

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ХАРКІВСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ
МІСЬКОГО ГОСПОДАРСТВА

М. В. Гринчак, А. Л. Шаповалов, К. В. Кузьмичова

Конспект лекцій

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В БУДІВНИЦТВІ І БАЗИ ДАНИХ

*(для студентів 3 курсу денної та заочної форм навчання напряму підготовки
6.060101 «Будівництво» спеціальності «Промислове і цивільне будівництво»)*

Харків – ХНАМГ – 2011

Гринчак М. В. Конспект лекцій з дисципліни «Інформаційні технології в будівництві і бази даних» (для студентів 3 курсу денної та заочної форм навчання напряму підготовки 6.060101 «Будівництво» спеціальності «Промислове і цивільне будівництво») / М. В. Гринчак, А. Л. Шаповалов, К. В. Кузьмичова; Харк. нац. акад. міськ. госп-ва. — Х.: ХНАМГ, 2011 – 66 с.

Автори: к. т. н., доц. М. В. Гринчак
к. т. н., доц. А. Л. Шаповалов
К. В. Кузьмичова

Рецензент: к. ф.- м. н., доц. О. Б. Костенко

Рекомендовано кафедрою ПМ і ІТ
протокол № 5 від 4 січня 2010 р.

© М. В. Гринчак, А. Л. Шаповалов,
К. В. Кузьмичова, ХНАМГ, 2011

Зміст

Зміст.....	3
1. ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ.....	4
1.1. Поняття інформаційної технології.....	4
1.2. Види інформаційних технологій.....	6
1.3. Інформаційні технології кінцевого користувача.....	9
1.4. Мережні інформаційні технології.....	10
1.5. Інтеграція інформаційних технологій.....	12
2. ОСНОВИ СИСТЕМНОГО ПІДХОДУ В ПРОЕКТУВАННІ.....	15
2.1. Принципи системного підходу в проектуванні.....	15
2.2. Рівні проектування.....	18
3. ОСОБЛИВОСТІ Й ПРИНЦИПИ ПОБУДОВИ САПР.....	22
3.1. Мета створення САПР.....	22
3.2. Склад САПР.....	22
3.3. Основні принципи побудови САПР.....	24
3.4. Стадії створення САПР.....	25
3.5. Відображення процесу проектування в програмне забезпечення САПР.....	27
3.6. Специфіка інформаційного забезпечення САПР.....	30
4. СУЧАСНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ.....	31
4.1. Структура САПР.....	33
4.2. Введення в CALS-технології.....	35
5. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ САПР ТЕХНІЧНИХ ОБ'ЄКТІВ.....	38
5.1. Моделювання об'єктів і процесів.....	38
5.2. Класифікація математичних моделей об'єктів $F(x,y)$ у САПР.....	39
6. ОПТИМІЗАЦІЯ В САПР - ТЕХНОЛОГІЇ МСAD.....	43
6.1. Основи теорії.....	43
6.2. Види завдань математичного програмування.....	44
6.3. Оптимізація в середовищі МСAD:.....	45
Функції Minimize й Maximize.....	45
7. ПОНЯТТЯ ПРО БАЗИ ДАНИХ ТА СИСТЕМИ КЕРУВАННЯ НИМИ.....	46
7.1. Класифікація баз даних. Основні засоби обробки даних.....	46
7.2. Основні види моделей БД.....	47
7.3. Реляційні бази даних та їх особливості. Види зв'язку між реляційними базами.....	48
7.4. СКБД Access та їхня структура.....	50
7.5. Операції з таблицями. Фільтрація даних.....	51
7.6. Запити до бази даних. Види запитів та технологія їх створення.....	53
7.7. Створення форм. Їх види, властивості та структура.....	55
7.8. Звіти, їх створення, призначення та використання.....	58
7.9. Макроси та їх конструювання.....	59
8. КОМП'ЮТЕРНА ГРАФІКА. ПРОГРАМИ CORELDRAW ТА AUTOCAD.....	61
8.1. Растрові й векторні зображення.....	61
8.2. Об'єктно-орієнтовний підхід в Corel Draw.....	62
8.3. Опис AutoCAD.....	63
ІНФОРМАЦІЙНО-МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	64

1. ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

1.1. *Поняття інформаційної технології*

Технологія прийняття рішень завжди мала інформаційну основу, хоча обробка даних і здійснювалася вручну. Однак із впровадженням засобів обчислювальної техніки в процес керування з'явився спеціальний термін - інформаційна технологія.

