

ОСНОВИ МОДЕЛЮВАННЯ ПОВЕДІНКИ. ЕЛЕМЕНТИ ГРАФІЧНОЇ НОТАЦІЇ ДІАГРАМИ ДІЯЛЬНОСТІ

Мета

Набути практичних навичок побудови діаграм діяльності (Activity Diagram) UML для моделювання поведінки системи або бізнес-процесів.

Завдання

- Ознайомитися з графічними елементами діаграм діяльності та їх призначенням.
- Розробити власну діаграму діяльності для конкретного процесу чи сценарію.
- Відпрацювати навички визначення послідовності дій, розгалужень та паралельних потоків у процесі.
- Навчитися аналізувати поведінку системи через візуальне моделювання.
- Підвищити вміння використовувати UML-нотацію для документування процесів та комунікації в команді.

Час

4 години.

Література:

Запитання для самоперевірки до/після лабораторної роботою

Стан дії

- Що таке стан дії в UML?
- Яка його роль у моделюванні поведінки системи?

Переходи

- Що показують переходи між станами на діаграмі діяльності?
- Які типи переходів існують?

Доріжки (секції)

- Що таке доріжки (swimlanes) на діаграмі діяльності?
- Для чого вони використовуються?

Об'єкти на діаграмі активності

- Що відображають об'єкти на діаграмі діяльності?
- Які приклади об'єктів можна навести?

Графічне зображення класів

- Які основні елементи класу зображуються на UML-діаграмі класів?
- Як графічно показуються атрибути та методи класу?

Визначення атрибутів

- Що таке атрибути класу?

- Які правила їх визначення у UML?

Типи класів

- Які основні типи класів існують у UML?
- Чим вони відрізняються один від одного?

Визначення методів

- Що таке методи класу?
- Яким чином вони зображуються на UML-діаграмі?

Теоретичні відомості

Перелік елементів діаграм активності / стану в plantUML

Елемент	Позначення PlantUML	Дефініція
Початок (Initial Node)	start	Позначає початок процесу чи сценарію.
Кінець (Final Node)	stop	Позначає завершення процесу чи сценарію.
Дія (Action / Activity)	:Назва дії;	Конкретна операція або крок у процесі, наприклад, «Додати рядок у OrderLine».
Рішення / розгалуження (Decision Node)	if (умова) then (так) ... else (ні) endif	Умова, за якою відбувається розгалуження потоку управління.
Об'єкт / результат (Object Node)	:Об'єкт;	Дані або об'єкт, що бере участь у процесі (наприклад, Order, OrderLine).
Доріжка (Swimlane / Partition)	`	Роль
Потік управління (Control Flow)	Стрілки автоматично при переході між діями	Вказує послідовність виконання дій.
Паралельний потік (Fork / Join)	fork ... fork again ... end fork	Позначає одночасне виконання кількох дій або об'єднання потоків.

Елемент	Позначення PlantUML	Дефініція
Початковий стан (Initial State)	[*]	Позначає початковий стан об'єкта.
Кінцевий стан (Final State)	[*] на кінці	Позначає завершення життя об'єкта або процесу.
Стан (State)	StateName	Конкретний стан об'єкта у системі, наприклад, Pending, Confirmed.
Перехід (Transition)	State1 --> State2 : подія / операція	Перехід між станами, що відбувається через певну подію або операцію.
Внутрішня дія (Entry / Exit / Do)	StateName : action	Дія, що виконується при вході (entry), виході (exit) або під час перебування у стані (do).
Розгалуження / об'єднання (Choice / Merge)	Через умовні переходи --> з if у PlantUML	Логічне розгалуження переходів між станами.
Підстан (Substate / Composite State)	state CompositeState { ... }	Групування станів у більш складний стан, що дозволяє деталізувати поведінку.

Завдання:

Завдання 1. Розглянути наведений приклад та проаналізувати приклади діаграми активності та відповідність станів та дій зображеніх на діаграмі UML до функцій системи.

