

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

методичні рекомендації до виконання
індивідуальних завдань

Запоріжжя 2025



УДК 004.41(072)
П78

Рекомендовано Науково-методичною радою
ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«МЕТІНВЕСТ ПОЛІТЕХНІКА»
(протокол № 6 від 28.02.2025 р.)

Укладач

Костіков О.А., канд. фіз.-мат. наук, доцент.

П78 Проєктування інформаційних систем та програмного забезпечення :
методичні рекомендації до виконання індивідуальних завдань (для
студентів комп'ютерних спеціальностей першого (бакалаврського) рівня
вищої освіти) / уклад. О. А. Костіков. Запоріжжя : ТОВ «ТЕХНІЧНИЙ
УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА». 2025.
22 с.

Методичні вказівки містять відомості щодо формування кейсу задач та критеріїв оцінювання індивідуальних завдань на платформі Moodle та приклади їх розв'язання з використанням CASE засобів, діаграм UML та обґрунтування прийнятих проєктних рішень. Матеріал навчального посібника має на меті підвищити якість виконання роботи, виробити навички проєктування програмних систем, в тому числі з використанням CASE засобів та мови моделювання UML.

Рекомендовано для студентів комп'ютерних спеціальностей та форм навчання першого (бакалаврського) рівня освіти.

УДК 004.41(072)

© ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ МЕТІНВЕСТ ПОЛІТЕХНІКА», 2025



зміст

ВСТУП	4
1 ІНДИВІДУАЛЬНІ ЗАВДАННЯ	5
1.1 Відомості щодо формування кейсу задач та критеріїв оцінювання індивідуального завдання на платформі Moodle	5
1.2 Кейс задач індивідуального завдання	6
2 РОЗВ'ЯЗАННЯ ТИПОВОГО ВАРІАНТУ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ	7
2.1 Приклад розв'язання завдання 1	7
2.2 Приклад розв'язання завдання 2	12
2.3 Приклад розв'язання завдання 3	13
2.4 Приклад розв'язання завдання 4	17
ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ	20



ВСТУП

Проектування інформаційних систем та програмного забезпечення – базовий курс підготовки майбутніх бакалаврів з комп'ютерних наук, який дозволяє опанувати і професійно використовувати знання та навички, необхідні для розв'язку практичних задач професійної діяльності, пов'язаної з моделюванням, проектуванням та розробкою програмного забезпечення. Дисципліна розглядає: 1) суть і призначення інформаційних систем; 2) стадії проектування і вимоги до процесу проектування інформаційних систем; 3) системний підхід до проектування інформаційних систем, топологію та архітектуру інформаційних систем; 4) структурну, об'єктно-орієнтовану та типову технологію проектування; 5) RAD-методологію, CASE-технологію створення й супроводу інформаційних систем, технологію RUP, паттерн-технологію.

Особливістю курсу є те, що в процесі вивчення дисципліни встановлюється зв'язок теоретичних знань, вмінь та навичок з технологіями проектування реальних програмних застосувань, особлива увага приділяється вирішенню загальних проблем, які виникають при проектуванні та розробці програмного забезпечення.

В рамках освоєння дисциплін «Проектування інформаційних систем та програмного забезпечення» за змістовними модулями першого семестру студентам пропонується виконати індивідуальні завдання з розв'язування прикладних задач за темами відповідних змістовних модулів. Представлені методичні рекомендації містять відомості щодо формування кейсу задач та критеріїв оцінювання індивідуального завдання на платформі Moodle та приклади їх розв'язання з використанням CASE засобів, діаграм UML та обґрунтуванням проектних рішень в обсязі, необхідному для виконання індивідуального завдання.



1 ІНДИВІДУАЛЬНІ ЗАВДАННЯ

1.1 Відомості щодо формування кейсу задач та критеріїв оцінювання індивідуального завдання на платформі Moodle

На платформі Moodle кейс задач індивідуальних завдань з дисципліни «Проектування інформаційних систем та програмного забезпечення» за змістовними модулями першого семестру формується рандомно системою шляхом автоматичного вибору окремого варіанта серед наявних в рамках відповідної категорії. Кожен студент отримує однаковий за кількістю та змістовністю кейс задач в межах відповідної спеціальності з зазначенням переліку завдань, що виносяться викладачем на виконання.

Індивідуальні завдання виконуються самостійно у зручний для студента час в межах терміну подачі роботи, передбачених у розділі «Розподіл балів за контрольними точками та графік їх виконання» та розміщується у відповідному розділі на платформі Moodle. Розв'язання кожного завдання завантажується у вигляді файлу з розширенням .docx або .pdf, або .jpg, або .png, або .txt .

Максимальна кількість балів вказана за кожне окреме завдання у системі Moodle та визначається в залежності від обґрунтування ходу розв'язання, правильності отриманого розв'язку та аналізу результату.

Використання штучного інтелекту (ШІ) не забороняється, оскільки пропозиції відомих застосунків ШІ суттєво залежать від обміркованої постановки питання і уточнюючих питань; однак в разі, якщо відповідь, отримана з використанням ШІ, містить суттєві похибки або не є комплексною, або не відповідає за усталеним оформленням, термінологією, або іншим вимогам до завдання, то оцінка за виконання знижується.

Перевірка індивідуального завдання виконується протягом тижня після завершення терміну подачі роботи. За побажанням студента при наявності похибок або виконання індивідуального завдання не в повному обсязі допускається доопрацювання до передостаннього тижня навчання.



1.2 Кейс задач індивідуальне завдання

Завдання 1. Розв'язання задач на аналіз вимог до програмного забезпечення, що розробляється.

