Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Звіт з лабораторних робіт

з дисципліни

«Моделювання програмного забезпечення»

Виконав:

студент 3-го курсу

групи ІП-53, ФІОТ

Сулима Олександр

Сергійович

Київ 2017

# Лабораторна робота № 1

Створення діаграми прецедентів

**Мета роботи:** отримати навички побудування діаграм прецедентів.

**Завдання:**

1. Створити головну діаграму прецедентів, вказавши на ній варіанти використання та акторів.
2. Додати відношення між акторами та варіантами використання.
3. Створити додаткову діаграму прецедентів.
4. Додати опис до акторів та варіантів використання.
5. Для кожного варіанту використання створити потік подій у вигляді окремого файлу та додати його до варіанту використання.

**Зміст звіту:**

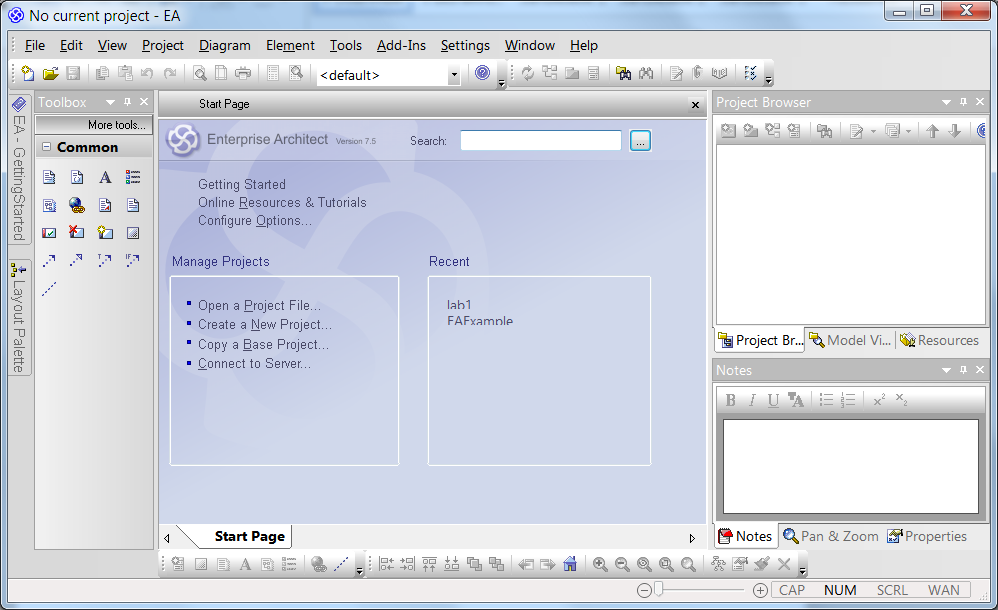
1. Створені діаграми прецедентів.
2. Скорочений опис кожного актору та прецеденту.
3. Опис потоку подій для кожного варіанту використання.

Виконання роботи

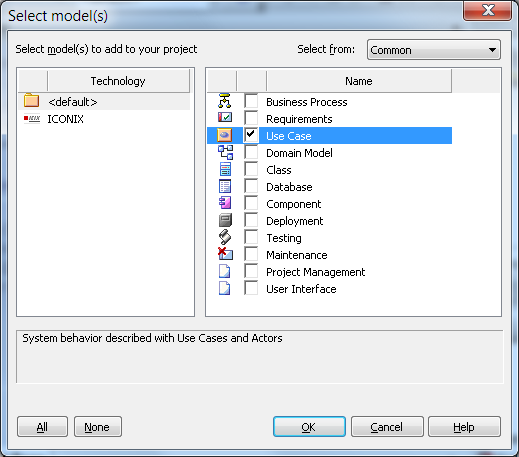
Предметна область: «Заклад швидкого харчування».

## Початкові дії – створення нового проекту.

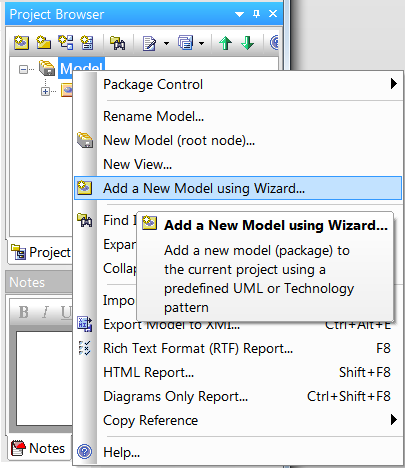
Робота з Enterprise Architect починається зі створення проекту. При запуску програми на екрані з’являється головне вікно програми.