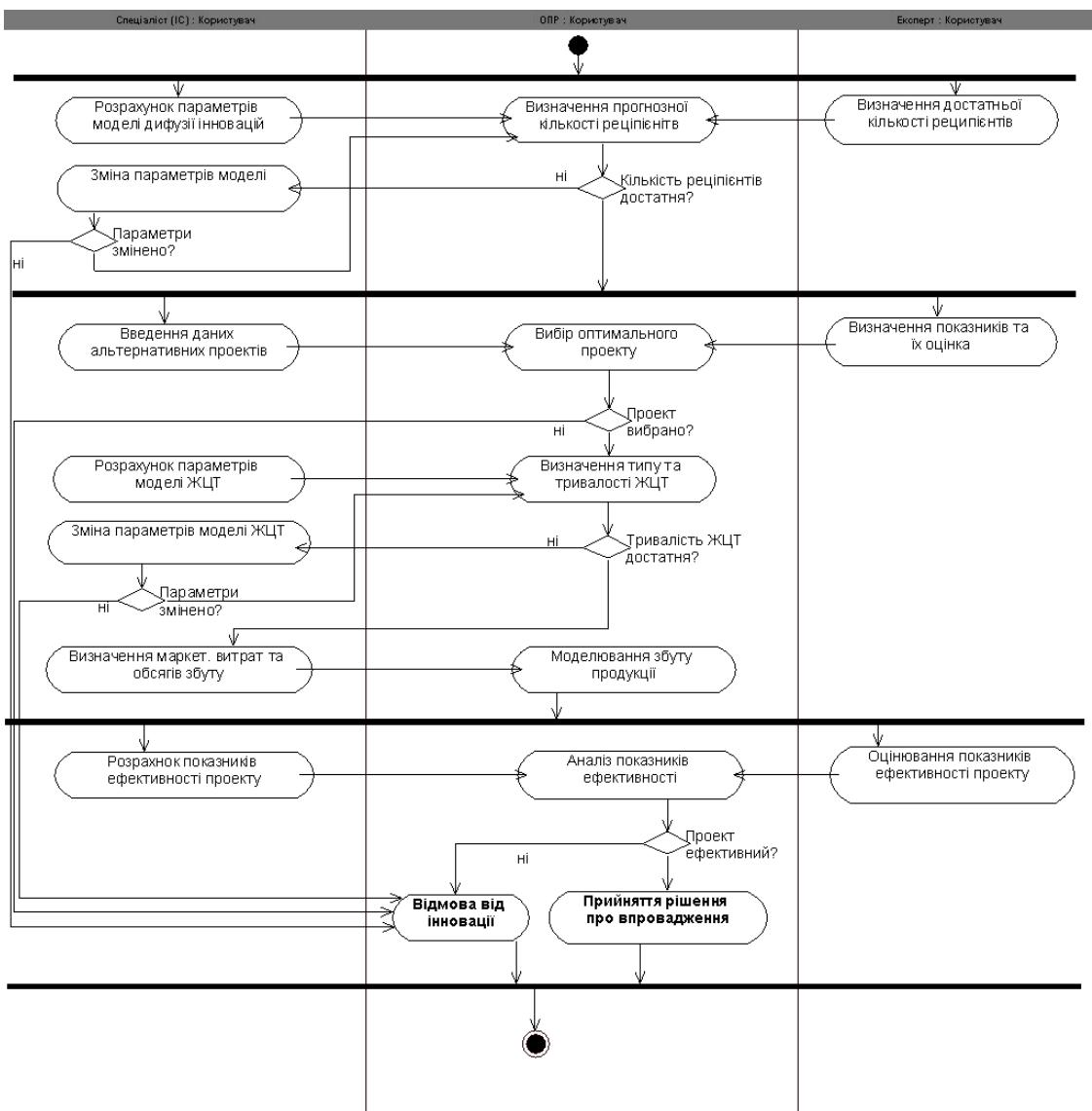


Рис. 1. Діаграма дій системи для процесу прийняття рішень щодо впровадження інноваційного продукту



