**Лекція 15. Методи об‘єктного аналізу та моделювання**

*Головна* ***мета об’єктно-орієнтованого аналізу******–*** *представити предметну область як множину об’єктів з властивостями і характеристиками, що достатні для їх ідентифікації, а також для завдання поведінки об’єктів у рамках вибраної системи понять і абстракцій.* На будь-якому етапі об’єктного аналізу всі поняття (сутності) предметної області – суть об’єкти. Кожен об’єкт – це унікальний елемент, який має принаймні одну властивість або характеристику й ідентифікується в множині об’єктів.

Предметна область сама є самостійним об’єктом або може бути об’єктом у складі іншої предметної області.

Аналіз предметної області проводиться за допомогою об’єктно-орієнтованих методів і відповідних стандартів. *Кінцева мета аналізу предметної області – визначення об’єктної моделі (ОМ), що містить у собі об’єкти та зв’язки (відношення) між ними.* При побудові ОМ виявляються функціональні задачі, формулюються вимоги до їх проектування і подання структури системи. Об’єктна модель, вимоги і задачі — необхідні умови побудови архітектури майбутньої системи.

***Огляд об’єктно-орієнтованих методів аналізу і побудови моделей.***

На даний час створено понад 50 об’єктно-орієнтованих методів, які застосовуються практично як механізми розроблення об’єктних моделей і побудови на їхній основі програмних систем. Головним поняттям цих методів є об’єкт, а також інші означення елементів предметної області, яка створюється.

Метод об’єктно-орієнтованого моделювання передбачає послідовне виконання двох етапів: об’єктно-орієнтованого аналізу та об’єктно-орієнтованого проектування.

*Об'єктний аналіз* - дослідження об'єктів предметної області (ПрО). Предметна область містить в собі ті об'єкти і взаємозв'язки між ними, які мають істотне значення при описі вимог та умов в розв'язанні задач. В процесі об'єктно-орієнтованого проектування визначають програмні об'єкти та способи їхньої взаємодії і схеми інформаційної бази, у т.ч. баз даних.

***Основні поняття об’єктно-орієнтованих методів аналізу****.*

*Об’єкт* – це абстрактний елемент, що має поведінку, обумовлену його характеристиками і відношеннями з іншими об’єктами предметної області.

Відповідно до теорії Фреге специфікацію об’єкта можна трактувати як трійку:

· **<ім’я об’єкта > <денотат > <концепт>;**

де <ім’я об’єкта> – ідентифікатор, рядок з літер і чисел;

· <денотат> – сутність реальної ПрО, що позначається цим ідентифікатором;

· <концепт > – семантика (зміст) денотата ПрО.

Схематично це можна подати за допомогою трикутника Фреге, котрий зображено на рис 1. В ньому містяться елементи реального світу, які мають такі властивості і характеристики:

· знак – ідентифікатор, який позначає денотат;

· денотат – сутність знаку, позначеного цим ідентифікатором;

· концепт – семантика денотату.

Вони визначаються на рівнях об'єктного аналізу із залученням математичних формалізмів їхнього опису та уточнення, які відрізняють один об’єкт від іншого.



Рис. 1. Подання об’єктів ПрО за трикутником Фреге

Об’єкт є іменована частина дійсної реальності з певним рівнем абстракції за наведеними характеристиками відносно вибраної ПрО. Як понятійна структура об’єкт відображає зміст концепту за об’єктним моделюванням предметної області. Одному об’єкту можуть відповідати кілька концептів залежно від вибраного рівня абстракції.

Об’єкт має зовнішню відмінності (наприклад, коричневий або білий стіл), що відрізняє його від інших об’єктів. Внутрішня особливість об’єкта (його структура, внутрішні характеристики) не впливає на зовнішню відмінність і для об’єктного моделювання не має значення.

*Сутність* – це семантично важливий об’єкт або значення об’єкта, що існує в ПрО і є абстрактним поняттям, інформацію про яке необхідно знати і/або зберігати. Ім’я сутності повинно бути унікальним в межах ПрО і може зображати тип або *клас* об’єктів. Сутність може мати синоніми (наприклад, аеропорт/аеродром).

*Концепт* – значення деякої сутності ПрО, позначається унікальним ім’ям. Головний, так званий батьківський концепт ПрО, визначається деяким набором загальних атрибутів. Концепт зображується графічно в об’єктній моделі або текстом.