Інформаційна технологія - процес, який використовує сукупність засобів і методів збору, обробки й передачі даних (первинної інформації) для одержання інформації нової якості про стан об'єкта, процесу або явища.

Мета інформаційної технології (ІТ) - постачання інформації для її аналізу людиною й прийняття на його підставі рішення до виконання якої-небудь дії.

Особливістю інформаційних технологій є те, що в них і предметом, і продуктом праці є інформація, а знаряддями праці - засоби обчислювальної техніки й зв'язку.

Нова інформаційна технологія - інформаційна технологія з "дружнім" інтерфейсом роботи користувача, яка використовує персональні комп'ютери й телекомунікаційні засоби.

Роль інформаційних технологій у розвитку економіки й суспільства

Поява найпростіших інформаційних технологій можна віднести до початку писемності. Це й наскальні малюнки, і знаки й малюнки на корі, на папірусі й т.д.

Винахід радіо, телебачення, а потім і комп'ютера, цифрових систем зв'язку й обчислювальних мереж, створення в 1978 році першого персонального комп'ютера й зовсім неймовірно й винятково швидке його поширення й дозволило новим, автоматизованим інформаційним технологіям (АІТ) впровадитися практично в усі області людської діяльності.

Властивості інформаційних технологій

Інформаційні технології мають наступні властивості:

- Складність (тому що велика кількість розробок).
- Цінність (тому що користувач може звертатися більшим масивам інформації у вигляді бази даних й інформаційної продукції широкої номенклатури).
- Різноманіття елементів (тобто включає весь набір елементів, необхідний для досягнення поставленої мети).
- Подільність (забезпечувати високий ступінь розподілу всього процесу інформації на етапи, операції, дії).
- Має регулятивний характер (тобто всі етапи, дії, операції технологічного процесу можуть бути стандартизовані й уніфіковані, що дозволяє більш ефективно здійснювати цілеспрямоване керування інформаційними процесами).
- Інтегрованість (взаємозв'язок з іншими програмними продуктами).
- Інтерактивність (діалоговий режим роботи з комп'ютером).
- Гнучкість процесу як зміни даних, так і постановки завдань.

Технологія спілкування з комп'ютером. Види користувальницького інтерфейсу

Користувальницький інтерфейс містить у собі три поняття:

- спілкування додатка з користувачем;
- спілкування користувача з додатком;
- мова спілкування.

Користувальницький інтерфейс представляє засіб взаємодії користувача з персональним комп'ютером (ПК).

У цей час широке поширені користувальницькі інтерфейси:

- командні
- графічні .

Командний користувальницький інтерфейс надає користувачеві можливість звертатися до ПК із деяким завданням (запитом), що надається деяким текстом (командою) на спеціальній командній мові (мові завдань).

Графічний користувальницький інтерфейс дозволяє

- розміщати на екрані безліч різних вікон, у які можна виводити інформацію незалежно;
- використовувати графічні об'єкти, називані піктограмами (або іконами), для позначення різних інформаційних об'єктів або процесів;
- використовувати екранний покажчик для вибору об'єктів (або їхніх елементів), розміщених на екрані.

Технології організації даних

Розрізняються наступні способи обробки даних:

1. централізований

На одному сервері перебуває єдина копія бази даних. Всі операції з базою даних (БД) забезпечуються цим сервером. Доступ до даних виконується за допомогою віддаленого запиту або віддаленої транзакції.

2. децентралізований

Децентралізована організація даних припускає розбивку інформаційної бази на декілька фізично розподілених. Кожен клієнт користується своєю базою даних, яка може бути або частиною загальної інформаційної бази, або копією інформаційної бази в цілому, що приводить до її дублювання для кожного клієнта.

3. розподілений

Розподілений спосіб обробки даних заснований на розподілі функцій обробки між різними ЕОМ, включеними в мережу.

4. інтегрований

Інтегрований спосіб обробки інформації передбачає створення інформаційної моделі керованого об'єкта, тобто створення розподіленої бази даних.

