

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**Методичні рекомендації  
до виконання кваліфікаційних робіт  
для студентів спеціальностей  
121 "Інженерія програмного забезпечення",  
122 "Комп'ютерні науки",  
126 "Інформаційні системи та технології"  
першого (бакалаврського) рівня**

**Харків  
ХНЕУ ім. С. Кузнеця  
2020**

УДК 004.415(07.034)

М54

**Укладачі:** Ю. Е. Парфьонов

І. О. Ушакова

Затверджено на засіданні кафедри інформаційних систем.

Протокол № 1 від 27.08.2019 р.

*Самостійне електронне текстове мережеве видання*

**Методичні** рекомендації до виконання кваліфікаційних робіт  
М54 для студентів спеціальностей 121 "Інженерія програмного забезпечення", 122 "Комп'ютерні науки", 126 "Інформаційні системи та технології" першого (бакалаврського) рівня [Електронний ресурс] / уклад. Ю. Е. Парфьонов, І. О. Ушакова. – Харків : ХНЕУ ім. С. Кузнеця, 2020. – 62 с.

Викладено питання організації дипломного проектування, наведено вимоги до структури дипломного проєкту, методичні рекомендації до розроблення його структурних елементів.

Рекомендовано для студентів спеціальностей 121 "Інженерія програмного забезпечення", 122 "Комп'ютерні науки", 126 "Інформаційні системи та технології" першого (бакалаврського) рівня.

**УДК 004.415(07.034)**

© Харківський національний економічний  
університет імені Семена Кузнеця, 2020

## Вступ

Дипломне проектування є одним із найважливіших видів самостійної роботи, яка завершує підготовку студентів за програмою бакалавра, а також – основою для проведення державної атестації бакалаврів.

Головним завданням дипломного проектування є підготовка студента до виконання завдань та обов'язків, що передбачені для первинних посад у певному виді економічної діяльності.

Дипломний проєкт бакалавра може бути початковим етапом виконання дипломного проєкту спеціаліста або магістра.

У методичних рекомендаціях викладено загальні вимоги до організації та проведення дипломного проектування, змісту, структури та обсягу дипломних проєктів, організації роботи над проєктом, оформлення та захисту дипломного проєкту.

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

Методичні рекомендації призначені для студентів, що навчаються за спеціальностями 121 "Інженерія програмного забезпечення", 122 "Комп'ютерні науки", 126 "Інформаційні системи та технології".

Вимоги до дипломного проектування повністю відповідають освітньо-професійній програмі за спеціальностями 121 "Інженерія програмного забезпечення", 122 "Комп'ютерні науки", 126 "Інформаційні системи та технології".

# 1. Мета й завдання дипломного проектування

Дипломне проектування – завершальний етап підготовки бакалаврів.

Дипломний проєкт – це кваліфікаційна робота, покликана здійснювати об'єктивний контроль за ступенем сформованості вмінь у випускників вирішувати типові завдання діяльності, які належать до проєктувальної (проєктно-конструкторської) та виконавської (технологічної, технічної) виробничих функцій.

Мета дипломного проектування – узагальнити та систематизувати знання і практичні навички студентів, які здобуті 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 проєктом студенти набувають навички з аналізу науково-технічної, нормативної та довідкової літератури, використання державних стандартів, складання пояснювальної записки до проєкту, практичного застосування знань під час ухвалення конкретних проєктних рішень.

Завданнями дипломного проектування є:

систематизація, закріплення та розширення теоретичних знань і практичних навичок за напрямом підготовки, застосування цих знань та навичок у процесі виконання конкретних завдань дипломного проєкту;

розвиток та закріплення навичок самостійної роботи;

удосконалення вміння користуватися сучасними системами програмування, вирішувати інженерні завдання з проектування інформаційних систем та їхніх елементів, використовуючи сучасні методології, інформаційні технології, проводити комп'ютерне моделювання, а також уміння обробляти і систематизувати результати досліджень, використовуючи комп'ютерну техніку та відповідні інструментальні засоби;

визначення відповідності рівня підготовки випускника вимогам освітньо-кваліфікаційної характеристики бакалавра, його готовності та спроможності до самостійної роботи в умовах ринкової економіки, сучасного виробництва, прогресу науки і техніки.

Виконуючи дипломний проєкт, студент має повною мірою використовувати набуті знання з інформаційних технологій та комп'ютерної техніки, інтелектуальних систем і баз знань, наявні пакети, методи та засоби математичної обробки інформації; поєднувати теоретичні знання з виробничим досвідом, отриманим під час проходження практики; використовувати

досягнення вітчизняної та світової науки і техніки; враховувати техніко-економічні показники функціонування створюваних програмно-інформаційних систем і комплексів; на високому теоретичному і професійному рівні виконувати проєктування обраних технічних рішень; грамотно, повно і водночас лаконічно викладати свої рішення в пояснювальній записці.

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

Дипломний проєкт є самостійною роботою студента. За всі розроблені в ньому проєктні рішення, а також правильність, обґрунтованість розрахунків і належне оформлення його матеріалів несе відповідальність автор.

До дипломного проєктування допускається студент, який пройшов повний курс навчання та склав усі передбачені навчальним планом заліки й екзамени, тобто виконав усі вимоги навчального плану з напямую підготовки.

## **2. Організація виконання дипломного проєкту**

Студенту може бути призначена тема дипломного проєкту з переліку рекомендованих тем. Також йому надається право самостійного вибору теми з урахуванням його схильностей і можливостей найбільш повно застосувати здобуті знання. Якщо тема пропонується студентом, то вона має бути обговорена й погоджена з керівником дипломного проєкту.

Для затвердження обраної теми студент подає заяву на ім'я завідувача кафедри інформаційних систем. Зразок заяви наведено в додатку А.

Після затвердження обраної теми студентові видається завдання на дипломний проєкт (додаток Б).

Безпосереднє керівництво дипломним проєктом покладається на викладачів кафедри, які призначаються завідувачем кафедри інформаційних систем.

Керівник дипломного проєкту видає студенту завдання на проєкт; допомагає студенту в складанні календарного плану; проводить консультації; контролює процес виконання проєкту відповідно до календарного плану; рекомендує студенту науково-технічну літературу і нормативно-довідкові джерела з теми проєкту; перевіряє матеріали роботи; здійснює попереднє заслуховування результатів виконання дипломного проєкту.

Дипломний проєкт студент виконує самостійно. Це вимагає чіткої організації його роботи з моменту вибору теми роботи й до її захисту.

На початковому етапі студент має попередньо ознайомитися з основними публікаціями за темою дипломного проєкту та скласти їхній список.

На основі вивчення літературних джерел, які мають охоплювати як монографії, підручники та навчальні посібники, статті у періодичних виданнях, так і патентні матеріали, науково-технічні звіти, реферативні видання, студент має чітко уявити собі, що зроблено в теоретичному та прикладному аспектах теми дипломного проєкту, а також докладно ознайомитися з аналогічними рішеннями у відповідній галузі. За результатами цієї роботи оформляється аналітичний огляд (порівняльний аналіз), із якого мають логічно випливати вибрані методики досліджень.

Після вивчення літературних джерел студент складає попередній план виконання дипломного проєкту, обговорює його з керівником. У процесі обговорення уточнюються вихідні дані для проєктування й строки, що регламентують роботу студента. Після цього студент складає уточнений план роботи над проєктом, погоджує його з керівником та приступає до проєктування. У процесі виконання дипломного проєкту студент має регулярно відвідувати консультації керівника, подавати йому на перевірку робочі матеріали відповідно до плану-графіка виконання етапів проєкту.

Контроль керівника дипломного проєкту не звільняє студента від повної відповідальності за обґрунтованість ухвалених рішень, дотримання стандартів, термінів виконання календарного плану.

На засіданнях кафедри інформаційних систем регулярно заслуховуються повідомлення керівників дипломних проєктів про хід виконання календарних планів. Студенти, що не дотримуються графіка виконання проєкту або значно відстають у його виконанні, запрошуються для звіту на засідання кафедри.

### **3. Структура, зміст та обсяг дипломного проєкту**

За своїм характером дипломні проєкти підрозділяються на такі типи:  
проєкти дослідницького характеру;  
проєкти практичного характеру.

Незалежно від типу дипломного проєкту до них висуваються певні загальні вимоги. Також кожен тип дипломного проєкту має свої особливості.

## Загальні вимоги до дипломних проєктів

Дипломний проєкт складається з пояснювальної записки та інших обов'язкових матеріалів (схеми, діаграми, графіки залежностей, таблиці, малюнки, лістинги програм тощо), що розробляються відповідно до завдання.

Обсяг пояснювальної записки становить 50 – 70 друкованих сторінок формату А4 (без додатків).

Загальну структуру пояснювальної записки дипломного проєкту та рекомендовану кількість сторінок наведено в табл. 1.

Пояснювальна записка виконується у друкований спосіб на аркушах паперу формату А4. Її текст має бути виконано з використанням гарнітури шрифту Times New Roman (кегель 14), із міжрядковим інтервалом 1,2.

Поля сторінок пояснювальної записки: 25 мм від лівого краю аркуша, 10 мм від правого краю аркуша, по 20 мм від верхнього та нижнього країв аркуша.

Абзацний відступ має бути однаковим упродовж усього тексту та дорівнювати 1,25 см.

Вирівнювання основного тексту проводиться "за шириною".

Докладні вимоги до оформлення пояснювальної записки наведено у відповідних методичних рекомендаціях [48].

Таблиця 1

### Структура пояснювальної записки дипломного проєкту

Структурні елементи пояснювальної записки	Кількість сторінок
Титульний аркуш	1
Завдання на дипломний проєкт	2
Реферат	1 – 2
Зміст	1
Перелік умовних скорочень	1
Вступ	1 – 2
Розділ 1	9 – 12
Розділ 2	14 – 19
Розділ 3	19 – 25
Висновки	1 – 2
Список використаних джерел	2 – 3
Додатки	

## **Особливості дипломних проєктів дослідницького характеру**

Проєкти дослідницького характеру спрямовані на розроблення нових або застосування наявних математичних моделей, методів чи алгоритмів для дослідження інформаційних процесів у галузі інформаційних технологій.

Вони мають містити елементи теоретичних або експериментальних досліджень.

У змістовній частині пояснювальної записки проєкту дослідницького характеру, зазвичай, мають знайти відображення такі питання:

1. Огляд за літературними джерелами стану досліджень за тематикою роботи, виявлення теоретичних передумов та можливих напрямів вирішення завдань дослідження.

2. Обґрунтування використання основних теоретичних закономірностей та співвідношень.

3. Опис методики досліджень.

4. Вирішення завдань дослідження на ПК (складання та налагодження програм для вирішення, отримання результатів та їхній аналіз).

5. Узагальнення результатів досліджень.

6. Загальні висновки з роботи з оцінкою застосування результатів досліджень.