*Атрибут* – це сутність концепту, що позначається ім’ям, унікальним у межах опису цього концепту.

*Відношення –* це абстракція зв’язків, що існують між різними видами об’єктів ПрО. Кожен зв’язок має унікальний ідентифікатор. Для формалізації зв’язків між концептами додаються допоміжні атрибути. Деякі зв’язки утворюються як наслідок існування інших зв’язків.

*Клас* – це множина об’єктів, що мають однакові властивості, зв’язки і методи.

*Предметна область* – це те, що аналізується з метою виділення специфічної множини понять (сутностей, об’єктів) і зв’язків між ними. На множині цих понять визначається простір проблем (problem space) і простір рішень (solution space).

*Простір проблем* – це абстрактні сутності, концепти та поняття ПрО, а *простір рішень* – це множина програмних реалізацій задач предметної області за поняттями предметної області, а саме, відповідні функціональні компоненти, що забезпечують розв’язок задач у цьому просторі. Об’єкт ПрО, як абстракція реального світу і понятійна структура, має поведінку, обумовлену властивостями даного об’єкта і його зв’язками з іншими об’єктами.

*Модель ПрО* – це сукупність понять, концептів, об’єктів і їхніх характеристик (атрибутів), а також множин синонімів і класифікованих зв’язків між об’єктами, що мають місці у просторі проблем предметної області і використовуються при проектування системи.

*Концептуальна модель* – це модель ПрО з концептів і понять без орієнтації на те, як вони подаються в конкретній системі.

Найбільш поширеним є подання ПрО за допомогою таких моделей:

– інформаційна модель системи;

– модель станів об’єктів, що може будуватися для будь-якого з об’єктів інформаційної моделі;

– модель процесів, що відображає процеси і дії, які відбуваються в системі при проходженні моделей станів через життєві цикли – одержання, породження і завершення подій у системі.

Відповідно ПрО аналізується в три етапи: інформаційне моделювання, моделювання станів, моделювання процесів. Як результат виконання цих процесів створюються ці три моделі.

Зв’язки об’єктів визначаються на процесі інформаційного моделювання, а поведінка – на процесі моделювання станів. Модель станів відображає динамічні стани об’єктів системи і їхню поведінку. На третьому процесі визначаються дії і процеси, що породжують події. Дії мають функціональну природу. Ціль моделювання процесів полягає в тому, щоб розділити процеси на дії, які разом визначають функціональний зміст системи.

Під **інформаційною моделлю** розуміється сукупність об’єктів (сутностей) ПрО, їхніх характеристик (атрибутів) і зв’язків між ними. Вона створюється за реляційним принципом: подання зв’язків між об’єктами і їхніми атрибутами у вигляді відношень.

Аналіз ПрО полягає у виявленні об’єктів, наданні їм унікальних імен, що відповідають важливим поняттям цієї предметної області. Об’єктами можуть бути:

– абстракції реально існуючих об’єктів ПрО;

– ролі як абстракції цілей або призначення людини в системі;

– взаємодії об’єктів, одержувані шляхом установлення зв’язків між ними і частинами системи;

– специфікації для подання правил, критеріїв і обмежень на застосування об’єктів у системі.

Таким чином, елементами інформаційної моделі можуть бути об’єкти, їхні атрибути й ідентифікатори, а також зв’язки між об’єктами.

Для об’єктів ПрО визначаються їхні характерні ознаки або властивості, що називають атрибутами. Кожен атрибут – це абстракція певної характеристики об’єкта, властива всім представникам класу об’єктів, яка одержує унікальне ім’я.

Розрізняються описові, додаткові атрибути та атрибути-посилання.

*Описовий атрибут* установлює реальну характеристику, що може визначатися одним з таких можливих способів:

– завданням числового діапазону;

– перерахуванням можливих значень, що може набувати атрибут;

– посиланням на документ, що визначає можливі значення;

– встановленням правил генерації припустимих значень.

*Додатковий атрибут* може набувати значень не в усіх об’єктах класу. Наприклад, для об’єктів класу «персональний комп’ютер» атрибут «тип монітора» є обов’язковим, а «тип принтера» — додатковим.

*Атрибут-посилання* визначає призначення або посилання на інший об’єкт. Наприклад, наукова стаття може містити у собі посилання на інші статті, книги тощо.

