

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ ШТУЧНОГО ІНТЕЛЕКТУ ТА РОБОТОТЕХНІКИ
КАФЕДРА ПРОГРАМНИХ І КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ

Методичні вказівки з дисципліни
ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ
(Теоретична частина. Частина перша)
Для студентів інституту штучного інтелекту та робототехніки

Перший (бакалаврський) рівень вищої освіти

Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма: Комп'ютерні технології автоматизації;

Одеса 2022

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ ШТУЧНОГО ІНТЕЛЕКТУ ТА РОБОТОТЕХНІКИ
КАФЕДРА ПРОГРАМНИХ І КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ**

Методичні вказівки з дисципліни
ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ
(Теоретична частина. Частина перша)
Для студентів інституту штучного інтелекту та робототехніки

Перший (бакалаврський) рівень вищої освіти

Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма: Комп'ютерні технології автоматизації;

Кількість год/кредитів ЄКТС за навчальним планом: 270/9

Схвалено на засіданні кафедри ПКІТ протокол
№7 від 26.01.2022 р.

Методичні вказівки з дисципліни «Обчислювальна техніка та комп'ютерні технології» (Теоретична частина) для студ. спец. 151 – «Автоматизація та комп'ютерно-інтегровані технології» ден. та заоч. форм навч./уклад.: В.О. Давидов - Одеса: ОП, 2022. - 31 с.

Укладачі: **В.О. Давидов**, канд. техн. наук

Зміст

1. Введення.....	5
1.1 Установка для аналізу.....	5
1.2 Інформаційні моделі	5
1.3 Моделі станів.....	5
1.4 Моделі процесів	5
1.5 Робочі продукти ООА.....	6
1.6 Короткий зміст	6
2 Концепції інформаційного моделювання	7
2.1 Об'єкти.....	7
2.2 Атрибути	8
2.3 Зв'язки.....	12
2.4 Формалізація зв'язку	14
2.5 Композиція зв'язків	14
2.6 Підтипи та супертипи	15
2.7 Робочі продукти	15
3 Життєві цикли об'єктів.....	16
3.1 Моделі поведінки в реальному світі	16
3.2 Життєві цикли та діаграма переходів у стани	17
3.3 Стану	19
3.4 Події.....	20
3.5 Дії.....	22
3.6 Переходи та таблиця переходів у стани.....	24
3.7 Таймери	26
3.8 Загальні форми життєвих циклів	27
3.9 Формування життєвих циклів	27
3.10 Життєві цикли для підтипів та супертипів	28
3.11 Коли формуються життєві цикли	29
3.12 Аналіз відмов.....	29
3.13 Робочі продукти	31

1. ВВЕДЕННЯ

Цей конспект є введенням в об'єктно-орієнтований аналіз, або ООА (Object-Oriented Analysis). Ми розглядаємо більшість матеріалу за книгою, Саллі Шлеєр та Стефана Меллор «Об'єктно-орієнтований аналіз: моделювання світу в станах».

1.1 Установка для аналізу

У формуванні типової великої системи програмного забезпечення аналітик, як правило, повинен розглядати ряд чітко визначених предметних областей або доменів. Кожен домен може розглядатися як окремий світ, населений власними концептуальними сутностями або об'єктами. Кожен домен може існувати більш менш незалежно від інших.

Деякі домени досить малі, щоб можна було аналізувати їх як єдине ціле, тоді як інші містять так багато об'єктів, що важко піддаються контролю та обробці. Після того як система розбита на домени та підсистеми, можна проводити аналіз. Незалежно одна від одної кожна підсистема (або маленький домен) аналізується у три етапи: інформаційне моделювання, моделювання станів та моделювання процесів.

1.2 Інформаційні моделі

Мета етапу інформаційного моделювання у тому, щоб ідентифікувати концептуальні сутності, чи об'єкти, які становлять підсистему для аналізу.

Зв'язки, властиві об'єктам, представляються на графічній моделі як з'єднання між об'єктами.

Закінчений опис чи визначення кожного об'єкта, атрибуту та зв'язку має бути підготовлено як документація для графічної моделі.

1.3 Моделі станів

Тепер, коли об'єкти та зв'язки ідентифіковані, ми звертаємось до дослідження їхньої поведінки у часі. В ООА кожен об'єкт і зв'язок може мати життєвий цикл - організовану схему поведінки.

Життєвий цикл формалізується в моделі станів: безлічі станів та подій. Стан є положення або ситуацію об'єкта, в яких застосовуються певні фізичні закони, правила та лінії поведінки. Подія є інцидент, який змушує об'єкт переходити з одного стану в інший.

Окремі моделі станів формуються для кожного об'єкта та зв'язку, які мають цікаву для нас динамічну поведінку. Зауважимо, що з кожним станом пов'язана певна діяльність. Ця діяльність, надалі звана дією, відбувається у той час, коли об'єкт досягає стану.

Щоб досягти узгодженого поведінки різних об'єктів, моделі станів взаємодіють між собою у вигляді подій: Для кожної підсистеми будуються окремі моделі взаємодій об'єктів.

Як тільки розроблені моделі взаємодії об'єктів для всіх підсистем у домені, для опису взаємодій подій між підсистемами може бути намальована модель взаємодії підсистем.

1.4 Моделі процесів

Все, що відбувається в системі, міститься в діях моделей станів. Тепер кожна дія визначається в термінах процесів та архівів даних об'єктів, де процес є фундаментальним модулем операції, а архів даних об'єкта відповідає даним (атрибутам) об'єкта в інформаційній моделі.

Кожна дія зображена графічно на діаграмі потоків даних дій – ДПДР. Зверніть увагу, що для кожної дії кожної моделі станів створюються окремі ДПДР.

Процеси дії можуть мати доступ як до даних об'єкта, у чию модель стану вони вкладені, і до даних інших об'єктів. Перегляд такого міжоб'єктного доступу до даних забезпечується моделлю доступу до об'єктів.

І нарешті, описи процесів розробляються для документування деталей будь-яких заплутаних процесів, що є ДПДР.

1.5 Робочі продукти ООА

Щоб описати взаємодію подій між різними підсистемами в межах домену, для кожного домену створюється модель взаємодії підсистем. Більшість робочих продуктів ООА необхідні на рівні підсистем: інформаційна модель, модель взаємодії об'єктів, модель доступу до об'єктів та допоміжні таблиці, описи та списки створюються для кожної підсистеми. Нижче підсистеми знаходяться об'єкти, які її складають: модель станів створюється для кожного об'єкта та зв'язку, які мають цікаву для нас динамічну поведінку. Дії кожної моделі станів забезпечують наступний рівень: діаграма потоків даних дій створюється кожному за стану кожної моделі станів. Нарешті опис процесу створюється для кожного складного процесу дії.

1.6 Короткий зміст

Ми починаємо з рівня підсистем у 2-му розділі, який є коротким викладом понять інформаційного моделювання. У 3-му розділі ми докладно обговорюємо, як формалізувати життєвий цикл кожного об'єкта, ідентифікованого інформаційної моделі. У 4-й розглядається предмет динамічних зв'язків. У 5-му розділі ми представляємо модель взаємодії об'єктів разом із методикою

розуміння та описи динаміки всієї підсистеми. У 6-й розглядаються діаграми потоків даних дій та модель доступу до об'єктів. У 7-му та 8-му розділах представляєш домени та підсистеми. І, нарешті, в 9-й ми розглядаємо один підхід для отримання об'єктно-орієнтованого проектування з ООА моделей.

2 КОНЦЕПЦІЇ ІНФОРМАЦІЙНОГО МОДЕЛЮВАННЯ

2.1 Об'єкти

Поняття об'єкту

Ми починаємо з визначення. В ТОВ об'єкт - це така абстракція безлічі предметів реального світу, що:

всі предмети у цьому безлічі - екземпляри - мають одні й самі характеристики;
всі екземпляри підпорядковані і узгоджуються з тим самим набором правил і ліній поведінки.

Кожен об'єкт моделі повинен бути забезпечений унікальним і значущим ім'ям, а також унікальним ключовим літералом: короткою формою імені об'єкта, яка може бути використана для кореляції інших елементів ООА моделі з об'єктом. У великій моделі для організації документації об'єкти, крім того, мають бути пронумеровані.

Об'єкт в ООА є один типовий, але невизначений екземпляр чогось у реальному світі.

В ООА немає спеціального терміна для сукупності існуючих екземплярів об'єкта.

Ідентифікація об'єктів

Об'єкти ідентифікуються шляхом розгляду концептуальних сутностей або "предметів", пов'язаних із аналізованою проблемою. Певні завдання передбачають створення об'єктів, що мають фізичний характер: у роботі аеропорту Ви, ймовірно, знаходитимете такі об'єкти, як Літак, Злітно-Посадкова Смуга та Вишка Управління. Інші завдання породжують велику кількість абстрактних об'єктів: система зв'язку може мати об'єкти типу Вихідне Повідомлення Фрейму Даних або Вхідне Повідомлення Підтвердження.

Незважаючи на те, що об'єкт може бути отриманий фактично з будь-чого, більшість об'єктів, з якими ми стикаємося, відносяться до наступних категорій:

- реальні об'єкти;
- ролі;
- інциденти;
- взаємодії;
- специфікації.

Реальні об'єкти - абстракції фактичного існування деяких предметів у фізичному світі:

Ролі - абстракції мети чи призначення людини, частини обладнання чи організації:

Інцидент - абстракція чогось того, що сталося або трапилося:

Нещасний випадок (у страхуванні).

Землетрус.

Вибори.

Постачання (компанією з перевезення).

Взаємодія - об'єкти, одержувані із відносин між іншими об'єктами:

З'єднання: стикування двох труб.

Контракт: угоду між двома сторонами.

Перехрестя: місце, де перетинаються дві чи більше вулиці.

Об'єкти-специфікації використовуються для представлення правил, стандартів або критеріїв якості (на відміну від реального об'єкта чи ролі, які відповідають цим стандартам).

Опис об'єктів

Опис має забезпечуватись для кожного об'єкта. Опис - це коротке інформативне твердження, яке дозволяє нам встановити, є реальний предмет екземпляром об'єкта чи ні. Опис об'єкта має бути заснований на абстракції, що точно пояснює подібність предметів реального світу.

Опис Об'єкта (Знімаючий Показ приладу): Знімаючи показання приладу - це штатний службовець-електрик, який може прочитати значення, що відображаються на електричних приладах, і зараз виконує це завдання. Хтось, який навчався на курсах зі зняття показань приладу, вважається за здатне прочитати його значення.

