

Лабораторна робота 3

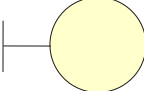


Дослідження процесу розробки програмного забезпечення. Архітектурне та детальне проектування.

Мета – отримати навички та єдині тактичні прийоми, якими повинні користуватися різні елементи системи.

Теоретичні відомості

Мета аналізу – дати опис поставленої задачі. Аналіз зосереджений не на формі, а на поведінці. У аналізі належить виявити модель, визначаючи класи і об'єкти (їх ролі, обов'язки та взаємодії), які формують модель предметної області (рис.1). Результатом аналізу є опис системи за допомогою сценаріїв. Щоб докладно описати очікувану поведінку системи, можна використовувати діаграми взаємодії, діаграми об'єктів та діаграми прецедентів. Класи аналізу завжди можна віднести до одного з типів: граничний, керуючий або сутності (табл.1).

Таблиця 1

Позначення стереотипу	Назва стереотипу	Опис
	<<boundary>>	Граничний елемент (рис.1) – абстракція, яка взаємодіє з зовнішнім оточенням системи: – виділяється для кожного зовнішнього агента (актора) для взаємодії з ним.
	<<entity>>	Сутність (рис.2) – абстракція для зберігання інформації: – в назві використовуються терміни з предметної області; – пасивна сутність; – ідентифікуються в потоці даних на діаграмах діяльності; – найчастіше використовуються для створення таблиць бази даних.
	<<control>>	Керуючий елемент (рис.3) – інкапсулює керування системою та прецедентами: – активна абстракція; – не має власної функціональності; – делегує відповідальність іншим класам та забезпечує ведення бізнес-правил.

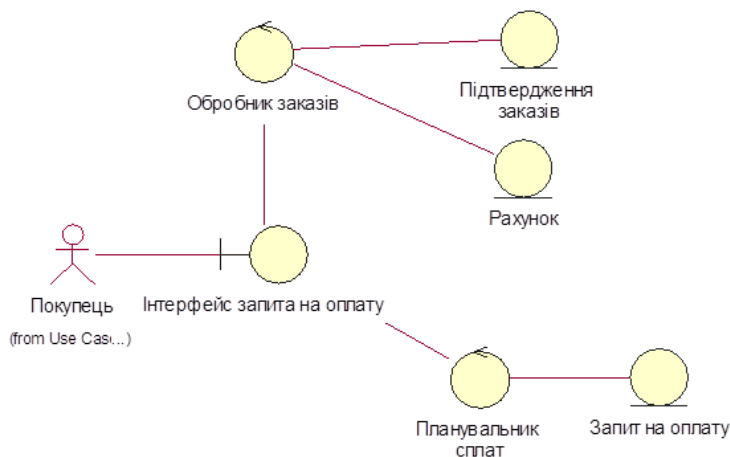


Рис. 1. Діаграма класів аналізу варіанту використання *Оплатити рахунок*

Мета проектування – створити архітектуру та єдині тактичні прийоми, якими повинні користуватися різні елементи системи.

Модель проектування — це об'єктна модель, яка описує фізичну реалізацію варіантів використання. В цій моделі основна увага надається тому, яким чином функціональні і нефункціональні вимоги разом з іншими обмеженнями, що відносяться до середовища реалізації, складають систему. Крім того, модель проектування є абстракцією реалізації системи і використовується як початкові дані для реалізації.

В моделі проектування варіанти використання реалізуються у вигляді класів проектування і їх об'єктів (рис.2). Ця реалізація представляється у вигляді кооперації і називається *проектом реалізації варіанту використання*.

Проект реалізації варіанту використання містить наступне:

1. текстовий опис потоку подій;
2. діаграми класів, які описують класи проектування, що приймають участь у реалізації варіанту використання;
3. діаграми взаємодії, які описують реалізацію конкретного потоку або сценарію у поняттях взаємодіючих об'єктів проектування.

