Н. В. Макарова. – М. : Финансы и статистика, 2003. – 768 с.
63. Маклаков С. В. Создание информационных систем с AllFusion Modeling Suit / С. В. Маклаков. – М. : Москва ДиалогМифи, 2003. – 432 с.
64. Мацяшек Л. Анализ требований и проектирование систем / Л. Мацяшек ; пер. с англ. – М. : Издательский дом "Вильямс", 2002. – 432 с.
65. Меняев М. Ф. Информационные технологии управления : книга 3 : системы управления организацией / М. Ф. Меняев. – М. : Омега-Л, 2003. – 464 с.
66. Методичні рекомендації до виконання комплексного курсового проекту для студентів спеціальності "Інформаційні управляючі системи і технології" всіх форм навчання / укл. І. О. Золотарьова, С. В. Мінухін, Ю. Е. Парфьонов та ін. – Х. : Вид. ХНЕУ, 2010. – 82 с.
67. Методичні рекомендації до оформлення звітів, курсових та дипломних проектів для студентів напряму підготовки 0804 "Комп'ютерні науки" всіх форм навчання / укл. І. О. Золотарьова, О. М. Беседовський, І. Л. Латишева та ін. – Х. : Вид. ХНЕУ, 2007. – 32 с. (укр. мов.)
68. Ойхман Е. Г. Реинжиниринг бизнеса: реинжиниринг организаций и информационные технологии / Е. Г. Ойхман, Е. В. Попов. – М. : Финансы и статистика 1997. – 336 с.

69. Орлова И. В. Экономико-математические методы и модели. Компьютерное моделирование. / И. В. Орлова, В. А. Половников. – М. : Инфра-М, 2011. – 368 с.
70. Орлов С. Технологии разработки программного обеспечения : учебник / С. Орлов. – СПб. : Питер, 2002. – 464 с.
71. Петров А. А. Компьютерная безопасность. Криптографические методы защиты / А. А. Петров. – М. : ДМК, 2000. – 448 с. : ил.
72. Петров В. Н. Информационные системы / В. Н. Петров. – СПб. : Питер, 2002. – 688 с.
73. Поповский В. В. Защита информации в телекоммуникационных системах : учебник : в 2 т. / В. В. Поповский, А. В. Персиков. – Х. : ООО "Компания СМИТ", 2006. – Т. 1. – 292 с.
74. Поповский В. В. Защита информации в телекоммуникационных системах : учебник : в 2 т. / В. В. Поповский, А. В. Персиков. – Х. : ООО "Компания СМИТ", 2006. – Т. 2. – 252 с.
75. Принципы проектирования и разработки программного обеспечения. Учебный курс MCSD / пер. с англ. – 2-е изд., испр. – М. : Издательско-торговый дом "Русская Редакция", 2002. – 736 стр.
76. РД 50-34.698-90. Руководящий документ за стандартизации. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов. – М. : Изд. стандартов, 1991. – 40 с.
77. Смирнова Г. Н. Проектирование экономических информационных систем : учебник / Г. Н. Смирнова, А. А. Сорокин, Ю. Ф. Тельнов ; под ред. Ю. Ф. Тельнова. – М. : Финансы и статистика, 2001. – 512 с.
78. Соммервилл И. Инженерия программного обеспечения. – 6-е изд. / И. Соммервилл. – М. : Издательский дом "Вильямс", 2002. – 624 с.
79. Татарчук М. І. Корпоративні інформаційні системи : навч. посібн. / М. І. Татарчук. – К. : ХНЕУ, 2005. – 291 с.
80. Ушакова І. О. Інформаційні системи і технології в статистиці / І. О. Ушакова. – Х. : Вид. ХНЕУ, 2006. – 164 с.
81. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації : навч. посібн. – Ч. 1 / І. О. Ушакова. – Х. : Вид. ХНЕУ, 2007. – 212 с.
82. Ушакова І. О. Основи системного аналізу об'єктів та процесів

комп'ютеризації : навч. посібн. – Ч. 2 / І. О. Ушакова. – Х. : Вид. ХНЕУ, 2008. – 324 с.

83. Ушакова І. О. Практикум з навчальної дисципліни "Основи системного аналізу об'єктів і процесів комп'ютеризації": навч.-практ. посібн.

/ І. О. Ушакова, Г. О. Плеханова. – Х. : Вид. ХНЕУ, 2010. – 344 с.