При підготовці описів об'єкта ми часто маємо звертатися як до реальних сутностей, так і до формальних об'єктів у моделі. Щоб допомогти читачеві відрізнити одне від одного, ми використовуємо прописну початкову літеру у всіх словах, що становлять ім'я об'єкта, та символи нижнього регістру при зверненні до екземплярів реального світу.

2.2 Атрибути

Поняття атрибуту

Предмети в реальному світі мають такі характеристики, як висота, температура, реєстраційний номер або положення. Кожна окрема характеристика, яка є загальною для всіх можливих екземплярів об'єкта, абстрагується як окремий атрибут.

Визначення. Атрибут - це абстракція однієї характеристики, яку мають усі абстраговані як об'єкт сутності. Кожен атрибут забезпечується ім'ям, унікальним у межах об'єкта.

Щоб звернутися до атрибута, пишуть <ім'я об'єкта>.<ім'я атрибута>, наприклад:

Літак. Розмах Крила.
Кішка. Вага.
Клапан.Статус Відкр/Закр.

Домени та значення атрибутів

Для будь-якого певного екземпляра атрибут може набувати значення. Припустимо, що ми маємо об'єкт Кішка з атрибутами Ім'я, Підлога, Вага, Колір, Темперамент. Тоді

Кішка Саллі:

Ім'я: Грусем.
Підлога: М.
Вага: 15.75 фунтів.
Колір: бура, у смужку.
Темперамент: неймовірно лінива.

Кішка Стіва:

Ім'я: Пакмен.
Підлога: Ж.
Вага: 9.25 фунтів.
Колір: черепаховий.
Темперамент: розпусна.

Діапазон допустимих значень, які атрибут може набувати, називається доменом. Домен повинен визначатися для кожного атрибута, як це виконується та пояснюється у наступному розділі.

Ідентифікатори

Ідентифікатор - це безліч одного або більше атрибутів, значення яких однозначно визначають кожен екземпляр об'єкта.

При використанні прикладу кішки атрибут Ім'я - задовільний ідентифікатор для об'єкта Кішка доти, доки ми дотримуємося стратегії надання кожній кішці власного унікального імені. Однак атрибут Вага не може бути ідентифікатором, тому що можливо, що є дві кішки з однаковою вагою.

Кожен об'єкт повинен мати ідентифікатор. Об'єкт може мати кілька ідентифікаторів, кожен з яких складається з одного або кількох атрибутів. Наприклад, об'єкт Аеропорт може мати атрибути:

Код Аеропорту.
Широта.
Долгота.
Місто.
Кількість Пасажирських Пропускних Пунктів.

Атрибут Код Аеропорту - ідентифікатор об'єкта Аеропорт, а комбінація Широти та Долготи - інший ідентифікатор Аеропорту.

Якщо об'єкт має кілька ідентифікаторів, один із них вибирається як привілейований.

Подання

Об'єкт може бути представлений разом зі своїми атрибутами кількома різними способами. Графічно об'єкт зображується у вигляді рамки, що містить ім'я, номер та ключовий літерал об'єкта та імена атрибутів. Атрибути, які складають привілейований ідентифікатор об'єкта, відзначаються зірочкою з лівого боку імені атрибута. Всі інші атрибути виділяються точкою або іншим зручним символом. В еквівалентному текстовому поданні привілейований ідентифікатор підкреслюється:

Таблична інтерпретація

Об'єкт інформаційної моделі може інтерпретуватися як таблиця. У цій інтерпретації кожен екземпляр об'єкта є рядком у таблиці. Рядок заповнюється значеннями атрибутів, що відповідають кожному екземпляру.

Ми вважаємо табличну інтерпретацію корисною для побудови прикладів та уявлень об'єкта для нетехнічної аудиторії. Зауважте, що таблиця не розглядається як завдання структури даних, яке використовується в реалізації: обережний проектувальник розпізнаватиме, які численні еквівалентні структури даних можуть бути відповідними варіантами різних ситуацій.

Кошка				
Имя	Пол	Вес	Цвет	Темперамент
Грусем	М	15,75	бурий в полосуку	невероятно-ленивый
Пакмен	Ж	9,25	черепаховый	распутный
Леппи	Ж	8,00	черепаховый	замкнутый
Мунчи	М	9,75	бело-оранжевый	нежный
Майнди	Ж	7,00	черепаховый	как у старой девы
Смадж	Ж	8,75	кремовый	пугливый
Диплер	Ж	8,25	кремовый	подвижный
Фанбелт	М	10,50	сиамский	сдеранный
Декард	М	10,25	чёрный	ленивый

Типи атрибутів

Атрибути можуть класифікуватися за належністю, одного з трьох різних типів::

- описові;
- вказівки;

- допоміжні.

Описові атрибути представляють факти, що властиво кожному екземпляру об'єкта:

Рахунок.Сальдо.
Джерело Електропостачання. Полярність.
Кішка.Вага.

Якщо значення описового атрибута змінюється, це говорить про те, що певний аспект екземпляра змінився, але сам екземпляр залишився колишнім. Наприклад, якщо Пакмен додала у вазі один фунт, то факт про цю кішку змінився, але кішка залишилася тією самою.

Атрибути, що вказують, використовуються для надання імені або позначення примірників. Імена зазвичай дещо довільні:

Рахунок.Номер.
Вантаж.Номер Накладний.
Рейс. Номер.

Атрибути, що вказують, часто (але не завжди) використовуються як ідентифікатор або як частина ідентифікатора.

Якщо значення вказівного атрибута змінюється, це говорить тільки про те, що нове ім'я дається тому ж самому екземпляру. Зміна імені Пакмен на Попкорн аж ніяк не змінює кішку.

Допоміжні атрибути використовуються для зв'язку екземпляра одного об'єкта з екземпляром іншого:

Ім'я Господаря позначає людину, якій належить ця кішка.

Рахунок.ID Клієнта вказує, який клієнт є власником цього рахунку.

Магніт.Джерело Електропостачання вказує, яке джерело електропостачання використовується для подачі напруги на цей магніт.

Якщо значення допоміжного атрибута змінюється, це свідчить, що тепер інші екземпляри пов'язані між собою. Якщо змінювати значення Магніт. Джерело електропостачання з PS10 на PS12, ми вказуємо, що тепер подача напруги на магніт здійснюється джерелом електропостачання PS12.

Опис атрибутів та доменів

Кожен атрибут вимагає опису, з якого стає зрозумілим, яку характеристику містить атрибут. Також має бути описаний домен кожного атрибуту.

Ці описи складаються залежно від типу атрибута, з яким маємо справу, трохи інакше.

Описові атрибути. Опис атрибута має встановлювати реальну характеристику, що абстрагується як атрибут. Воно може пояснювати, як визначається характеристика і чому вона є доречною для визначення об'єкта.

Опис домену можна давати:

- перерахування всіх можливих значень, які атрибут може приймати;
- посиланням на документ, що наводить можливі значення;
- формулюванням правила, що визначає, які значення допустимі;
- приведенням діапазону можливих значень.

Вказівні атрибути. Опис атрибута для вказівного атрибута встановлює форму вказівки (якщо форма доречна), організацію, яка призначає вказівку (якщо доречно) та ступінь, у якій ім'я може використовуватись як частина ідентифікатора. Якщо опис атрибута дає достатньо інформації про допустимі значення для вказівного атрибута, домен визначається таким чином:

Домен: Див. опис атрибута

або

Домен: Див. вище (коли передуює опис атрибута).

Крім того, домен вказівного атрибуту може визначатись перерахуванням або посиланням.

Допоміжні атрибути. Опис атрибута для допоміжного атрибута має вказувати реальне відношення, яке зберігається атрибутом. Наприклад, атрибут Магніт. Джерело електропостачання може бути описано як назва джерела електропостачання, що використовується в даний час для подачі напруги на цей магніт.

Значення, яке може приймати допоміжний атрибут, завжди буде таким же, як і значення атрибута, що діє як ідентифікатор для співвідносного об'єкта. Отже, опис домену завжди дається фразою: "Такий як <ім'я ідентифікуючого атрибута співвідносного об'єкта>", наприклад:

Атрибут: Магніт.Джерело Електропостачання.

Опис: . . .

Домен: Такий, як Джерело Електропостачання.

ID Джерела Електропостачання.

Правила атрибутів

Інформаційна модель полягає в реляційній моделі даних: поданні даних як відносин між ними. Реляційна модель визначається у конкретних твердженнях щодо форм, які ми використовуємо в інформаційній моделі, та значенні, яке ми маємо на увазі, коли призначаємо атрибут для об'єкта.

Перше правило. Один екземпляр об'єкта має одне-єдине значення для кожного атрибута в будь-який час.

У табличній інтерпретації об'єкта це правило вимагає, щоб існував один і лише один елемент даних у кожному перетині стовпця з рядком. Це унеможливорює складання таблиці зі структурою "повторюваної групи", як і таблицю з незаповненими осередками .

Друге правило. Атрибут не повинен містити жодної внутрішньої структури.

Третє правило. Коли об'єкт має складовий ідентифікатор, тобто. що складається з двох або більше атрибутів, кожен атрибут, який не є частиною ідентифікатора, представляє характеристику всього об'єкта, а не його частини, а тим більше не характеристику чогось іншого.

Четверте правило. Кожен атрибут, який не є частиною ідентифікатора, представляє характеристику примірника, зазначеного ідентифікатором, а не характеристику іншого атрибута - неідентифікатора.

Відповідно до четвертого правила, якщо Ви визначаєте об'єкт

Порція (ID Порції, ID Рецепту, Кількість Галлонів, Час Приготування),

Атрибут Порція. Час приготування повинен представляти фактичний час приготування порції, а не час приготування, що визначається рецептом.

2.3 Зв'язки

Поняття зв'язку

Відносини існують між різними видами предметів у світі.

Зв'язок - це абстракція набору відносин, що систематично виникають між різними видами предметів у реальному світі.

Реальні предмети, що беруть участь у відношенні, мають бути абстраговані як об'єкти.

Кожна зв'язок у моделі задається парою імен, які описують зв'язок із перспективи кожного об'єкта, що бере участь.

Власник Собаки володіє Собакою.
Собака належить Власнику Собаки.

Зауважте, що це одиночний зв'язок. Кожному зв'язку надається унікальний ідентифікатор виду R1, K2 і т.п.