Такий спосіб забезпечує максимальну зручність для користувача. З одного боку, бази даних передбачають колективне користування й централізоване керування. З іншого боку, обсяг інформації, розмаїтість розв'язуваних завдань вимагають розподілу бази даних. Особливістю цього способу є відділення технологічно й за часом процедури обробки від процедур збору, підготовки й введення даних.

Технологічний процес обробки даних

Процес обробки даних в інформаційній технології переслідує певну мету - рішення за допомогою ЕОМ обчислювальних завдань, які відображують функціональні завдання тієї системи, у якій ведеться керування.

Процес обробки може бути розбитий на ряд зв'язаних між собою процедур:

- організація обчислювального процесу;
- перетворення даних;
- відображення даних.

При обробці даних за допомогою ЕОМ розрізняють *три основних режими*.

1. Пакетний

При пакетному режимі обробки завдань програми з відповідними вихідними даними накопичуються у дисковій пам'яті ЕОМ, утворюючи "пакет". Обробка завдань здійснюється у вигляді їхнього безперервного потоку.

2. Поділу часу

Режим поділу часу реалізовується шляхом виділення для виконання завдань певних інтервалів часу, названих квантами. Протягом одного кванта обробляється одне завдання, потім виконання першого завдання припиняється із запам'ятовуванням отриманих проміжних результатів й у наступний квант обробляється друге завдання й т.д.

3. Реального часу

Режим реального часу використовується при обробці даних в інформаційних технологіях, призначених для керування фізичними процесами

В ЕОМ використовуються також режими, названі однопрограмними й мультипрограмними. У режимі поділу часу використовується варіант мультипрограмного режиму.

1.2. Види інформаційних технологій

Класифікація інформаційних технологій

Інформаційні технології в цей час можна класифікувати по ряду ознак:

- способу реалізації в ІС;
- ступеня охоплення ІТ прикладних завдань;
- класам реалізованих технологічних операцій;
- типу користувальницького інтерфейсу;
- варіантам використання мережі ЕОМ;
- предметної області, яка обслуговується.

По способі реалізації ІТ в ІС відзначають традиційно сформовані й нові інформаційні технології. Традиційні ІТ насамперед існували в умовах централізованої обробки даних

По ступені охоплення ІТ серед завдань керування виділяють електронну обробку даних, коли з використанням ЕОМ без перегляду методології й організації процесів керування ведеться обробка даних з рішенням окремих завдань.

По класах реалізованих технологічних операцій ІТ розглядаються, власно кажучи, в програмному аспекті й містять:

- текстову обробку,
- електронні таблиці,

- автоматизовані банки даних,
- обробку графічної й звукової інформації, мультимедійні й інші системи.

По типі користувальницького інтерфейсу можна розглядати ІТ з погляду можливостей доступу користувача до інформаційних й обчислювальних ресурсів.

Варіанти використання мережі ЕОМ. У цей час спостерігається тенденція до об'єднання різних типів інформаційних технологій у єдиний комп'ютерно - технологічний комплекс, що зветься інтегрованим.

Предметна технологія

Предметна технологія являє собою послідовність технологічних етапів по модифікації первинної інформації в результатну.

Наприклад, технологія бухгалтерського обліку припускає надходження первинної документації, що трансформується у форму бухгалтерської проводки. Остання, змінюючи стан аналітичного обліку, приводить до зміни рахунків синтетичного обліку, а далі й балансу.

Предметна технологія й інформаційна технологія впливають один на одного. Так, наприклад, наявність пластикових карток носія фінансової інформації принципово змінює предметну технологію, надаючи такі можливості, які без цього носія просто були відсутні.

Забезпечуючі й функціональні інформаційні технології

Забезпечуючі ІТ - технології обробки інформації, які можуть використовуватися як інструментарій у різних предметних областях для рішення різних завдань.

Функціональна ІТ являє собою таку модифікацію забезпечуючі ІТ, при яких реалізується яка-небудь із предметних технологій.

Наприклад, забезпечуючі технології - текстові й табличні процесори, а спеціальні функціональні технології - табличні процесори, СКБД, експертні системи.

Особливе місце займають мережні технології, які забезпечують взаємодію багатьох користувачів.

Поняття розподіленої інформаційної технології

Мережа - це сукупність програмних, технічних, комунікаційних засобів, які забезпечують ефективний розподіл обчислювальних ресурсів.