Завдання 2. Розв'язання задач на побудову діаграми діяльності до варіантів використання проєктованої системи.

Завдання 3. Розв'язання задач на визначення класів сутностей та побудову діаграми класів проєктованої системи.

Завдання 4. Розв'язання задач на аналіз варіанту використання розроблюваної системи, який включає визначення класів аналізу, що беруть участь у реалізації цього варіанту використання та побудову діаграм послідовностей для представлення їх взаємодії.

2 РОЗВ'ЯЗАННЯ ТИПОВОГО ВАРІАНТУ ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

2.1 Приклад розв'язання завдання 1

Завдання 1. Для системи автоматизації діяльності магазину відеопрокату виконати аналіз вимог. Скласти глосарій предметної області, розробити діаграми варіантів використання, створити сценарії варіантів використання.

Розв'язання.

Аналіз вимог до системи здійснюється на основі постановки задачі.

Постановка задачі.

Щойно відкритий відеомагазин має намір запропонувати прокат відеокасет і дисків широкому загалу. Директор магазину сповнений рішучості почати свою діяльність, спираючись на підтримку комп'ютерної системи. Директор уже навів довідки про деякі програмні пакети для малого бізнесу, які могли б стати в нагоді для налагодження і подальшого розвитку. Для надання допомоги у виборі пакету магазин найняв бізнес-аналітика, який займається визначенням і специфікацією вимог.

Спочатку асортимент магазину складає близько тисячі відеокасет і п'ятсот відеодисків. Запас вже замовлений у одного постачальника, проте, для майбутніх замовлень директор має намір вдаватися до послуг більшого числа постачальників. Все відеокасети і диски забезпечені штрих-кодом, так що сканер, інтегрований в систему, може підтримувати операції видачі напрокат і повернення відеофільмів. Членські картки клієнтів також забезпечені штрих-кодом.

Існуючі клієнти мають можливість резервувати відео таким чином, щоб комплект відеофільмів був зібраний до певної дати. Система повинна володіти гнучким пошуковим механізмом для відповідей на запити клієнтів, включаючи питання, що стосуються фільмів, яких немає в асортименті магазину (але які він може замовити на прохання клієнта).

Система повинна задовольняти наступним вимогам:

1. Перед видачою відео в оренду система встановлює особистість клієнта та перевіряє його репутацію шляхом сканування його членської картки відеомагазину.

2. Відеокасету чи диск можна просканувати, щоб отримати його опис та ціну після запити цієї відеопродукції клієнтом.

3. Клієнт сплачує номінальну плату до того, як відео можна взяти напрокат. Оплата може бути готівкою або кредитною картою.

4. Система перевіряє всі умови прокату відео, визнає, що транзакція може продовжуватися, та роздруковує квитанцію для замовника

Глосарій предметної області

Глосарій предметної області створюється на основі опису системи



автоматизації діяльності магазину відеопрокату і постановки завдання. Словник призначений для опису термінології області, в якій буде працювати програмне забезпечення. Виділяються терміни, їм дається опис, розрахований на широке коло читачів (користуватися цим описом будуть всі особи, зацікавлені в розробці системи). Словник складається українською мовою. Терміни супроводжуються перекладом на англійську на той випадок, якщо термін буде використаний в моделі системи як назва класу, пакета і т.д. Глосарій наведено в табл. 1.

Таблиця 1 – Глосарій предметної області.

Клієнт (Customer)	Особа, яка користується послугами відеомагазину, бере напрокат відеопродукцію та платить за це готівкою або кредитною карткою
Співробітник (Employee)	Користувач системи. Виконує сканування членської картки користувача, відеопродукції, роздруковує квитанцію для замовника видає відео напрокат та приймає назад від клієнтів
Відео носій (Video medium)	Матеріал, фізичні властивості якого використовуються для зберігання відео (наприклад, магнітна стрічка, компакт-диски, тощо)
Членська картка (membership card)	Картка, що видається постійним клієнтам. Дозволяє брати відеопродукцію в оренду без попередньої оплати

Діаграми варіантів використання

Функціональні вимоги до системи моделюються і документацію у вигляді варіантів використання (use case). Варіант використання (use case) - зв'язний елемент функціональності, що надається системою при взаємодії з діючими особами. Дійова особа (actor) - роль, виконувана елементами зовнішнього оточення системи, провідними себе по відношенню до системи однаковим чином.

В контексті роботи з вимогами варіанти використання трактуються наступним чином:

- варіант використання фіксує угоду між учасниками проекту щодо поведінки системи;
- варіант використання описує поведінку системи при різних умовах, коли система відповідає на запит одного з учасників, званого основною дійовою особою;
- основна діюча особа ініціює взаємодію з системою, щоб досягти своєї власної мети. Система відповідає, дотримуючись інтересів усіх дійових осіб (основного і другорядних).

Із постановки задачі та глосарію можна виділити список діючих осіб та їх інтересів:

Співробітник – сканує членську картку клієнта, сканує відео носій,



роздруковує квитанцію, приймає оплату

Клієнт – сплачує оренду або готівкою або кредитною картою.

Проектування програмного продукту здійснювалось за допомогою CASE засобу Visual Paradigm. Visual Paradigm(VP) - це система управління вимогами, що підтримує повний цикл розробки програмного продукту - аналіз, дизайн архітектури, розробку програмного коду, тестування і розміщення продукту на стороні замовника. VP також забезпечує підтримку версійності і одночасної роботи команди користувачів над одним проектом. Діаграма варіантів використання, розроблена у VP, наведена на рис. 1.