Подання

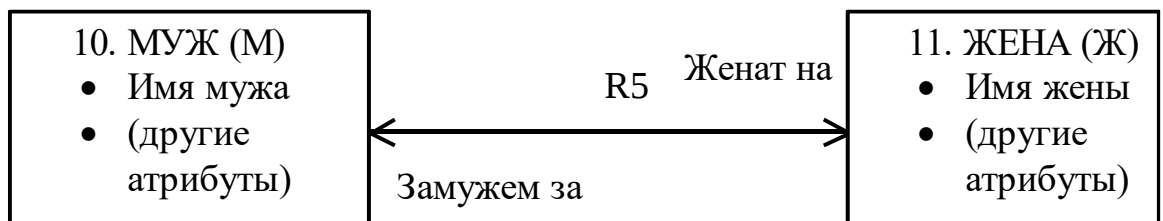
Зв'язок представляється графічно лінією між співвідносними об'єктами (рис.2.3.1). Лінія позначається ідентифікатором зв'язку. Назви зв'язку поміщаються кожному кінці лінії зв'язку, як показано малюнку.



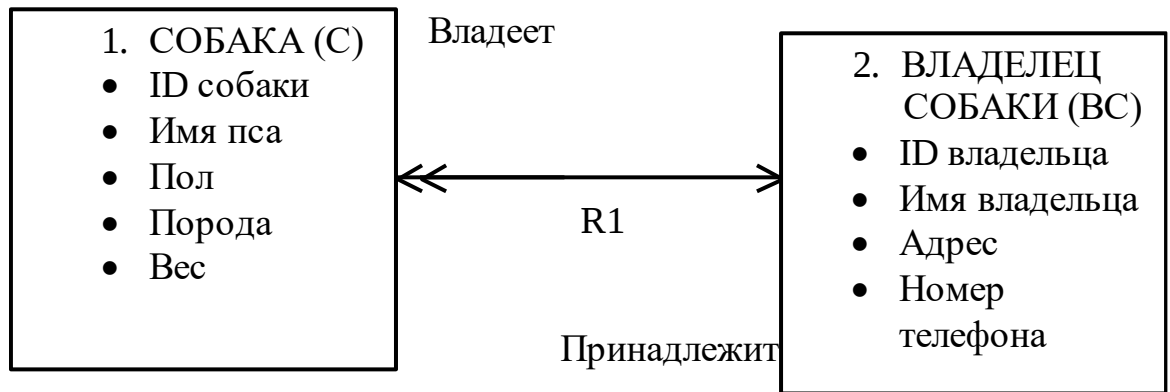
Безумовні зв'язки

Існує три фундаментальні види зв'язку: один-до-одного, один-до-багатьом і багато-до-багатьом. Ці фундаментальні види називаються безумовними формами, бо участі у зв'язку потрібен кожен екземпляр обох об'єктів.

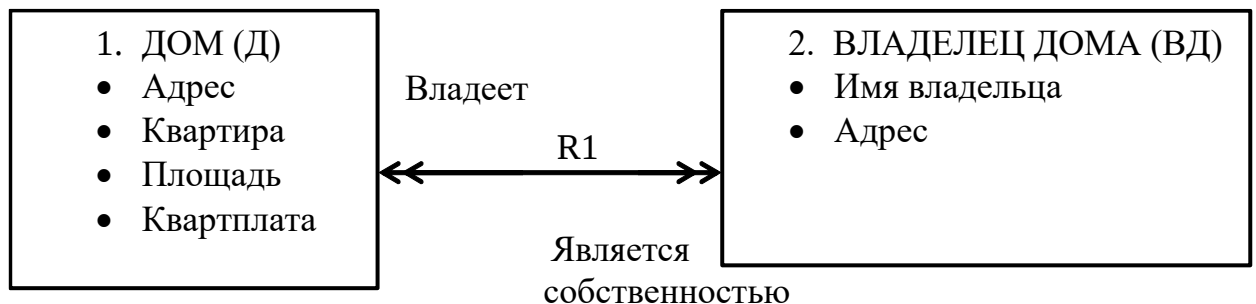
Зв'язок один до одного існує, коли один екземпляр одного об'єкта пов'язаний з єдиним екземпляром іншого.



Зв'язок одним-багатьом існує, коли один екземпляр деякого об'єкта пов'язаний з одним або більше екземпляром іншого і кожен екземпляр другого об'єкта пов'язаний тільки з одним екземпляром першого.



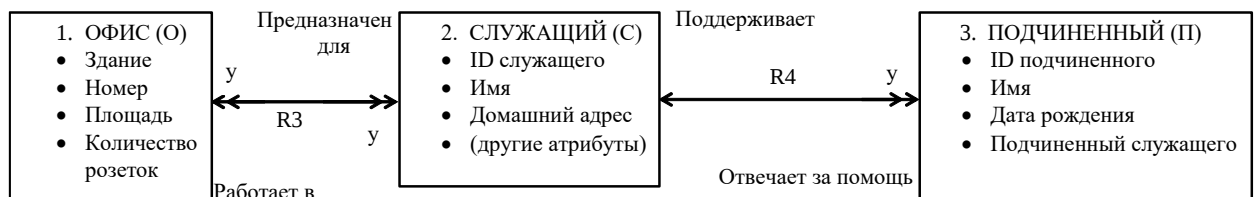
Зв'язок багато хто існує, коли один екземпляр деякого об'єкта пов'язаний з одним або великою кількістю екземплярів іншого, і кожен екземпляр другого об'єкта пов'язаний з одним або більше екземпляром першого.



Терміни один-до-одного, один-до-багатьом і багато-багатьом є формулюваннями множинності зв'язку. Зауважимо, що множинність позначається графічно: одна стрілка на кінці означає один екземпляр, а подвійна стрілка – один або більше екземпляр.

Умовні форми

У безумовних зв'язках, обговорених вище, участі у зв'язку був потрібен кожен екземпляр об'єктів. В умовному зв'язку можуть існувати екземпляри об'єктів, які не беруть участі у зв'язку. Ця ситуація зображується приміщенням літери "У" (від слова "умовна") поблизу фрази зв'язку, яка іноді є істинною. Зв'язок R4 на рис. умовна лише з одного боку, оскільки можуть бути службовці без підлеглих, але може бути підлеглим без співвідносного із нею службовця.



Якщо зв'язок умовний з обох сторін - мається на увазі, що можуть існувати екземпляри обох об'єктів, які не беруть участь у зв'язку, - він називається біумовним. І тут маркер " У " вміщено обох фразах зв'язку. Зв'язок R3 на рис. біумовна, тому що деякі службовці не працюють в жодному з офісів і деякі офіси не призначені для жодного з службовців.

Опис зв'язку

Усі зв'язки вимагають опису. Опис має забезпечувати:

- ідентифікатор зв'язку;
- формулювання імен зв'язку з погляду кожного об'єкта, що бере участь;
- вид зв'язку (її множинність та умовність);
- формулювання підстави абстракції;

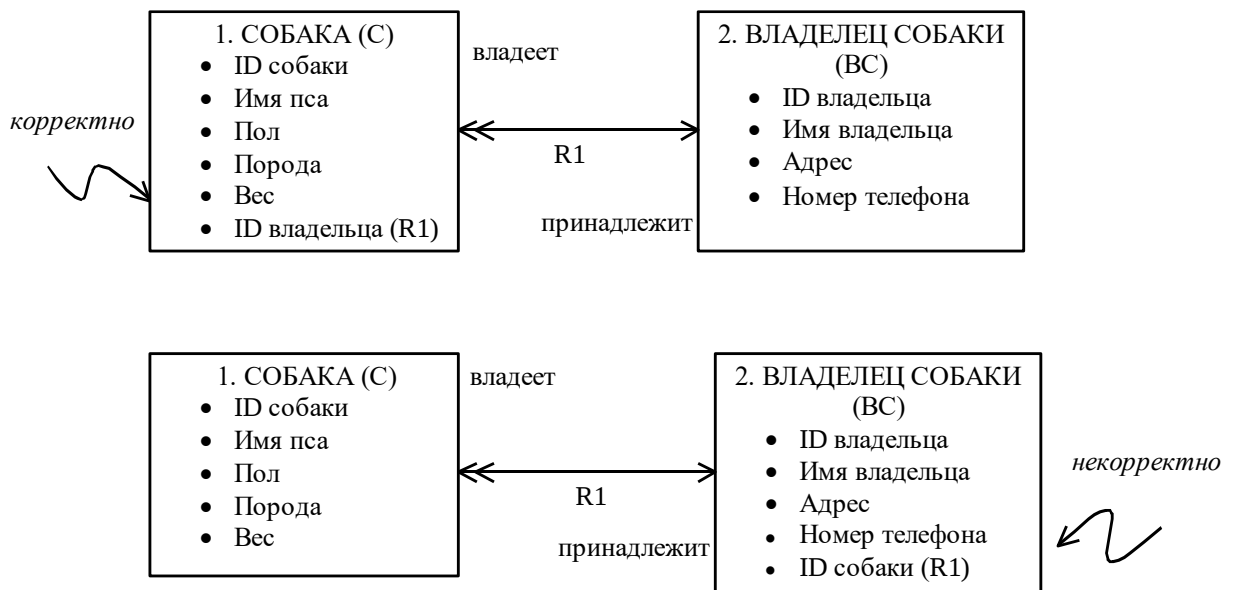
- формулювання того, як зв'язок був формалізований (тема, яка обговорюється в наступному розділі).

2.4 Формалізація зв'язку

Мета зв'язку полягає в тому, щоб дозволити нам встановити зв'язок екземпляра одного об'єкта з екземпляром іншого. Це виконується розміщенням допоміжних атрибутів у відповідних об'єктах моделі. Коли це виконано, говорять, що зв'язок формалізована в даних.

Для формалізації зв'язку один до одного допоміжні атрибути можуть бути додані до будь-якого об'єкта (але не до обох).

Для формалізації зв'язку допоміжні атрибути повинні бути додані до об'єкта на стороні "багато", оскільки розміщення такого допоміжного атрибута на стороні "один" порушуватиме третє правило атрибутів.



Для формалізації зв'язку багато хто створює асоціативний об'єкт - окремий об'єкт, який містить посилання на ідентифікатори кожного з екземплярів, що беруть участь. Тоді асоціативний об'єкт обробляється як об'єкт із власними правами, ім'ям, номером, ключовим літералом, описом. Подібно до іншого, асоціативний об'єкт може мати додаткові атрибути і брати участь у зв'язках з іншими об'єктами.