Однією з найважливіших мережних технологій є розподілена обробка даних. Персональні комп'ютери стоять на робочих місцях, тобто на місцях виникнення й використання інформації. Вони з'єднані каналами зв'язку. Це надало можливість розподілити їхні ресурси по окремих функціональних сферах діяльності й змінити технологію обробки даних у напрямку децентралізації.

У розподілених системах використовуються три інтегровані технології:

- технологія клієнт-сервер;
- технологія спільного використання ресурсів у рамках глобальних мереж;
- технологія універсального користувальницького спілкування у вигляді електронної пошти.

Основна ідея технології клієнт-сервер полягає в тому, щоб сервери розташувати на потужних машинах, а додатки клієнтів, які використовують мову запитів, - на менш потужних машинах. Таким чином будуть задіяні ресурси могутнішого сервера й менш потужних машин клієнтів. Ввід-вивід до бази заснова-

ний не на фізичному дробленні даних, а на логічному, тобто сервер відправляє клієнтам не повну копію бази, а тільки логічно необхідні порції. У технології клієнт-сервер програми клієнта і його запити зберігаються окремо від СКБД. Сервер обробляє запити клієнтів, вибирає необхідні дані із БД, посилає їх клієнтам по мережі, робить відновлення інформації, забезпечує цілісність і схоронність даних.

Розглянемо основні *види технології розподіленої обробки даних*.

1. Технологія клієнт-сервер, орієнтована на автономний комп'ютер, тобто й клієнт, і сервер розміщені на одній ЕОМ

2. Технологія клієнт-сервер, орієнтована на централізований розподіл. При використанні цієї технології клієнт одержує доступ до даних одиночного віддаленого сервера, дані можуть тільки зчитуватися.

3. Технологія клієнт-сервер, орієнтована на локальну обчислювальну мережу.

4. Технологія клієнт-сервер, орієнтована на зміни даних в одному місці. Вилучені сервери не зв'язані між собою мережею ЕОМ. Клієнт може змінювати дані тільки у своїй локальній базі.

5. Технологія клієнт-сервер, орієнтована на зміну даних у декількох місцях. На відміну від попередньої технології тут є сервер-координатор. Це створює передумови розробки розподіленої СКБД.

6. Технологія клієнт-сервер, орієнтована на розподілену СКБД. Вона забезпечує стратегію розбивки й дублювання, дозволяє одержати більш швидкий доступ до даних. Розподілена СКБД забезпечує незалежність клієнта від місця розміщення сервера, глобальну оптимізацію, розподілений контроль цілісності бази, розподілене адміністративне керування.

Об'єктно - орієнтовані інформаційні технології

Об'єктно - орієнтований підхід заснований на систематичному використанні моделей для мовно-незалежної розробки програмної системи, на основі її прагматики. Прагматика визначається метою розробки програмної системи: для керування роботою аеропорту, для обслуговування чемпіонату світу з футболу й т.п.. У формулюванні мети беруть участь предмети й поняття реального миру, які мають відношення до розроблювальної програмної системи.

При використанні об'єктно - орієнтованого підходу ці предмети й поняття замінюються їхніми моделями, тобто певними формальними конструкціями, які представляють їх у програмній системі.

1.3. Інформаційні технології кінцевого користувача **Комп'ютерні офісні технології**

База даних. Обов'язковим компонентом будь-якої технології є база даних. В автоматизованому офісі база даних концентрує в собі дані про виробничу систему фірми так само, як у технології обробки даних на операційному рівні.

Текстовий процесор. Це вид прикладного програмного забезпечення, призначений для створення й обробки текстових документів.

Табличний процесор. Він так само, як і текстовий процесор, є базовою складовою інформаційної культури будь-якого співробітника й автоматизованої офісної технології.

Електронна пошта. Електронна пошта (E-mail), ґрунтуючись на мережному використанні комп'ютерів, дає можливість користувачеві одержувати, зберігати й відправляти повідомлення своїм партнерам по мережі.

Аудіопошта. Це пошта для передачі повідомлень голосом.

Електронний календар. Він надає ще одну можливість використати мережний варіант комп'ютера для зберігання й маніпулювання робочим розкладом керівників й інших працівників організації.

Комп'ютерні конференції й телеконференції. Комп'ютерні конференції використовують комп'ютерні мережі для обміну інформацією між учасниками групи, які вирішують певну проблему. Телеконференція містить у собі три типи конференцій: аудіо, відео й комп'ютерну.