В об’єктах є один або кілька атрибутів, значення яких дозволяють однозначно виділити екземпляр об’єкта в даному класі (наприклад, табельний номер співробітника, номер паспорта й ін.).

В інформаційній моделі, а також в багатьох мовах програмування посилання на атрибут можуть уточнюватися ім’ям класу, яке записується зліва від імені атрибута і відділяється від нього крапкою, а атрибути – зв’язками, що визначаються за такими правилами:

– кожен об’єкт – екземпляр одного класу або більш ніж одного класу, характеризується набором значень своїх атрибутів,

– ідентифікатор об’єкта може складатися з кількох імен атрибутів, розділених крапками. Наприклад, *викладач****.****стаж–роботи****.****заробітна–плата*.

Між об’єктами предметної області можуть існувати семантичні *зв’язки*. Наприклад, у певному розумінні студент зв’язаний з викладачем, який викладає в його групі. *Зв’язок* — це абстракція певної змістовної залежності між об’єктами. Як правило, зв’язки встановлюються між об’єктами одного або різних класів і характеризуються кількістю екземплярів об’єктів, що одночасно можуть брати участь у цих зв’язках.

Зв’язки між об’єктами класифікуються за множинністю. Відповідно до цієї класифікації виділяють три різновиди зв’язків:

* *один до одного* (1:1) існує тоді, коли у зв’язку беруть участь по одному екземпляру від цих об’єктів (наприклад, проект ведеться менеджером, менеджер веде один проект);
* *один до багатьох* (1:*N*), існує тоді, коли один екземпляр об’єкта деякого класу може бути зв’язаний одночасно більш ніж з одним екземпляром іншого або того самого класу (наприклад, проект має виконавців, виконавці зайняті у проекті);
* *багато до багатьох* (M:N) існує тоді, коли у зв’язку можуть брати участь по декілька екземплярів об’єктів з кожного класу, тобто один або більше екземплярів одного класу зв’язані з одним або декількома екземплярами іншого або того самого класу (наприклад, проект має виконавців, виконавці зайняті одночасно у кількох проектах).

Ці зв’язки можуть бути статичними (постійними) – такі, що не змінюються або змінюються рідко, і динамічними, що можуть змінюватися під час функціонування системи. Зв’язки між об’єктами з часом можуть еволюціонувати істотно впливати на хід розв’язання задачі. Для таких випадків зв’язку будується асоціативний об’єкт і визначається модель станів цього об’єкта шляхом додавання атрибута, що фіксує поточний стан.

Серед дій, що супроводжують переходи об’єктів у інші стани, повинні бути дії зі створення нового екземпляра асоціативного об’єкта, якщо нова пара екземплярів вступає в зв’язок, або зі знищення, якщо об’єкт або зв’язок перестають існувати.

Крім зв’язків розглянутих типів, між класами об’єктів ПрО може існувати відношення *успадкування*, що дозволяє визначити їх спільності та розбіжності. Коли клас B містить у собі усі атрибути й операції класу A і, можливо, має ще додаткові атрибути або операції, він (клас B) називається *підкласом* або *нащадком*, а клас A – *суперкласом,* або *предком*. Класи можуть утворювати ієрархію успадкувань довільної глибини, в яких кожний відповідає певному рівню абстракції і є узагальненням класу–нащадку та конкретизацію класу– предку. Наприклад, клас «число» має підкласи: цілі, дійсні, комплексні числа. Ці підкласи успадковують операції суперкласу, а саме, операції додавання, віднімання тощо. Але кожний підклас має свої особливості виконання цих операцій.

У структурному аналізі і проектуванні використовуються різні моделі, що описують функціональну структуру системи як ієрархію діаграм потоків даних, які, в свою чергу, описують асинхронний процес перетворення інформації від її введення в систему до видачі споживачеві. **Діаграма потоків даних** (англ. *Data Flow Diagram -* DFD) — модель проектування, графічне представлення «потоків» даних в [інформаційній системі](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), яку також використовують для візуалізації процесів обробки даних в структурному проектуванні. Практично, будь-який клас систем успішно моделюється за допомогою DFD-орієнтованих методів, оскільки вони із самого початку створювалися як засіб проектування інформаційних систем. Аналогічні методи пропонує м**етодологія SADT (**англ. *structured analysis and design technique***)**, яка являє собою сукупність методів, правил і процедур, призначених для побудови функціональної моделі об'єкта будь-якої предметної області. Функціональна модель **SADT** відображає функціональну структуру об'єкта, тобто вироблені їм дії та зв'язку між цими діями. SADT - як засіб моделювання систем взагалі має багатший набір елементів, що адекватно відображають специфіку таких систем.