Структура змістовної частини дипломних проєктів дослідницького характеру має бути погоджена з керівником відповідного проєкту. Приклад змістовної частини дипломного проєкту дослідницького характеру наведено в додатку В.

## **Особливості дипломних проєктів практичного характеру**

Проєкти практичного характеру спрямовані на аналіз, моделювання, прогнозування інформаційних процесів або керування такими процесами в економіці (як приклад, можна розглядати окреме підприємство, організацію, галузь чи географічний регіон). Вони передбачають вибір конкретної предметної області для аналізу й дослідження. Виконання таких проєктів передбачає аналіз предметної області на основі вивчення спеціальної літератури та ознайомлення з інформаційними процесами безпосередньо на підприємствах та в організаціях.

Змістовна частина пояснювальної записки такого проєкту має містити:

1. Опис напрямів діяльності підприємства, організаційної структури підприємства, структурного підрозділу.

2. Аналіз предметної області, наявних бізнес-проблем об'єкта проєктування, опис бізнес-процесів.



3. Аналіз наявних програмних продуктів, що реалізують функції предметної області.

4. Специфікацію вимог до програмного забезпечення.

5. Опис проєктних і технічних рішень стосовно проєктування бази даних, архітектури програмної системи, її тестування та розгортання.

Структуру змістовної частини дипломних проєктів практичного характеру, якої необхідно дотримуватися, наведено в додатку Г.

## **4. Методичні рекомендації до розроблення структурних елементів пояснювальної записки**

Загальними вимогами до тексту пояснювальної записки є логічна послідовність викладення матеріалу, чіткість і конкретність викладення теоретичних і практичних результатів роботи, сутності постановки завдання та мети роботи, методів дослідження, ухвалених рішень, доведеність висновків і обґрунтованість рекомендацій. У тексті пояснювальної записки необхідно дотримуватися єдиної термінології. Вона не має бути перевантажена малоінформативним матеріалом, описом загальновідомих даних, виведенням формул тощо. Необхідно посилатися на джерела інформації. У тексті пояснювальної записки має бути наведений використаний математичний апарат та результати виконаних розрахунків за допомогою ПК.

Текст пояснювальної записки не слід викладати від першої особи, краще використовувати безособову форму (наприклад, "обчислюється", "знаходимо") за всім текстом у визначеному відмінку й часі.

Під час викладення матеріалу не слід використовувати:

розмовні звороти;

жаргонні слова та звороти;

різні терміни для позначення одного поняття;

іншомовні слова та терміни за наявності в українській мові рівнозначних слів і термінів;

скорочення слів і словосполучень, крім встановлених правилами орфографії та нормативними документами.

Першою сторінкою пояснювальної записки є **титульний аркуш**. Він містить такі дані:

відомості про виконавця роботи – юридичну особу (організацію) або фізичну особу;

повна назва документа;

підписи відповідальних осіб;

рік складання пояснювальної записки;

Зразок титульного аркуша дипломного проєкту наведено в додатку Д.

**Реферат** – це короткий виклад змісту пояснювальної записки, що містить основні фактичні відомості та висновки, необхідні для початкового ознайомлення з нею. *Реферат виконується українською та англійською мовами.*

Реферат має бути стислим, інформативним і містити відомості, які дозволяють ухвалити рішення про доцільність читання пояснювальної записки.

Реферат має містити:

відомості про обсяг пояснювальної записки, кількість ілюстрацій, таблиць, додатків, кількість джерел згідно з переліком посилань (усі відомості наводять, охоплюючи дані додатків);

текст реферату;

перелік ключових слів.

Текст реферату має відбивати подану в пояснювальній записці інформацію в такій послідовності:

об'єкт дослідження або розроблення;

мета роботи;

методи дослідження та апаратура;

результати та їхня новизна;

основні технологічні й техніко-експлуатаційні характеристики та показники;

взаємозв'язок з іншими роботами;

рекомендації щодо використання результатів роботи;

галузь застосування;

значущість роботи та висновки;

прогнози припущення про розвиток об'єкта дослідження або розроблення.

Реферат належить виконувати обсягом не більш як 500 слів і, бажано, щоб він уміщувався на одній сторінці формату А4.

Ключові слова призначені для розкриття сутності проекту та для розповсюдження інформації про розробку. Їх розміщують після тексту реферату. Перелік ключових слів містить від 5 до 15 слів (словосполучень), надрукованих великими літерами в називному відмінку в рядок через коми.

Приклади рефератів наведено в додатку Е.

**Зміст** містить: вступ; назви всіх розділів, підрозділів та пунктів основної частини пояснювальної записки; висновки; перелік посилань; назви додатків і номери сторінок, які містять початок матеріалу.

Приклад змісту пояснювальної записки наведено в додатку Ж.

Якщо в пояснювальній записці використовуються маловідомі скорочення, нові символи, позначення і таке інше, то в ній має бути **перелік умовних скорочень**, який подається у вигляді окремого списку, що розміщують перед вступом. Незважаючи на це, за першої появи цих елементів у тексті документу надають їхню розшифровку.

Якщо в роботі спеціальні терміни, скорочення, символи, позначення повторюються менше трьох разів, перелік не складають, а їхню розшифровку наводять у тексті під час першого згадування.

Приклад переліку умовних скорочень наведено в [48].

**Вступ** – це віддзеркалення роботи, тому слід ретельно його опрацювати. Краще формувати вступ після виконання основного тексту пояснювальної записки.

*У вступі до дипломного проекту практичного характеру* необхідно ідентифікувати та сформулювати проблему бізнесу, яка виникла на підприємстві, обґрунтувати актуальність теми проекту для вирішення цієї проблеми на основі розроблюваного модуля або системи. Коротко охарактеризувати функціональність модуля або системи. Необхідно охарактеризувати технічну та програмну платформи розроблення автоматизованого модуля. Потрібно сформулювати мету й завдання проекту, визначити об'єкт і предметну область проектування. Також необхідно навести інформацію щодо засобів проектування, які використовувалися в дипломному проекті, та можливих галузей застосування результатів, отриманих у дипломному проекті.

*У вступі до дипломного проекту дослідницького характеру* викладається:

оцінювання сучасного стану вирішення проблеми;  
актуальність роботи;

мета та завдання роботи;  
об'єкт та предмет дослідження;  
методи дослідження;  
наукове та практичне значення роботи;  
відомості щодо публікацій за темою роботи.

**Висновки** до роботи – це резюме за результатами всієї роботи. Ця частина має особливу важливість, оскільки тут мають бути наведені підсумкові результати роботи.

У пояснювальній записці мають міститися висновки за кожним етапом виконаного проєкту та по роботі загалом, які необхідно співвіднести з метою і завданням на дипломне проєктування.

Необхідно зазначити практичну цінність результатів роботи, дати рекомендації для подальшого вдосконалення об'єкта проєктування. Зазначаючи практичну цінність отриманих результатів, важливо окреслити ступінь їхньої 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 проєкт впроваджений на підприємстві, то до дипломного проєкту додається довідка або акт про впровадження.

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

**Список використаних джерел** має містити відомості про літературні джерела, використані в процесі розроблення проєкту.

Список літератури – це реєстр використаних джерел за темою проєкту в якнайширшому значенні. Тому не слід обмежуватися лише цитованою літературою. У список варто вміщувати всі матеріали, які були прочитані, переглянуті, проаналізовані в процесі роботи над дипломним проєктом і стосуються його теми. Бажано подавати джерела якомога повніше, пам'ятаючи, що бібліографічний список до проєкту – це підсумок вивчення проблеми і передумова подальших наукових досліджень.

Список використаних джерел оформлюється згідно з ДСТУ 8302:2015 "Бібліографічне посилання. Загальні вимоги та правила складання" [11].

Список використаних джерел має містити не менше 30 джерел. Він подається мовою оригіналу, розміщується в алфавітному порядку

прізвищ перших авторів або назв та нумерується в порядку їхнього зростання. Нумерація безперервна.

Роботи одного автора розташовані за алфавітом назв або в хронології їхнього написання. Алфавітний список розташований за алфавітом у такій послідовності:

література зведеного кириличного алфавіту, для джерел на мовах із кириличною графікою (російська, українська, болгарська та ін.);

література в латинському алфавіті;

електронні ресурси в тій же послідовності що й друковані видання (спочатку кирилицею, а потім латиницею).

Список використаних джерел обов'язково має містити прізвище та ініціали автора, повну назву джерела, місто видавництва, видавництво та рік видання, кількість сторінок чи посилання на сторінки тощо. Загальний обсяг книги в сторінках вказується, якщо посилання на неї проводиться повністю, сторінки (від ... до) відмічаються, якщо посилання відносяться до окремої частини літературного джерела.

У **додатках** вміщують матеріал, який є необхідним для повноти пояснювальної записки, але не може бути послідовно розміщений в її основній частині через великий обсяг або з інших причин.

Ілюстрації (діаграми бізнес-процесів, сценарії діалогів та ін.), таблиці, проміжні математичні докази, формули та розрахунки, текст допоміжного характеру тощо можуть бути оформлені у вигляді додатків.

#### **4.1. Рекомендації щодо розроблення розділів пояснювальної записки дипломного проєкту дослідницького характеру**

В основній частині роботи розглядається методика й техніка дослідження, узагальнюються його результати. Зміст розділів основної частини має відповідати темі дипломного проєкту та повністю її розкривати. Ця частина роботи має показати вміння студента стисло, логічно й аргументовано викладати матеріал.

Основна частина роботи має містити інформацію, що відбиває мету, завдання, сутність, методику й основні результати виконаних досліджень:

**Розділ 1.** Формулювання наукової задачі, аналіз стану її вирішення за матеріалами вітчизняних і закордонних джерел.

**Розділ 2.** Аналіз методів, моделей, методик, наявного методологічного та методичного забезпечення щодо завдання, яке вирішується. Обґрунтування вибору методу(ів), методики(ик), показників ефективності та інструментів дослідження.

**Розділ 3.** Виконання практичних досліджень, аналіз отриманих результатів із графічним та табличним відображенням одержаних переваг (поліпшень) основних характеристик (параметрів). Результати використання розробленого програмного продукту або наявних інструментальних засобів для отримання та аналізу результатів за обраною моделлю. Аналіз практичної цінності отриманих результатів зі вказівкою відповідної галузі щодо їхнього впровадження та напрямів (перспектив) подальшого розвитку.

## **4.2. Рекомендації щодо розроблення розділів пояснювальної записки дипломного проєкту практичного характеру**

### **РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ <НАЗВА ПРЕДМЕТНОЇ ОБЛАСТІ>**

Метою розділу 1 є проведення детального аналізу проблеми, яка виникла на об'єкті управління (підприємстві) при веденні бізнесу, вибір шляхів її вирішення.

У підрозділі 1.1. **Коротка характеристика об'єкта управління <назва об'єкта управління>** треба:

коротко описати напрями діяльності об'єкта управління (підприємства, організації);

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

розробити схему організаційної структури підрозділу (підрозділів), пов'язаних із визначеними бізнес-процесами.

