



ЛЕКЦІЯ 13

Розробка інформаційної моделі

- **Програмна система** розглядається як сукупність визначеного ряду доменів проблемних галузей, кожний з яких є окремим світом, населеним своїми об'єктами, і котрий аналізується незалежно від інших.
- Продуктом аналізу домену є три складові, які будуються за три етапи:
 1. онтологія домену, яку автори даного методу називають інформаційною моделлю системи або інформаційною моделлю домену;
 2. модель станів об'єктів, визначених у складі інформаційної моделі (або онтології);
 3. модель процесів, які супроводжують переходи з одного стану в інший.

- Завданням першого етапу є виявлення суттєвих об'єктів і встановлення зв'язків (відношень) між ними. Для цього ми маємо,
 1. знайти такі об'єкти
 2. дати їм унікальні та значущі назви.

- Ці дії суто суб'єктивні, єдина рекомендація щодо цього, яку надали автори методу, полягає в переліку категорій, серед яких доцільно проводити пошук.

- Це такі категорії:
- реальні предмети світу, котрі фізично втілені, наприклад, Іван Іванович, стілець у кабінеті, Дніпро;
- абстракції фізичних предметів світу, наприклад, людина, тварина, літак, кабель, дім, сад;
- ролі предметів, тобто абстракції їхнього призначення або метивикористання. Наприклад, для домену університет значущими є ролі – декан, ректор, студент, викладач; для домену хімічне виробництво –
- очисна споруда, каталізатор, відстійник; для домену адміністрація міста – платник податків, виборець, мер;
- інциденти, тобто абстрактні події, котрі впливають на зміну стану об'єкта, наприклад, паводок, вибори, залік, прогул;
- взаємодії, тобто об'єкти, які характеризують відношення об'єктів, наприклад, контракт, перехрестя доріг або вулиць, угода; специфікації, тобто подання правил, стандартів, критеріїв якості, обмежень користування.

- Склавши список виявлених об'єктів, для кожного з них потрібно визначити його характерні ознаки або властивості, які в інформатиці називають атрибутами.
- **Кожний атрибут** є абстракцією однієї з характеристик об'єкта, котрі властиві всім представникам класу об'єктів.
- За атрибутом закріплюємо ім'я, унікальне в межах класу. Зазначимо, що вдало вибрані імена (як кажуть, мнемонічні імена, тобто такі, котрі передають сутність об'єктів, які вони позначають) є важливим чинником для розуміння програм.

- Для кожного з визначених атрибутів задаються його можливі значення (типи значень).

- Способи опису можливих значень можуть бути такі:
 - числовий діапазон;
 - перелік можливих значень;
 - посилання на документ, у якому визначено можливі значення;
 - правила генерації значень.

- Для об'єкта визначається ідентифікатор – це один або кілька атрибутів, значення чи сукупність значень яких точно вирізняють екземпляр об'єкта з-поміж інших у класі.

- Прикладом ідентифікатора можуть бути:
 - назва або ім'я об'єкта,
 - табельний номер працівника (бо серед них можливі носії однакових прізвищ та імен),
 - номер паспорта,
 - код платника податків,
 - номер автомобіля.

- Сукупність атрибутів, які становлять ідентифікатор, може залежати від області визначення об'єкта.
- Так, якщо йдеться про котів однієї родини, то зазвичай кличка kota є унікальною в родині, якщо ж ідеться про котів одного двору, то, можливо, доведеться уточнити кличку kota його масть (Васько.рудий) або ім'ям його хазяїна (Васько.Олена)
- *Посилання на ідентифікатор подається як перелік через крапку атрибутів, котрі входять до складу ідентифікатора, як це видно з наведених вище прикладів. Посилання на атрибут при необхідності може уточнюватися класом, поданим через крапку, наприклад: викладач, стаж-роботи, літак.розмах-крил, собака.порода.*

- Подання інформаційної моделі в цьому методі базується на відомій реляційній моделі даних.
- Атрибути об'єктів подаються як атрибути відношень за такими правилами:
 1. кожний екземпляр об'єкта одночасно обов'язково має одне значення (тобто значення не може бути відсутнім або невизначеним);
 2. атрибут є одновимірним і не має внутрішньої структури або кількох значень одночасно;
 3. якщо до ідентифікатора входить кілька імен атрибутів, усі вказані імена атрибутів, окрім першого, належать до першого вказаного імені, яким є ім'я об'єкта.