Асоціативний об'єкт може використовуватися для формалізації не тільки зв'язку багатьом, але й будь-якого іншого зв'язку. Як ми побачимо в 4-му розділі, зв'язок з динамічною поведінкою має бути формалізована за допомогою асоціативного об'єкта.

2.5 Композиція зв'язків

Деякі зв'язки утворюються як неминучий наслідок існування інших зв'язків. Це буде показано з прикладу.

Розглянемо випадок Університету А. У цьому закладі кожному студенту необхідно визначити одну основну тему досліджень, яку представляє деякий факультет, який спеціалізується на цій темі. Кожен факультет має у своєму штаті професорів.

В Університеті А студент обирає професора, який під час навчання буде його керівником. Професор може бути з будь-якого факультету, тому студент, який спеціалізується на математиці, може мати керівником професора фізики.

Тепер розглянемо Університет В. Правила тут такі ж, як у випадку з Університетом А, за винятком того, що студент повинен обирати професора на факультеті спеціалізації. Інформаційна модель в Університеті В дещо відрізняється, оскільки зв'язок R3 – логічне наслідок зв'язків R1 та

R2: якщо студентом керує професор, студент повинен обов'язково спеціалізуватися у навчанні на факультеті професора.

Кажуть, що такий зв'язок утворений композицією (як композиція функцій у математиці). Зв'язок, утворений композицією, не може бути формалізований у допоміжних атрибутах, тому що відношення між екземплярами (у цьому випадку студенти та факультети) вже задано зв'язками між студентами та професорами і далі між професорами та їхніми факультетами.

Зв'язок, утворений композицією, позначається на моделі як $R3 = R1 + R2$, що означає, що R3 є логічним наслідком R1 і R2.

2.6 Підтипи та супертипи

У багатьох завданнях Ви знаходите чіткі спеціалізовані об'єкти, які мають певні загальні атрибути. У цьому випадку ми можемо абстрагувати більш загальний об'єкт для представлення характеристик, які спільно використовуються спеціалізованими об'єктами.

Атрибути, спільні для всіх об'єктів підтипів, знаходяться в об'єкті супертипу. Об'єкти підтипу будуть також мати додаткові атрибути, щоб забезпечувати більш спеціалізовані абстракції, що надаються кожним підтипом.

У конструкції підтип-супертип один реальний екземпляр є комбінацією екземпляра супертипу та екземпляра лише одного підтипу. На відміну від деяких об'єктно-орієнтованих мов програмування, ТОВ не дозволяє створення екземпляра супертипу без створення екземпляра одного підтипу та навпаки.

2.7 Робочі продукти

Для інформаційної моделі підсистеми розробляються три робочі продукти.

Діаграма інформаційної структури. Діаграма інформаційної структури – це графічне уявлення інформаційної моделі. На цей малюнок іноді посилаються як на діаграму "сутність-зв'язок" чи просто як на інформаційну модель.

Опис об'єктів та атрибутів. Опис об'єктів та атрибутів документує списки всіх об'єктів та атрибутів у моделі та забезпечує організовану сукупність описів об'єктів, атрибутів та доменів.

Опис зв'язків. Опис зв'язків документує переліки кожного зв'язку моделі разом із описом зв'язку.

3 ЖИТТЄВІ ЦИКЛИ ОБ'ЄКТІВ

3.1 Моделі поведінки в реальному світі

Коли ми звернемося до розгляду динамічної поведінки предметів у світі, то зауважимо, що у світі зазвичай мають термін життя. Часто щось створюється чи з'являється, еволюціонує через певні стадії свого існування і потім вмирає чи зникає. Класичний приклад - електрична лампочка: після виготовлення вона встановлюється в патроні, де потім то вмикається, то вимикається. Іноді вона може бути переміщена з одного освітлювального патрона до іншого. Зрештою, вона перегорає і її викидають. Всі електричні лампочки протягом терміну свого існування йдуть однією і тією ж простою, але цікавою для нас **моделі динамічної поведінки**

- Багато предметів протягом термінів життя проходять через різні стадії. Літак, наприклад, може бути запаркований на стоянці, пересуватися до злітно-посадкової смуги, летіти тощо.



Мал. Модель поведінки літака.

- Порядок еволюціонування предмета через стадії саме формує характерну рису поведінки відмітного ознаки цього предмет;
- Реальний предмет знаходиться в одній-єдиній стадії моделі поведінки в будь-який час. Літак не може одночасно бути запаркованим на стоянці та залітати.
- Предмети еволюціонують відбійної стадії до іншої стрибкоподібно, частково залежить від вибору визначення чи сприйняття стадій. Ми визначаємо, що політ буде стадією поведінки літака, коли крила відірвуться або землі.
- У схемі поведінки дозволені в повному обсязі еволюції між стадіями. Деякі еволюції забороняються законами фізики, деякі статутами тощо. Закони фізики не допускають еволюціонування літака від польоту безпосередньо до паркування на стоянці.
- У реальному світі існують інциденти, які змушують предмети еволюціонувати (або вказують, що вони еволюціонували) між стадіями. Запущений двигун літака змушує

літак еволюціонувати від запаркованого на стоянці до переміщеного на злітно-посадкову смугу.

3.2 Життєві цикли та діаграма переходів у стани

Повернемося до опису об'єкта.

Об'єкт - це абстракція безлічі предметів реального світу, така що:

- всі предмети в даному множині - екземпляри - мають одні й самі характеристики;
- всі екземпляри підпорядковані і узгоджуються з одними і тими самими правилами.

Оскільки всі екземпляри об'єкта повинні слідувати тим самим правилам поведінки, то, абстрагуючи групу подібних предметів для отримання об'єкта, ми абстрагуємо їх загальну модель поведінки в життєвий цикл типового об'єкта. Життєвий цикл забезпечує формалізоване опис моделі поведінки, що спільно використовується всіма екземплярами.

Життєві цикли на моделі станів

Життєвий цикл виявляється як модель станів. В аналізі головним чином використовуються два види моделей станів, в ТОВ ми використовуємо форму Мура. **Модель станів Мура:**

1. з безлічі станів. Кожен стан представляє стадію у життєвому циклі типового екземпляра об'єкта;
2. з багатьох подій. Кожна подія є інцидентом або вказівкою на те, що відбувається еволюціонування;
3. з правил переходу (або переходів; для стислості). Правило переходу визначає, який новий стан досягається, коли з екземпляром у цьому стані відбувається певна подія;
4. з дій. Дія - діяльність чи операція, які мають бути виконані, коли екземпляр досягає стану. Одна дія пов'язується з кожним станом.

Модель станів може бути представлена в діаграмній формі, відомої як діаграма переходів у стани, що ілюструється наступним прикладом.

приклад. Припустимо, ми плануємо створити компактну, недорогу мікрохвильову піч - однохвилинну мікрохвильову печі. Концепція виробу включає такі елементи:

1. Є єдина кнопка управління доступна для користувача печі. Якщо дверцята печі закриті і Ви натиснете кнопку, піч готуватиме їжу (тобто силовий елемент перебуватиме під напругою) протягом 1 хв.
2. Якщо Ви натискаєте кнопку під час роботи печі, Ви отримуєте додаткову хвилину. Наприклад, якщо у Вас залишилося ще 40 с для приготування їжі і натискаєте кнопку двічі, час приготування збільшується до 2 хв 40 с.
3. Натискання кнопки при відкритих дверях не має жодного ефекту.
4. Усередині печі є електрична лампочка. Під час роботи печі вона повинна бути включена, щоб Ви могли подивитися через скло дверцята печі та побачити, наприклад, що страва згоріла. Щоразу, коли дверцята відчинені, електрична лампочка повинна бути увімкнена, щоб Ви могли бачити їжу або вимити піч.
5. Ви можете зупиняти процес приготування їжі відчиненням дверцят.
6. Якщо Ви закриваєте її, електрична лампочка гасне. Це нормальна конфігурація, якщо їжу помістили до пічки, але ще не натиснули на контрольну кнопку.
7. Якщо час закінчився (їжа готується протягом заздалегідь встановленого часу), то вимикаються як силовий елемент, так і електрична лампочка. Тоді видається звуковий сигнал, який повідомляє, що їжа готова.

У цьому є кілька істотних інцидентів, які впливають працювати печі:

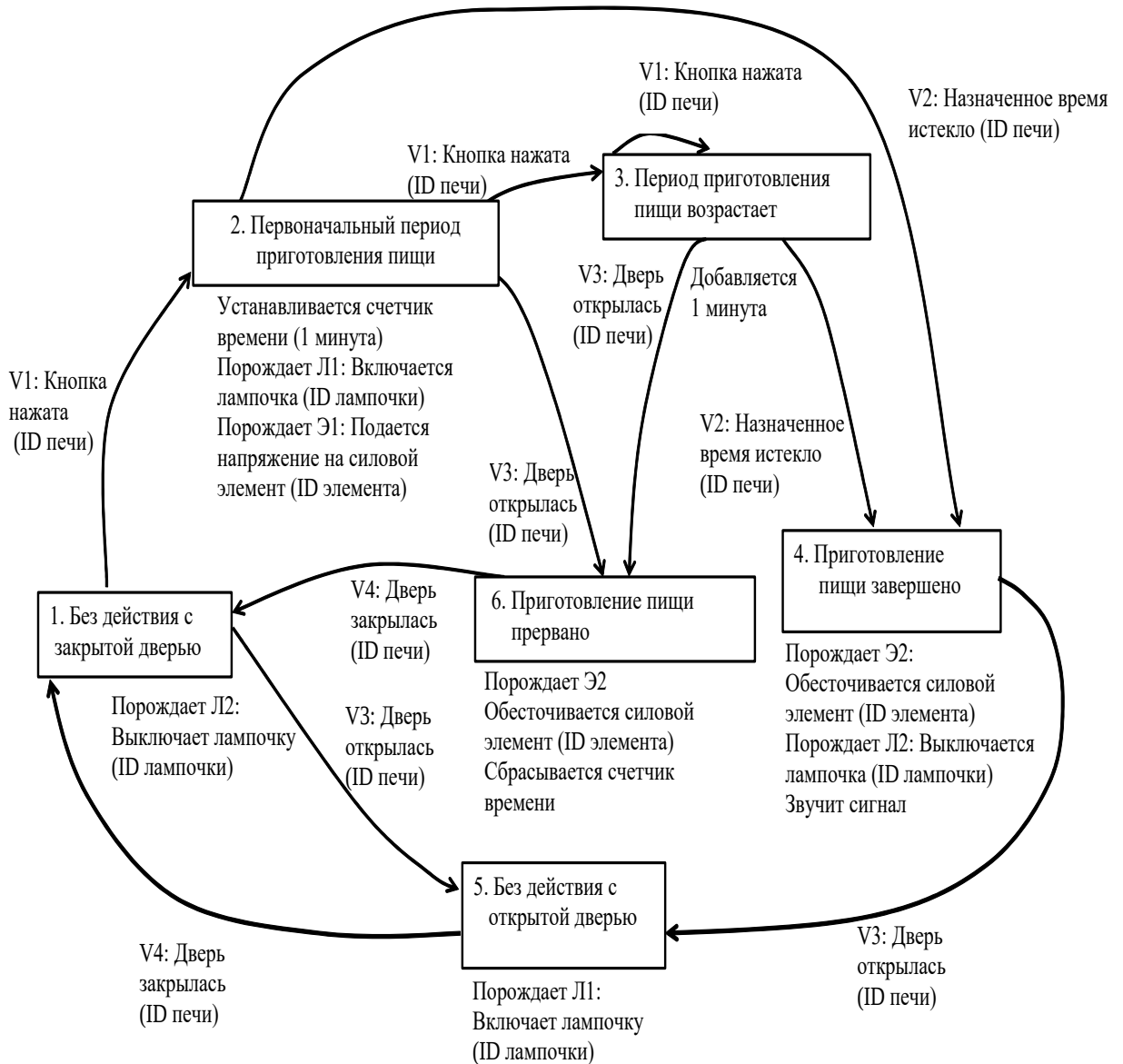
Відкриття дверей.