У підрозділі 1.2. **Опис предметної області <назва предметної області>** необхідно:

визначити склад функцій, що входять до бізнес-процесу;

розробити схему управління бізнес-процесом та описати його (табл. 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 вказати 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 транзакційну складову бізнес-процесу, яка забезпечує збирання, 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 має 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 1.3. Огляд і аналіз наявних аналогів, що реалізують функції предметної області, необхідно:**

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

навести в таблиці порівняльну характеристику програмних продуктів за такими характеристиками (табл. 3):

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

Таблиця 3

### Порівняльна характеристика програмних продуктів

Фірма-розробник	<Назва фірми-розробника 1>	<Назва фірми-розробника 2>	...
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 їхніх рішень під час розроблення проєктних рішень проєкту.

## **РОЗДІЛ 2. СПЕЦИФІКАЦІЯ ВИМОГ ДО МОДУЛЯ (СИСТЕМИ)**

Метою розділу 2 є розроблення і детальна специфікація вимог до модуля (системи), що розробляється.

### **Підрозділ 2.1. 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 лягають в основу об'єктної моделі (для об'єктно-орієнтованих застосувань) і моделі даних (для генерації схеми бази даних).



Глосарій необхідно подати у вигляді табл. 4.

Таблиця 4

### Глосарій

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

**Підрозділ 2.2. Розроблення варіантів використання** містить:

діаграму варіантів використання;

специфікацію варіантів використання;

розкадровку варіантів використання;

специфікацію функціональних та нефункціональних вимог.

**Пункт 2.2.1. Діаграма варіантів використання** містить діаграму варіантів використання, виконану за допомогою певного CASE-інструмента.

Діаграма варіантів використання відображає функціональність, яка буде реалізована в програмному продукті. 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 під час специфікації варіанта використання серед усієї функціональності системи виділяють лише ту функціональність, яка:

**корисна** конкретному кінцевому користувачеві;

дозволяє отримувати користувачеві **конкретні закінчені результати**.

Перш ніж приступити власне 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 щодо 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 їхніх функціональних обов'язків, розмежування доступу, способів використання інформаційної системи.

### **Пункт 2.2.2. Специфікація варіантів використання**

Кожен варіант використання повинен мати опис. У дипломному проєкті необхідно навести опис варіантів використання, що реалізують основну функціональність (зазвичай, крім ведення довідників):

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

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

#### **Дійові особи:**

основна дійова особа – це ім'я ролі основного актора або його опис; учасники – перелік інших акторів-учасників варіанта використання.

**Передумова** – варіант використання, який має бути обов'язково виконаний, щоб можна було виконати цей варіант. Передумова описує стан, у якому система має перебувати до початку виконання варіанта.

**Тригер** – подія предметної області, що викликає використання прецеденту (варіанта використання). Іноді тригер передує першому кроку варіанта використання, а іноді сам є першим кроком.

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

**Постумова** – варіант використання, який обов'язково має бути виконаний після виконання цього варіанта; це стан, у якому система має перебувати після закінчення виконання варіанта; це те, що гарантується акторам-учасникам, незалежно від успіху виконання цього варіанта. 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.

Опис варіантів використання необхідно оформити у вигляді таблиці (табл. 5).

## Варіант використання &lt;Назва&gt;

Контекст використання	
Дійові особи	
Передумова	
Тригер	
Сценарій	
Постумова	

**Підрозділ 2.3. Специфікація функціональних та нефункціональних вимог**

Специфікацію функціональних вимог навести в таблиці (табл. 6).

**Пріоритет** заповнюється аналітиком, показує пріоритет реалізації вимоги для клієнта. Використовується під час управління проектом і визначає пріоритет розроблення. Можливі значення: обов'язкове, рекомендоване, опційне (за вибором).

**Специфікація функціональних вимог**

Ідентифікатор вимоги	Назва вимоги (варіанта використання)	Атрибути вимог		
		Пріоритет	Трудність	Контакт/ Виконавець

**Трудність** заповнюється менеджером проекту, показує рівень трудовитрат, пов'язаних із реалізацією вимоги. Використовується під час управління проектом і визначає пріоритет розроблення. Трудність виконання вимоги може виражатися у вигляді трудомісткості та вказувати кількість людино-днів, потрібних для її реалізації або у вигляді значень шкали: висока, середня, низька.

**Контакт** заповнюється аналітиком, ідентифікує людину, яка може надати необхідну інформацію про вимогу (ім'я контакту). Використовується

для гарантії, що розробники можуть отримати інформацію, необхідну їм для реалізації вимоги. Замість **Контакту** часто використовується атрибут **Виконавець**.

Якщо в проєкті як контакт виступає одна людина – цю колонку можна опустити, але треба ідентифікувати контакт один раз у тексті цього пункту.

Специфікацію нефункціональних вимог навести в таблиці (табл. 7).

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 відповідності до загальних стандартів застосовності, наприклад, до стандартів інтерфейсу користувача IBM або стандартів графічного інтерфейсу користувача Microsoft для Windows.

Таблиця 7

## Специфікація нефункціональних вимог

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

### 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 у днях, місяцях або роках.

*Середнє напрацювання до ремонту* показує, як довго системі дозволяють працювати до того, коли має 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 кількості помилок на тисячу рядків коду або кількості помилок у функціональній одиниці.

### **3. Робочі характеристики.**

Мають бути виділені характеристики продуктивності системи. Долучіть конкретні характеристики швидкодії. Де можливо, потрібно зробити посилання на пов'язані варіанти використання за ім'ям.

*Швидкодія для транзакції* (середнє значення, максимальне).

*Продуктивність* (наприклад, кількість транзакцій за секунду).

*Місткість* (наприклад, кількість замовників або транзакцій, які може розміщувати система).

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

*Використання ресурсів:* пам'яті, дискового простору, комунікацій тощо.

### **4. Експлуатаційна придатність.**

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

### **5. Проєктні обмеження.**

Мають містити всі проєктні обмеження до формованої системи. Проєктні обмеження становлять рішення, які були сформульовані як обов'язкові та мають твердо витримуватися. Прикладами можуть бути мови програмування, вимоги до технології програмування, обов'язкове використання інструментальних засобів розроблення, архітектурні й конструктивні обмеження купованих компонентів, бібліотек класів і т. д.

### **6. Вимоги до документації, призначеної для користувача, і до системи допомоги.**

Описують вимоги, якщо вони є, до інтерактивної документації користувача, до системи довідки, до попереджувальних повідомлень тощо.

### **7. Куповані компоненти.**

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

### **8. Інтерфейси.**

Визначають інтерфейси, які мають бути підтримані застосунком. Опис має містити адекватну специфіку, протоколи, порти і логічні адреси

тощо так, щоб програмне забезпечення могло бути розроблене і перевірене на відповідність вимогам інтерфейсів.

### **8.1. Інтерфейси користувача.**

Описуються інтерфейси користувача, які мають бути реалізовані програмним забезпеченням.

### **8.2. Апаратні інтерфейси.**

Визначаються всі апаратні інтерфейси, які мають бути підтримані програмним забезпеченням, охоплюючи логічну структуру, фізичні адреси, очікувану поведінку тощо.

### **8.3. Програмні інтерфейси.**

Описуються програмні інтерфейси з іншими компонентами програмної системи. Це можуть бути куповані компоненти, багато разів використовувані 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 має 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 мають виконуватися програмним забезпеченням.

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

Описуються всі необхідні юридичні застереження, гарантії, оголошення про авторське право, право спадкоємства, торговельні 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 та регулювальні стандарти, стандарти якості, промислові стандарти щодо застосовності, здатності до взаємодії, інтернаціоналізації, відповідності операційній системі тощо.

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

**У підрозділі 2.4. Проєктування інтерфейсу користувача** необхідно розробити графічний інтерфейс користувача застосунку у вигляді вайрфрейму або мокапу, які використовуються для отримання схвалення зацікавлених осіб, щодо запропонованої концепції.

**Інтерфейс користувача** (UI – англ. User interface) – інтерфейс, що забезпечує передачу інформації між користувачем-людиною і програмно-апаратними компонентами комп'ютерної системи.

Метою проєктування інтерфейсу є отримання ранньої реакції користувачів на запроповану концепцію системи. Як інструментальні засоби використовуються Marvel, Pencil Project, Use Your Interface, UX Myths, Mobile Patterns, Good UI, Microsoft Visio, Microsoft PowerPoint тощо.

**Вайрфрейм** (англ. wireframe, каркас) – структурна схема розташування елементів інтерфейсу з прикладом контенту (текстом, ілюстраціями, таблицями) і виділенням акцентів, яка частково відображає роботу з продуктом. Вайрфрейм – це образ дизайну низької точності. Він має чітко показувати:

- основні групи контенту (Що?);

- структуру інформації (Де?);

- опис і базову візуалізацію взаємодії між інтерфейсом і користувачем (Як?).

Вайрфрейм не містить графічного оформлення (візуального дизайну). Елементи інтерфейсу подають у спрощеному вигляді, наприклад, використовуючи "наповнювачі" – прямокутники, пересічені лініями хрест нахрест для зображень. Тому вайрфрейми зазвичай називають даними низької точності (lo-fi). Схема доповнюється текстовим документом із описом логіки роботи продукту і взаємодії користувача з ним.

**Мокап** (англ. mockup, макет) – це середньо- або високодеталізоване статичне уявлення дизайну. Передає структуру інформації, візуалізує зміст і демонструє основні функціональні можливості у вигляді статичних зображень. Дозволяє зрозуміти, як буде виглядати кінцевий продукт.

Мокап – неклікабельний, але більш візуально оформлений макет (практично вже дизайн). Він має:

- показувати інформаційну структуру;

- візуалізувати контент;

- демонструвати базову функціональність у статиці;

- дати можливість оцінити візуальну сторону проєкту.

## **РОЗДІЛ 3. ПРОЄКТНІ ТА ТЕХНІЧНІ РІШЕННЯ**

Розділ 3 призначений для проектування та розроблення програмної системи, що призначена для автоматизації бізнес-задач із використанням сучасних інформаційних технологій.

### **Підрозділ 3.1. Математична (логічна) постановка**

У цьому підрозділі виконується формалізація задач, у результаті якої визначається склад змінних, констант і їхня класифікація, види обмежень на змінні та математичні залежності між змінними. Встановлюється клас, до якого належать розв'язувані задачі, і наводиться порівняльний аналіз методів розв'язання для вибору найбільш ефективного методу. Наводиться обґрунтування прийнятих припущень і передумов під час формалізації та вибору методу розв'язання. Визначається загальна послідовність розв'язання задач. У цьому ж підрозділі наводяться результати теоретичних досліджень, опис розроблених алгоритмів, аналіз їхньої ефективності тощо.