ІДЕНТИФІКАТОРИ ОБ'ЄКТІВ

- Об'єкти нумеруються. Об'єкт зображається прямокутною рамкою, всередині якої подається номер та ім'я об'єкта, а також імена його атрибутів
- Наприклад

1. Собака
- Ім'я
- Порода
- Стать
- Вік

2. Адреса
- Вулиця
- Номер будинку
- Номер квартири

3. Особа
- Прізвище
- Ім'я
- По батькові
- Номер паспорта

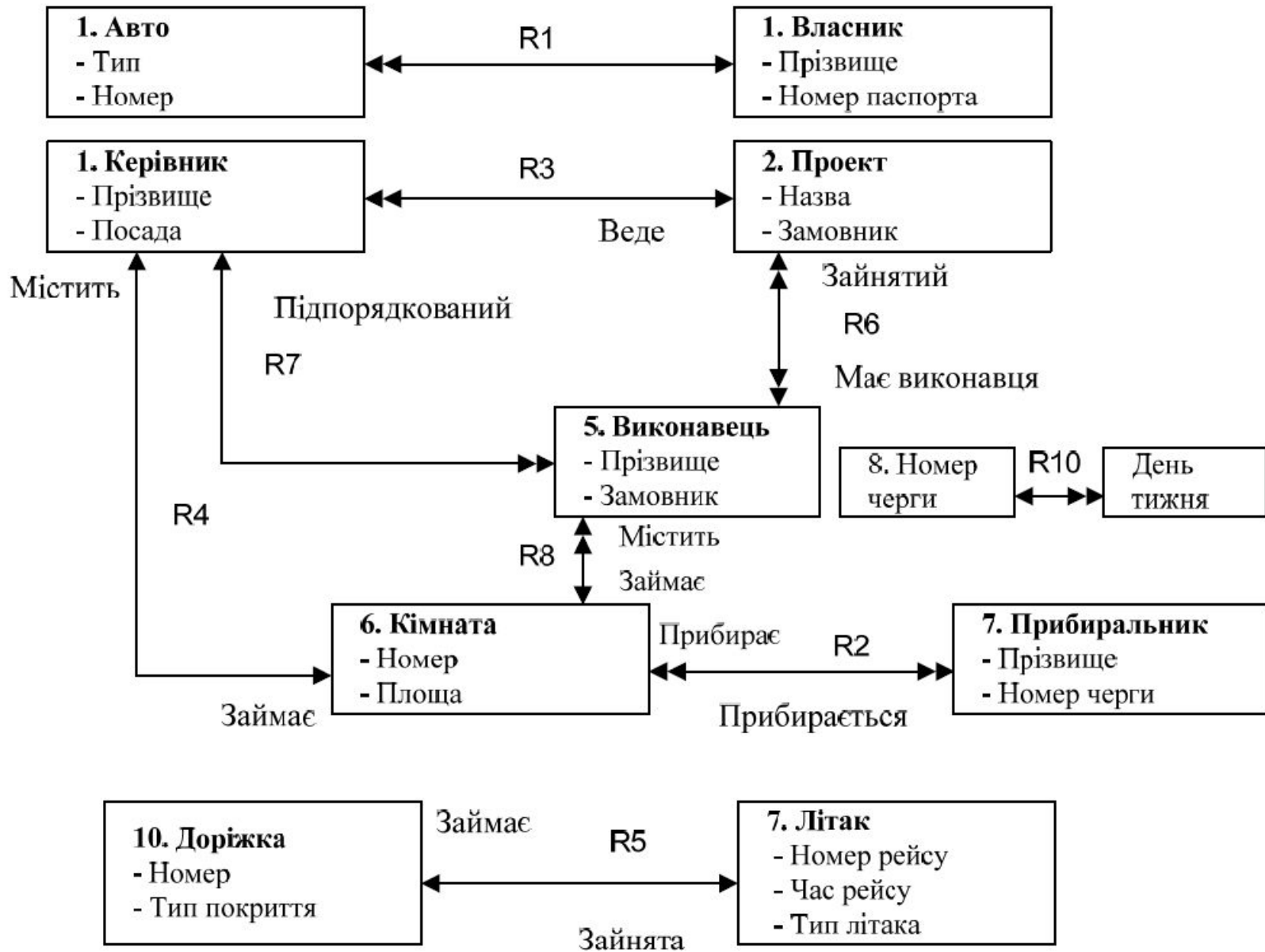
- Визначивши склад класів об'єктів домену та властиві їм атрибути, розглянемо зв'язки (відношення) між об'єктами домену.
- Об'єкти одного класу можуть брати участь у бінарних, тобто попарних зв'язках з об'єктами іншого або того самого класу.