Закривання дверей.
Натискання кнопки управління.

Завершення інтервалу для приготування їжі.

Ці інциденти абстрагуються як події.

На рис. показана діаграма переходів у стани (ДПС) для однохвилинної мікрохвильової печі. Стан представляється рамками, кожна з яких позначається відповідним ім'ям стану. Переходи зображуються дугами, що з'єднують два стани. Кожен перехід позначений подією, що його викликає. Дія, пов'язана зі станом, описується під рамкою стану.



Мал. Модель станів для однохвилинної мікрохвильової печі.

Координація життєвих циклів

Життєві цикли різних об'єктів часто скоординовані одне з одним. Наприклад, у разі мікрохвильової печі силовий елемент та електрична лампочка мають життєві цикли, які нерозривно пов'язані зі станом самої печі. Модель станів печі, що породжує подію Подається напруга та подія Знеструмується, які змушують силовий елемент переміщатися через свою модель станів. Подібним чином піч породжує події, які наказують електричній лампочці вмикатися та вимикатися.

Моделі стану проти кінцевих автоматів

Хоча всі екземпляри об'єкта спільно використовують ту саму модель станів, іноді необхідно розрізнити модель станів, що виконується одним екземпляром (скажімо, силовим елементом 92) та іншим (силовим елементом 256). Ми використовуємо термін, кінцевий автомат, щоб звертатися до виконання моделі станів приватним екземпляром. Ви можете розглядати кінцевий автомат як деякий зразок моделі станів, що виконується одним екземпляром .

3.3 Стану

Стан - це положення об'єкта, в якому застосовується певний набір правил, ліній поведінки, розпоряджень та фізичних законів. Стани показуються рамками на ДПС.

Номери та назви станів

Кожному стану дається ім'я, що є унікальним у межах цієї моделі стану. Рамки стану на ДПС позначені іменами станів. Кожному стану надається номер, теж унікальний у межах моделі станів. Насамперед номери станів використовуються в табличному поданні Доділи станів (описується в розділі 3.6) і не обов'язково наказують порядок, в якому екземпляр перебуватиме в станах. Зауважте, що номери станів наводяться на ДПС.

Стан створення

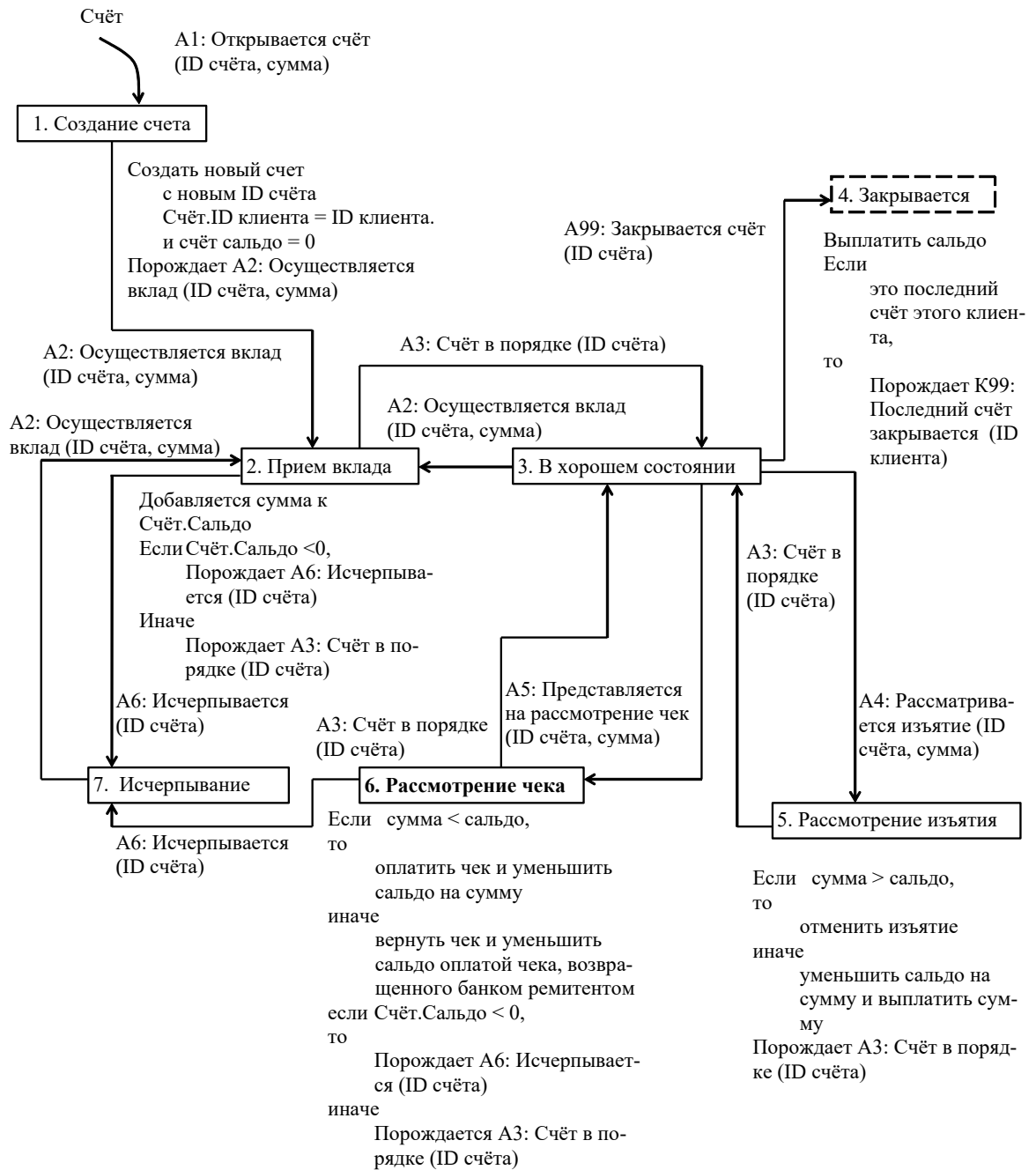
Деякі моделі станів мають один або більше стан, де екземпляр з'являється вперше. Такі стани відомі під назвою стану створення. Стан створення показано на ДПС як рамки, намальованої звичайним способом. Однак перехід у стан створення позначається як перехід не зі стану.

Заключний стан

У деяких моделях станів один (або, можливо, кілька) стан є кінцем життєвого циклу екземпляра. Такі стани відомі як останні стани. Заключний стан може бути однією з двох ситуацій.

Примірник стає нерухомим. Примірник продовжує існувати, але немає наступного цікавого динамічного поведінки. Немає спеціального позначення, яке використовується на ДПС для заклучного стану, в якому екземпляр стає нерухомим.

Примірник припиняє існування. Як тільки екземпляр досягає цього стану, він відразу зникає і, отже, те саме відбувається з його життєвим циклом. Такий стан позначається на ДПС рамкою, намальованою штриховою лінією, як показано на рис.



Мал. Модель станів для об'єкта Рахунок

поточний стан

У будь-який конкретний момент різні екземпляри об'єкта можуть бути у різних станах. Стан, у якому знаходиться екземпляр, відомий як поточний стан. Поточний стан є атрибутом, який зазвичай називається поточним станом чи статусом. Це описовий атрибут, його домен визначається перерахуванням станів об'єктної моделі станів.

3.4 Події

Подія- це абстракція інциденту або сигналу у реальному світі, який повідомляє нам про переміщення чогось у новий стан. В абстрагуванні події визначаються чотири аспекти:

- значення;

- призначення;
- мітка;
- дані події.

Значення

Значення події полягає у короткій фразі, яка повідомляє, що відбувається у реальному світі. У випадку з мікрохвильовою піч ми маємо такі значення для подій: кнопка натиснута, приготування їжі завершено, двері відчинені та двері зачинені.

Призначення

Призначення події- Модель станів {не автомат), яка приймає подію. Може бути один-єдиний такий приймач для цієї події.

Мітка

Унікальна позначка повинна забезпечуватися для кожної події. Мітки потрібні для того, щоб відрізнити події одна від одної, особливо якщо Ви маєте справу з окремими подіями подібного значення.

Форма мітки події довільна. Щоб робити короткі мітки події, ми зазвичай використовуємо комбінацію буква-номер, наприклад, для мікрохвильової печі:

- V1: Кнопка натиснута.
- V2: Призначений час минув.
- V3: Двері відчинилися.
- V4: Двері зачинилися.

Більшість аналітиків вважають корисним використовувати певний вид угоди для позначення міток події. Ось дві цілком прийнятні для цього схеми.

Позначення, що ґрунтується на призначенні. Визначте позначки події так, щоб усі події, які приймаються об'єктом, починалися з ключового літералу.

Позначення "внутрішній-зовнішній". Використовуйте особливий символ (зазвичай E - від external, тобто зовнішній) для позначення події, що відбувається між двома об'єктами. Для подій, які породжуються об'єктом і які до нього спрямовані, використовуйте ключовий літерал цього об'єкта.