Якщо задача є оптимізаційною або задачею прогнозування, то потрібно обґрунтувати вибір економіко-математичної моделі вирішення задачі. Економіко-математична модель охоплює цільову функцію для обраного критерію ефективності та систему обмежень. Необхідно описати послідовність розрахунку результатних показників відповідно до логіки реалізації економіко-математичної моделі.

Якщо задача розв'язується на основі алгоритму прямого лічення, необхідно навести розрахункові формули у вигляді математичної залежності вихідних показників, що розраховуються, від вхідних.

Якщо задача не має математичного формулювання її розв'язання, необхідно навести опис логіки послідовних дій у вигляді виконуваних функцій обробки інформації з задачі (добору, порівняння).

Математична і логічна моделі розв'язання задачі мають бути описані з достатнім ступенем деталізації, щоб за ними можна було написати програму розрахунку показників вихідних повідомлень.

### **Підрозділ 3.2. Проектування структури бази даних**

У цьому підрозділі необхідно розробити інформаційне забезпечення для розроблюваної системи, виконати логічне та фізичне проектування БД, проявити творчий підхід до вибору технічного та програмного забезпечення інформаційної системи, СКБД, CASE-засобів проектування БД.



Початковими даними для проєктування бази даних є:  
опис функціональної схеми підприємства, для якого виконується проєкт;  
виявлення місця та функцій конкретного підрозділу в структурі усєї організації (підприємства);

опис технології обробки інформації у рамках цієї предметної області конкретного підрозділу з зазначенням вхідної та вихідної інформації, вирішуваних завдань, функцій і регламенту роботи виконавців обробки інформації, періодичності виконання функцій із обробки інформації;

аналіз наявних засобів автоматизації обробки інформації у рамках цієї предметної області (апаратне та програмне забезпечення, СКБД);

обґрунтування необхідності та можливості розроблення модуля автоматизації обробки інформації і бази даних цього модуля як складового елемента загальної інфраструктури даних організації;

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

Логічна модель (logical data model) бази даних визначає: набір підтримуваних типів структур даних; набір допустимих операцій над підтримуваними структурами даних; набір загальних правил цілісності даних, що явно або неявно визначають коректний стан бази даних або їхні зміни;

Виділяють такі моделі баз даних:

ієрархічна (hierarchical model);

мережева (network model);

реляційна (RDBMS, relational DBMS) – горизонтальне і вертикальне зберігання (row / column store);

постреляційні моделі, NoSQL (Not only SQL) бази даних: об'єктно-орієнтовані (OODBMS, object-oriented DBMS); сховища ключів і значень (KV-store, key-value store, KVP, key-value pairs); документо-орієнтовані (document-oriented model); графові бази даних (graph databases); стовпцеві бази даних;

багатостороннє зберігання (polyglot persistence).

БД складається з двох частин: транзакційної та аналітичної. Характерними рисами транзакційної частини є:

- 1) реляційна структура (переважно);
- 2) можливість накопичення значних обсягів 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 розроблення аналітичної частини БД.

Аналітична частина БД використовується в процесі виконання оперативного аналізу інформації і розроблення моделей для систем підтримки прийняття рішень. Характерними рисами аналітичної частини БД є:

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

У процесі розроблення транзакційної частини БД можна використати структурне або об'єктно-орієнтоване моделювання, застосовуючи відповідні CASE-інструменти: ErWin Data Modeler, IBM Rational Software тощо.

Під час побудови сховища даних необхідно обґрунтувати вибір моделі зберігання (кубічна модель MOLAP або ROLAP: "зірка" або "сніжинка").

Якщо тема дипломного проєкту пов'язана зі створенням складних транзакційних баз даних для інформаційної системи, рекомендується в цей підрозділ долучати проєктування концептуальної, логічної та фізичної моделей даних. В інших випадках можна обмежитися проєктуванням логічної і фізичної моделей даних.

Якщо в дипломному проєкті використовуються постреляційні бази даних, зміст цього підрозділу погоджується з керівником дипломного проєкту.

### **Пункт 3.2.1. Концептуальне інфологічне проєктування**

У цьому пункті виконується побудова моделі даних, незалежної від СКБД, яка охоплює створення словника даних і глобальної інфологічної моделі даних.

**Словник даних.** На основі аналізу вхідних та вихідних документів 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 вигляду, зручного для проєктування. Для цього складають словник даних (табл. 8).

Таблиця 8

### **Словник даних**

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

Елементи даних словника наводяться в алфавітному порядку за графою "Найменування елемента" з метою подальшого вилучення омонімів та дублювальних елементів. Якщо словник вміщує багато елементів, його виносять у додаток. У полі "Призначення елемента" необхідно вказати: елемент збереження є фактичним чи обчислюваним.

Під час проєктування **глобальної інфологічної моделі даних** необхідно здійснити виявлення еквівалентних сутностей та їхнє злиття, виявлення категорій і синтез узагальнювальних сутностей, виявлення й усунення дублювання атрибутів і зв'язків. Будується графічне подання глобальної моделі у вигляді ERD (нотація IDEF1X), діаграми класів або інших нотацій.

Для всіх сутностей розроблюваної системи слід навести специфікації обмежень цілісності та операційних правил, а саме:

- 1) обмеження атрибутів сутностей (табл. 9);
- 2) обмеження кортежів;
- 3) обмеження унікальності;
- 4) динамічні обмеження;
- 5) інші обмеження;
- 6) операційні правила;
- 7) правила посилальної цілісності.

Таблиця 9

### Обмеження атрибутів сутностей

№ п/п	Ім'я атрибута або агрегату	Межі / допустимі значення	Структура (формат)	Умова	Значення за замовчуванням

### Пункт 3.2.2. Проєктування логічної моделі даних

Проєктування логічної моделі даних містить: обґрунтування вибору СКБД, графічне подання логічної моделі у вигляді ERD (в нотації IDEF1X), діаграми класів тощо.

Обґрунтування властивостей моделі бази даних містить: опис засобів, інструментів, методів, які застосовувалися для забезпечення таких властивостей бази даних: функціональна повнота; мінімальна надмірність; цілісність бази даних (таблична, посилальна цілісність, забезпечувана

ключами і тригерами тощо); узгодженість; актуальність; безпека; відновлюваність; логічна та фізична незалежність; ефективність.

### **Пункт 3.2.3. Проєктування фізичної моделі даних**

Проєктування фізичної моделі даних охоплює опис розроблення моделі даних на фізичному рівні як результат генерації команд SQL-скрипту щодо створення основних об'єктів зберігання в БД, таких як: таблиці, подання, ключі тощо.

### **Підрозділ 3.3. Проєктування програмного забезпечення**

У цьому підрозділі пояснювальної записки має бути:

1. *Для об'єктно-орієнтованих програмних систем* – UML-діаграма класів (Class Diagram), що реалізують основну бізнес-логіку програмної системи, та її короткий опис, у якому щонайменше необхідно навести призначення кожного класу. *Для інших програмних систем* – UML-діаграма діяльності (Activity Diagram), яка відбиває основну бізнес-логіку програмної системи, та її короткий опис.

2. UML-діаграма станів, у яких можуть знаходитися елементи графічного інтерфейсу користувача, та її опис.

3. Посилання на лістинг програми, де міститься вихідний код, який відповідає UML-діаграмам, наведеним у пункті 1. Сам лістинг програми має знаходитися в одному з додатків до пояснювальної записки.

Далі знаходяться методичні рекомендації щодо розроблення архітектури програмної системи.

### **Рекомендації з розроблення UML-діаграми класів (Class Diagram), що реалізують основну бізнес-логіку програмної системи.**

Програмна система означає програмне забезпечення, що розробляється. Це може бути крупна колекція з безлічі компонентів програмного забезпечення, одна програма або частина програми.

Для уявлення статичної структури моделі програмної системи призначена UML-діаграма класів. Вона може відбивати, зокрема, 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 слугує для позначення безлічі об'єктів, які мають однакову структуру, поводження 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 має 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 проєктованої 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 із будь-якого іншого класу пакета, у якому визначена діаграма.
2. Захищений (protected). Атрибут із цією областю видимості недоступний для всіх класів, за винятком підкласів цього класу.
3. Закритий (private). Атрибут із цією областю видимості недоступний для всіх інших класів без винятку.

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

Кратність атрибута характеризується загальною кількістю конкретних атрибутів певного 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 цієї 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 має бути унікальним у межах цього класу.

Крім внутрішнього устрою або структури класів на відповідній діаграмі вказуються різні відносини між класами. Водночас 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.

**Рекомендації з розроблення UML-діаграми діяльності (Activity Diagram), яка відбиває основну бізнес-логіку програмної системи.**

Діаграми діяльності використовуються, щоб показати потік управління в системі та кроки, що виконуються в процесі виконання варіанта використання. Використовуючи діаграми діяльності, можна моделювати послідовні та паралельні дії. Діаграма діяльності фокусується на умовах та послідовності виконання деякого потоку.

Діаграма діяльності зображує потік управління від стартової точки до кінцевої точки, показує різні шляхи прийняття рішень, які існують під час виконання діяльності. Вона використовується для моделювання робочого процесу з урахуванням умов, обмежень, послідовних та одночасних дій.

Щоб розробити діаграму діяльності, необхідно визначити:

1. Початковий стан та кінцевий стан.
2. Проміжні дії, необхідні для досягнення кінцевого стану із початкового стану.
3. Умови або обмеження, які спонукають систему змінити потік управління.

**Рекомендації з розроблення UML-діаграми станів (State Diagram) 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. Здебільшого, 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 під час зміни стану.

Для формального опису СУП доцільно використовувати UML-діаграми станів.

Діаграма станів зв'язує події й стани. У ході виникнення 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 процесі розроблення інтерфейсу користувача необхідно керуватися методичними рекомендаціями, викладеними далі.

### **Рекомендації щодо розроблення інтерфейсу користувача.**

Користувальницький інтерфейс є своєрідним комунікаційним каналом, за яким здійснюється взаємодія користувача й комп'ютера.

Щоб створити ефективний інтерфейс, потрібно розуміти, які завдання будуть вирішувати користувачі за допомогою цієї програми і які вимоги до інтерфейсу можуть виникнути в користувачів.

Загальні принципи проєктування інтерфейсу користувача:

1. Програма має допомагати виконувати завдання.

Це значить, що інтерфейс має бути легким для освоєння й не створювати перед користувачем перешкоду, яку він повинен буде подолати, щоб розпочати роботу.

2. Під час роботи із програмою користувач не має відчувати дискомфорт.

Для реалізації цього принципу необхідно:

забезпечити перевірку результатів якомога більшої кількості "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 вимкнення виведення інформаційних повідомлень;

добре продумувати зміст повідомлень, що виводяться користувачеві.