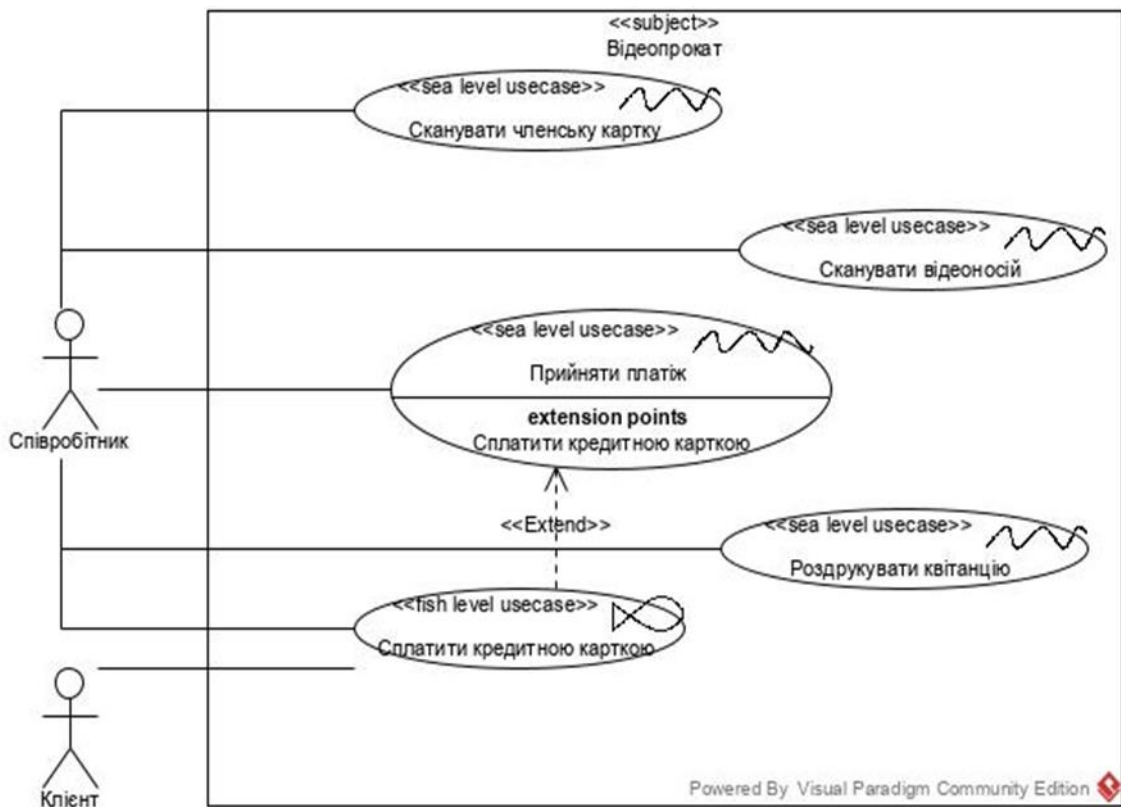


Рисунок 1 – Діаграма варіантів використання

На діаграмі у подвійних лапках вказується стереотип варіанту використання. Варіант використання зі стереотипом <<sea level usecase>> означає, що даний прецедент відноситься до «рівня моря», тобто відповідає меті одного з користувачів системи. Стереотип <<fish level usecase>> мають прецеденти, які можуть ініціювати декілька акторів.

Варіанти використання визначають на основі функціональних вимог до системи. Відповідність вимог варіантам використання(use case) та акторам наведено у табл. 2.

Таблиця 2 – Відповідність вимог варіантам використання

№ вимоги	Вимога	Актори	Варіанти використання
1	Перед видачою відео в оренду система встановлює особистість клієнта та перевіряє його репутацію шляхом сканування його членської картки відеомагазину	Співробітник	Сканувати членську картку
2	Відеокасету чи диск можна просканувати, щоб отримати його опис та ціну після запиту цієї відеопродукції клієнтом.	Співробітник	Сканувати відеоносій
3	Клієнт сплачує номінальну плату до того, як відео можна взяти напрокат. Оплата може бути готівкою або кредитною карткою.	Співробітник, Клієнт	Прийняти платіж, сплатити кредитною карткою
4	Система перевіряє всі умови прокату відео, визнає, що транзакція може продовжуватися, та роздруковує квитанцію для замовника	Співробітник	Роздрукувати квитанцію

Сценарії варіантів використання

Функціональні вимоги докладно фіксуються в описах варіантів використання. Описи складаються спеціальним чином, щоб зменшити ймовірність неправильного тлумачення і полегшити сприйняття тексту. Кожен опис включає в себе:

- а) короткий опис, що є стислим оглядом варіанти використання;
- б) основний потік подій, що описує взаємодію системи і дійових осіб, при якому досягається мета основної діючої особи;
- в) альтернативні потоки, що описують обробку помилок і виняткових



ситуацій;

г) підлеглі потоки, які полегшують опис основного і альтернативних потоків;

д) перед і постумови.

Кожен потік подій задається перенумерованим набором кроків. Використовуються кроки трьох типів: дія системи (наприклад, «Система запитує ім'я користувача та пароль»); реакція дійової особи («Користувач вводить ім'я і пароль»); управління потоком («Виконання переходить на початок основного потоку»). Структура пропозицій, що описують кроки, однакова: підмет і присудок, інші частини мови. Від неї відходять лише при описі циклів і розгалужень.

В табл. 3 наведено варіант використання «Прийняти платіж» з діаграми прецедентів на рис. 1.