Структурний аналіз - метод дослідження статичних (сталих) характеристик ієрархічно впорядкованої системи шляхом виділення в ній підсистем і елементів різного рівня і визначення відносин і зв'язків між ними.

Засоби структурного аналізу приблизно однакові з погляду функціональних можливостей засобів моделювання. При цьому одним з основних критеріїв вибору того чи іншого методу є ступінь володіння ним з боку консультанта або аналітика.

Найбільш поширеним засобом моделювання даних є модель "сутність - зв'язок" (Entity-Relationship Model - ERM). Вона вперше була введена П. Ченом у 1976 р. Ця модель традиційно використовується у структурному аналізі і проектуванні, проте, по суті, це підмножина об'єктної моделі предметної області. Один з різновидів моделі "сутність - зв'язок" використовується в методі IDEF1-X, що належить сімейству стандартів IDEF, і реалізується у низці поширених CASE-засобів (зокрема, AH Fusion ERwin Data Modeler).

Розглянемо більш детально ці засоби.

*Модель потоків даних* (Data Flow Model – DFM) використовується для опису процесів обробки даних у системі й містить у собі:

– ієрархічний набір діаграм потоків даних (Data Flow Diagram – DFD);

– опис елементарних процесів, потоків даних, **сховищ даних** і зовнішніх сутностей.

Сховище даних (англ. data warehouse) - предметно орієнтований, інтегрований, незмінний набір даних, що підтримує хронологію і здатний бути комплексним джерелом достовірної інформації для оперативного аналізу та прийняття рішень.

Діаграми потоків данихDFD є ієрархією функціональних процесів, пов'язаних потоками даних. Мета такого представлення - показати, як кожен процес перетворює свої вхідні дані у вихідні, а також виявити відношення між цими процесами. Для побудови DFD традиційно використовуються дві різні нотації, відповідні методам Йордона - ДеМарко і Гейна - Серсона. Ці нотації відрізняються одна від одної графічним зображенням символів.

Кожна DFD відбиває проходження даних через систему залежно від рівня та призначення діаграми. DFD перетворює вхідні потоки даних (входи) у вихідні потоки даних (виходи). Як правило, процеси, що виконують такі перетворення, створюють і використовують дані зі сховища даних.

Першим кроком при побудові ієрархії DFD діаграм є побудова контекстних діаграм, які показують, як система буде взаємодіяти з користувачами та іншими зовнішніми системами. При проектуванні простих систем достатньо однієї контекстної діаграми, яка має зіркову топологію, в центрі якої розміщується основний процес, з’єднаний з джерелами і приймачами інформації.

Для складних систем будується ієрархія контекстних діаграм, яка визначає взаємодію основних функціональних підсистем проектованої системи як між собою, так і з зовнішніми вхідними і вихідними потоками даних і зовнішніми об’єктами. При цьому контекстна діаграма верхнього рівня містить набір підсистем, з’єднаних потоками даних. Контекстну діаграми наступного рівня деталізують вмістиме і структуру підсистем.

***Метод SADT*** є сукупністю правил і процедур, призначених для побудови функціональної моделі об'єкта певної предметної області. Функціональна модель SADT відображає функціональну структуру об'єкта, тобто його дії і зв'язки між цими діями. Метод SADT розроблений Дугласом Россом у 1969 р. для моделювання штучних систем середньої складності. Цей метод успішно використовувався у військових, промислових і комерційних організаціях США для вирішення широкого кола завдань, таких як довгострокове і стратегічне планування, автоматизоване виробництво і проектування, розробка ПЗ для оборонних систем, управління фінансами і матеріально-технічним постачанням тощо. Метод SADT підтримується Міністерством оборони США, яке було ініціатором розробки сімейства стандартів IDEF (Icam DEFinition), які є основною частиною програми ІСАМ (Integrated Computer-Aided Manufacturing - інтегрована комп'ютеризація виробництва), що проводиться за ініціативою BBC США. Це стандарти ***IDEF-0*** - ***IDEF-14.***