Широко відомі евристичні правила авторитетного американського фахівця в галузі проєктування інтерфейсів Якоба Нільсена [52]:

1. Видимість стану системи.
2. Відповідність між системою й реальним світом.
3. Управління користувачами та свобода їхніх дій.
4. Несуперечність і стандарти.
5. Запобігання помилок.
6. Впізнання, а не згадування.
7. Гнучкість і ефективність використання.
8. Естетичний і мінімалістський дизайн.
9. Допомога користувачам розпізнавати й виправляти помилки.
10. Довідка й документація.

### **Проєктування 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 – це форми, призначені для введення даних. Під час їхнього розроблення основну увагу варто приділити швидкості їхнього використання. Основне правило – якщо користувач збирається ввести в базу даних велику кількість записів, то він не має 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.

Ще одна важлива частина розроблення форм – створення змістовних і ефективних меню. От деякі важливі рекомендації:

1. Дотримуйтеся стандартних угод про розташування пунктів меню, прийнятих в операційній системі.
2. Групуйте пункти меню в логічному порядку й за змістом.
3. Для угруповання 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. Поміщайте на панель інструментів часто використовувані команди меню.

Далі наведені деякі рекомендації із проєктування Web-інтерфейсу користувача:

1. Мінімізуйте зусилля, які необхідно зробити користувачеві для ухвалення 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. Під час створення декількох сторінок поєднайте 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 однаково з візуальної точки зору.
5. Варто бути дуже обережним щодо 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 внесення в продукт нових функцій, спочатку необхідно вирішити, як буде розширюватися навігація для ефективного внесення нових екранів.

### **Підрозділ 3.4. Тестування програмної системи**

Тестування програмного продукту – це процес, що містить всі активності життєвого циклу, як динамічні, так і статичні, які стосуються планування, підготовки та оцінювання програмного продукту з метою визначити,

що вони відповідають описаним вимогам, показати, що вони підходять для заявлених цілей і для визначення дефектів [16]. Під дефектом розуміється вада в компоненті або системі, яка може привести компонент або систему до неможливості виконати потрібну опцію, наприклад, неправильний оператор або визначення даних. Дефект, виявлений під час виконання, може привести до відмов компонента або системи.

Завдання тестування – визначення умов, за яких проявляються дефекти системи, і протоколювання цих умов.

Мета застосування процедури тестування програмного коду – мінімізація кількості дефектів у кінцевому продукті.

Тестування само собою не може гарантувати повної відсутності дефектів у програмному коді системи. Однак, у сполученні з процесами верифікації й валідації, спрямованими на усунення суперечливості й неповноти проєктної документації (зокрема, вимог на систему), грамотно організоване тестування дає гарантію того, що система задовольняє вимоги і поводить відповідно до них у всіх передбачених ситуаціях.

#### **Модульне тестування.**

У ході модульного тестування кожен модуль тестується як на відповідність вимогам, так і на відсутність проблемних ділянок програмного коду, які можуть викликати відмови й збої в роботі системи.

#### **Інтеграційне тестування.**

Окремі модулі рідко функціонують самі по собі, тому наступне завдання після тестування окремих модулів – це тестування коректності взаємодії декількох модулів, об'єднаних у єдине ціле. Таке тестування називають інтеграційним.

#### **Системне тестування.**

Після завершення інтеграційного тестування всі модулі системи є погодженими за інтерфейсами і функціональністю. Починаючи із цього моменту, можна переходити до системного тестування, тобто тестування поведінки системи загалом як єдиного об'єкта.

Вхідною інформацією для проведення системного тестування є два класи вимог: функціональні й нефункціональні.

Функціональні вимоги явно описують, що система має робити і які перетворення вхідних даних виконувати.

Нефункціональні вимоги визначають властивості системи, прямо не пов'язані з її функціональністю. Прикладом таких властивостей може

слугувати максимальний час відгуку системи на запит користувача, мінімальний час безперебійної роботи системи й ін.

Системне тестування проводиться в кілька етапів, на кожному з яких використовується один із видів системного тестування, наприклад:

### **1. Функціональне тестування.**

Цей вид системного тестування призначений для підтвердження того, що вся система загалом поводить ся відповідно до очікувань користувача, формалізованих у вигляді системних вимог. У ході функціонального тестування перевіряються всі функції системи з погляду її користувачів (як людей, так і інших програмних систем). Також необхідно перевірити функціональну повноту користувальницького інтерфейсу й коректність виведення інформації.

За функціонального тестування система розглядається як "чорний ящик".

Критерії повноти функціонального тестування:

1. Усі функціональні вимоги мають бути протестовані.
2. Усі класи припустимих вхідних даних мають коректно оброблятися системою.
3. Усі класи неприпустимих вхідних даних мають бути відкинуті системою, водночас не має порушуватися стабільність її роботи.
4. У тестових прикладах мають генеруватися всі можливі класи вихідних даних системи.
5. Під час тестування система має побувати у всіх своїх внутрішніх станах, пройшовши всі можливі переходи між станами.

### **2. Тестування безпеки.**

Якщо програмна система призначена для зберігання або обробки даних, які становлять таємницю певного роду (наприклад, комерційну), то до властивостей такої системи, пред'являються підвищені вимоги. Виконання цих вимог перевіряється під час тестування безпеки системи.

У ході використання цього виду системного тестування перевіряється:

1. Організація авторизації й аутентифікації користувачів.
2. Коректність реєстрації в "журналі аудита" всіх подій системи, пов'язаних із безпекою.
3. Архітектура системи з погляду забезпечення захисту інформації від несанкціонованого доступу.
4. Наявність і повнота опису засобів забезпечення безпеки в документації на програмну систему.

## **Документування процедури тестування.**

Основне призначення документації, створеної під час тестування, – забезпечення гарантій того, що процес тестування виконується з необхідною якістю й усі аспекти поведження системи протестовані.

Перелік необхідної документації:

1. Тест-вимоги.
2. Тест-план.
3. Звіт про тестування.

Тест-вимоги розробляються на підставі системних і функціональних вимог до програмної системи. У них докладно описується, які аспекти поведження системи мають бути протестовані, щоб упевнитися в її коректному функціонуванні, і на підставі якого зовнішнього ефекту можна переконатися, що функціональність, яка перевіряється, реалізована правильно.

Тест-вимоги мають бути достатніми для побудови тест-плану перевірки програмної системи без знайомства з її програмним кодом.

Структура тест-вимог має дотримуватися структури розділу функціональних вимог на систему. Здебільшого, одній системній або функціональній вимозі відповідає мінімум одна тест-вимога.

Класифікація тест-вимог за призначенням:

1. Контроль вхідних даних.
2. Обробки помилок уведення даних і обробки інформації.
3. Отримання основного результату.
4. Оформлення й виведення результатів.

Для кожної тест-вимоги має існувати можливість перевірки – виконується ця вимога в реалізованій системі чи ні.

На підставі тест-вимог створюється тест-план – документ, що містить докладний покроковий опис того, як мають бути протестовані тест-вимоги. На відміну від тест-вимог, у тест-плані описуються конкретні способи перевірки функціональності системи.

Переважно, тест-план складається з окремих тестових прикладів, кожен із яких перевіряє деяку функцію або набір функцій системи. Для кожного тестового прикладу однозначно визначається критерій успішного проходження, за допомогою якого можна судити про відповідність поведження системи заданому.

Структура тест-плану має відповідати структурі тест-вимог.

Кожен пункт тест-плану має містити:

1. Посилання на вимогу(и), що перевіряється цим пунктом.
2. Конкретне значення вхідних даних.
3. Очікувану реакцію програми (тексти повідомлень, значення результатів).
4. Опис послідовності дій, необхідних для виконання пунктів тест-плану.

Під час ручного тестування зручним є подання тест-плану у вигляді текстового документа, у якому окремі розділи становлять описи тестових прикладів. Кожен тестовий приклад має містити перерахування послідовності дій, які необхідно виконати для проведення тестування, а також очікувані відгуки системи на ці дії.

За результатами виконання тестів створюється звіт про виконання тестування. Він є основним джерелом для висновку про ступінь відповідності протестованої системи вимогам. Такий звіт як мінімум має містити інформацію про кожен виконаний тестовий приклад і результат його виконання (успіх або невдача). Іноді тест-план сполучають зі звітом про проведення тестування, додаючи до нього інформацію про отриману реакцію системи й збіг (розбіжності) отриманих результатів з очікуваними. Наприкінці опису кожного тестового прикладу додається інформація про те, чи пройдений тестовий приклад загалом. Наприкінці всього тест-плану, сполученого зі звітом, міститься графа "Тестових прикладів пройдено / усього", у яку заноситься число пройдених тестових прикладів і загальна їхня кількість.

Приклад тест-плану, сполученого зі звітом про проведення тестування, наведено в додатку І.

У ході розроблення програмної системи студент має провести модульне, інтеграційне та системне тестування.

**У підрозділі 3.4** пояснювальної записки має знаходитися опис процедур системного тестування та їхніх результатів.

Мають бути описані такі види системного тестування:

1. Функціональне тестування.
2. Тестування безпеки.

Для опису процедури тестування мають бути складені такі документи:

1. Тест-вимоги.
2. Тест-плани, сполучені зі звітами про проведення тестування.

У звіті про проведення тестування вказуються як позитивні, так і негативні результати виконання окремих тестів.

Однак, загальний результат тестування програмної системи має бути позитивним, бо в протилежному випадку вона не відповідає певним вимогам. Для цього за необхідності проводяться додаткові заходи щодо виправлення помилок та тестування. Вони також мають бути описані в цьому пункті пояснювальної записки.

### **Підрозділ 3.5. Розгортання програмного продукту**

Розгортання – це процес поширення готового застосунку або компонента для установки на інші комп'ютери. У цьому підрозділі необхідно навести:

1. Опис вимог до апаратних та програмних засобів, необхідних для функціонування розробленого програмного продукту.
2. Опис дій щодо його інсталяції на комп'ютері користувача (послідовність дій, етапи інсталяції, скріншоти тощо).

## **5. Порядок подання до захисту та захист дипломного проєкту**

### **5.1. Попередній захист дипломного проєкту**

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

- доповідь із презентацією за повністю виконаним проєктом;
- демонстрація роботи програмного продукту.

Мета попереднього захисту – перевірка готовності студента до захисту відповідно до вимог випускової кафедри, оцінювання обсягу поданого проєкту і якості його виконання й оформлення. Незалежно від ступеня готовності проєкту, студент має з'явитися на попередній захист.

Для проведення попереднього захисту випускова кафедра визначає склад комісій та складає графік попереднього захисту.

На попередній захист подаються:

- оформлене і своєчасно затверджене завдання;
- повністю оформлена, але не переплетена пояснювальна записка з підписами студента, керівника і консультантів на завданні;

готовий програмний продукт і відеоролик його роботи;  
план доповіді та презентація, погоджені з керівником.

На підставі доповіді студента, його відповідей на питання, результатів перевірки пояснювальної записки, презентації комісія визначає рекомендації студенту щодо:

доповіді;  
відповідей на запитання;  
змісту пояснювальної записки;  
оформлення пояснювальної записки;  
презентації;  
демонстрації програмного продукту.