Відеотекст. Він заснований на використанні комп'ютера для одержання відображення текстових і графічних даних на екрані монітора.

Автоматизоване робоче місце - засіб автоматизації роботи кінцевого користувача

Автоматизоване робоче місце (АРМ) можна визначити як сукупність інформаційно-програмно-технічних ресурсів, які забезпечують кінцевому користувачеві обробку даних й автоматизацію управлінських функцій у конкретній предметній області.

Створення автоматизованих робочих місць припускає, що основні операції по нагромадженню, зберіганню й переробці інформації покладають на обчислювальну техніку, а користувач виконує частину ручних операцій й операцій, що вимагають творчого підходу при підготовці рішень.

АРМ мають проблемно-професійну орієнтацію на конкретну предметну область. Професійні АРМ є головним інструментом спілкування людини з обчислювальними системами, відіграючи роль автономних робочих місць, інтелектуальних терміналів великих ЕОМ, робочих станцій у локальних мережах.

Ефективним режимом роботи АРМ є його функціонування в межах локальної обчислювальної мережі як робоча станція. Особливо доцільний такий варіант, коли потрібно розподіляти інформаційно-обчислювальні ресурси між декількома користувачами.

1.4. Мережні інформаційні технології

Можливості мережі Інтернет

Інтернет являє собою глобальну комп'ютерну мережу, яка містить гігантський обсяг інформації по будь-якій тематиці, доступної для всіх бажаючих, і надає великий спектр інформаційних послуг. Кожна локальна мережа називається вузлом або сайтом, а юридична особа, що забезпечує роботу сайту - провайдером. Сайт складається з декількох комп'ютерів - серверів, кожний з яких призначений для зберігання інформації певного типу й у певному форматі. Кожен сайт і сервер мають унікальні імена.

Інтернет надає наступний інформаційний сервіс загального призначення:

1. Електронна пошта(E-mail) надає кожному абоненту електронну адресу, що є аналогом поштової адреси. За допомогою E-mail користувач може пересилати й одержувати повідомлення й файли довільного виду.

2. Доступ до інформаційних ресурсів. Є кілька видів інформаційних ресурсів Інтернету, які розрізняються характером інформації, способом її організації, методами роботи з нею. *В Інтернеті є наступні інформаційні системи:*

➤ **World Wide Web (WWW)** - всесвітня інформаційна павутина, у якій інформація складається зі сторінок. Інформація в WWW організована у формі гіпер-тексту. Це означає, що в документі існують спеціальні елементи - текст або малюнки, названі гіпертекстовими посиланнями, щиклик мишею на які виводить на екран інший документ, на який указує дане посилання.

➤ **FTP (File Transfer Program)** - система, яка існує для пересилання файлів. Файли стають доступними для роботи тільки після копіювання на власний комп'ютер.

➤ **Система телеконференції – Use Net** являє собою сукупність документів, згрупованих по певних темах.

3. IRC (Internet Relay Chat) - обмін інформацією в режимі реального часу. Цей режим нагадує селекторний зв'язок, те, що набирається користувачем, негайно відтворюється на екрані одного або відразу декількох абонентів.

Адресація й протоколи

Комп'ютер, підключений до Інтернету, використовує для зв'язку з іншими комп'ютерами мережі спеціальний протокол TCP/IP (Transfer Control Protocol/Internet Protocol), який називають хостом. Для ідентифікації кожного хосту в мережі є два способи адресації, які завжди діють спільно.

Перший спосіб адресації називається **IP-адресою**, це аналогічно телефонному номеру. IP-адреса призначається провайдером і складається із чотирьох груп цифр (чотирьох байтів), розділених крапками, закінчується крапкою і має, наприклад, вид: 123.45.67.91, де числа в кожній групі можуть приймати значення від 0 до 255.

Другий спосіб ідентифікації комп'ютерів називається **системою доменних імен або DNS (Domain Name Service)**. DNS - імена призначаються провайдером й, наприклад, мають такий вигляд - *win. smtp. dol. ru*. Наведене вище повне

доменне ім'я складається із чотирьох розділених точками простих доменів. Так, у наведеному прикладі DNS-імені *домени мають наступний сенс:*

- **ru.** - домен провайдера, у цьому випадку позначає всі хости в Росії.
- **dol** - домен провайдера, позначає комп'ютери локальної мережі російської фірми Demos.
- **smtп** - домен групи серверів Demos, який обслуговує систему електронної пошти.
- **win** - ім'я конкретного комп'ютера.