***IDEF-0*** - це методологія функціонального моделювання. За допомогою наочної графічної мови система представляється у вигляді набору взаємопов'язаних функцій. ***IDEF-1*** - методологія моделювання інформаційних потоків, що дозволяє відображати та аналізувати їх структуру і взаємозв'язки. ***IDEF-1Х (***IDEF1 Extended***)*** - методологія побудови реляційних структур на основі моделі "сутність - зв'язок". ***IDEF-2*** - методологія динамічного моделювання розвитку систем. ***IDEF-3*** - методологія документування процесів, що відбуваються в системі і використовуються, наприклад, при дослідженні технологічних процесів. ***IDEF-4*** – методологія побудови об’єктно-орієнтованих систем.

Метод SADT реалізовано саме в одному зі стандартів цього сімейства - IDEF-0, який був затверджений як федеральний стандарт США в 1993 р.

Моделі SADT (IDEF0) традиційно використовуються для моделювання організаційних систем (бізнес-процесів). Метод SADT успішно функціонує тільки при описі добре специфікованих і стандартизованих бізнес-процесів у зарубіжних корпораціях, тому він і прийнятий у США як типовий. Перевагами застосування моделей SADT для опису бізнес-процесів є:

* повнота опису бізнес-процесу (управління, інформаційні і матеріальні потоки, зворотні зв'язки);
* жорсткі вимоги методу, що забезпечують отримання моделей стандартного вигляду;
* відповідність підходу до опису процесів стандартам ISO 9000.

На вітчизняних підприємствах бізнес-процеси почали формуватися і розвиватися порівняно недавно. Вони слабо типізуються, тому розумніше орієнтуватися на менш жорсткі моделі.

Метод моделювання IDEF-3, що є частиною сімейства стандартів IDEF, розроблено у 1980 р. для закритого проекту Міноборони США. Цей метод призначений для таких моделей процесів, у яких важливо зрозуміти послідовність виконання дій і взаємозалежності між ними. Хоча IDEF-3 і не досяг статусу федерального стандарту США, він набув значного поширення серед системних аналітиків як доповнення до методу функціонального моделювання IDEF-0 (моделі IDEF-3 можуть використовуватися для деталізації функціональних блоків IDEF-0, що не мають діаграм декомпозиції). Основою моделі IDEF-3 слугує сценарій процесу, що виділяє послідовність дій і під-процесів аналізованої системи.

SADT – це сукупність правил і процедур, призначених для побудови функціональної моделі предметної області, яка відображає функціональну структуру, функції і дії, а також зв’язки між ними. На стадії проектування моделі системи зображаються у вигляді діаграм або екранних форм і відображають структуру або архітектуру системи, а також схеми програм.

Метод SADT базується на наступних концепціях:

– **графічне зображення** структури з поданням функцій блоками, а інтерфейсів дугами, що, відповідно, входять у блок і виходять з нього (рис.2).

Схема (грец. Σχήμα - образ, вид) - спрощене зображення, викладення чогось у загальних, основних рисах. Кресленик, що передає основну ідею конструкції машини, приладу тощо за допомогою умовних позначень.

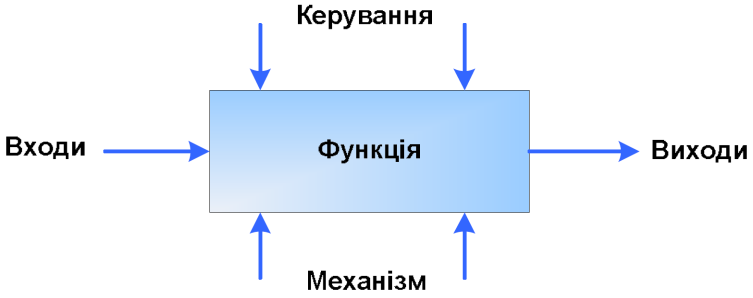


Рис. 2. Структура моделі

– блоків може бути від 3 до 6 на кожному рівні декомпозиції;

– взаємодія блоків описується обмеженнями, які визначають умови керування й виконання функцій;

– унікальність позначок і найменувань;

– незалежність функціональної моделі від організаційної структури колективу розробників.