Таблиця 3 – Сценарій варіанту використання «Прийняти платіж»

Варіант використання	Прийняти платіж
Короткий опис	Цей варіант використання дозволяє працівникові приймати плату від Клієнта за прокат відео. Оплата може бути здійснена готівкою кредитною картою.
Актори	Співробітник, Клієнт
Передумови	Клієнт бажає взяти напрокат відео, він / вона має дійсний членський квиток та відео, що він замовляє, доступне для прокату.
Основний потік	Випадок використання починається, коли Клієнт вирішить оплатити за прокат відео та пропонує оплату готівкою чи картою. Співробітник робить запит до системи, щоб відобразити плату за оренду разом з даними про клієнта та відео. Якщо Клієнт пропонує готівкову оплату, Працівник приймає готівку, повідомляє систему, що платіж отримано. Система зберігає дані про здійснення платежу. Якщо Клієнт пропонує сплатити кредитною картою, Співробітник зчитує інформацію з картки, просить Клієнта ввести PIN-код картки, вибрати кредитний рахунок і здійснити платіж. Після підтвердження сплати в електронному вигляді постачальником картки, система записує платіж як здійснений. На цьому прецедент закінчується

Варіант використання	Прийняти платіж
Альтернативні потоки	Клієнт не має достатньо коштів і не пропонує сплатити карткою. Співробітник запрошує систему перевірити рейтинг клієнта (отриманий з історії платежів замовника). Співробітник вирішує, чи брати давати відео в оренду без оплати або з частковою оплатою. В залежності від цього рішення Співробітник скасовує транзакцію (і варіант використання припиняється) або приймає часткову оплату (і варіант використання продовжується) Картка Клієнта не зчитується сканером. Після трьох невдалих спроб, Співробітник вводить номер картки вручну. Варіант використання продовжується.
Постумови	Якщо прецедент закінчився успішно, запис про платіж занесено до бази даних системи. Інакше, стан системи залишається незмінним.

2.2 Приклад розв'язання завдання 2

Завдання 2. Для одного із варіантів використання системи автоматизації діяльності магазину відеопрокату розробити діаграму діяльності.

Розв'язання.

Для побудови діаграми діяльності виділимо дії згідно варіантів використання. Дії, відповідні варіантам використання, наведені в табл. 4

Таблиця 4 – Дії, що відповідають варіантам використання

№ з/п	Зміст варіанту використання	Дії
1	Співробітник робить запит до системи, щоб відобразити плату за оренду разом з даними про клієнта та відео.	Показати подробиці транзакції
2	Якщо Клієнт пропонує готівкову оплату, Працівник приймає готівку, повідомляє систему, що платіж отримано. Система зберігає дані про здійснення платежу.	Прийняти задаток готівкою, Підтвердити транзакцію

№ з/п	Зміст варіанту використання	Дії
3	Якщо Клієнт пропонує сплатити кредитною картою, Співробітник зчитує інформацію з картки, просить Клієнта ввести PIN-код картки, вибрати кредитний рахунок і здійснити платіж. Після підтвердження сплати в електронному вигляді постачальником картки, система записує платіж як здійснений.	Зчитати картку, Прийняти номер картки, Вибрати рахунок картки, Підтвердити транзакцію
4	Клієнт не має достатньо коштів і не пропонує сплатити картою. Співробітник запрошує систему перевірити рейтинг клієнта (отриманий з історії платежів замовника). Співробітник вирішує, чи брати давати відео в оренду без оплати або з частковою оплатою. В залежності від цього рішення Співробітник скасовує транзакцію (і варіант використання припиняється) або приймає часткову оплату (і варіант використання продовжується)	Перевірити рейтинг клієнта, Відмовитись від транзакції, Дозволити оренду без оплати, Дозволити оренду з частковою оплатою
5	Картка Клієнта не зчитується сканером. Після трьох невдалих спроб, Співробітник вводить номер картки вручну. Варіант використання продовжується.	Ввести номер картки вручну

Діаграма діяльності для варіанта використання «Прийняти платіж» наведена на рис. 2.

2.3 Приклад розв'язання завдання 3

Завдання 3. На основі вимог до системи автоматизації діяльності магазину відеопрокату та глосарію цієї предметної області визначити класи-сутності та побудувати діаграму класів.

Розв'язання.

Класи-сутності виділяємо на основі вимог до системи та глосарію предметної області. Кожен термін глосарію може бути потенційним кандидатом на клас ключової абстракції. В табл. 5 наведені класи, визначені на основі вимог до системи.