Для створення нового проекту необхідно натиснути File-New Project та вказати шлях збереження проекту. Після цього програма автоматично запропонує обрати тип початкових діаграм проекту. Для створення діаграми прецедентів позначте пункт «Use Case».

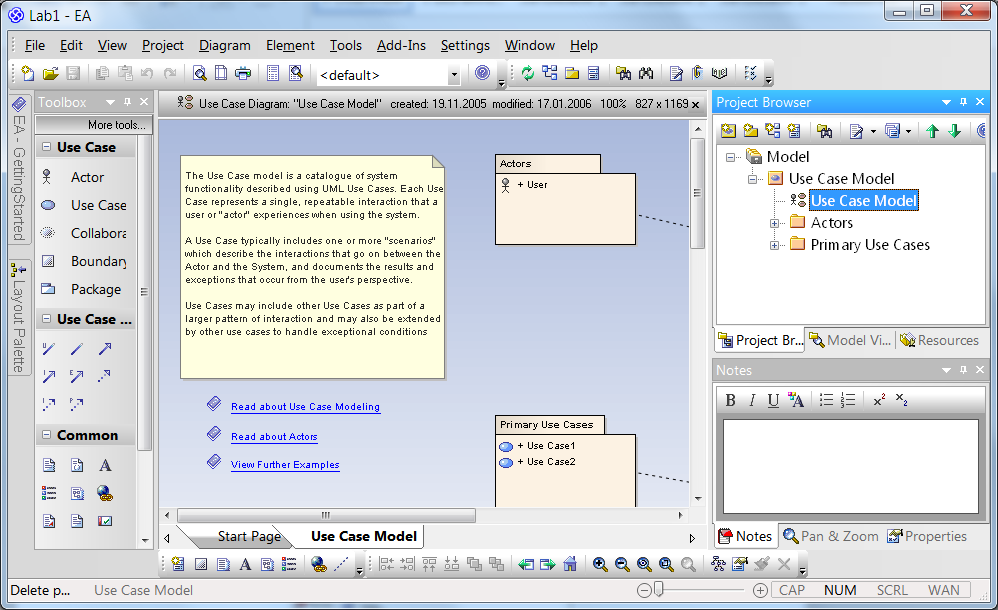


Для того, щоб додати нову діаграму до вже існуючого проекту, необхідно у вікні «Project Browser», що розташоване у правій частині головного вікна, натиснути правою кнопкою миші по проекту та у контекстному меню обрати пункт «Add a new model using wizard».



## Заповнення діаграми прецедентів (Use Case)

Щоб відкрити діаграму для редагування необхідно у вікні «Project browser» двічі натиснути на необхідну діаграму. Обрана діаграма розгорнеться у центральній частині головного вікна. Кожна створена діаграма за замовчанням містить довідникову інформацію, яку можна видалити з діаграми без будь-яких наслідків.



Для обраної предметної області (виробництво та оптова продаж меблів) знадобляться наступні дійові особи (Actor):

1. Робітник(кухня) – співробітник, який слідкує за наявністю запасів продукції та приймає замовлення до виконання.
2. Робітник(сервіс) – співробітник, який спілкується з відвідувачами, реєструє замовлення, проводить збір коштів та видає готове замовлення відвідувачу.
3. Інструктор – співробітник, який кожного дня отримує від менеджера зміни новий робочий план та надає необхідні інструкції робітникам, також інструктор повідомляє менеджера у разі виникнення незвичайних ситуацій.
4. Бухгалтер – співробітник, який веде документацію та своєчасно перераховує заробітню плату всім робітникам. Також у разі “невиходу” робітника на зміну - проводить оформлення лікарняного.
5. Менеджер зміни – співробітник, який проводить контроль виробництва і слідкує за якістю продукції, вносячи зміни у приготування замовлення(за потребою).
6. Менеджер ресторану – директор компанії, слідкує за діями менеджерів і корегує їх роботу. Займається піаром ресторану.