Метод SADT застосовується при моделюванні широкого кола систем, для яких визначаються вимоги й функції, а потім проводиться їхня реалізація. Засоби SADT можуть застосовуватися при аналізі функцій у діючій програмній системі, а також при визначенні способів їхньої реалізації.

Результат проектування в методі SADT – модель, що складається з діаграм, фрагментів текстів і глосарію з посиланнями один на одного. Всі функції й інтерфейси зображаються діаграмами у вигляді блоків і дуг. Місце з’єднання дуги з блоком визначає тип інтерфейсу. Керуюча інформація позначається дугою, яка входить у блок зверху, у той час як інформація, що піддається обробці, вказується з лівої сторони блоку, а результати виходу – з правої сторони. Механізм, що здійснює операцію (людина або автоматизована система), задається дугою, що входить у блок знизу.

Одна з найбільш важливих переваг методу SADT – поступова деталізація моделі системи в міру додавання функцій і діаграм, що уточнюють цю модель.

Найпоширеніші засоби **моделювання даних** – діаграми «сутність-зв’язок» (ER-діаграми), запропоновані Баркером, як застосування класичної ER-моделі Чена.

Відно́влення да́них (англ. Data recovery) - процес порятунку або відновлення доступу до даних, що зберігаються на будь-якому носії пристрою запам'ятовування, коли вони не можуть бути доступні в звичайному режимі.

Моделюва́ння да́них (англ. Data modeling) в програмній інженерії - це процес створення моделі даних для інформаційна система з застосуванням формальних підходів.

В ER-діаграмах визначаються сутності (множини однотипних об’єктів) ПрО, їхні властивості (атрибути) і залежності (зв’язки). Сутність (Entity) – реальний або уявлюваний об’єкт, що має істотне значення для області. Кожна сутність й її екземпляр мають унікальні імена. Сутність має такі властивості:

– один або кілька атрибутів, які або належать сутності, або успадковуються через зв’язок (Relationship);

– довільну кількість зв’язків з іншими сутностями моделі.

*Зв’язок* – це асоціація між двома сутностями ПрО. У загальному випадку кожен екземпляр сутності-батька асоційований з довільною кількістю екземплярів успадкованої сутності (нащадка), а кожен екземпляр сутності-нащадка асоційований з одним екземпляром сутності-батька. Таким чином, екземпляр сутності-нащадка може існувати тільки при наявності сутності-батька. Для зв’язків можуть встановлюватися обмеження на кількість екземплярів сутності, що беруть участь у зв’язку. Наприклад, одному екземпляру однієї сутності може відповідати не більше ніж один екземпляр іншої.

Концептуальною основою об'єктно орієнтованого аналізу і проектування ПЗ (ООАП) є ***об'єктна модель.*** її основні принципи (абстрагування, інкапсуляція, модульність та ієрархія) і поняття (об'єкт, клас, атрибут, операція, інтерфейс тощо) найчіткіше сформульовані Г. Бучем у його фундаментальних працях.

Більшість сучасних методів ООАП базуються на використанні мови **UML**. Уніфікована мова моделювання UML (Unified Modeling Language) є мовою для визначення, подання, проектування і документування програмних систем, організаційно-економічних систем, технічних систем та інших систем різної природи. UML містить стандартний набір діаграм і нотацій найрізноманітніших видів.

UML - це наступник того покоління методів ООАП, які з'явилися в кінці 1980-х і на початку 1990-х років. Створення UML фактично розпочалося в кінці 1994 p., коли Граді Вуч і Джеймс Рамбо почали роботу щодо об'єднання їх методів Booch і ОМТ (Object Modeling Technique) під егідою компанії Rational Software. До кінця 1995 р. вони створили першу специфікацію об'єднаного методу, названого ними Unified Method. Тоді ж у 1995 р . до них приєднався автор методу OOSE (Object-Oriented Software Engineering) Івар Якобсон. Таким чином, UML є прямим об'єднанням і уніфікацією методів Г. Буча, Д. Рамбо і Г. Якобсона, проте доповнює їх новими можливостями. Головними при розробці UML були такі цілі:

* надати користувачам готову до використання виразну мову візуального моделювання, що дозволяє їм розробляти осмислені моделі й обмінюватися ними;
* передбачити механізми розширюваності і спеціалізації для розширення базових концепцій;
* забезпечити незалежність від конкретних мов програмування і процесів розробки;
* забезпечити формальну основу для розуміння цієї мови моделювання (мова має бути одночасно точною і доступною для розуміння, без зайвого формалізму);
* стимулювати зростання ринку об'єктно орієнтованих інструментальних засобів;
* інтегрувати кращий практичний досвід.