Допуск до захисту можливий у разі позитивного оцінювання за обома видами попереднього захисту (пояснювальна записка; програмна частина). Студенти, що не пройшли попередній захист, не допускаються до захисту.

## **5.2. Подання дипломного проєкту до захисту**

Після попереднього захисту та усунення недоліків закінчений дипломний проєкт подається керівнику проєкту. Він остаточно перевіряє відповідність виконаної роботи завданню та відповідним вимогам, складає подання (письмовий відгук), в якому дає характеристику роботи студента.

Цілком оформлений дипломний проєкт, що підписаний його керівником, проходить нормоконтроль. Після нормоконтролю дипломний проєкт підписує завідувач кафедри. Дипломний проєкт направляється на рецензування.

## **5.3. Захист дипломного проєкту**

Не пізніше ніж за добу до захисту студент подає дипломний проєкт секретарю екзаменаційної комісії (ЕК). Обов'язковим є роздатковий матеріал щодо виконаної роботи для кожного члена ЕК, який містить роздруківку слайдів презентації.

У ЕК можуть бути подані інші матеріали, які характеризують наукову та практичну цінність виконаного дипломного проєкту, а саме:

друковані статті за темою роботи;  
документи, які характеризують практичну цінність розробки студента;

документи, що вказують на практичне застосування роботи (підписані офіційними особами);

макети, зразки виробів тощо.

Захист дипломних проєктів проводиться на засіданні ЕК.

Захист одного дипломного проєкту, переважно, не має перевищувати 30 хвилин. Для доповіді щодо проєкту студенту надається не більше 10 хвилин.

Захист комплексного дипломного проєкту, здебільшого, планується і проводиться на одному засіданні ЕК, причому студенту, який захищається першим, доручається доповісти як про загальну частину роботи, так і про індивідуальну частину зі збільшенням (за необхідності) часу на доповідь. Усі студенти, які виконували комплексну роботу, мають бути повною мірою обізнані із загальною частиною роботи і готові до запитань членів комісії не тільки з індивідуальної, а й із загальної частини роботи.

Доповідь студента має складатися з трьох основних частин, а саме: вступу, основної частини та висновків.

У вступі необхідно зазначити актуальність теми проєкту, дати загальний аналіз стану проблеми і сформулювати основні завдання, з вирішенням яких було пов'язано виконання проєкту.

В основній частині доповіді у стислій формі необхідно навести звіт про зміст виконаних розробок, показати ефективність ухвалених технічних рішень, навести короткий звіт з отриманих результатів.

У заключній частині доповіді необхідно зробити загальні висновки і дати рекомендації щодо можливої сфери застосування об'єкта проєкування, перелічити публікації за темою роботи, навести відомості про впровадження.

Доповідь має супроводжуватися посиланнями на електронну презентацію, яка демонструється студентом. Презентація має містити такі слайди:

титульний слайд із вихідними даними щодо дипломного проєкту;

зміст презентації (з посиланнями на відповідні слайди);

актуальність теми та мета дипломного проєкту;

модель організаційної структури підприємства, підрозділу підприємства;

модель управління бізнес-процесом;

UML-діаграма варіантів використання;

вайрфрейм або мокап проєкту інтерфейсу користувача;

математична (логічна) постановка;



заповнені форми вихідних та вхідних документів, діаграми, карти;  
логічна та фізична моделі бази даних;

UML-діаграма класів (Class Diagram), що реалізують основну бізнес-логіку програмної системи, або UML-діаграма діяльності (Activity Diagram), яка відбиває основну бізнес-логіку програмної системи;

UML-діаграма станів (State Diagram), у яких можуть знаходитися елементи графічного інтерфейсу користувача;

результати функціонального тестування програмного забезпечення та тестування його безпеки;

використані інструментальні засоби та технології;

висновки за результатами дипломного проєкту;

апробація результатів дипломного проєкту;

заключний слайд.

Під час захисту може додатково використовуватися демонстраційний матеріал у вигляді відеоролика.

Після доповіді студент стисло відповідає на запитання членів ЕК. Далі зачитується рецензія. Студенту надається можливість відповісти на зауваження рецензента.

Після закінчення захисту всіх заявлених студентів комісія проводить закрите обговорення кожного захисту й оцінює його відповідно до критеріїв оцінювання. Водночас береться до уваги рівень виконаної роботи та розробленого програмного продукту, якість оформлення пояснювальної записки, рівень наукової, практичної та теоретичної підготовки студента, ритмічність роботи над проєктом, наявність публікацій, виступів на конференціях тощо.

Результати захисту дипломного проєкту доводяться до відома студентів після завершення роботи ЕК.

## Рекомендована література

### Основна

1. Методичні рекомендації до виконання курсового проекту з навчальної дисципліни "Організація баз даних та знань" / уклад. Л. А. Павленко, О. В. Тарасов, М. Ю. Лосєв та ін. – Харків : Вид. ХНЕУ, 2013. – 84 с.

2. Ушакова І. О. Проектування інформаційних систем : практикум / І. О. Ушакова. – Харків : Вид. ХНЕУ ім. С. Кузнеця, 2015. – 250 с.

### Додаткова

3. Автоматизация проектирования вычислительных систем. Языки, моделирование и базы данных / ред. М. Брейер. – Москва : Мир, 2015. – 463 с.

4. Вендров А. М. Практикум по проектированию программного обеспечения экономических информационных систем / А. М. Вендров. – Москва : Финансы и статистика, 2017. – 192 с.

5. Вигерс К. Разработка требований к программному обеспечению / К. Вигерс, Дж. Битти. – Санкт-Петербург : БХВ-Петербург, 2018. – 736 с.

6. Воройский Ф. С. Основы проектирования автоматизированных библиотечно-информационных систем / Ф. С. Воройский. – Москва : ФИЗМАТЛИТ, 2016. – 384 с.

7. Гагарина Л. Г. Технология разработки программного обеспечения : учеб. пособ. / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул ; под ред. проф. Л. Г. Гагариной. – Москва : ИД "ФОРУМ" ; НИЦ "Инфра-М", 2013. – 400 с.

8. Глухих М. И. Программная инженерия. Обеспечение качества программных средств методами статического анализа : учеб. пособ. / М. И. Глухих, В. М. Ицыксон. – Санкт-Петербург : Изд-во Политехн. ун-та, 2011. – 150 с.

9. Глушаков С. В. Базы данных / С. В. Глушаков, Д. В. Ломотько. – Харьков : Фолио, 2017. – 504 с.

10. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт. – 6-е изд. – Киев : Диалектика, 2015. – 784 с.

11. ДСТУ 8302:2015 "Бібліографічне посилання. Загальні положення та правила складання". – Київ : Держстандарт України, 2007. – 52 с.
12. ДСТУ ISO/IEC TR 9126-4:2012 (ISO/IEC TR 9126-4:2004, IDT) Програмна інженерія. Якість продукту. Частина 4. Метрики якості. – Київ : Мінекономрозвитку України, 2013. – 49 с.
13. ДСТУ ISO/IEC 9126-1:2013 (ISO/IEC 9126-1:2001, IDT) Програмна інженерія. Якість продукту. Частина 1. Модель якості. – Київ : Мінекономрозвитку України, 2014. – 20 с.
14. Зегжда Д. П. Основы безопасности информационных систем / Д. П. Зегжда, А. М. Ивашко. – Москва : Горячая линия – Телеком, 2017. – 452 с.
15. Ипатова Э. Р. Методологии и технологии системного проектирования информационных систем / Э. Р. Ипатова, Ю. В. Ипатов. – Москва : Флинта, 2016. – 256 с.
16. Канер С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / С. Канер, Дж. Фолк, Е. Нгуен ; пер. с англ. – Киев : Издательство "Диа-Софт", 2011. – 544 с.
17. Котляров В. П. Основы тестирования программного обеспечения : учеб. пособ. / В. П. Котляров, Т. В. Коликова. – Москва : Интернет-Университет Информационных Технологий ; БИНОМ. Лаборатория знаний, 2009. – 285 с.
18. Кулаков К. А. Основы тестирования программного обеспечения : учеб. пособ. / К. А. Кулаков, В. М. Димитров. – Петрозаводск : Изд-во ПетрГУ, 2018. – 57 с.
19. Куликов С. С. Тестирование программного обеспечения: Базовый курс / С. С. Куликов. – 2-е изд. – Минск : Четыре четверти, 2017. – 312 с.
20. Липаев В. В. Тестирование компонентов и комплексов программ : учебник / В. В. Липаев. – Москва : СИНТЕГ, 2010. – 400 с.
21. Макгрегор Дж. Тестирование объектно-ориентированного программного обеспечения / Дж. Макгрегор, Д. Сайкс. – Киев : Диасофт, 2002. – 432 с.
22. Макконнелл С. Совершенный код. Мастер-класс / С. Макконнелл ; пер. с англ. – Москва : Издательско-торговый дом "Русская редакция" ; Санкт-Петербург : Питер, 2005. – 896 с.
23. Мидоу Ч. Анализ информационных систем : монография / Ч. Мидоу. – Москва : Прогресс, 2015. – 400 с.

24. Мюллер Р. Дж. Базы данных и UML. Проектирование / Р. Дж. Мюллер. – Москва : ЛОРИ, 2017. – 420 с.
25. Плаксин М. А. Тестирование и отладка программ для профессионалов будущих и настоящих / М. А. Плаксин. – 2-е изд. (эл.). – Москва : БИНОМ. Лаборатория знаний, 2013. – 167 с.
26. Постолиит А. Visual Studio .NET: разработка приложений баз данных / А. Постолиит. – Санкт-Петербург : БХВ, 2014. – 544 с.
27. Раскин Дж. Интерфейс: новые направления в проектировании компьютерных систем / Дж. Раскин. – Москва : Символ-плюс, 2017. – 272 с.
28. Редько В. Н. Базы данных и информационные системы / В. Н. Редько, И. А. Басараб. – Москва : Знание, 2017. – 341 с.
29. Сеницын С. В. Верификация программного обеспечения : курс лекций / С. В. Сеницын, Н. Ю. Налютин. – Москва : Интуит НОУ, 2016. – 446 с.
30. Слепцов А. И. Автоматизация проектирования управляющих систем гибких автоматизированных производств / А. И. Слепцов, А. А. Юрасов. – Москва : Техника, 2015. – 110 с.
31. Сухомлинов А. И. Разработка информационных систем : учеб. пособ. / А. И. Сухомлинов. – Москва : Проспект, 2015. – 112 с.
32. Ульман Дж. Базы данных на Паскале / Дж. Ульман. – Москва : Машиностроение, 2015. – 368 с.
33. Уэлдон Дж.-Л. Администрирование баз данных / Дж.-Л. Уэлдон. – Москва : Финансы и статистика, 2017. – 207 с.
34. Фаулер М. Рефакторинг. Улучшение существующего кода / М. Фаулер ; пер. с англ. – Санкт-Петербург : Символ-Плюс, 2003. – 432 с.
35. Хаббард Дж. Автоматизированное проектирование баз данных / Дж. Хаббард. – Москва : Мир, 2016. – 296 с.
36. Хансен Г. Базы данных: разработка и управление / Г. Хансен, Дж. Хансен. – Москва : Бином, 2015. – 704 с.
37. Черников Б. В. Оценка качества программного обеспечения: Практикум : учеб. пособ. для вузов / Б. В. Черников, Б. Е. Поклонов ; под ред. Б. В. Черникова. – Москва : Инфра-М ; Форум, 2012. – 400 с.
38. Шаймарданов Р. Б. Моделирование и автоматизация проектирования структур баз данных / Р. Б. Шаймарданов. – Москва : Радио и связь, 2017. – 120 с.
39. Шнайдер Р. Microsoft SQL Server 6.5. Проектирование высокопроизводительных баз данных / Р. Шнайдер. – Москва : ЛОРИ, 2017. – 361 с.