Розглянемо можливості, які повинна задовольняти автоматизована система:

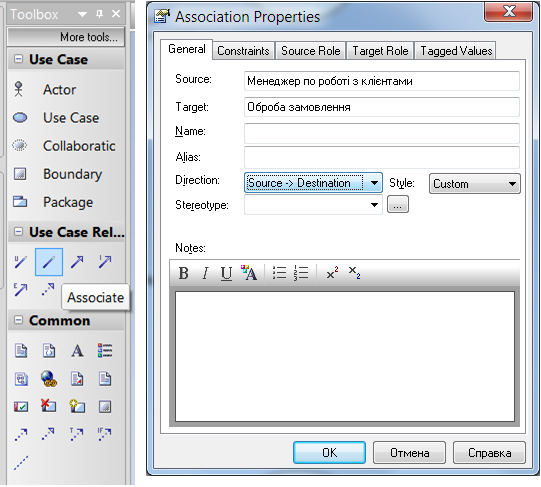
1. Робітник кухні використовує систему для перегляду актуальних замовлень та готує продукцію. Якщо таких замовлень немає - проводить поповнення запасів.
2. Робітник сервісу чекає на відвідувачів, у разі замовлення - надсилає перелік з замовлення робітнику кухні і проводить збір коштів. Як тільки замовлення готове до видачі - робітник сервісу пакує його та видає відвідувачу.
3. Інструктор проводить остаточну перевірку цілосності замовлення та організовує робітників згідно з робочим планом.
4. Бухгалтер не з’являється на робочій станції. В свою чергу він організовує всю роботу з документами. У разі коректної роботи бухгалтера - всі робітники отримують заробітню плату вчасно і без затримок.
5. Менеджер зміни час від часу заходить на кухню та проводить ревізію продукції та підтримання чистоти. За порядок та контроль виробництва перед менеджером ресторану відповідає саме менеджер зміни.
6. Менеджер ресторану являється директором закладу. Він робить все можливе, аби заклад приносив гарний дохід.

На базі вищесказаного можна виділити наступні прецеденти:

|  |  |
| --- | --- |
| Робітник(кухня) | |
| Обробка та приготування замовлення | Готує замовлення. |
| Своєчасне поповнення запасів | Перевіряє та поповнює запаси продуктів та сповіщує інструктора в разі недостачі. |
| Робітник(сервіс) | |
| Прийом замовлення | Перегляд та затвердження запитів відвідувача. |
| Збір коштів | Збір коштів відвідувача згідно замовленню. |
| Видача замовлення | Збір та видача замовлення відвідувачу. |
| Отримання робочого плану | Вивчення робочого плану, отримуваного від інструктора. |
| Інструктор | |
| Отримання робочого плану | Ведення списку наявних матеріалів та інструментів |
| Перевірка цілосності замовлення | Створення та редагування запитів щодо придбання продуктів відвідувачем. |
| Бухгалтер | |
| Оформлення лікарняних | Створення та редагування робочого плану робітників, оформлення лікарняних бланків. |
| Видача зарплат | Затвердження зарплат, переведення на рахунки робітників. |
| Менеджер зміни | |
| Якість продукції | Перевірка якості продуктів на кухні. |
| Контроль виробництва | Контролювання якості виробництва. |
| Менеджер ресторану | |
| Ревізія | Ревізія всіх галузей ресторану. |

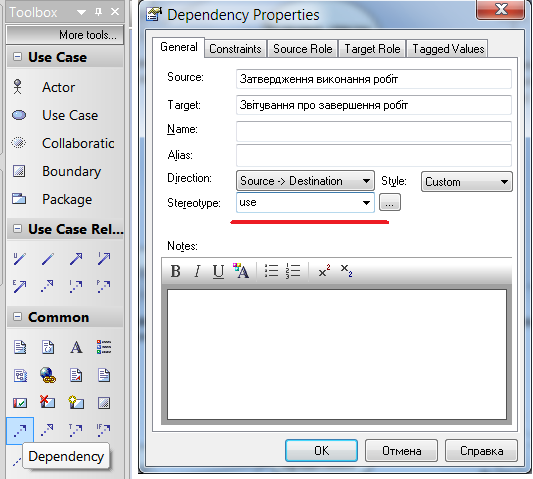
Для створення діаграми використовуються елементи Actor (діюча особа) та Use Case (Прецедент). Ці елементи розташовані у лівій частині головного вікна програми.