UML прийнятий на озброєння практично всіма найбільшими компаніями - виробниками ПЗ (Microsoft, Oracle, IBM, Hewlett-Packard, Sybase тощо). Крім того, практично всі світові виробники CASE-засобів, крім IBM Rational Software, підтримують UML у своїх продуктах (Oracle Designer, Together Control Center (Borland), AllFusion Component Modeler (Computer Associates), Microsoft Visual Modeler). Діаграма (diagram) — графічне представлення сукупності елементів моделі у формі зв'язного графа, вершинам і ребрам (дугам) якого приписується певна семантика. Нотація канонічних діаграм - основний засіб розробки моделей на мові UML.

У нотації мови UML визначені наступні види канонічних діаграм:

* варіантів використання (use case diagram)
* класів (class diagram)
* кооперації (collaboration diagram)
* послідовності (sequence diagram)
* станів (statechart diagram)
* діяльності (activity diagram)
* компонентів (component diagram)
* розгортання (deployment diagram)

Перелік цих діаграм і їх назви є канонічними в тому сенсі, що є невід'ємною частиною графічної нотації мови UML. Більш того, процес ООАП нерозривно пов'язаний з процесом побудови цих діаграм. При цьому сукупність побудованих таким чином діаграм є самодостатньою в тому сенсі, що в них міститься вся інформація, яка необхідна для реалізації проекту складної системи.

Кожна з цих діаграм деталізує і конкретизує різні уявлення про модель складної системи в термінах мови UML. При цьому діаграма варіантів використання є найбільш загальною концептуальною моделлю складної системи, яка є початковою для побудови всіх останніх діаграм. Діаграма класів, за своєю суттю, логічна модель, що відображає статичні аспекти структурної побудови складної системи.

Діаграми кооперації і послідовностей є різновидами логічної моделі, які відображають динамічні аспекти функціонування складної системи. Діаграми станів і діяльності призначені для моделювання поведінки системи. І, нарешті, діаграми компонентів і розгортання служать для представлення фізичних компонентів складної системи і тому відносяться до її фізичної моделі.

**Umbrello** - The UML Modeller <https://umbrello.kde.org/>

<https://www.quality-assurance-group.com/top-10-bezkoshtovnyh-onlajn-redaktoriv-dlya-stvorennya-uml-diagram-na-probu/> - перелік безкоштовних on-line редакторів

*Для самостійного вивчення* *(2 години)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література*.

1. Буч Гради, Максимчук Роберт А., Энгл Майкл У., Янг Бобби Дж., Коналлен Джим, Хьюстон Келли А.. Объектно-ориентированный анализ и проектирование с примерами приложений. М.: Вильямс, 2008.- 720с.
2. Буч Г., Рамбо Дж., Джекобсон А. Язьік UML. Руководство пользователя.: Пер. с англ. - М.: ДМК, 2006. - 432с. URL: <http://www.dut.edu.ua/uploads/l_1508_23942623.pdf>, <http://bourabai.kz/dbt/uml/content.html>
3. Лавріщева К.М. Програмна інженерія. Електронний підручник. URL: <http://csc.knu.ua/uk/library/books/lavrishcheva-6.pdf>.
4. Соммервиль И.. Инженерия программного обеспечения, 6 изд. – И.д. "Вильямс", 2013. URL: <https://www.studmed.ru/download/sommervill-ian-inzheneriya-programmnogo-obespecheniya_4935164f089.html>

*Контрольні запитання.*

1. Визначте задачі аналізу предметної області.
2. На які етапи можна розкласти аналіз предметної області?
3. Надайте визначення інформаційної моделі та визначте, що може бути її об’єктами.
4. Які атрибути можуть бути у об’єктів предметної області?
5. Чим характеризуються семантичні *зв’язки* між об’єктами предметної області ?
6. З якою метою використовують DFD – діаграми?
7. В чому полягає метод SADT?
8. В чому сутність ER - діаграм?
9. Які цілі ставились при розробці UML?