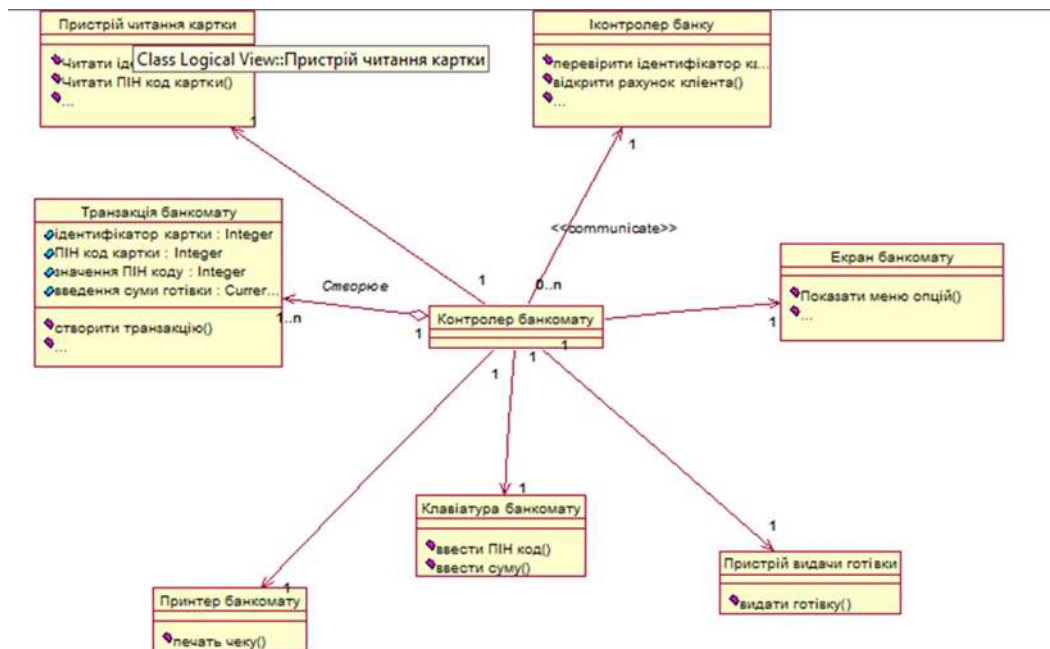


Рис 2. Модель проектування

Клас проектування — це найбільш наближена до реальності абстракція класу або схожій конструкції реалізації системи. Семантика класу наведена на рисунку 3. Ця абстракція є найближчою до реальності із наступних причин:

- Мова, що використовується для опису класу проектування, однакова з мовою програмування для реалізації. Відповідно, операції, параметри, атрибути, типи та інші подробиці визначаються з використанням синтаксису вибраної мови програмування.
- Часто задається видимість атрибутів і операцій класу проектування. Наприклад, в мові C++ для цього використовуються ключові слова *public*, *protected* і *private*.
- Відносини з іншими класами, в яких бере участь даний клас проектування, часто одержують явний вираз при реалізації цього класу. Наприклад, узагальнення у всіх стереотипах узагальнення має семантику, яка відповідає узагальненню (або наслідуванню) у мові програмування. Таким чином, узагальнення і агрегації часто відображаються на відповідні змінні (атрибути) реалізації, що відповідають посиленням на об'єкти.

- Методи (тобто реалізації операцій) класу проектування прямо відображаються на відповідні методи класів реалізації (тобто код). Методи, що визначаються в ході проектування, часто визначаються на природній мові або на псевдокоді і можуть бути у такому вигляді використані в анотації до реалізації цих методів.
- Клас проектування часто задається стереотипом, який напряду відображається в конструкцію відповідної мови програмування.
- Клас проектування може бути реалізований — і реалізується — у вигляді інтерфейсу, якщо це поняття існує у вибраній мові програмування.
- Клас проектування може бути активним. Це означає, що об'єкти класу використовуватимуть свій власний потік керування, працюючи паралельно з іншими активними об'єктами. Проте зазвичай класи проектування неактивні. Це означає, що всі їх об'єкти «запускаються» в єдиному адресному просторі під керуванням іншого активного об'єкту. Деталі семантики цього процесу залежать від мови програмування і технологій паралельної або розподіленої обробки, що використовуються в ньому.

Опис класів	
Властивість класу проектування	Опис
1	2
Назва	GameController
Відношення	Відношення асоціації від класу SceneMenu та SceneGame.
Операції	Menu_sceneGameWithKindOfGame(kind : integer) : void Game_sceneMenu() : void CreateMainMenu() : void update() : void GenerateCards() : void GenerateDots() : void
Атрибути	Власні.
Назва	SceneMenu
Відношення	Відношення асоціації до класу GameController. Відношення композиції від класу MenuItem.
Операції	ShowAnimated (isAnimated : bool) : void HideAnimated (isAnimated : bool) : void GameSelected (kindOfGame : integer) : void CreateMenuItem() : void eventAnimationEnd() : void onEnterTransitionDidFinish() : void
Атрибути	countOfGames : integer items[] : MenuItem
Назва	MenuItem
Відношення	Відношення композиції до класу SceneMenu.
Операції	ShowAnimated (isAnimated : bool) : void HideAnimated (isAnimated : bool) : void eventMenuItemSelected() : void eventCreationSuccessful() : void eventAnimationEnded() : void
Атрибути	anOfCards[] : Card typeOfGame : integer
Назва	SceneGame
Відношення	Відношення композиції із класом CardNode та DotNode. Відношення агрегації із класами FacebookController, EmailController та Narrations.
Операції	ShowAnimated (isAnimated : bool) : void HideAnimated (isAnimated : bool) : void ShowCongratulations() : void ShowNarrations(text : string, sound : string) : void CalculateTouchLocation() : void setCurrentCard(c : CardNode) : void getCard() : CardNode isCardAndDotEquals() : bool CreateEmailController() : void CreateFacebookController() : void eventAnimationsEnded() : void ShowAlert(text : string) : void eventAnimationEnded() : void
Атрибути	Cards[] : CardNode Dots[] : DotNode

Рис. 3 Семантика (опис) класу.

Один з результатів проектування – архітектура системи. Архітектура об'єктно-орієнтованої системи виражає структуру класів і об'єктів. Тому здебільшого використовуються діаграми класів та об'єктів.

Завдання

1. Побудувати модель аналізу реалізації варіантів використання системи, що розробляється (лабораторна робота 2).
2. Побудувати модель проектування, яка містить проект реалізації варіантів використання, основна частина яких трасується з аналізу реалізації варіантів використання.
3. Висвітлити семантику класів та об'єктів, які застосовуються для чіткого розподілу обов'язків між частинами реалізації.
4. Уточнити діаграму класів для відображення проектних рішень стосовно узагальнення, агрегації, використанні, реалізації.
5. Визначити архітектуру системи, що розробляється.
6. Представити модель проектування та проектних рішень у результуючій діаграмі класів, об'єктів та діаграмі пакетів.

Контрольні питання