Рис. 2. Діаграма дій процесу контролю впровадження інновації

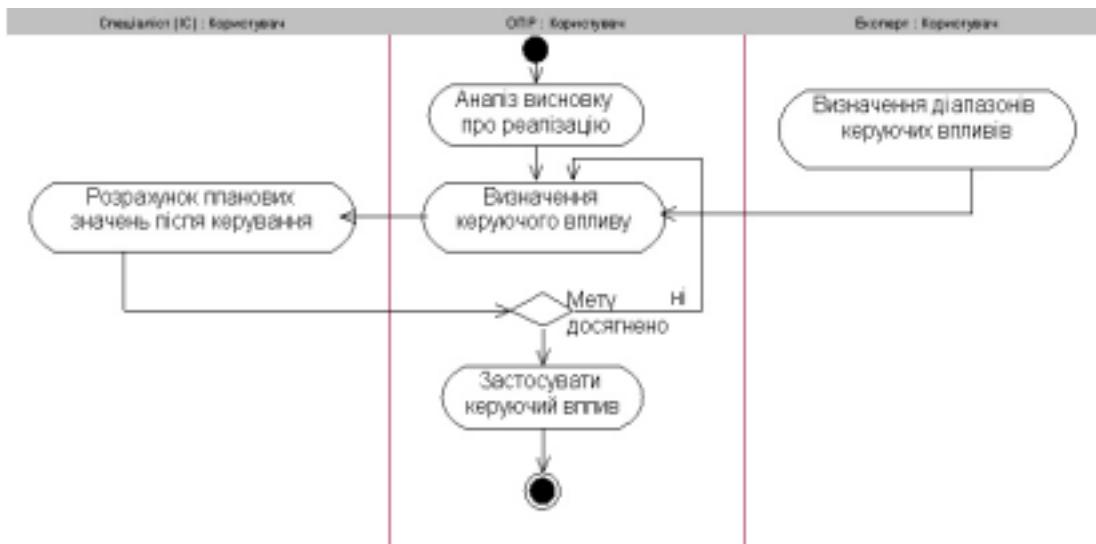


Рис. 3. Діаграма дій процесу керування впровадженням інновації

Завдання 2. Побудувати діаграми активності системи за власним аріантам. Для цього:

Аналіз готової Use Case діаграми

- Ознайомитися з наявною діаграмою Use Case.
- Визначити ключові сценарії (варіанти використання), які потрібно деталізувати у діаграмі активності.

2. Визначення доріжок виконання (Swimlanes)

- Визначити ролі користувачів або підсистеми, які взаємодіють із системою.
- Для кожної ролі створити відповідну доріжку на діаграмі активності.
- Доріжки відображають, хто виконує яку дію.

3. Визначення станів та дій системи

- Виділити дії або стани системи, що відповідають **реальним алгоритмам роботи програмного забезпечення**.
- Під «реальними алгоритмами» розуміються **конкретні операції та функції, які виконує система на рівні об'єктів та бази даних**, наприклад:

- Не просто «додати продукт», а «зберегти новий рядок у об'єкті Замовлення в БД», з вказанням структури даних (назва продукту, кількість, ціна).
- **Оновлення стану замовлення** після підтвердження оплати.
- **Видалення запису** або зміну статусу об'єкта у системі.

- Позначити початкові та кінцеві стани.

4. Побудова послідовності дій системи

- Побудувати діаграму активності на основі сценаріїв Use Case, відобразивши **логічну послідовність виконання операцій системи**.
- Використати розгалуження (Decision), паралельні потоки (Fork/Join) за потреби.
- Визначити потоки управління та взаємодію між доріжками.