Зв'язки ОБ'ЄКТІВ

- Розглянемо кілька прикладів зв'язку:
 1. власник авто має авто, а авто належить власнику;
 2. прибиральник прибирає кімнату, а кімната прибирається ним;
 3. проект ведеться керівником, керівник веде проект;
 4. керівник займає кімнату, кімната належить керівнику;
 5. літак займає доріжку аеродрому, доріжка зайнята літаком;
 6. проект має виконавців, виконавці зайняті в проекті;
 7. керівник керує виконавцем, виконавець підпорядкований керівнику;
 8. виконавець займає кімнату, кімната містить виконавця.
- Кожна фраза фіксує можливість екземпляра певного класу об'єктів бути у певному відношенні (або зв'язку) з екземпляром іншого класу (приклади 1–5) або того самого класу (приклад 6).
- Суттєвою рисою наведених зв'язків є число екземплярів об'єктів, які можуть одночасно брати в ньому участь.

- Розрізняються три фундаментальних види зв'язку:
 1. один до одного ($1 : 1$), коли у зв'язку беруть участь по одному екземпляру з кожного боку. Одночасно на одній злітній смузі може бути тільки один літак, і один літак може займати тільки одну смугу;
 2. один до багатьох ($1 : n$), коли один екземпляр об'єкта певного класу може підтримувати відношення одночасно з декількома екземплярами об'єктів іншого або того самого класу;
 3. багато до багатьох ($m : n$), коли у зв'язку можуть брати участь по декілька екземплярів об'єктів з кожного боку.

Зв'язки об'єктів



- Інформаційна модель проблеми на наступних фазах життєвого циклу розробки програмної системи відображається на структури баз даних. Власне, таке відображення є продуктом проектних рішень з реалізації зв'язків, задекларованих як вимоги до розробки.
- Є одне відношення, яке має особливу вагу для подання онтологій. Це відношення успадкування, за допомогою якого виражаються спільності та розбіжності між визначеними класами об'єктів. Зазвичай, відношення успадкування подаються на окремих діаграмах – на так званих діаграмах класів. На рис. 3.3 наведено приклади таких діаграм. При цьому діаграму інформаційної моделі супроводжують неформальним описом усіх об'єктів, їхніх атрибутів та зв'язків, в яких об'єкти беруть участь.

Зв'язки об'єктів

а)



б)



в)



- Модель, яку буде розглянуто нижче, має відображати динаміку змін, котрі відбуваються в стані об'єктів, або, як кажуть, динаміку їхньої поведінки. Зазначимо, що всі екземпляри одного класу об'єкта, за визначенням поняття клас, мають однакову поведінку.
- Нагадаємо базові поняття моделі динаміки поведінки об'єктів:
 - стан об'єкта визначається поточними значеннями окремих його атрибутів, а стан домену – сукупністю станів його об'єктів;
 - стан об'єкта змінюється внаслідок того, що відбулися певні події або з'явилися певні стимули;
 - зміна стану супроводжується певними процесами, котрі визначено для кожного стану як такі, що мають відбутися після досягнення цього стану.

МОДЕЛЬ СТАНІВ

- У методі, який ми розглядаємо, запропоновано одразу дві альтернативні нотації для фіксації динамічних аспектів вимог як поведінки визначених об'єктів.
 1. – графічна – називається діаграмою переходів у стани (ДПС).
 2. – таблична – називається таблицею переходів у стани (ТПС).

- Обидві нотації базуються на автомати Мура, відомому з теорії автоматів. Згідно з цим методом, побудову моделі станів починаємо з того, що серед визначених інформаційною моделлю класів об'єктів виділяємо ті, котрі мають динамічну поведінку, тобто змінюють свій стан з плином часу, або, як кажуть, мають життєвий цикл від створення екземпляра об'єкта через зміни його станів і до зникнення об'єкта.

МОДЕЛЬ СТАНІВ

- Для відображення поведінки таких об'єктів у вимогах до розробки потрібно:
- визначити множину станів, в яких об'єкт може перебувати, при тому кожний стан є абстракцією стану, в котрому може перебувати кожен з екземплярів класу об'єктів;
- визначити множину інцидентів або подій, які спонукають екземпляри класу змінювати свій стан;
- визначити для кожного із зафіксованих станів правила переходу, котрі вказують, в який новий стан перейде екземпляр даного класу, якщо певна подія з визначеної для класу множини подій відбудеться тоді, коли він перебуває в даному стані;
- визначити для кожного з визначених станів дії або процеси, які потрібно виконати при набутті даного стану.