Дані події

Подія слід розглядати як сигнал управління, який може (і зазвичай це робить) переносити дані. Ці дані наводяться для дії при досягненні об'єктом певного стану. Два різні типи даних - ідентифікуючі та додаткові - можуть переноситися подією.

Ідентифікуючі дані. Ідентифікуючі дані - це безліч одного або більше атрибутів, що включають привелірований ідентифікатор об'єкта-призначення. Ідентифікують дані визначають екземпляр і, отже, кінцевий автомат, який збирається приймати подію.

Додаткові дані. Додаткові дані складаються з одного або більше атрибутів, які переносяться подією, крім ідентифікуючих даних. Додаткові дані можуть наводити атрибути як об'єкта-призначення, а й іншого об'єкта.

Дані подій та стану

Існує ряд правил, що пов'язують дані події зі станами, які можуть бути досягнуті як результат одержання події.

Правило тих самих даних. Усі події, які викликають перехід у певний стан, повинні нести одні й ті самі події.

Це правило гарантує, що дія отримуватиме ті самі дані події незалежно від того, яка подія викликає перехід.

Правило стану нестворення. Якщо подія може викликати перехід у стан, який не є станом створення, то ідентифікатор екземпляра, до якого подія застосовується, має переноситись як дані.

Це правило гарантує, що дія зможе визначити, який з існуючих примірників здійснив перехід.

Правило стану створення. Подія, що викликає перехід у стан створення, не несе ідентифікатора, якщо модель стану створює ідентифікатор у дії стану створення.

Абстрагування подій з інцидентів

Один інцидент у реальному світі може бути абстрагований у більш ніж одну подію. Наприклад, припустимо, що Ви маєте залізницю, в якій залізнична мережа поділена на певні сегменти дороги, і що місцезнаходження поїзда визначається як сегмент дороги, який займає головний вагон. Тоді реальний інцидент переміщення поїзда в інший сегмент міг би породити дві прийнятні події: для поїзда - інформує, що нового сегмента було досягнуто, - і для сегмента дороги, який щойно став зайнятим.

Специфікація події проти екземплярів події

Коли подія (VI: наприклад, кнопка натиснута) з'являється на ДПС, вона є поняття типового, але невизначеного інциденту - в цьому випадку натиснута будь-яка кнопка будь-якої мікрохвильової печі. Це поняття має відрізнятися від приватної події VI: кнопка печі 5 натиснута в 15 год, другий раз - в 15 год 10 хв, і кнопка печі 25 натиснута в 21 год. Формально обидва поняття розглядаються як подія, їхнє значення зазвичай ясно з контексту. Однак, коли необхідно їх розрізнити за допомогою термінології, ми використовуємо поняття специфікація події по відношенню до типової невизначеної події (натиснута будь-яка кнопка будь-якої печі) і екземпляр події по відношенню до певної події (натиснута ця сама кнопка печі 5 в 15 год 10 хв).

3.5 Дії

Дії та екземпляри

Дія - це діяльність або операція, яка має бути виконана екземпляром, коли він досягає стану. Одна дія пов'язується з кожним станом.

Оскільки всі екземпляри об'єкта спільно використовують ту саму модель станів, дії повинні бути визначені таким чином, щоб вони могли бути виконані будь-яким екземпляром.

Коли дія починається, це забезпечується даними події, що передаються подією, що спричинила перехід. Дані події містять достатньо інформації, щоб ідентифікувати існуючий екземпляр, від імені якого виконується дія. У разі стану нестворення дані події включають ідентифікатор примірника. У разі стану створення дія буде створювати новий екземпляр, що базується на інформації, переданій даними події. Щойно екземпляр створено, дія продовжується до виконання додаткової роботи.

Що може робити дію

Дія зазвичай читає та записує атрибути того екземпляра, для якого воно виконується. Дія також може:

1. Виконувати будь-яке обчислення.

2. Створювати подію для будь-якого екземпляра будь-якого об'єкта (включаючи екземпляр, що виконується в даний час).

3. Породжувати подію для чогось поза областю аналізу: оператора, апаратного пристрою або об'єкта в іншій підсистемі.

4. Створювати, видаляти, встановлювати, скидати або зчитувати таймер (детальніше про таймери в розділі 3.7).

5. Читати та записувати атрибути інших екземплярів власного об'єкта.

6. Читати та записувати атрибути екземплярів інших об'єктів.

Зверніть увагу, що роздільна здатність читати або записувати атрибути інших об'єктів у моделях аналізу не порушує принцип проектування - інкапсуляцію. Одна з цілей аналізу полягає в тому, щоб визначити, які доступи до даних потрібні між об'єктами. Потреба звертатися до даних іншого об'єкта виявляється у моделях аналізу як читання чи запис цих даних.

Що має діяти

Щоб гарантувати несуперечність моделей станів як єдиного цілого, на дії мають бути накладені певні зобов'язання.

Збереження екземпляра несуперечливим. Дія повинна залишати дані, що описують власний екземпляр, несуперечливими. Якщо дія модифікує атрибут власного екземпляра, вона має модифікувати всі атрибути, які є обчислювально залежними від першого атрибута. Наприклад, припустимо, що ми маємо два об'єкти:

Газ В Ємності Для Газа (Назва Газа, ID Ємності (R), Температура, Тиск).

Ємність для Газа (ID Ємності, Об'єм).

Тоді, якщо дія Газа В Ємності Для Газа модифікує атрибут температури, вона повинна модифікувати атрибут тиску.

Гарантування несуперечності зв'язків. Якщо дія створює або видаляє екземпляри власного об'єкта, вона повинна гарантувати, що будь-які зв'язки, що включають екземпляри, не суперечать правилам, встановленим на інформаційній моделі. Наприклад, розглянемо два об'єкти:

Клієнт (ID Клієнта, Адреса).

Рахунок (Номер Рахунку, Сальдо, ID Клієнта (B)).

Між клієнтом та його рахунком (рахунками) існує зв'язок одним-багатьом. Тому якщо дію Рахунку видаляє останній рахунок, яким володіє певний клієнт, то необхідно або видалити цього клієнта безпосередньо, або породити подію для цього екземпляра клієнта, яке повідомить йому, що він є экс-клієнтом, тобто подія повинна давати вказівку клієнту видалити себе самостійно.

Збереження підтипів н супертипів несуперечливими. Дія повинна залишати підтипи та супертипи несуперечливо наповненими. Тому, якщо дія створює екземпляр об'єкта супертипу, воно має створювати екземпляр одного з об'єктів підтипу. Подібним чином, якщо дію видаляє екземпляр підтипу, вона повинна видаляти відповідний екземпляр супертипу.

Модифікація атрибуту поточного стану. Дія повинна модифікувати атрибут поточного стану, щоб відповідати стану, яку екземпляр займає в даний момент як результат виконання дії. Але деякі аналітики та проектувальники опускають цей крок із ДПС за домовленістю, оскільки це мають робити всі дії.

Опис дії

Опис має забезпечуватись для кожної дії. Якщо описи не надто довгі, вони розміщуються на ДПС безпосередньо під відповідною рамкою стану, інакше вони можуть бути представлені в окремому документі.

Події описуються на псевдокодї, вибраному аналітиком. Ми не наказували закінчений синтаксис для псевдокода, але знайшли корисним встановити такі умовні позначення.

1. Для того, щоб створити подію, пишемо

Породжує <мітка події>:<значення події> (<дані події>).

Наприклад:

Породжує E1: подається напруга силовий елемент (ID елемента).

2. Вказівка атрибута як <назва об'єкта><назва атрибута>

Наприклад:

Рахунок. ID Клієнта.

3. Звернення до величини атрибута певного примірника, як <назва об'єкта> <назва атрибута> (ідентифікатор).

Наприклад:

Рахунок.Сальдо(ID рахунки).

4. Позначення певного значення для атрибута укладенням значення лапки.

Наприклад:

Статус(ID Печі):= "приготування їжі перервано".

Дії, події та час

В ООА робляться певні припущення про час:

1. Тільки одна дія даного кінцевого автомата може виконуватись у будь-якій точці часу. Якщо дію розпочато, вона має завершитися перш, ніж кінцевим автоматом цього екземпляра буде отримано іншу подію.

2. Дії в різних кінцевих автоматах можуть виконуватись одночасно.

3. Події ніколи не губляться.

4. Якщо подія породжена для екземпляра, який в даний час виконує дію, ця подія не буде прийнята, доки дія не закінчена.

5. Кожна подія припиняється, коли вона представляється кінцевому автомату, подія тоді зникає як подія.

Запам'ятовування подій

Наслідком припущення "подія використовується, коли вона отримана" може бути ситуація, коли модель стану приймає подію, яка буде впливати на подальшу поведінку об'єкта, але на яку не можна реагувати доти, доки екземпляр перебуває у поточному стані. У цій ситуації подію необхідно запам'ятати або у даних, або у станах.

3.6 Переходи та таблиця переходів у стани

Правила переходів

Кожна модель станів має набір правил переходів (або переходів, для стислості), які визначають, яке досягається новий стан, якщо конкретна подія відбувається у певному стані. Перехід представляється на ДПС стрілкою, спрямованої від стану до іншого. Стрілка позначається подією (позначкою та значенням), що змушує екземпляр еволюціонувати від стану до його нащадка.

Як альтернатива правила переходів можуть бути представлені в таблиці переходів станів. У таблиці переходів у стани (ТПС) кожен рядок представляє один із можливих станів моделі станів і кожен стовпець - подія, яка має цю модель станів як призначення. Осередки таблиці заповнюються визначення того, що відбувається, коли екземпляр у цьому стані (рядок) приймає конкретну подію (стовпець).

Наповнення осередків

Існує три різні типи наповнення осередку в ТПС:

1. Новий стан.
2. Подія ігнорується.
3. Не може статися.

Новий стан. Осередок заповнюється ім'ям нового стану, в який переходить екземпляр, коли він у стані, що визначається рядком, приймає подію, що визначається стовпцем. Новий стан точно відповідає стрілці переходу на ДПС.

Зауважте, що подія може змушувати екземпляр переходити в стан, у якому він перебуває. У цьому випадку новий стан, звичайно, той самий, що і стан, що визначається рядком. Тим не менш, вважається, що це реальний перехід, і він змушує екземпляр повторно виконувати дію.

Подія ігнорується. Якщо об'єкт відмовляється реагувати на певну подію, коли він перебуває у певному стані, у відповідний осередок записують "подію ігнорується". Коли подія ігнорується, екземпляр залишається в колишньому стані і не виконує повторної дії. Таким чином, наповнення "подія ігнорується" відрізняється від переходу до того ж стану.