40. Шоу А. Логическое проектирование операционных систем : монография / А. Шоу. – Москва : Мир, 2016. – 360 с.
41. Яргер Р. Дж. MySQL и mSQL: Базы данных для небольших предприятий и Интернета / Р. Дж. Яргер, Дж. Риз, Т. Кинг. – Москва : Символ-Плюс, 2014. – 560 с.
42. Ярочкин В. Безопасность информационных систем / В. Ярочкин. – Москва : Ось-89, 2015. – 320 с.
43. Developing Information Systems: Practical guidance for IT professional / P. Thompson, D. Paul, A. Paul et al. – London : BCS Learning & Development Limited, 2014. – 206 p.
44. Information Systems Development: Advances in Theory, Practice, and Education / O. Vasilecas, A. Caplinskis, G. Wojtkowski et al. – Berlin : Springer Science & Business Media, 2015. – 544 p.
45. Information Systems Development / W. W. Song, Sh. Xu, Ch. Wan et al. – Berlin : Springer Science & Business Media, 2010. – 576 p.
46. Lampathaki F. Business process modelling. Business process reengineering / F.Lampathaki, S. Koussouris, J. Psarras. – Zografou : NTUA, 2013. – 89 p.
47. Standard glossary of terms used in Software Testing. Version 2.3. – Edinburgh : ISTQB, 2014. – 53 p.

## **Інформаційні ресурси**

48. Вимоги до оформлення курсових і дипломних проектів : методичні рекомендації для студентів галузей знань 12 "Інформаційні технології" та 18 "Виробництво та технології" [Електронний ресурс] / уклад. А. А. Гаврилова, С. П. Євсєєв, Г. П. Коц та ін. – Харків : ХНЕУ ім. С. Кузнеця, 2018. – 49 с.
49. Інформаційний портал CRM [Електронний ресурс]. – Режим доступу : [www.crm.com.ua](http://www.crm.com.ua).
50. Корпоративне управління [Електронний ресурс]. – Режим доступу : <http://www.corporation.com.ua>.
51. Сайт Держкомстату України [Електронний ресурс]. – Режим доступу : [www.ukrstat.gov.ua](http://www.ukrstat.gov.ua).
52. 10 usability heuristics for user interface design / Jakob Nielsen. – April 24, 1994 [Electronic resource]. – Access mode : <https://www.nngroup.com/articles/ten-usability-heuristics>.

53. Сайт компанії IBM [Електронний ресурс]. – Режим доступу : <http://www.ibm.com/ua-en>.
54. Сайт компанії Microsoft [Електронний ресурс]. – Режим доступу : <http://www.microsoft.com>.
55. Agile alliance [Electronic resource]. – Access mode : <http://www.agilealliance.org>.
56. ITC Online [Electronic resource]. – Access mode : <http://itc.ua>.
57. Object Management Group [Electronic resource]. – Access mode : <http://www.omg.org>.
58. SCRUM [Electronic resource]. – Access mode : <http://www.scrum.org>.
59. The official UML Web site [Electronic resource]. – Access mode : <http://www.uml.org>.
60. The official web site of Gartner, Inc. [Electronic resource]. – Access mode : <http://www.gartner.com>.
61. Try QA [Electronic resource]. – Access mode : <http://tryqa.com>.

# Додатки

Додаток А

## Зразок заяви студента на затвердження теми дипломного проєкту

Завідувачу кафедри  
інформаційних систем  
доценту Ушаковій І.О.  
студента 4 курсу <номер> групи  
<ПІБ студента>

### ЗАЯВА

Прошу затвердити мені тему дипломного проєкту <Назва теми>.

<Дата>                      <Підпис студента>      <ПІБ студента>

Керівник

дипломного проєкту

<Посада

наук. ступінь, вчене звання>      <Підпис керівника>      <ПІБ керівника>

## Зразки завдань на дипломний проєкт

### Перша сторінка завдання для спеціальності 122 "Комп'ютерні науки"

#### ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ

Факультет	<u>Економічної інформатики</u>
Кафедра	<u>Інформаційних систем</u>
Освітній ступінь	<u>Бакалавр</u>
Спеціальність	<u>122 "Комп'ютерні науки"</u>

#### ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних систем  
\_\_\_\_\_ доц. І. О. Ушакова  
" " \_\_\_\_\_ 20 р.

### ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ СТУДЕНТУ Педану Олегу Анатолійовичу

**1. Тема проєкту:** "Розроблення чат-бота для підтримки навчання студентів на основі технології Microsoft Bot Framework"

керівник проєкту: Ушакова Ірина Олексіївна, к. е. н., доцент  
затверджені наказом ректора від " " \_\_\_\_\_ 20 р. № \_\_\_\_\_

**2. Строк подання студентом проєкту:** \_\_\_\_\_

**3. Вихідні дані до проєкту:** ДСТУ щодо обробки інформації, літературні джерела, матеріали практики.

**4. Зміст розрахунково-пояснювальної записки** (перелік питань, які потрібно розробити):

Розділ 1. Аналіз предметної області "Розроблення чат-бота для підтримки навчання студентів"

Розділ 2. Специфікація вимог до модуля "Розроблення чат-бота для підтримки навчання студентів"

Розділ 3. Проєктні та технічні рішення

**5. Перелік графічного матеріалу:**

модель організаційної структури підприємства та його підрозділу, модель бізнес-процесів, UML-діаграма варіантів використання, UML-діаграма класів, UML-діаграма станів, фізична та логічна моделі бази даних, форми вхідних та вихідних документів, екранні форми програмного продукту.



**Перша сторінка завдання для спеціальності  
121 "Інженерія програмного забезпечення"**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

<b>Факультет</b>	<u>Економічної інформатики</u>
<b>Кафедра</b>	<u>Інформаційних систем</u>
<b>Освітній ступінь</b>	<u>Бакалавр</u>
<b>Спеціальність</b>	<u>121 "Інженерія програмного забезпечення"</u>

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інформаційних систем  
\_\_\_\_\_ доц. І. О. Ушакова  
" " \_\_\_\_\_ 20 р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЄКТ СТУДЕНТУ  
<ПІБ студента>**

**1. Тема проєкту:** " \_\_\_\_\_ "

керівник проєкту: ПІБ керівника, вчене звання, ступінь  
затверджені наказом ректора від " " \_\_\_\_\_ 20 р. №

**2. Строк подання студентом проєкту:** \_\_\_\_\_

**3. Вихідні дані до проєкту:** ДСТУ щодо обробки інформації, літературні джерела, матеріали практики.

**4. Зміст розрахунково-пояснювальної записки** (перелік питань, які потрібно розробити):

Розділ 1. <Назва>

Розділ 2. <Назва>

Розділ 3. <Назва>

**5. Перелік графічного матеріалу:**

<Перелік> .

Перша сторінка завдання для спеціальності  
126 "Інформаційні системи та технології"

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

Факультет	<u>Економічної інформатики</u>
Кафедра	<u>Інформаційних систем</u>
Освітній ступінь	<u>Бакалавр</u>
Спеціальність	<u>126 "Інформаційні системи та технології"</u>

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інформаційних систем  
\_\_\_\_\_ доц. І. О. Ушакова  
" " \_\_\_\_\_ 20 р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЄКТ СТУДЕНТУ  
<ПІБ студента>**

**1. Тема проєкту:** " \_\_\_\_\_ "

керівник проєкту: ПІБ керівника, вчене звання, ступінь

затверджені наказом ректора від " " \_\_\_\_\_ 20 р. №

**2. Строк подання студентом проєкту:** \_\_\_\_\_

**3. Вихідні дані до проєкту:** ДСТУ щодо обробки інформації, літературні джерела, матеріали практики.

**4. Зміст розрахунково-пояснювальної записки** (перелік питань, які потрібно розробити):

Розділ 1. <Назва>

Розділ 2. <Назва>

Розділ 3. <Назва>

**5. Перелік графічного матеріалу:**

<Перелік> .

**Друга сторінка завдання для спеціальностей  
121 "Інженерія програмного забезпечення", 122 "Комп'ютерні  
науки", 126 "Інформаційні системи та технології"**

**6. Консультація розділів дипломного проєкту**

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

**7. Дата видачі завдання:** " " 20 р.

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів дипломного проєкту	Строк виконання етапів проєкту	Примітка
1	Розроблення плану дипломного проєкту, ознайомлення з літературними джерелами за темою		
2	Аналіз предметної області		
3	Розроблення специфікації вимог до модуля		
4	Розроблення проєктних та технічних рішень		
5	Перевірка чернетки дипломного проєкту та внесення зміни до неї керівником		
6	Перевірка якості дипломного проєкту на "Антиплагіат"		
7	Оформлення дипломного проєкту		
8	Подання Голові Екзаменаційної комісії щодо захисту дипломного проєкту		

**Приклад змістовної частини дипломного проєкту  
дослідницького характеру**

1. АНАЛІЗ СУЧАСНОГО СТАНУ ДОСЛІДЖЕНЬ З ПРОБЛЕМИ СЕГМЕНТУВАННЯ БАЗИ ДАНИХ КЛІЄНТІВ
  - 1.1. Необхідність сегментування бази даних клієнтів
  - 1.2. Аналіз наявних методів сегментування
2. МЕТОДИКА ВИБОРУ МЕТОДУ СЕГМЕНТУВАННЯ БАЗИ ДАНИХ КЛІЄНТІВ
  - 2.1. Критерії вибору методу сегментування бази даних клієнтів
  - 2.2. Етапи методики вибору методу сегментування бази даних клієнтів
3. РОЗРОБЛЕННЯ ПРАКТИЧНИХ РЕКОМЕНДАЦІЙ ЩОДО ВИБОРУ МЕТОДУ СЕГМЕНТУВАННЯ БАЗИ ДАНИХ КЛІЄНТІВ
  - 3.1. Опис бази клієнтів, що аналізується
  - 3.2. Аналіз результатів сегментування бази даних клієнтів
  - 3.3. Висновки щодо вибору методу сегментування