Для зв’язування акторів з їх діяльністю слід використовувати стрілку «associate», що розташована на панелі «toolbox». Щоб задати напрям стрілки необхідно після встановлення зв’язку двічі натиснути мишею по стрілці на діаграмі та задати пункт Direction

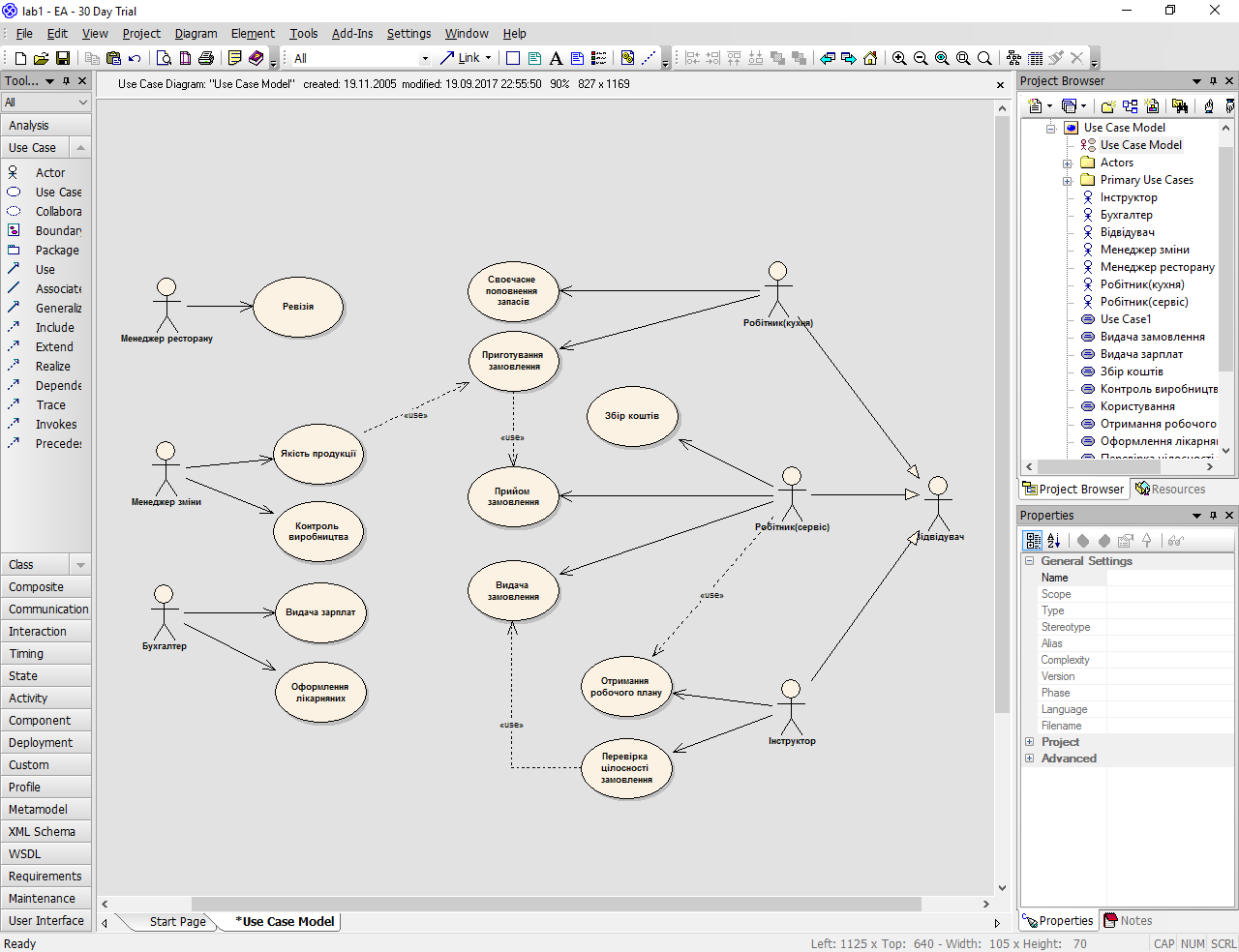


Для того, щоб вказати, що даний актор є підтипом іншого актору використовується стрілка «Generalize»/ Вона розташована поряд зі стрілкою «associate».

Для зауваження залежності одного прецеденту від іншого можна скористуватися стрілкою «dependency», задавши її тип як «Use».



Готова діаграма має наступний вигляд:



## Потоки подій для заданих прецедентів.

Потоки подій для прецеденті будуть описуватися наступним чином:

* X.1 передумова
* X.2 головний потік
* X.3 під-потік
* X.4 альтернативні потоки
* X.3 пост-умови

X – число від 1 до кількості прецедентів

Потік подій для прецеденту «Обробка замовлення»:

1. Передумова

Якщо замовлення оформлюється на нового клієнта, то під-потік «Додати нового клієнта» повинен здійснюватися до початку цього етапу.