Зауважте, що хоча подія ігнорується (не викликає перехід), подія використовується моделлю станів.

Не може статися. Якщо подія неспроможна статися, коли екземпляр перебуває у певному стані, то відповідну осередок записують: " не може статися " . Наповнення "не може статися" призначається для випадків, коли ця подія просто не може відбутися у реальному світі.

Роль таблиці переходів у стани

Хоча ТПС та ДПС містять багато однакової інформації, ми вважаємо, що обидві форми необхідні. Діаграмна форма ДПС істотно розуміння життєвого циклу об'єкта. У нашому випадку дуже важко зрозуміти (або розробити) модель життєвого циклу, працюючи лише з ТПС. З іншого боку, безперечно те, що ТПС - чудове уявлення для перевірки закінченості та несуперечності правил переходів. Заповнення ТПС вимагає від Вас розгляду ефекту кожної комбінації стан-подія.

3.7 Таймери

Поняття таймера

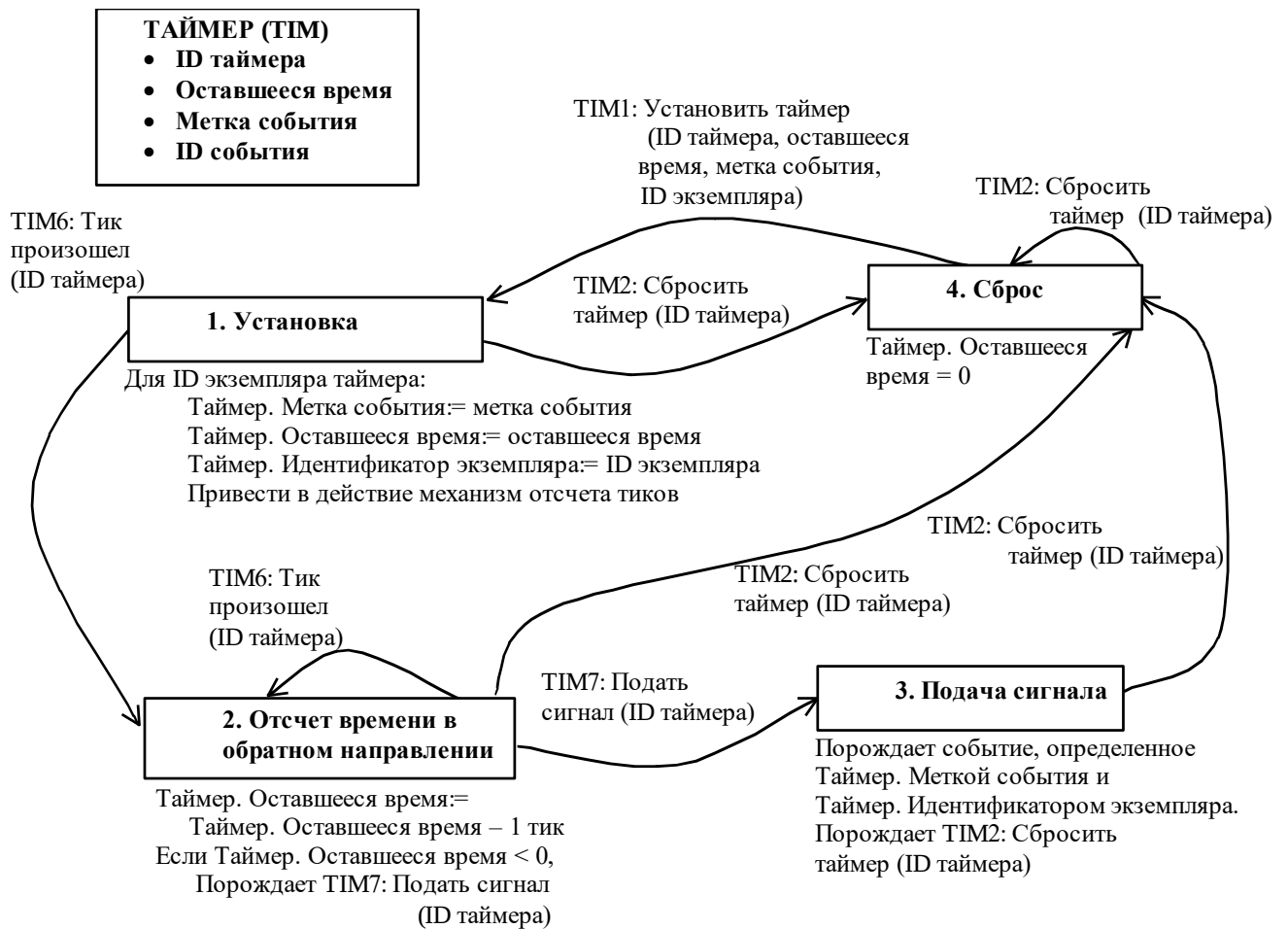


Рис. Об'єкт Таймер із своїм життєвим циклом.

Таймер- це механізм, який може використовуватись дією для створення події через певний час у майбутньому. Цей механізм є частиною формалізації -ООА і забезпечується у формі об'єкта Таймер рис. Атрибути Таймера є:

- ID Таймера – ідентифікатор власне таймера;
- Час, що залишився - час, який повинен пройти до того моменту, коли таймер подасть сигнал;
- Мітка події - позначка події, яка створюється, коли таймер подає сигнал;
- ID екземпляра – ідентифікатор, який передається з породженою подією.

7

Всі таймери діють відповідно до моделі станів Таймера, яка передбачає простий механізм рахунку в зворотному напрямку.

Умовні позначення псевдокоду для Таймерів

Наступні умовні позначення можуть бути використані в описі дії, яка використовує таймер.

1. Створення таймера записують:
ID таймера: = Створюваний таймер.
2. Для встановлення таймера породжують подію
TIM1: Встановити таймер(ID таймера, час, що залишився, позначка події, ID екземпляра).

Наприклад, для мікрохвильової печі
породжує TIM1: Встановити таймер(таймер печі, 60 с, V2, ID печі).

3. Для скидання таймера породжують подію
TIM2: Скинути таймер (ID таймера).

4. Для читання інтервалу часу, що залишився, на таймері записують так:
ще залишилося := Прочитати час, що залишився (ID таймера).

5. Видалення таймера записують так: Видалити таймер (ID таймера).

Використання таймера

Таймери зазвичай служать для сигналізації про закінчення заданого проміжку часу, необхідного інструментом або виробничим процесом, як показано на прикладі мікрохвильової печі. Крім того, таймер може використовуватися для управління ітераційними або безперервними операціями (показано в розділі 5.7) та для захисту проти відмов у прийнятих подіях (розділ 3.12).

3.8 Загальні форми життєвих циклів

Незважаючи на те, що модель станів для об'єкта може мати майже будь-які форми, деякі загальні схеми можуть бути ідентифіковані. Ми знайшли корисне дати їм назви, щоб мати можливість звертатися до цих схем.

Циркулярний цикл життя. Циркулярні життєві цикли зазвичай виникають, коли об'єкт має операційний цикл поведінки.

Життєвий цикл народження-смерть. Об'єкт має життєвий цикл народження-смерть, коли екземпляри створюються та знищуються в межах часу життя аналізованої системи. Життєвий цикл народження-смерть характеризується одним або більше станом створення та одним або більше заключним станом, у яких екземпляр руйнується. Об'єкт Рахунок, досліджений раніше, забезпечує приклад життєвого циклу народження-смерть.

У малому варіанті цієї схеми екземпляр не видаляється, а лише стає неактивним у заключному стані.

3.9 Формування життєвих циклів

Багато хто навчився користуватися моделями станів для запису зовнішньої поведінки пристрою або системи. Таке використання моделей станів наказує Вам розцінювати пристрій як "чорну скриньку", внутрішні процеси якої невідомі. Стани формуються відповідно до того, що може спостерігатися зовні, а дії опускаються, оскільки вони невидимі зовні.

Навпаки, моделі станів ООА прагнуть пояснення подробиць поведінки зсередини пристрою. Їхня мета є метою аналізу: викласти деталі поведінки таким чином, щоб Ви могли зрозуміти, чому екземпляр повинен поводитися певним способом і які дії для цього потрібні. Це використання моделей станів іноді називається використанням "білого (або світлого) скриньки".

Залежно від того, чи потрібна перспектива білої або чорної скриньки, моделі абсолютно відрізняються одна від одної.

Ціль. При ідентифікації станів важливо пам'ятати, що кожен стан має на меті. На рис.3.5.2 бак-змішувач має однакову конфігурацію в Змішуванні Продукту та стані Промивання: змішувач увімкнений, і всі клапани закриті. Тим не менш, цілі станів відрізняються і відображаються в назвах, які ми вибираємо для станів.

Контекст Кожен стан має контекст. Значення деякого стану може бути встановлене станами, які передують йому та йдуть за ним.

Повні фотографії. Не намагайтеся формувати стани, що відображають обмежені знімки. Події. Зверніть увагу на події. Стан мікрохвильової печі Приготування Їжі Завершено і Без Дії З Зачиненими Дверями різні, тому що дії повинні відрізнятися, щоб піч працювала відповідно до вимог.

3.10 Життєві цикли для підтипів та супертипів

У конструкції супертип-підтип одиночний екземпляр на інформаційній моделі представляється в об'єкті як супертипу, так і підтипу. Цей факт накладає певні додаткові правила формування життєвих циклів об'єктів супертипу і підтипу. Існує кілька різних випадків, кожен із яких описаний нижче.

Міграція підтипів

Коли екземпляр об'єкта супертипу мігрує між об'єктами підтипу, життєвий цикл може бути зображений лише на рівні супертипу чи підтипу.

Життєвий цикл супертипу, випадок міграції. Всі події, які робить цей життєвий цикл, мають своїм призначенням модель станів Бака-Змішувача.

Життєвий цикл підтипу, випадок міграції. Циркулярний життєвий цикл супертипу може бути розбитий на два цикли народження-смерть – по одному для кожного підтипу. Це виконується призначенням ряду послідовного станів для кожного підтипу. Зауважте, що події тепер необхідно спрямовувати до життєвих циклів підтипів.

Немігруючі підтипи

Коли екземпляр супертипу залишається екземпляром одного підтипу протягом усього існування, ми маємо нагоду неміграції. Тоді життєві цикли слід формувати відповідно до таких міркувань.