Таким чином, за своєю організацією й внутрішньою структурою DNS-імена нагадують повний шлях до конкретного файлу в дереві каталогів і файлів. Одне з розходжень полягає в тому, що домен більш високого рівня в DNS - імені перебуває праворуч.

Особливе значення мають імена доменів самого верхнього рівня, які стоять у повному імені праворуч. Вони зафіксовані міжнародною організацією Inter NIC і *будуються по регіональній й організаційній ознаці, наприклад:*

- **com** - комерційні;
- **edu** - освітні;
- **au** - Австралія;
- **ua** - Україна.

Домени можуть сполучати географічні й організаційні рівні, наприклад: **com.ua.** - сервери України.

Звичайно імена доменів мають 3 - 4 рівня вкладення.

Засіб організації інформації на конкретному хості й ідентифікація розміщеного на ньому певного інформаційного ресурсу здійснюються за допомогою системи адресації, названої URL (Universal Resource Locator). Наприклад, URL може мати вигляд: ***http://home.microsoft.com/inti/ru/www_tour.html***. *Елементи адреси означають:*

- **http://** - тип протоколу, указує, що адреса ставиться до хосту, який є WWW-сервером
- **home.microsoft.com** - доменне ім'я хосту;
- **/inti/ru/** - підкаталоги ru кореневого каталогу inti хоста;
- **www_tour.html** - ім'я файлу.

URL складається з латинських букв і деяких символів: тире «-», підкреслення «_», тильди «~» і не може містити пробілів.

Для ідентифікації адресата електронної пошти застосовується система E-mail-адрес, наприклад виду **vatbul@dol.ru**. У цій адресі **vatbul** - ім'я адресата, символ **@** - ознака кінця імені адресата, **dol.ru** - доменне ім'я провайдера, на одному з комп'ютерів якого буде зберігатися "поштова скринька" користувача.

Крім перерахованих загальносїтьових протоколів Інтернету для ряду інформаційних служб існують свої протоколи.

Особливості роботи зі службами Інтернету

World Wide Web

Служба World Wide Web (WWW) являє собою велику мережу серверів HTTP (HyperText Transfer Protocol - протокол передачі гіпертексту), що передають файли через Інтернет.

Відзначимо основні особливості WWW, які забезпечили її високу популярність:

- гіпертекстова організація інформаційних елементів - сторінок WWW. Це означає те, що сторінка WWW може містити спеціальні елементи - гіпертекстові посилання, щиглик на яких викликає перехід до сторінки, на яку вказує дане посилання;
- можливість включення в сторінки WWW сучасних мультимедійних засобів - графіків, звуку, анімації, а також інших засобів для художнього оформлення сторінок;
- наявність безкоштовного, гарного й досить простого програмного забезпечення;
- наявність гарних пошукових систем, які дозволяють досить швидко відшукувати необхідну інформацію;
- можливість швидкого переміщення вперед по вже переглянутих сторінках;
- наявність засобів забезпечення надійності й конфіденційності інформаційного обміну.

Робота з ресурсами WWW забезпечується спеціальними програмами, які називаються броузерами

Файлові бази даних. FTP (File Transfer Protocol) - це гігантське сховище файлів різного типу: текстових, електронних таблиць, програм, даних, графічних, звукових й інших, які зберігаються на FTP-серверах.

Електронна пошта. Електронна пошта (E-mail) забезпечує швидку передачу повідомлень і файлів конкретному адресатові або списку адресатів, і забезпечує доступ до інших ресурсів Інтернет. Для відправлення й одержання повідомлень по E-mail використовуються як поштові програми, так і програми, які поставляють разом із броузерами.

Конференції Usenet. Конференції організуються на спеціальному мережному сервері NNTP (Network News Transfer Protocol), за допомогою якого забезпечується можливість організації колективних дискусій по будь-якій тематичі для всіх передплатників послуг цього сервісу.

1.5. Інтеграція інформаційних технологій

Загальне поняття про систему штучного інтелекту

Крім інформаційно-обчислювальних завдань, рішення яких здійснюється за заданим алгоритмом, в останнє десятиліття комп'ютерні технології усе активніше намагаються застосувати для реалізації інтелектуальних процесів, тобто, процесів пошуку рішення, при яких кінцевий результат непередбачений і є плодом логічних самостійних висновків і висновків комп'ютера.