МОДЕЛЬ СТАНІВ

- Графічна нотація для подання інформації передбачає таке:
 1. кожному стану, визначеному для класу об'єктів, присвоюють назву та порядковий номер;
 2. кожній визначеній події присвоюють унікальну мітку та назву;
 3. на діаграмі ДПС стан позначається рамкою, яка містить номер та назву стану;
 4. перехід від стану до стану зображається спрямованою дугою, позначеною міткою та назвою події, яка зумовила перехід;
 5. початковий стан позначається стрілкою, що веде до відповідної йому рамки, і є станом, в якому екземпляр об'єкта з'являється вперше (ініціалізується). Допускається кілька початкових станів на ДПС;
 6. заключний стан визначає кінець життєвого циклу екземпляра об'єкта, що може настати в одному із двох випадків – або екземпляр існує далі, але його поведінка втрачає динамічний характер, і тоді такий стан позбавляється спеціальної позначки або екземпляр зникає, і тоді заключний стан позначається пунктирною рамкою;
 7. під рамкою зазначаються дії, які має виконати екземпляр об'єкта, коли він набуває відповідного рамці стану.

МОДЕЛЬ СТАНІВ

- Зазначимо, що при застосуванні ТПС дії, відповідні станам, позначаються окремою нотацією.
- Зупинимось коротко на можливих діях при зміні станів. Дії виконуються екземпляром, котрий змінює стан. Подія, яка викликає зміни стану, є сигналом керування, що зазвичай передає якісь дані. Вони мають нести досить інформації, щоб визначити екземпляр класу, котрий змінює стан (або створити новий екземпляр) і забезпечити даними відповідні дії. Різновиди дій такі:
 - оброблення інформації, яку несе подія;
 - зміна певного атрибута об'єкта;
 - обчислення;
 - генерація події для деякого екземпляра деякого класу (можливо, для самого себе);
 - генерація події, що має передаватися зовнішнім щодо даного домену об'єктам, як-от людина-оператор, інша система, фізичний прилад тощо;
 - прийом повідомлення про події від зовнішніх об'єктів;
 - взаємодія з двома специфічними об'єктами — таймером та системним годинником.