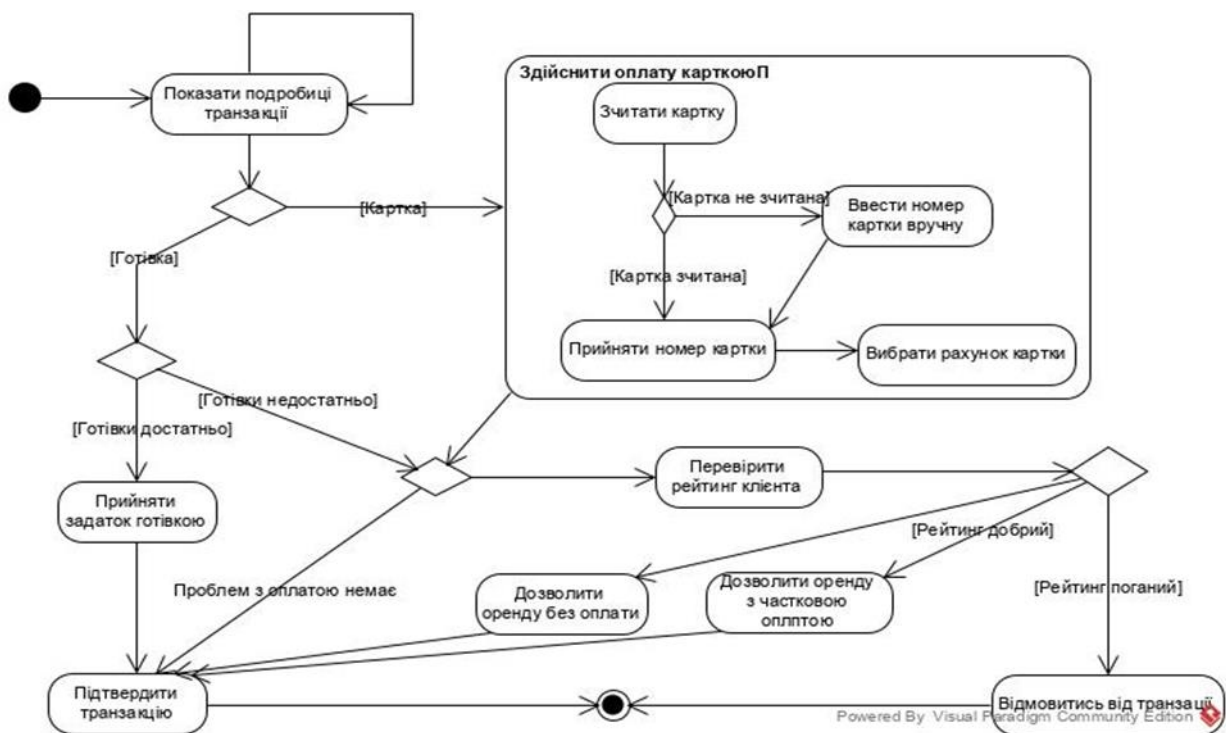


Рисунок 2 - Діаграма діяльності use case «Прийняти платіж»

Таблиця 5 – Визначення класів-сутностей системи

№ ВИМОГИ	Вимога	Клас сутність
1	Перед видачою відео в оренду система встановлює особистість клієнта та перевіряє його репутацію шляхом сканування його членської картки відеомагазину.	Video(Відео), Customer (Клієнт), MembershipCard (Членська картка)
2	Відеокасету чи диск можна просканувати, щоб отримати його опис та ціну після запиту цієї відеопродукції клієнтом.	Videotape (Відеострічка), VideoDisk (Відеодиск), Customer (Клієнт), Rental (Оренда)
3	Клієнт сплачує номінальну плату до того, як відео можна взяти напрокат. Оплата може бути готівкою або кредитною картою.	Customer (Клієнт), Rental (Оренда), Video(Відео), Payment(Оплата)
4	Система перевіряє всі умови прокату відео, визнає, що транзакція може продовжуватися, та роздруковує квитанцію для замовника	Rental (Оренда), Receipt (Квитанція)



При аналізі табл. 5 виникають наступні питання:

Яка різниця між класами Video та Videotape/Videodisk?

Чи є Video просто узагальненою назвою Videotape/Videodisk?

Якщо так, чи потрібно ввести клас Movie(Фильм) для опису відеоконтенту?

Введемо клас Movie. Клас Video буде батьківським класом для класів Videotape / Videodisk, що містить їх спільні властивості. Також необхідно додати клас Employee (Співробітник)

Визначимо атрибути класів-сутностей. На цьому етапі не всі атрибути класів можуть бути визначені. Атрибути класів наведені в табл. 6:

Таблиця 6 – Атрибути класів

Ім'я	Опис	Тип
Class Customer		
memberStartDate	Дата вступу в групу користувачів магазину відеопрокату	String(Строковий)
customerName	Ім'я клієнта	String(Строковий)
customerAddress	Адреса клієнта	String(Строковий)
customerPhone	Телефон клієнта	String(Строковий)
standingIndicator	Рейтинг клієнта. Приймає значення від «А» до «Е». Значення «А» відповідає самому високому рейтингу, «Е» - найнижчому	Char
Class Rental		
rentalDate	Дата початку оренди відеопродукції	Date(дата)
rentalDuration	Кількість днів оренди	int(цілий)
rentalCharge	Орендна плата	float (число з рухомою комою)
Class Payment		
paymentDate	Дата оплати	Date(дата)
paymentAmount	Сума оплати	float (число з рухомою комою)

Продовження таблиці 6

Ім'я	Опис	Тип
Class MembershipCard		
cardIssueDate	Дата випуску карти	Date(дата)
cardExpiryDate	Дата закінчення строку дії картки	Date(дата)
Class Receipt		
receiptDate	Дата видачі квитанції	Date(дата)
receiptAmount	Сума квитанції	float (число з рухомою комою)
Class Video		
videoCondition	Умови відеопрокату	Byte(байт)
numberCurrentlyAvailable	Число копій фільму, що наразі доступні	int(цілий)
Class Movie		
movieTitle	Назва фільму	String(Строковий)
director	Режисер фільму	String(Строковий)
isInStock	Знаходиться на складі	Boolean(Булевий)
Class Videotype		
isTapedOver	Чи була стрічка перезаписана	Boolean(Булевий)
Class Videodisk		
differentLanguages	Наявність різних мов	Boolean(Булевий)
differentEndings	Альтернативна фінальна сцена фільму	Boolean(Булевий)
Class Employee		
employeeName	Ім'я співробітника	String(Строковий)
employeeAddress	Адреса співробітника	String(Строковий)

Діаграма класів наведена на рис. 3.