- Приклад результату

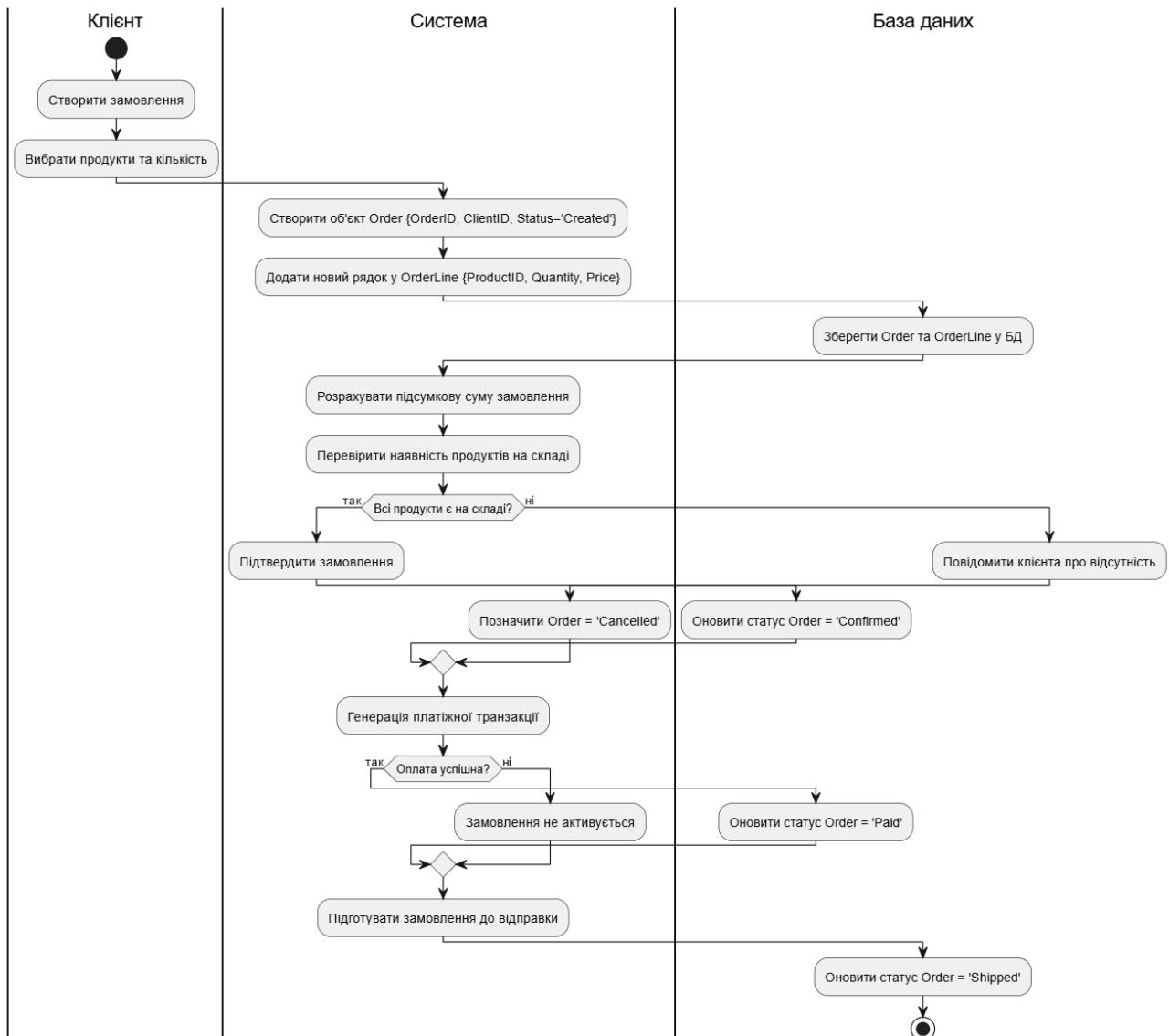


Рис. 4. Діаграма дій замовлення товарів побудована в VS Code

```
@startuml
```

```
|Клієнт|
```

```
start
```

```
:Створити замовлення;
```

```
:Вибрати продукти та кількість;
```

```
|Система|
```

```
:Створити об'єкт Order {OrderID, ClientID, Status='Created'};
```

```
:Додати новий рядок у OrderLine {ProductID, Quantity, Price};
```

|База даних|

:Зберегти Order та OrderLine у БД;

|Система|

:Розрахувати підсумкову суму замовлення;

:Перевірити наявність продуктів на складі;

if (Всі продукти є на складі?) then (так)

:Підтвердити замовлення;

|База даних|

:Оновити статус Order = 'Confirmed';

else (ні)

:Повідомити клієнта про відсутність;

|Система|

:Позначити Order = 'Cancelled';

endif

|Система|

:Генерація платіжної транзакції;

if (Оплата успішна?) then (так)

|База даних|

:Оновити статус Order = 'Paid';

else (ні)

|Система|

:Замовлення не активується;

endif

|Система|

:Підготувати замовлення до відправки;

|База даних|

:Оновити статус Order = 'Shipped';

stop

@enduml

Лістінг 1. Приклад PlantUML дефінації

Результат:

Готова діаграма активності UML, яка показує операції та функції на рівні програмного забезпечення, їх послідовність та взаємодію об'єктів, базуючись на сценаріях з готової Use Case діаграми.