## Структура змістовної частини дипломного проєкту практичного характеру

№ п/п	Назва структурного елемента
1.	АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ <НАЗВА ПРЕДМЕТНОЇ ОБЛАСТІ>
1.1.	Коротка характеристика об'єкта управління <назва об'єкта управління>
1.2.	Опис предметної області <назва предметної області>
1.3.	Огляд і аналіз наявних аналогів, що реалізують функції предметної області
2.	СПЕЦИФІКАЦІЯ ВИМОГ ДО МОДУЛЯ (СИСТЕМИ)
2.1.	Глосарій
2.2.	Розроблення варіантів використання
2.2.1.	Діаграма варіантів використання
2.2.2.	Специфікація варіантів використання
2.3.	Специфікація функціональних та нефункціональних вимог
2.4.	Проектування інтерфейсу користувача
3.	ПРОЄКТНІ ТА ТЕХНІЧНІ РІШЕННЯ
3.1.	Математична (логічна) постановка
3.2.	Проектування структури бази даних
3.2.1.	Концептуальне інфологічне проектування
3.2.2.	Проектування логічної моделі даних
3.2.3.	Проектування фізичної моделі даних
3.3.	Проектування програмного забезпечення
3.4.	Тестування програмної системи
3.5.	Розгортання програмного продукту

**Зразок титульного аркуша дипломного проєкту**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**ФАКУЛЬТЕТ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ**

**КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ**

Пояснювальна записка

До дипломного проєкту

Бакалавра

на тему: "Назва теми"

Виконав: студент 4 курсу,  
групи <група>,  
спеціальності  
<№ спеціальності> <назва  
спеціальності>  
<ПІБ студента>

Керівник: <науковий ступінь>,  
<вчене звання>, <ПІБ>

## Приклади рефератів

### 1. Для проєктів дослідницького типу

#### РЕФЕРАТ

Пояснювальна записка до дипломного проєкту: 81 с., 34 рис., 6 табл., 2 додатка, 35 джерел.

Об'єктом дослідження є методи математичного моделювання інформаційних систем.

Метою роботи є дослідження підходів до розроблення імітаційних моделей процесів функціонування інформаційних систем та автоматизація процесу імітаційного моделювання.

Методами розроблення обрано метод аналізу для дослідження наявних методів моделювання, метод синтезу для поєднання переваг наявних методів моделювання, методи моделювання для подання та дослідження процесів функціонування інформаційних систем, метод порівняльного аналізу для оцінювання адекватності моделі процесів функціонування інформаційних систем.

У результаті виконання роботи обґрунтований раціональний підхід до розроблення імітаційних моделей процесів функціонування інформаційних систем та розроблений програмний засіб автоматизації імітаційного моделювання, який дозволяє створювати імітаційні моделі процесів функціонування інформаційних систем та досліджувати їх.

Результати дослідження можуть бути використані в науково-дослідницьких закладах та підрозділах підприємств, що займаються розробленням імітаційних моделей.

ІНФОРМАЦІЙНА СИСТЕМА, Е-МЕРЕЖА, АДЕКВАТНІСТЬ МОДЕЛІ, ІМІТАЦІЙНА МОДЕЛЬ, МАТЕМАТИЧНА СХЕМА, МЕТОДИ МОДЕЛЮВАННЯ

## ABSTRACT

The bachelor's thesis report: 81 pages, 34 figures, 6 tables, 2 appendices, 35 sources.

The object of the research is the methods of mathematical modeling of information systems.

The purpose of the work is to research the approaches to the development of simulation models of the functioning processes of information systems and automation of the simulation process.

The research methods are analysis to research existent methods of modeling, synthesis to unite the advantages of the available methods of modeling, modeling to present and research the functioning processes of information systems, comparison to estimate the adequacy of the models.

As a result of the work a rational approach to the development the simulation models of the functioning processes of the information systems has been substantiated. Also a software product has been developed, which help to create simulation models of the functioning processes of information systems and research the models.

The results of the research can be used in research establishments and the departments of enterprises which develop simulation models.

INFORMATION SYSTEM, E-NET, MODEL ADEQUACY, SIMULATION MODEL, MATHEMATICAL SCHEME, METHODS OF MODELING



## 2. Для проєктів практичного характеру

### РЕФЕРАТ

Пояснювальна записка до дипломного проєкту: 90 с., 30 рис., 20 табл., 12 додатків, 55 джерел.

Об'єктами проєктування є функціональні елементи, архітектура, інформаційне і програмне забезпечення модуля моніторингу складських поставок ТОВ "СІГМА".

Мета проєктування – створення модуля "Моніторинг складських поставок".

Метод проєктування – використання програмних систем ARIS Toolset, IBM Rational, Microsoft Visual Studio.

Створений модуль дозволяє підвищити достовірність обліку товарів на складі, обґрунтованість та оперативність прийняття рішень під час управління поставками товарів і їхніми запасами на складі.

Результати розробки можуть бути впроваджені на торгових підприємствах.

СИСТЕМА УПРАВЛІННЯ ЛАНЦЮГАМИ ПОСТАВОК, ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОЄКТУВАННЯ, CASE-ДІАГРАМИ, БАЗА ДАНИХ, МОНІТОРИНГ СКЛАДСЬКИХ ПОСТАВОК, WEB-СЛУЖБА

## ABSTRACT

The bachelor's thesis report: 90 pages, 30 figures, 20 tables, 12 appendices, 55 sources.

The objects of designing are functional elements, architecture and software of a module which provides monitoring of warehouse supplies.

The purpose of designing is to create "The monitoring of warehouse supplies" software module for Sigma Ltd.

The method of designing is using the ARIS Toolset, IBM Rational, Microsoft Visual Studio software systems.

The advantage of the developed module is the increasing of reliability of goods accounting, validity and timeliness of decision-making.

The obtained results can be applied at commercial enterprises.

SUPPLY CHAIN MANAGEMENT SYSTEM, OBJECT ORIENTED DESIGN, CASE DIAGRAMS, DATABASE, MONITORING OF WAREHOUSE SUPPLIES, WEB SERVICE

## Зразок оформлення змісту

## ЗМІСТ

ВСТУП .....	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ <НАЗВА ПРЕДМЕТНОЇ ОБЛАСТІ> ....	7
1.1. Коротка характеристика об'єкта управління <назва об'єкта управління> .....	7
1.2. Опис предметної області <назва предметної області> .....	10
1.3. Огляд і аналіз наявних аналогів, що реалізують функції предметної області .....	14
2. СПЕЦИФІКАЦІЯ ВИМОГ ДО МОДУЛЯ (СИСТЕМИ).....	19
2.1. Глосарій .....	19
2.2. Розроблення варіантів використання .....	21
2.2.1. Діаграма варіантів використання .....	21
2.2.2. Специфікація варіантів використання.....	24
2.3. Специфікація функціональних та не функціональних вимог .....	29
2.4. Проектування інтерфейсу користувача.....	32
3. ПРОЄКТНІ ТА ТЕХНІЧНІ РІШЕННЯ .....	35
3.1. Математична (логічна) постановка .....	35
3.2. Проектування структури бази даних .....	38
3.2.1. Концептуальне інфологічне проектування .....	38
3.2.2. Проектування логічної моделі даних .....	40
3.2.3. Проектування фізичної моделі даних .....	43
3.3. Проектування програмного забезпечення .....	45
3.4. Тестування програмної системи.....	49
3.5. Розгортання програмного продукту .....	55
ВИСНОВКИ .....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	71
Додаток А. Організаційна структура підприємства .....	75
Додаток Б. Бізнес-процеси предметної області .....	80

## Приклад тест-плану, сполученого зі звітом про проведення тестування

### Тестовий приклад: № 1.

Призначення: перевірка того, що перед початком передачі даних перевіряється обліковий запис користувача й у випадку введення імені користувача за замовчуванням під час перебування системи в стані "Максимальний захист" передача даних не відбувається.

Тест-вимога, що перевіряється: 1.1.

Передумови для тесту: система має бути переведена в стан "Максимальний захист", а її налаштування повинні мати значення за замовчуванням.

Критерій проходження тесту: всі реальні значення збігаються з очікуваними.

№ п/п	Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
1	Запустити термінальний клієнт і ввести IP-адресу 127.0.0.1	Має з'явитися запрошення: TRANSFER>	Запрошення TRANSFER> з'являється	Так
2	Запустити процес передачі даних за допомогою введення команди SEND DATA	Має з'явитися запрошення: Enter your credentials... Login:	Запрошення Enter your credentials... Login: з'являється	Так
3	Увести ім'я користувача <b>default</b>	Має з'явитися повідомлення: <b>Password:</b>	Повідомлення <b>Password:</b> з'являється	Так
4	Увести пароль <b>default</b>	Має з'явитися повідомлення: <b>Default user blocked - system set to High security.</b>	З'являється повідомлення: <b>Your credentials accepted. Data transfer started.</b>	Ні

Відмітка про проходження тесту (пройдений/не пройдений):  
не пройдений.

## Зміст

Вступ.....	3
1. Мета й завдання дипломного проєктування.....	4
2. Організація виконання дипломного проєкту .....	5
3. Структура, зміст та обсяг дипломного проєкту.....	6
4. Методичні рекомендації до розроблення структурних елементів пояснювальної записки.....	9
4.1. Рекомендації щодо розроблення розділів пояснювальної записки дипломного проєкту дослідницького характеру.....	13
4.2. Рекомендації щодо розроблення розділів пояснювальної записки дипломного проєкту практичного характеру .....	14
5. Порядок подання до захисту та захист дипломного проєкту.....	38
5.1. Попередній захист дипломного проєкту .....	38
5.2. Подання дипломного проєкту до захисту .....	39
5.3. Захист дипломного проєкту .....	39
Рекомендована література.....	42
Основна .....	42
Додаткова .....	42
Інформаційні ресурси .....	45
Додатки.....	47

НАВЧАЛЬНЕ ВИДАННЯ

**Методичні рекомендації  
до виконання кваліфікаційних робіт  
для студентів спеціальностей  
121 "Інженерія програмного забезпечення",  
122 "Комп'ютерні науки",  
126 "Інформаційні системи та технології"  
першого (бакалаврського) рівня**

*Самостійне електронне текстове мережеве видання*

Укладачі: **Парфьонов** Юрій Едуардович  
**Ушакова** Ірина Олексіївна

Відповідальний за видання *І. О. Ушакова*

Редактор *В. Ю. Степаненко*

Коректор *В. Ю. Степаненко*

План 2020 р. Поз. № 58 ЕВ. Обсяг 62 с.

---

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

---

*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру  
ДК № 4853 від 20.02.2015 р.*