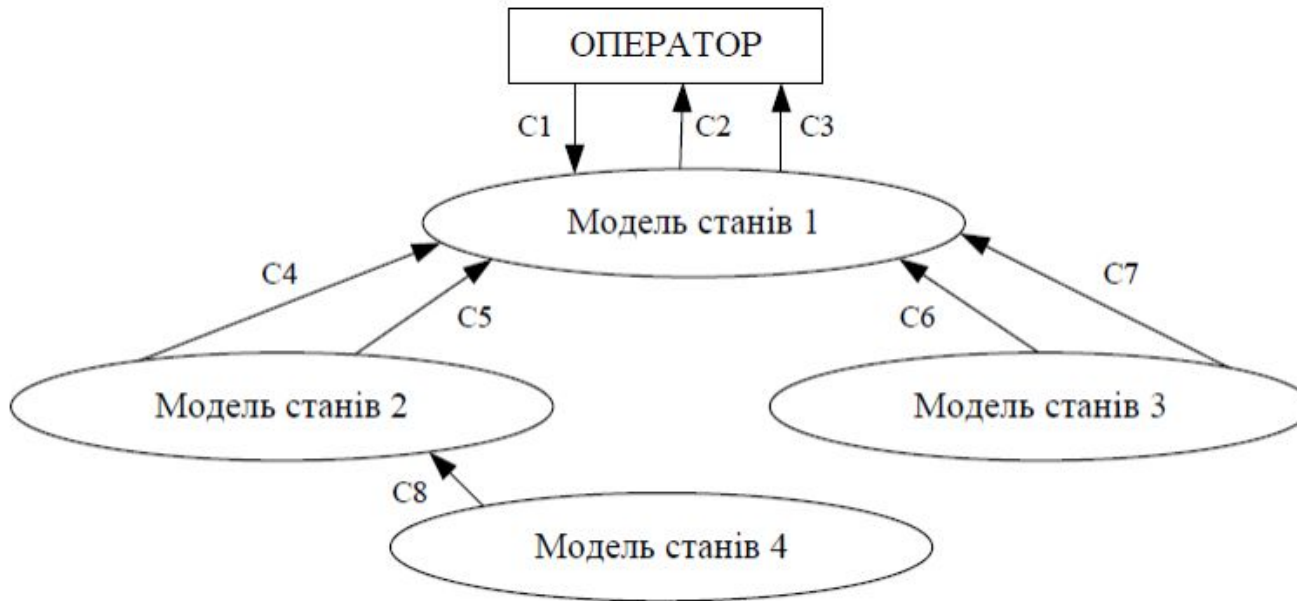
МОДЕЛЬ СТАНІВ

- Таймер – це механізм вимірювання інтервалу часу, який вважається вбудованим у даний метод системним об'єктом, що не потребує визначення.
- Атрибутами цього об'єкта є:
 1. унікальний ідентифікатор екземпляра таймера;
 2. залишок часу (інтервал часу, через який буде подано сигнал про настання певної події);
 3. мітка події, яка настане, коли залишок часу буде дорівнювати нулю;
 4. ідентифікатор екземпляра об'єкта, для якого встановлюється таймер.
- Екземпляр таймера встановлюється для окремого екземпляра певного керованого об'єкта (наприклад, бак накопичувача, духовка шафа печі, шахматист Іванчук) для повідомлення про настання події, даними якої є значення атрибутів таймера. Окремі події передбачено для скидання таймера на нуль та знищення таймера.

МОДЕЛЬ СТАНІВ

- Якщо вибрати між ДПС і ТПС, то аргументом на користь
- 1. першої нотації – ДПС – є її *наочність та визначення дій*,
- 2. друга з них – ТПС – *дозволяє зафіксувати всі можливі комбінації стан – подія і забезпечити повноту та несуперечність подання вимог.*

□



МОДЕЛЬ ПРОЦЕСІВ

- Модель станів об'єктів, за допомогою якої висуваються вимоги до поведінки системи, передбачає у своєму складі опис певних дій, котрі супроводжують зміни станів об'єктів.
- Дії є алгоритмами, що виконуються системою як реакції на події і визначають її функціональність.
- Розуміння вимог до системи передбачає і розуміння зазначених вище дій, інколи досить складних.
- Способом, що пропонується для подолання труднощів розуміння дій у даному методі, є декомпозиція їх на окремі складові, які отримали назву процесів.
- Послідовність процесів, що виконуються, утворює потік керування; водночас процеси під час виконання обмінюються даними, що утворюють потоки даних; два зазначені типи потоків пропонується використовувати як моделі алгоритмів дій системи, для подання котрих у даному методі передбачено спеціальну нотацію, якій присвоєно назву діаграми потоків даних дій

□ Як джерела даних допускаються:

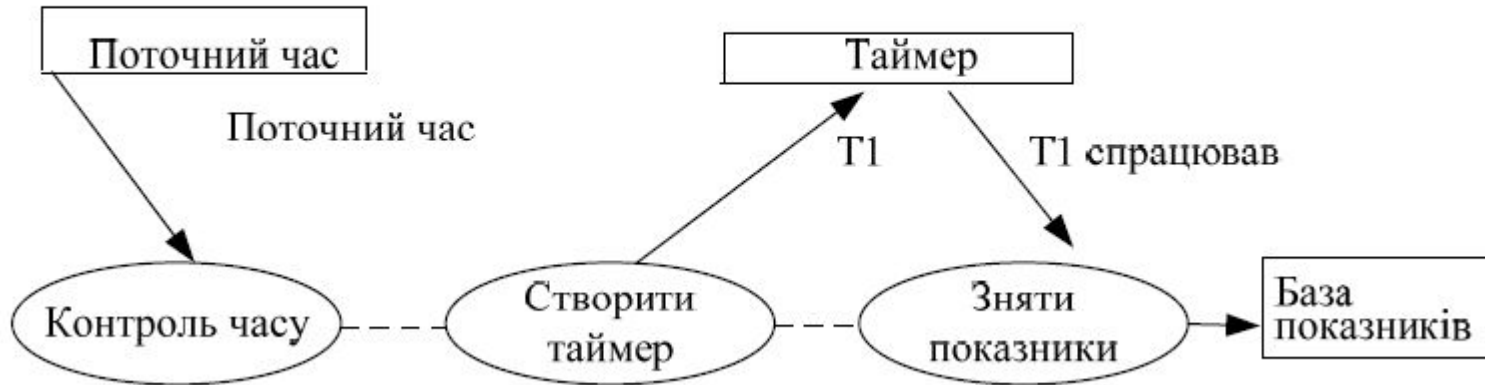
1. атрибути об'єктів (що звичайно зберігаються в архівах – файлах або базах даних, які існують і після завершення роботи системи);
2. системний годинник як показник системного часу;
3. таймери;
4. дані подій;
5. повідомлення зовнішніх об'єктів (людей-операторів, приладів тощо).