Життєвий цикл супертипу, випадок неміграції. Якщо динамічна поведінка екземпляра супертипу залежить від підтипу, якому належить, життєвий цикл формується лише на рівні супертипу і події, управляючі життєвим циклом, спрямовані до супертипу. Зверніть увагу на те, що в діях такого життєвого циклу є деяка if-логіка. Ця логіка буде необхідна, якщо у підтипів є власні атрибути, які отримують значення протягом життєвого циклу, або якщо підтипи входять у різні зв'язки, які повинні підтримуватися діями,

Життєвий цикл підтипу, випадок неміграції. Якщо динамічна поведінка екземпляра залежить повністю від того, якого підтипу він належить, життєві цикли формуються окремо для кожного підтипу. Події, що управляють життєвими циклами, необхідно направити до кожного окремого підтипу.

Зрощування. І, нарешті, можливо, що здебільшого життєвого циклу екземпляра його поведінка залежить від підтипу, якого він належить, але іншій частини залежить.

Цей вид поведінки міг би бути правильно документований окремою та закінченою моделлю станів для кожного підтипу. Проте в цьому випадку виникне проблема супроводу, коли та сама інформація - спільно використовувана частина життєвого циклу - повинна точно підтримуватися в моделі станів кожного підтипу. Далі як альтернатива аналітик може створювати загальну частину життєвого циклу такої ж моделі станів для супертипних та індивідуальних частин життєвого циклу, як моделі станів для кожного підтипу. При цьому спостерігається таке:

- події, що викликають переходи до (або всередині) загальної частини життєвого циклу, спрямовані на об'єкт супертипу;

- події, що викликають переходи до (або всередині) підтипних частин життєвих циклів, спрямовані на підтипи.

3.11 Коли формуються життєві цикли

Хоча у принципі, всі об'єкти мають життєві цикли, формувати моделі станів для формалізації життєвих циклів необхідно лише для деяких об'єктів. Наступні пропозиції повинні допомогти Вам вирішити, який об'єкт потребує цієї обробки. Будь ласка, зверніть увагу, що це лише керівні принципи: як завжди Ви повинні вміти самостійно розумітися на питаннях у тій конкретній ситуації, в якій Ви опинилися.

Головний керівний принцип є наступним. Якщо об'єкт має нас цікавити і має відношення до справи динамічне поведінка, то формалізуйте його життєвий цикл, інакше не робіть цього.

Створення чи знищення під час виконання. Якщо екземпляри об'єкта створюються або знищуються (або архівуються, видаляються, робляться марними тощо) під час виконання, життєвий цикл слід формалізувати. Приклад такого об'єкта є Рахунок.

Міграція між підтипами. Якщо екземпляри об'єкта мігрують від одного підтипу до іншого, життєвий цикл слід формалізувати.

Нагромадження атрибутів. Якщо екземпляри набувають значень для своїх атрибутів протягом еволюціонування через свої життєві цикли, життєвий цикл слід формалізувати.

Операційний цикл обладнання. Орієнтований обладнання об'єкт, який виявляє чіткий операційний цикл, повинен мати формалізований життєвий цикл. Робот-свердлювач я бак-змішувач – приклади таких об'єктів.

Об'єкт виробляється чи виникає поетапно. Об'єкт, який виникає крок за кроком, повинен мати формалізований цикл життя.

Завдання чи запит. Об'єкт, що є запитом або завданням, які необхідно виконати, повинен, ймовірно, мати формалізований життєвий цикл.

Динамічні зв'язки. Якщо екземпляри об'єкта вступають у зв'язку та виходять із них, життєвий цикл слід формалізувати.

Пов'язані життєві цикли. Якщо об'єкт є устаткуванням, нормальне функціонування чи збій якого змушують інший об'єкт еволюціонувати через його життєвий цикл, життєві цикли обох об'єктів необхідно формалізувати.

Пасивні засоби обслуговування. Якщо об'єкт є засобами обслуговування, які є повністю пасивними, його життєвий цикл, швидше за все, не повинен формалізуватися.

Об'єкти специфікації. Якщо об'єкт є специфікацією, стандартом якості або рецептом, його життєвий цикл, ймовірно, не повинен формалізуватися.

3.12 Аналіз відмов

Анормальна поведінка

У прикладах, розглянутих досі, ми досліджували лише нормальну чи бажану поведінку різних об'єктів. У розділі ми обговорюємо тему анормального поведінки. Це всеосяжний термін, який включає поняття помилок, відмов та незвичайної та (або) небажаної поведінки. Дослідження анормальної поведінки часто називають аналізом відмов. Судження, наведені тут, не претендують на те, щоб бути всебічними: аналіз відмов - велика тема і дуже проблемно орієнтована. У цьому розділі ми зосередили увагу на прикладах, що стосуються систем реального часу.

Мета аналізу відмов полягає в тому, щоб вивчити вплив певних видів збоїв та помилок та оцінити використовувані стратегії. Більш високою метою є зазвичай підтримка чи відновлення управління індустріальним чи іншим зовнішнім процесом. В результаті аналіз відмов часто веде

до затвердження спочатку не очікуваних вимог, встановлення додаткових датчиків, формалізації виробничих процедур на підприємстві тощо.

Аналіз відмов в ООА

В ООА аномальна поведінка формалізується в моделях станів тим самим способом, що й нормальне. Для цього рекомендується наступна процедура. По-перше, моделюють нормальну, бажану поведінку об'єктів у моделях станів, як ми робили в цьому розділі. По-друге, додають стани, події та дії до моделей нормальної поведінки, щоб охопити аномальну поведінку, що виникає у зовнішньому процесі чи устаткуванні. І нарешті, розширюють виникаючі в результаті моделі станів, щоб врахувати відмови силових приводів та датчиків.

Ключовою проблемою в розширенні моделей станів, що охоплюють аномальну поведінку, є, звичайно, ідентифікація джерел помилок та відмов. Ми знаходили корисними два підходи: що базується на дослідженні зовнішнього процесу, що ґрунтується на систематичному наскрізному контролі ООА моделей. Вони описані у наступних розділах.

Відмови у зовнішньому процесі та інтерфейсному обладнанні

Ось деякі великі питання, які розглядаються у пошуках джерел відмови у зовнішньому процесі.

Відмова чи збій обладнання. Ці відмови включають такі події, як втрата потужності, заїдання клапана, вимикання рубильника, підвищення температури тощо. Багато умов такого типу будуть відчутні, вони досить добре відомі та прості для ідентифікації. Менш простими будуть невідчутні умови на кшталт витоків.

Помилка персоналу. Помилки персоналу включають такі помилки, як напрямок поїзда неправильним шляхом або відкриття клапана в невідповідний час (за допомогою ЕОМ або вручну).

Проблеми, пов'язані з часом. Завдання змішування, розглянуте раніше, є чудовим прикладом. Припустимо, що продукт змішування залишається в баку-змішувачі протягом тривалого часу, тоді як бак-сховище вже підготовлено. Чи можливо, що інгредієнти можуть почати обстоюватися і потрібен інший цикл змішування?

Відмови датчика. Датчик може працювати неправильно: наприклад, вказує, що рубильник вимкнений, коли фактично це не так. Вивчення додатково отриманих даних допоможе виключити ситуацію, коли відмови датчика призводять до некоректного тлумачення положення, пов'язаного з роботою обладнання. У випадку з баком-змішувачем ми, можливо, перевіркою датчиків рівня зможемо визначити, відкритий або закритий клапан для випуску продукту в бак-змішувачі і в бак для зберігання.

Відмова силових приводів. Відмова силового приводу відбувається тоді, коли ЕОМ видає команди зовнішнього устрою, але він відповідає. Деякі відмови силових приводів можуть бути виявлені за допомогою таймера та відповідного виміряного значення. Створіть команду для зовнішнього пристрою та встановіть таймер на інтервал часу, після якого пристрій має відповісти. Якщо пристрій відповідає перш, ніж таймер закінчується, що доводиться відповідним зміною виміряного значення, силовий привід працював правильно і таймер повинен бути скинутий. Якщо таймер закінчується, перш ніж відповідатиме пристрій, то, можливо, сталася відмова силового приводу.

Ідентифікації відмов за допомогою ООА моделей

Ми можемо певною мірою систематично ідентифікувати джерела відмов дослідженням ООА моделей стану нормальної поведінки. Ось деякі питання та керівні засади для ідентифікації потенційних джерел відмов.

1. Який інцидент у реальному світі для кожного стану міг би змусити екземпляр переходити в новий стан? Наприклад, якби промивний вхідний клапан повинен був відкритися, коли Бак-змішувач перебуває у стані Змішування, продукт у баку-змішувачі зіпсувався б розчином для промивання. Отже, необхідно створити новий стан для опису цього стану.

2. Чи гарантується наступ кожної події? У відповідь на це питання будуть виявлені певні помилки датчиків та силових приводів, а також деякі операційні помилки.

3. Якщо подія відбувається, то чи це нормально?

4. Чи гарантується, що кожна подія відбувається до певної міри своєчасно? Що відбувається із процесом, якщо подія затримується?

5. Чи гарантується, що кожна дія має очікуваний ефект за всіляких обставин?

Заключне зауваження

Перед тим, як закінчити обговорення теми аналізу відмов, ми хотіли б зробити таке застереження. Варто виникнути темі аномальної поведінки, як просто необхідно буде враховувати та досліджувати міріади можливих відмов, від реальних та прийнятних до нездійснено штучних. Аномальна поведінка може бути дуже заплутаною, в результаті моделі стають все більш громіздкими і складними, у той час як кількість аномальних випадків, що розглядаються, продовжує збільшуватися. Зараз саме до місця визначити інженерне рішення, прийнятне в нашій ситуації, щоб не розширювати об'єкт дослідження.

3.13 Робочі продукти

Моделі станів

Кожен об'єкт, що має модель станів, документується наданням: діаграми переходів у стани (ДПС);

- таблиці переходів у стани (ТПС);
- описи кожної дії на діаграмі переходів у стані (потрібно лише якщо дії займають занадто багато місця і не можуть поміститися на ДПС).

Список подій

Список подій - це список всіх подій, визначених всім моделям станів.

Ми виявили, що електронна таблиця, база даних або CASE-засіб є надзвичайно корисними для підтримки списку подій. Інструменти такого типу роблять його простим для сортування подій у різному порядку, що полегшує розуміння та перевірку моделей станів. Зокрема, розглядають сортування списку подій:

- в алфавітному порядку за позначкою події;
- за вихідною моделлю станів;
- за моделлю станів призначення.