На діаграмі на рис. 3 зображені асоціації між класами Video, Rental, Employee, Payment, Receipt, Customer, Movie. Асоціація показує, що об'єкти однієї сутності (класу) пов'язані з об'єктами іншої сутності. Якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. На кінцях асоціації вказана кратність ролі асоціації, яка означає, скільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці.

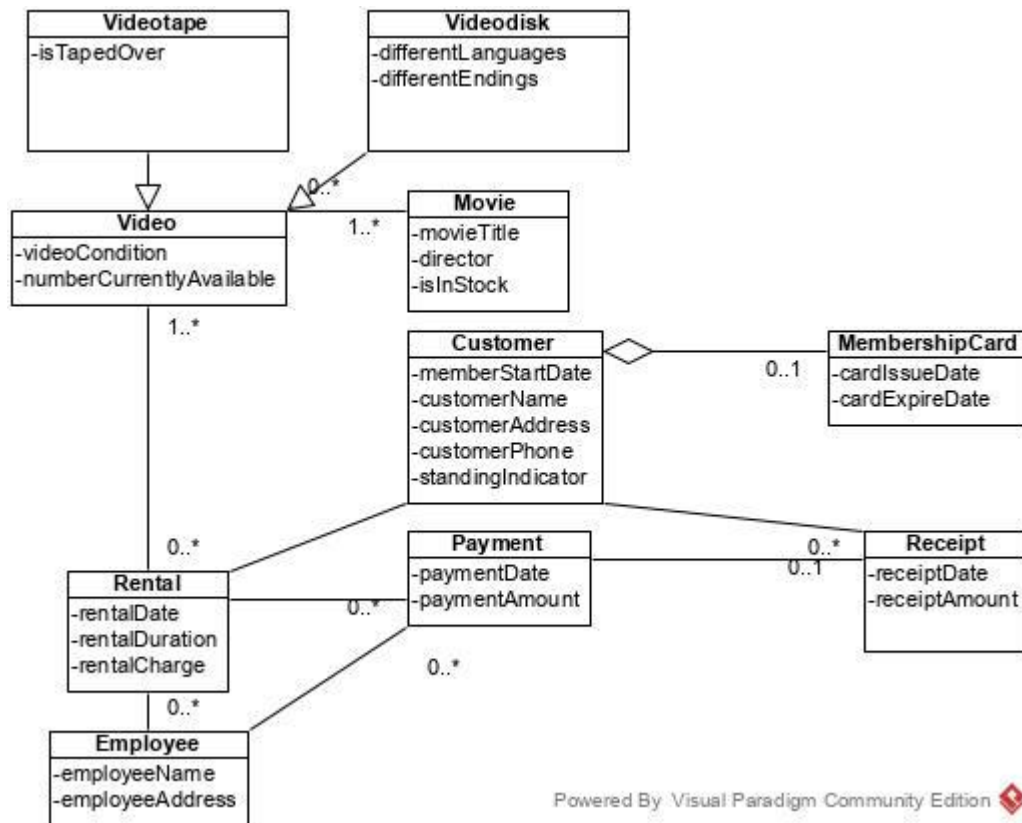



Рисунок 3 – Діаграма класів.

З діаграми на рис. 3 видно, що об'єкт класу Customer може бути зв'язаним з багатьма екземплярами класу Rental. Це означає, що клієнт може взяти декілька фільмів в оренду. Також кратність асоціації 0..* на кінці Payment між класами Rental і Payment означає, що оренда може бути оплачена декількома платіжками. Також можливо взяти в оренду відео без попередньої оплати. Це дозволяється альтернативним потоком в специфікації варіантів використання (табл. 3). Також на діаграмі (рис.3) показано, що класи VideoTape та Videodisk пов'язані з класом Video відношенням спадкоємства, тобто являються нащадками цього класу. Це відношення показано лінією з незакрашеною стрілкою на кінці. У даному випадку спадкування означає, що атрибути videoCondition та numberCurrentlyAvailable доступні екземплярам класів VideoTape та VideoDisk. Класи Customer та MembershipCard пов'язані між собою відношенням агрегації, що показано стрілкою з незакрашеним ромбом на кінці. Це означає, що Customer може бути членом групи клієнтів магазину відеопрокату.

2.4 Приклад розв'язання завдання 4

Завдання 4. Виконати аналіз одного з варіантів використання системи автоматизації діяльності магазину відеопрокату. Визначити класи аналізу, необхідні для реалізації цього варіанту використання і побудувати



послідовностей для одного з потоків подій із його сценарію.

Розв'язання. Ідентифікуємо класи аналізу. Класи аналізу відображають функціональні вимоги до системи і моделюють типи об'єктів. Сукупність класів аналізу являє собою початкову концептуальну модель системи. Ця модель проста і дозволяє зосередитися на реалізації функціональних вимог, не відволікаючись на деталі реалізації, забезпечення ефективності та надійності. Для вирішення цих питань згодом модель аналізу буде трансформована в проектну модель. В ході аналізу варіантів використання в їх потоках подій виявляються класи трьох типів:

граничні класи (boundary classes), що є посередниками при взаємодії системи з дійовими особами та з апаратною базою;

класи-сутності (entity classes), що відповідають за зберігання даних;

керуючі класи (control classes), які реалізують бізнес-логіку і забезпечують координацію поведінки об'єктів в системі.