84. Фаулер М. UML в кратком изложении. Применение стандартного языка объектного моделирования / М. Фаулер, К. Скотт. – М. : Мир, 1999. – 191 с.

85. Шафер Д. Ф. Управление программными проектами: достижение оптимального качества при минимуме затрат / Д. Ф. Шафер, Р. Т. Фатрел, Л. И. Шафер. – М. : Издательский дом "Вильямс", 2003. – 1136 с.

86. Якобсон А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо. – СПб. : Питер, 2002. – 496 с.

87. Столлингс В. Криптография и защита сетей: принципы и практика ; 2-е изд. / В. Столлингс ; пер. с англ. – М. : Издательский дом "Вильямс", 2001. – 672 с. : ил.

88. Щеглов А. Ю. Защита компьютерной информации от несанкционированного доступа / А. Ю. Щеглов. – СПб. : Наука и Техника, 2004. — 384 с. : ил.

89. Вербіцький О. В. Вступ до криптології / О. В. Вербіцький. – Львів : ВНТЛ, 1998. – 248 с.

90. Цифровая стеганография / В. Г. Грибунин, И. Н. Оков, И. В. Туринцев и др. – М. : СОЛОН-Прес, 2002. – 272 с.

91. Чмора А. Л. Современная прикладная криптография / А. Л. Чмора. – М. : Гелиос АРВ, 2001. – 256 с. ил.

92. Хорошко В. А. Методы и средства защиты информации / В. А. Хорошко, А. А. Чекатков. – К. : Юниор, 2003. – 504 с

93. Хомоненко А. Д. Базы данных : учебник для высших учебных заведений / А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев. – СПб. : Корона, 2004. – 736 с.

94. Чубук В. В. Бази даних у питаннях і відповідях : навч. посібн. / В. В. Чубук, Р. М. Чен, Л. А. Павленко, В. І. Клименко. – Х. : Вид. ХНЕУ, 2004. – 288с.

Ресурси мережі Інтернет

95. Приклади оформлення бібліографічного опису у списку джерел, який наводять у дисертації, і списку опублікованих робіт, який наводять в авторефераті. [Електронний ресурс]. Бюлетень ВАК України, №3, 2008 (Форма 23, С. 9 – 13)). – Режим доступу : http://www.kntu.kr.ua/doc/bibl_opis.pdf; http://www.kntu.kr.ua/doc/bibl_opis.pdf.

96. Самое интересное о разработке программного обеспечения [Электронный ресурс]. – Режим доступа : <http://www.maxkir.com/>.

97. CNews. Издание о высоких технологиях [Электронный ресурс]. – Режим доступа : <http://www.cnews.ru/>.

98. Интернет-университет информационных технологий [Электронный ресурс]. – Режим доступа : <http://www.intuit.ru/>.

99. Информационный портал CRM [Электронный ресурс]. – Режим доступа : www.crm.com.ua.

100. CITForum.ru. IT-портал [Электронный ресурс]. – Режим доступа : <http://www.citforum.ru/>.

101. Корпоративне управління [Електронний ресурс]. – Режим доступу : <http://www.corporation.com.ua/>.

102. Корпоративный менеджмент [Электронный ресурс]. – Режим доступа : <http://www.cfin.ru/>.

103. Корпорация "Галактика". Информационные технологии управления [Электронный ресурс]. – Режим доступа : <http://www.galaktika.ru/>.

104. Открытые системы [Электронный ресурс]. – Режим доступа : <http://www.osp.ru/>.

105. Офіційний сайт Держкомстату України. – Режим доступу : www.ukrstat.gov.ua.

106. Портал "Информационно-коммуникационные технологии в образовании" [Электронный ресурс]. – Режим доступа : <http://www.ict.edu.ru/>.

107. Портал "Профессионал управления проектами" [Электронный ресурс]. – Режим доступа : <http://www.pmpprofy.ru/>.

108. Сайт компанії IBM. – Режим доступу : <http://www.ibm.com/ru/ru/>.

109. Сайт компанії Gartner, Inc. (NYSE: IT). – Режим доступа : <http://www.gartner.com/>.

110. Сайт компанії Interface: Internet and software company. – Режим

доступа : <http://www.interface.ru/>.

111. Сайт компании Microsoft. – Режим доступа : <http://www.microsoft.com/>.