1. Головний потік

Прецедент починає виконуватися , коли менеджер підключається до системи і вводить своє ім'я і пароль. Система перевіряє правильність пароля (Е- 1) і виводить можливі варіанти дій: «Нове замовлення», «Змінити замовлення», «Видалити замовлення».

Якщо обрана операція «Нове замовлення» , S-1: виконується потік «Додавання нового замовлення».

Якщо обрана операція «Змінити замовлення» S-2: виконується потік «Редагування замовлення».

Якщо обрана операція «Видалити замовлення», S-3: виконується потік «Видалення замовлення».

1. Під-потік

S-1: «Додавання нового замовлення».  
        Система відображає каталог, в якому робітник сервісу повинен вибрати все, що скаже відвідувач. Система відображає поле для вибору клієнта і список можливих страв. Робітник заповнює бланк замовлення (E-2). Система запам'ятовує введені дані, передає їх робітнику на кухні і роздруковує рахунок для оплати. Потім прецедент починається спочатку.  
        S-2: «Редагування замовлення».  
        Система відображає діалогове вікно, що містить список замовлень і поле для введення номера замовлення . Інструктор вибирає необхідне замовлення зі списку або вводить номер замовлення в поле (Е-3)​​. Система відображає інформацію про дане замовлення. Інструктор робить необхідні зміни (Е-2). Система запам'ятовує введені дані. Потім прецедент починається спочатку.  
        S-3: «Видалення замовлення».  
        Система відображає діалогове вікно, що містить список замовлень і поле для введення номера замовлення . Менеджер вибирає необхідне замовлення зі списку або вводить номер замовлення в поле (Е-3)​​. Система видаляє вказане замовлення (Е-4). Потім прецедент починається спочатку.  
        Альтернативні потоки

Е-1: відвідувач передумав над замовленням; Повинен повторити введення або завершити прецедент.

Е-2: вказаної страви немає в наявності. Інструктор повинен змінити страву або завершити прецедент.

E-3: введено неправильний номер замовлення. Інструктор повинен повторити введення або завершити прецедент.

Е-4: система не може видалити замовлення. Інформація зберігається, система видалить замовлення пізніше. Виконання прецеденту триває.

Потік подій для прецеденту «Ревізія»

1. Головний потік

Прецедент починає виконуватися, коли менеджер ресторану підключається до системи і вводить своє ім'я і пароль і обирає пункт “меню менеджера”. Відкриваються всі відеокамери ресторану. Таким чином менеджер слідкує і перевіряє всі робітників ресторану.

1. Під-потік

S-1: «Централізувати камеру»  
        Система відображає діалогове вікно, що містить тільки дану камеру. Менеджер заповнює зазначені поля (Е-1). Система запам'ятовує введені дані. Потім прецедент починається спочатку.

S-2: «Позначити»  
        Система ставить мітку на робітнику. Система виписує чек-сумму, яка потрапляє до інструктора для прийняття мір. Потім прецедент починається спочатку.

1. Альтернативні потоки

Е-1: введено неправильне ім'я або пароль. Користувач повинен повторити введення або завершити прецедент.

Питання

1. Можливі варіанти зв’язків між прецедентами.

* У мові UML визначено такі типи відносин: залежність , асоціація , узагальнення і реалізація . Ці відносини є основними єднальними конструкціями UML і також як сутності застосовуються для побудови моделей. Залежність (Dependency) – це семантичне відношення між двома сутностями, при якому зміна однієї з них, незалежної, може вплинути на іншого. Асоціація (Association) – структурне ставлення, що описує сукупність смислових або логічних зв'язків між об'єктами. Узагальнення (Generalization) – це відношення, при якому об'єкт нащадок може бути підставлений замість об'єкта предка. При цьому, відповідно до принципів об'єктно-орієнтованого програмування, нащадок успадковує структуру і поведінку свого предка. Реалізація (Realization) є семантичним відношенням між класифікаторами, при якому один класифікатор визначає зобов'язання, а інший гарантує його виконання.

2) Що таке нотації і метамодель в UML?

* Нотація - сукупність графічних об'єктів, які використовуються в моделях. Як приклад на діаграмах показано, як в нотації діаграми класу визначають поняття і предмети типу «клас», «ассоціацція», «множинність» і т. д.  
  Метамодель - діаграма, що визначає нотацію. Метамодель допомагає зрозуміти, що таке добре організована і синтаксично правильна, модель.

3) http://alice.pnzgu.ru:8080/~zsa/sql/titan\_zsa/uml\_htm\_gol/gl\_02.htm