Правило виділення граничних класів: для кожного зв'язку між дійовою особою і варіантом використання створюється або призначається граничний клас, який відповідає за дане взаємодія. Правило виділення класів-сутностей: класи-сутності - це, як правило, класи, що представляють ключові абстракції системи. Правило виділення керуючих класів: для кожного варіанту використання створюється відповідальний за його реалізацію клас управління. Створимо в пакеті Analysis Model граничний клас RentalWindow, призначений для організації взаємодії користувача з системою та клас CustomerVerifier, який є керуючим класом, що відповідає за бізнес логіку. Додамо в кооперацію AcceptPayment нову діаграму класів Accept Payment VOPC, яка включає класи, що використовуються у прецеденті «AcceptPayment». Діаграма класів показана на рис. 4. Клас RentalWindow має стереотип «boundary», клас CustomerVerifier - «control», інші класи мають стереотип «entity».

Додамо в кооперацію AcceptPayment нову діаграму послідовності Verify Customer Subflow. Діаграма послідовності показана на рис. 5. Діаграма містить об'єкти чотирьох класів Employee, RentalWindow, CustomerVerifier та Customer. Процес визначення рейтингу клієнта починається, коли об'єкт класу Employee посилає повідомлення checkRating. Після отримання цього повідомлення RentalWindow показує інформацію щодо рейтингу відповідного клієнта. Це означає, що RentalWindows об'єкт має посилання на релевантний Customer об'єкт. Відповідно, RentalWindow посилає повідомлення класу CustomerVerifier повідомлення verifyRating, аргументом якого є об'єкт Customer.

CustomerVerifier є керуючим об'єктом, відповідальним за програмну логіку. CustomerVerifier визиває власний метод findInCache для отримання покажчика на об'єкт Customer. Після цього він посилає повідомлення об'єкту Customer повідомлення getStandingInd() для отримання його рейтингу. Значення рейтингу повертається до RentalWindow. Після цього



Rental Window посилає повідомлення displayRating щоб показати рейтинг клієнта співробітнику.

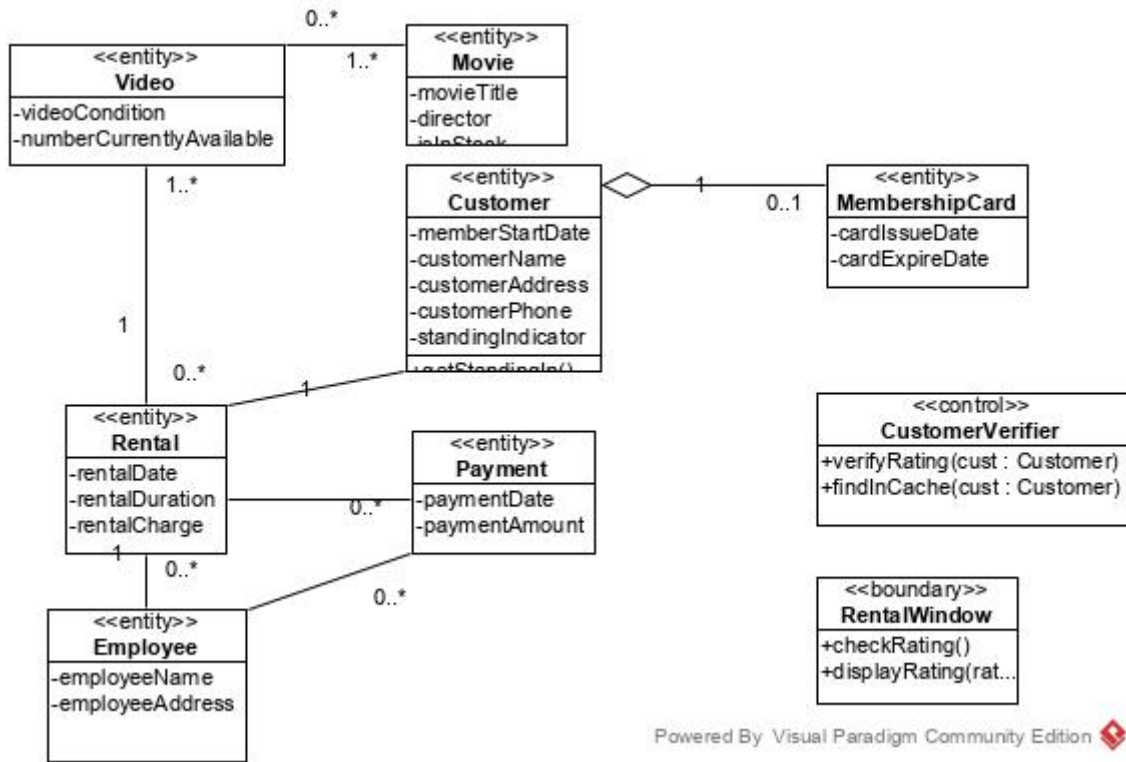


Рисунок 4 - Діаграма класів кооперації «AcceptPayment» AcceptPaymentVOPC.

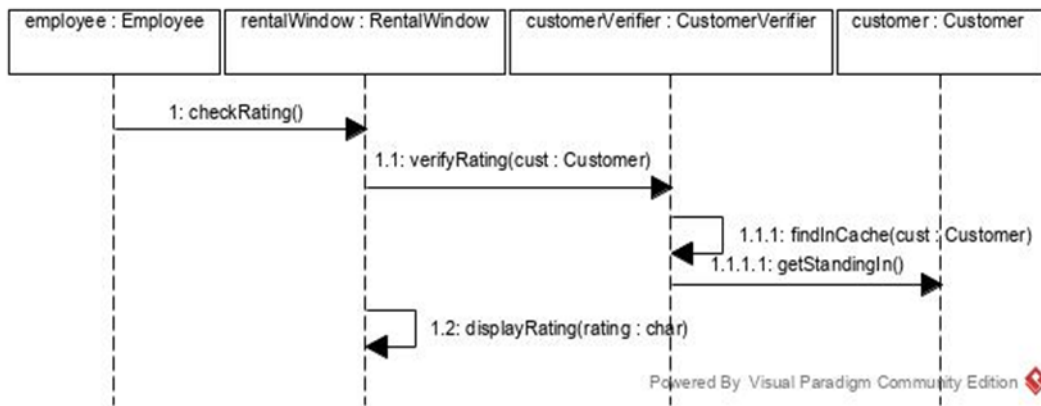


Рисунок 5 – Діаграма послідовності для діяльності «Verify customer» системи відеопрокату.

ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Авраменко В. С., Авраменко А. С. Проектування інформаційних систем. Черкаси : Черкаський національний університет ім. Б. Хмельницького, 2021. 434 с.
2. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) : навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / уклад.: О. С. Коваленко, Л. М. Добровська. Київ : КПІ ім. Ігоря Сікорського, 2020. 192 с.
3. Якість програмного забезпечення та тестування: базовий курс : навчальний посібник для бакалаврів галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення» / за ред. С. Я. Крепич, І. Я. Співак. Тернопіль : ФОП Паляниця В. А., 2020. 478 с.
4. Pressman R. Maxim B. Software Engineering: A Practitioner's Approach-9 th Edition. NY : McGraw-Hill Education, 2020. 705 p.
5. ISO12207:2008. Systems and software engineering – Software Life Cycle Processes. [Чинний від 2008-02-01] – II, 122 с.
6. IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology.
7. Software Engineering Body of Knowledge (SWEBOK). URL: <https://www.computer.org/education/bodies-of-knowledge/software-engineering> (дата звернення: 20.08.2024).
8. Unified Modeling Language 2.0. URL: <http://www.omg.org/spec/UML/2.0/> (дата звернення: 20.08.2024).
9. Maciaczek L. A. Requirement Analysis and System Design. Harlow, England : Pearson Education Limited, 2007. 464 p.
10. IEEE Std 1233-1996. Guide for Developing of System Requirements Specifications. URL: https://www2.seas.gwu.edu/~mlancast/cs254/IEE_STD_1233-Requirements_Spec.pdf
11. Холодняк Ю. С., Костіков А. А., Подлесний С. В., Капорович С. В. Розвиток комп'ютерних програм спрощеного розрахунку на міцність двотаврових балок. Вісник Донбаської державної машинобудівної академії, 2020. № 1 (48). С. 117- 127.
12. Kholodnyak Yu., Kostikov A., Podlesny S., Kaporovich S. The software enhancement for simplified strength computation of I-beam. Перспективні технології та прилади, 2019. № 17. С. 146- 53.
13. Software Quality: The Next Big Thing in Software Engineering and Quality. Cham : Springer Nature, 2022. 114 с. (Серія "14th International Conference on Software Quality, SWQD 2022, Vienna, Austria, May 17–19, 2022, Proceedings"). URL: <https://read.kortext.com/library/books/1921624>
14. Wooten E. Software Design 311 Success Secrets - 311 Most Asked Questions On Software Design - What You Need To Know. Emereo Publishing, 2014. 197 с. URL: <https://read.kortext.com/library/books/1320517>



15. Основи тестування програмного забезпечення : Prometheus : URL: https://prometheus.org.ua/course/course-v1:LITS+115+2017_T4 (дата звернення: 20.08.2024).
16. Вступ до програмної інженерії : Coursera : веб-сайт. URL: <https://www.coursera.org/learn/introduction-to-software-engineering> (дата звернення: 20.08.2024).
17. Інженерія програмного забезпечення : Coursera : веб-сайт. URL: <https://www.coursera.org/specializations/software-engineering> (дата звернення: 20.08.2024).
18. Життєвий цикл розробки програмного забезпечення : Coursera : веб-сайт. URL: <https://www.coursera.org/specializations/software-developmentlifecycle> (дата звернення: 20.08.2024).
19. Основи тестування та валідації програмного забезпечення : Coursera : вебсайт. URL: <https://www.coursera.org/learn/foundations-of-software-testing-andvalidation> (дата звернення: 20.08.2024).
20. Аудит, контроль та забезпечення якості інформаційних систем : Coursera : веб-сайт. URL: <https://www.coursera.org/learn/information-systems-audit> (дата звернення: 20.08.2024).
21. Міністерство освіти і науки України : веб-сайт. URL: <https://mon.gov.ua/> (дата звернення: 20.08.2024).
22. Національна бібліотека України ім. Вернадського. : веб-сайт. URL: www.nbuv.gov.ua (дата звернення: 20.08.2024).
23. Національна бібліотека України імені Ярослава Мудрого. : веб-сайт. URL: <https://nlu.org.ua/> (дата звернення: 20.08.2024).
24. Kortext : веб-сайт. URL: <https://kortext.com/> (дата звернення: 20.08.2024).
25. Research4life : веб-сайт. URL: <https://portal.research4life.org/> (дата звернення: 20.08.2024).
26. Інституційний репозитарій ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА» : веб-сайт. URL: <https://dspace.mipolytech.education/home> (дата звернення: 20.08.2024).
27. Центральна державна науково-технічна бібліотека гірничо-металургійного комплексу України : веб-сайт. URL: <http://cgntb.dp.ua/> (дата звернення: 20.08.2024).



Навчально-методичне видання

Костіков Олександр Анатолійович

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**методичні рекомендації до виконання
індивідуальних завдань**

Самостійне електронне мережеве видання
Публікується в авторській редакції