112. BYTE-Россия – журнал для ИТ-профессионалов [Электронный ресурс]. – Режим доступа : <http://www.bytemag.ru/>.

113. ERP-эксперт – Всё о ERP, ERP II, MRP, MRP II [Электронный ресурс]. – Режим доступа : <http://erp-expert.narod.ru/index.htm>.

114. ITC Online [Электронный ресурс]. – Режим доступа : <http://itc.ua/>.

115. The official UML Web site [Электронный ресурс]. — Режим доступа : <http://www.uml.org>.

116. Ten Usability Heuristics : Web-сайт Якоба Нильсена. [Электронный ресурс]. – Режим доступа : <http://www.useit.com/papers/>.

Додатки

Додаток А

Пояснювальна записка. Титульний аркуш

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
Факультет економічної інформатики
Кафедра інформаційних систем

Допускаю до захисту:

Завідувач кафедри

д. е. н., проф. Пономаренко В. С.

_____ 20__ р

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

**"РОЗРОБЛЕННЯ МОДУЛЯ "КОНТРОЛЬ РОЗРАХУНКІВ ЗА ПОСЛУГИ
ТЕПЛОПОСТАЧАННЯ" АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ
СИСТЕМИ ТОВ "БРОК-ЕНЕРГІЯ""**

освітньо-кваліфікаційний рівень "спеціаліст"

спеціальність 7.080401 – інформаційні управляючі системи і технології

Керівник проекту

к.е.н., проф. кафедри ІС

І. О. Іванов

Консультант

П. П. Петренко

Виконавець

Студент 5 курсу 5 групи

Р. С. Турчак

Харків, 2011

Зразок заяви студента на затвердження теми дипломного проекту

Завідувачу кафедри
інформаційних систем
проф. Пономаренку В.С.
студента __ курсу __ групи

ЗАЯВА

Прошу затвердити тему дипломного проекту _____

База практики: _____

Дата

(підпис студента)<підпис студента>

Керівник
дипломного проекту
звання, ПІБ керівн.>

(підпис)

_____<посада, наук. ступінь,

Пояснювальна записка. Завдання на дипломний проект

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Факультет економічної інформатики
Кафедра інформаційних систем
Спеціальність "Інформаційні управляючі системи і технології"

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ (підпис) _____ (ініціали, прізвище)
"___" _____ 20__ р.

ЗАВДАННЯ

на дипломний проект освітньо-кваліфікаційного рівня
"Спеціаліст"

Кваліфікація – спеціаліст (інформаційні управляючі системи і технології)

студенту _____
(прізвище, ім'я, по батькові)

1. Тема проекту: _____

_____ затверджена наказом по університету від "___" _____ 20__ р. № _____

2. Термін здачі студентом закінченого проекту: "___" _____ 20__ р

3. Вхідні дані до проекту:

_____ 4. Зміст пояснювальної записки (перелік питань, що підлягають розробленню):

_____ 5. Перелік графічного матеріалу:

6. Консультанти по проекту із зазначенням розділів проекту, що їх стосуються

Найменування розділу	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання	Позначка керівника про виконання

7. Дата видачі завдання "___" _____ 20__р.

Завдання підготував

Керівник дипломного проекту _____
 (підпис) (ініціали, прізвище)

Студент _____
 (підпис) (ініціали, прізвище)

Зразок відгуку керівника дипломного проекту

Відгук
керівника дипломного проекту
освітньо-кваліфікаційного рівня "Спеціаліст"

виконаного на тему:

студентом (кою)

(прізвище, ім'я, по батькові)

Керівник дипломної роботи

(посада, вчені звання, ступінь)

(підпис)

(ініціали, прізвище)

Зразок рецензії керівника дипломного проекту

Рецензія
на дипломний проект
освітньо-кваліфікаційного рівня "Спеціаліст"

виконаного на тему:

студентом (кою)

(прізвище, ім'я, по батькові)

Рецензент

(посада, вчені звання, ступінь)

(підпис)

(ініціали, прізвище)

Печатка установи, організації рецензента (*тільки для зовнішнього рецензента*)

Зразок оформлення реферату

Реферат

Пояснювальна записка: 90 сторінок, 30 рисунків, 20 таблиць, 12 додатків, 55 джерел.