МОДЕЛЬ ПРОЦЕСІВ

- Правила побудови діаграми потоків даних дій (ДПДД) подано нижче:
- кожному з ДПС станів може відповідати тільки одна ДПДД;
- процес зображається на ДПДД як овал, всередині якого подано зміст або назву процесу;
- потоки даних зображено як стрілки, на яких вказуються ідентифікатори даних, що передаються від процесу до процесу; напрямом стрілки до овалу позначає дані, які є входами до процесу, напрямом від овалу – виходи;
- джерела даних зображено як прямокутні рамки чи рамки з відкритими сторонами;
- якщо джерелами даних є архівні об'єкти, відповідні потоки маркуються назвами атрибутів об'єктів, що передаються потоками, при цьому назва відповідного об'єкта може не вказуватися;
- потоки даних від таймера маркуються назвою таймера;
- потоки даних від системного годинника маркуються назвами показників часу (час, година, хвилина, день тощо);
- подія, повідомлення про яку отримує процес, зображається як стрілка, котра маркується назвами даних подій;
- якщо процес, який створив подію, та процес, який приймає повідомлення про подію, обидва належать до тієї самої ДПДД, відповідний потік пов'язує такі процеси;
- якщо подія, яку створив процес певної ДПДД, передається до процесу з іншої ДПДД, для першого із вказаних процесів вона позначається стрілкою, котра веде від процесу в "нікуди", а для другого – до процесу з "нізвідки", причому обидва рази стрілка маркується даними події, які передаються.

МОДЕЛЬ ПРОЦЕСІВ

- Процеси розрізняються за такими типами:
 1. так званий аксесор, що здійснює доступ до архівів;
 2. генератор подій;
 3. перетворювач даних (обчислення);
 4. перевірка умов.
- Потоки керування на ДПДД позначаються пунктирними стрілками. Якщо процес являє собою перевірку певної умови, при виконанні котрої здійснюється передавання керування до іншого процесу, то відповідний потік керування зображається перекресленою пунктирною стрілкою.



МОДЕЛЬ ПРОЦЕСІВ

- До ДПДД додається неформальний опис функцій процесів, які входять до її складу. Нотація для опису подробиць дії процесів у даному методі не регламентується і залежить від смаків авторів.

- Після побудови всіх ДПДД для всіх об'єктів системи доцільно побудувати загальну таблицю процесів для станів, до якої входять такі колонки:
 - ідентифікатор процесу;
 - тип процесу (див. вище);
 - назва процесу;
 - назва стану, в якому визначено процес;
 - назва дії стану.

- Створення такої таблиці має декілька цілей:
 1. таблиця дає можливість перевірити несуперечність назв та ідентифікаторів процесів,
 2. перевірити повноту визначених подій та процесів,
 3. перевірити, чи всі визначені події генеруються певним процесом і чи всі згенеровані події обробляються певним процесом.
 4. Крім того, наявність такої таблиці дає можливість виявити процеси, спільні для кількох дій чи станів і уніфікувати їх.

- За даним методом, результатом проведеного аналізу вимог до створюваної програмної системи є такі продукти:
 - 1. Інформаційна модель системи (онтологія) у формі:
 - діаграми сутність – зв'язок;
 - опису об'єктів та їхніх атрибутів (подається неформально);
 - опису зв'язків між об'єктами (подається неформально).

- 2. Модель поведінки об'єктів системи у формі:
 - діаграми переходів у стани (ДПС) або таблиці переходів у стани (ТПС);
 - опису дій ДПС (подається неформально);
 - опису подій ДПС (подається неформально).

- 3. Модель процесів для станів об'єктів у формі:
 - діаграми потоків даних дій (ДПДД);
 - таблиці процесів станів;
 - опису процесів (подається неформально).

- Сукупність перелічених продуктів вважається достатньою для переходу до процесу проектування системи.

ДЯКУЮ ЗА УВАГУ!