Об'єктами проектування є комплекси функціональних елементів, архітектури, інформаційного і програмного забезпечення модуля моніторингу складських поставок.

Предметною областю проектування є система організаційно-економічного управління складською діяльністю ТОВ "СІГМА".

Мета проектування – створити проект модуля "Моніторинг складських поставок", який становить технічна документація з докладним описом усіх проектних рішень. Створений проект дозволяє підвищити достовірність обліку товарів на складі, обґрунтованість та оперативність прийняття рішень при управлінні поставками товарів і їх запасами на складі.

Пояснювальна записка дипломного проекту містить результати розробки модуля "Моніторинг складських поставок". Проведений аналіз предметної області, розроблені моделі організаційної структури, бізнес-процесів та варіантів використання програмного продукту з використанням CASE-засобу ARIS Toolset. Розроблена специфікація функціональних та не функціональних вимог до програмного продукту з використанням CASE-інструментів IBM Rational.

Описана архітектура системи, математична постановка, діаграма класів формування вихідних документів з вхідних. Спроектована модель БД предметної області з використанням CASE-інструментів IBM Rational. Розроблений програмний додаток в середовищі Microsoft Visual Studio. Наведені результати тестування програмного продукту. Пророблені питання безпеки життя і діяльності людини.

Результати розробки можуть бути впроваджені на торгових підприємствах.

Системи управління ланцюгами поставок, об'єктно-орієнтоване проектування, case-діаграми, моделювання предметної області, база даних, моніторинг складських поставок, запаси, web-сервіс, віртуальна

Зразок переліку умовних позначень і скорочень

Перелік умовних позначень і скорочень

англ. – англійська

БД – база даних

год. – година

КІС – корпоративна інформаційна система

теор. – теоретичний

Зразок діаграм опису предметної області

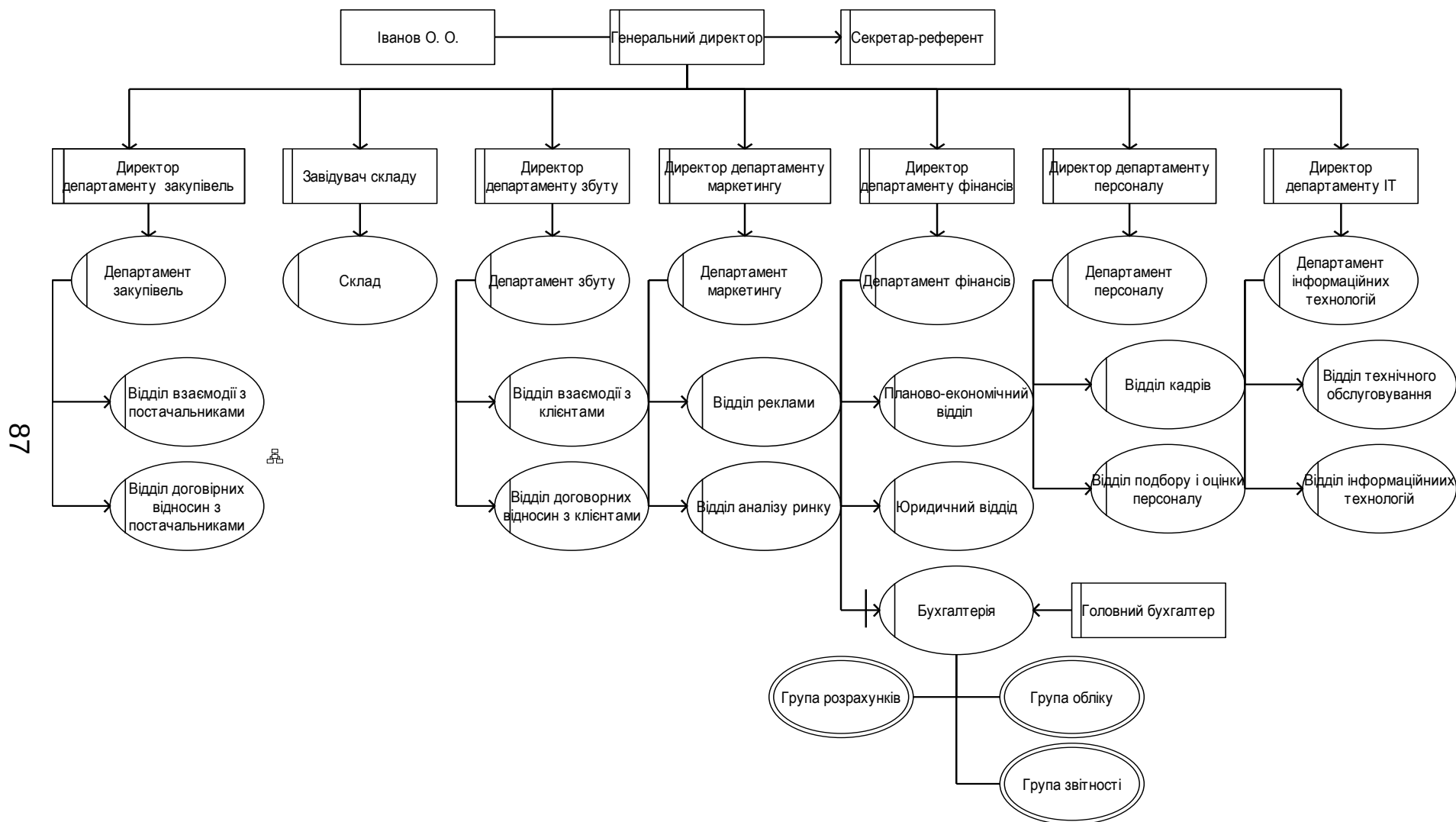


Рис. К.1. Організаційна структура підприємства дрібнооптової торгівлі

Зразок діаграм опису предметної області



Рис. Л.2. Організаційна схема відділу по взаємодії з постачальниками

Зразок діаграм опису предметної області

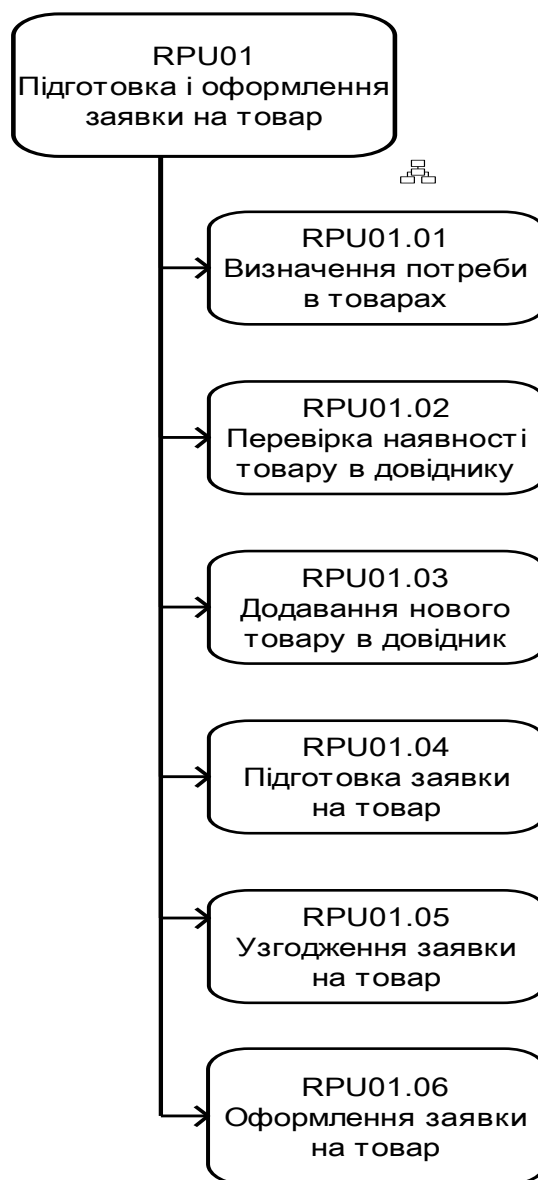


Рис. К.3. Діаграма "Дерево функцій"

Зразок діаграм опису предметної області

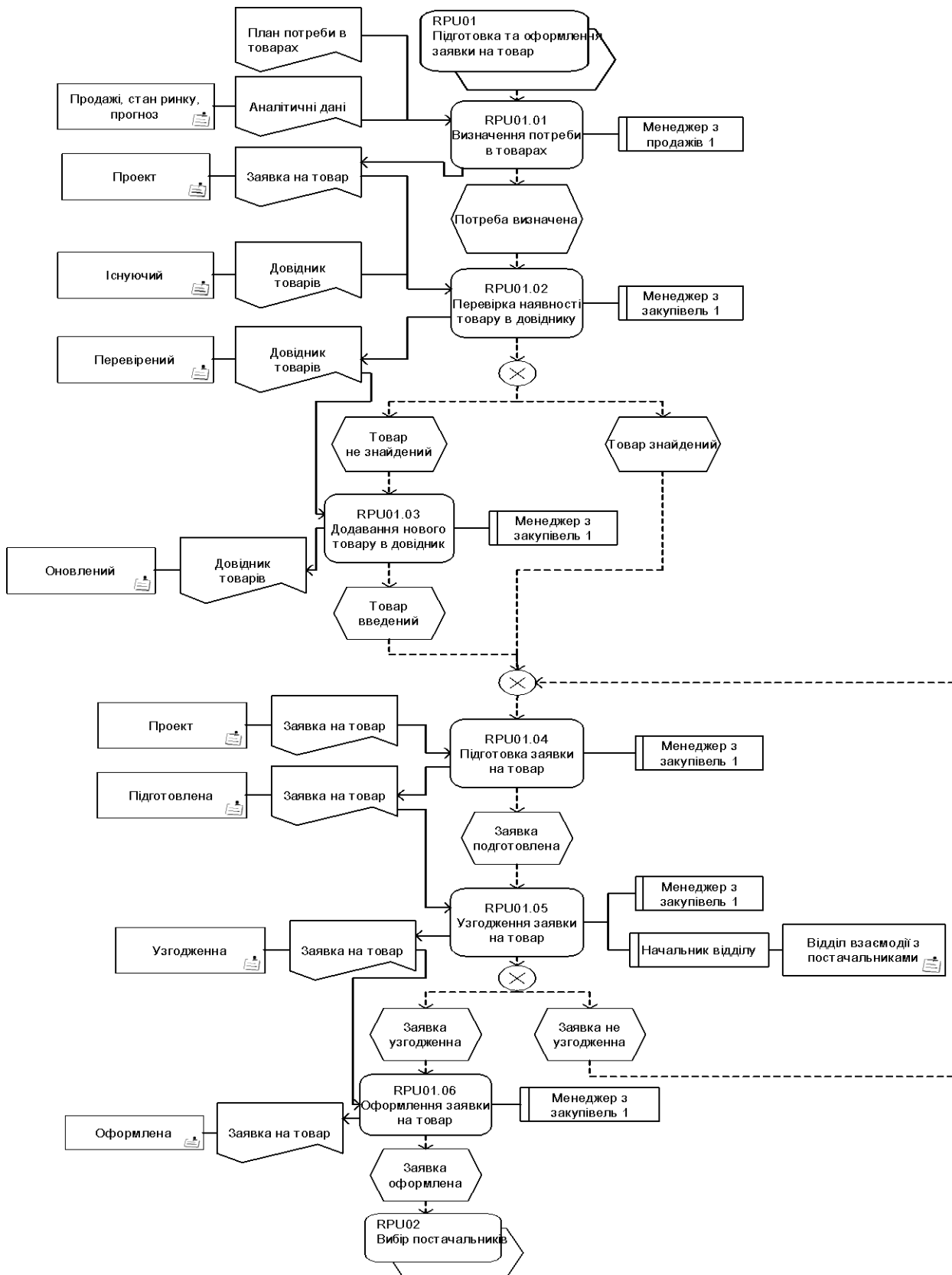


Рис. К.4. Процес "PR01. Підготовка та оформлення заявки на товар"

Приклад діаграми зв'язку вхідних документів з вихідними даними

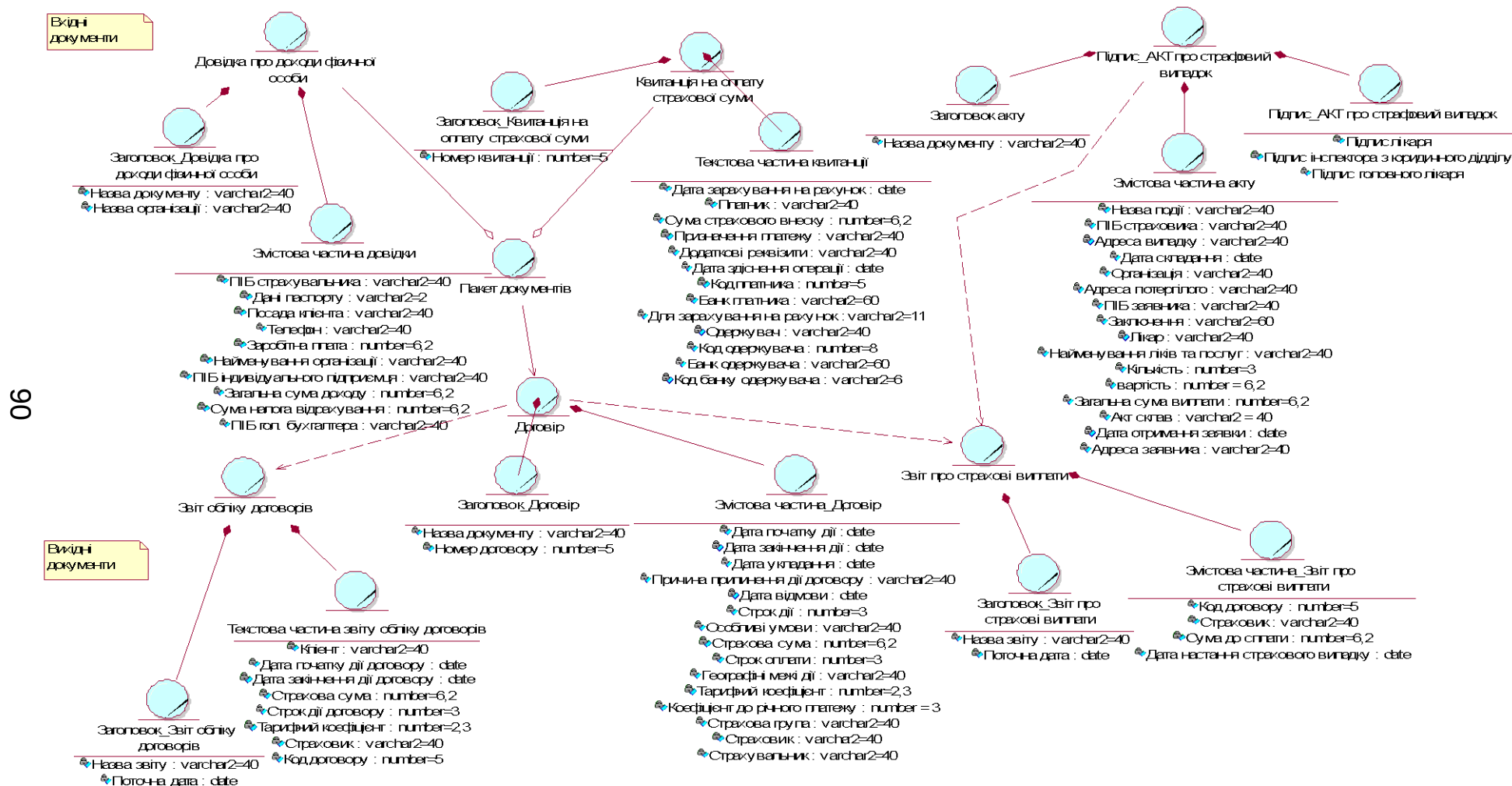


Рис. Л.1. Діаграма зв'язку вхідних документів з вихідними даними для предметної області <...> на базі бізнес клас-сутності бізнесу

Приклад структурної схеми програмної системи**Рис. М.1. Структурна схема розробленої програмної системи**

НАВЧАЛЬНЕ ВИДАННЯ

**Методичні рекомендації
до виконання дипломного проекту
для студентів спеціальностей
"Інформаційні управляючі системи
і технології",
"Комп'ютерний еколого-економічний моніторинг"
усіх форм навчання**

Укладачі: **Золотарьова** Ірина Олександрівна
Ушакова Ірина Олексіївна
Мінухін Сергій Володимирович та ін.

Відповідальний за випуск **Пономаренко В. С.**

Редактор **Бутенко В. О.**

Коректор **Бутенко В. О.**

План 2011 р. Поз. № 204.

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

Ум.-друк. арк. 5,75. Обл.-вид. арк. 7,2. Тираж прим. Зам. №

Видавець і виготівник — видавництво ХНЕУ, 61001, м. Харків, пр. Леніна, 9а

*Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк № 481 від 13.06.2001 р.*