Програмні системи, які реалізують алгоритми, для яких не існує формальної моделі рішення, називають евристичними й відносять до класу інтелектуальних систем (або систем штучного інтелекту).

Принципова відмінність інтелектуальних систем від традиційних комп'ютерних програм полягає в наступному:

- До творчих, інтелектуальних завдань відносять завдання, для яких поки що не існує формальної моделі рішення, наприклад, гра в шахи.
- Завдання штучного інтелекту - це такі завдання, у яких формалізується не процес рішення, а процес пошуку рішення. У зв'язку із цим розвиваються такі напрямки, як експертні системи, база знань, нейронні мережі, нейрокомп'ютери, DM - системи.

Експертні системи

Найбільш широке застосування методи штучного інтелекту (ШІ) знайшли в програмах, названих експертними системами (ЕС).

Особливою рисою цих програм є здатність накопичувати знання й досвід кваліфікованих професіоналів (експертів) в будь-якій вузькій предметній області. Потім за допомогою знань, накопичених в ЕС, фахівці з не дуже високою кваліфікацією можуть вирішувати складні завдання на стільки ж високому рівні, як й експерти, іншими словами, дані програми підтримують засобами автоматизації не тільки й не стільки обчислювальні функції, скільки функції розумові, інтелектуальні, допомагаючи користувачеві приймати рішення в складних неоднозначних ситуаціях.

Всі експертні системи є системами штучного інтелекту, але не всі системи штучного інтелекту є ЕС. Наприклад, програма розрізнення друкованого тексту або голосових повідомлень ставиться до інтелектуальних систем, але не є експертною системою, оскільки вирішити таке завдання здатна будь-яка людина.

Для ЕС характерною рисою є наявність мети функціонування, яка складається в розв'язанні складних проблем, рішення яких під силу фахівцеві високої кваліфікації - експертіві.

Спеціалізовані продукційні системи знайшли широке використання в різних галузях людської діяльності. Однак найбільш популярними ЕС є оболонки Guru, KEE, LOOPS, які можуть бути наповнені будь-якими предметними знаннями.

Географічні інформаційні системи

Географічні інформаційні системи (ГІС) - це комп'ютерні системи, що дозволяють ефективно працювати із просторово розподіленою інформацією. Вони є закономірним розширенням концепції баз даних, доповнюючи їх наочністю подання й можливістю вирішувати завдання просторового аналізу.

Практично в будь-якій сфері діяльності ми зустрічаємося з інформацією такого роду, представленій у вигляді мап, планів, схем, діаграм й ін. Це може бути схема метро або план будинку, мапа екологічного моніторингу території або схема взаємозв'язків між офісами компанії, атлас земельного кадастру або карта природних ресурсів і багато чого іншого. ГІС надає можливість накопичувати й аналізувати подібну інформацію, оперативно знаходити потрібні відомості й відображувати їх у зручному для використання виді.

Працююча ГІС містить у собі п'ять ключових складових:

- апаратні засоби;
- програмне забезпечення;
- дані;
- виконавці;
- методи.

Апаратні засоби - це комп'ютер, на якому запущена ГІС.

Програмне забезпечення ГІС містить функції й інструменти, необхідні для зберігання, аналізу й візуалізації географічної (просторової) інформації. *Ключовими компонентами програмних продуктів є:*

- інструменти для уведення й оперування географічною інформацією;
- система керування базою даних (DBMS або СКБД);
- інструменти підтримки просторових запитів, аналізу й візуалізації (відображення);
- графічний користувальницький інтерфейс (GUI або ГІП) для легкого доступу до інструментів і функцій.

Дані. Це, мабуть, найбільш важливий компонент ГІС. У процесі керування просторовими даними ГІС інтегрує просторові дані з іншими типами й джерелами даних, а також може використати СКБД, застосовані багатьма організаціями для впорядкування й підтримки наявних в їхньому розпорядженні даних.

Виконавці. Користувачами ГІС можуть бути як технічні фахівці, які розробляють і підтримують систему, так і звичайні співробітники (кінцеві користувачі), яким ГІС допомагає вирішувати поточні щоденні справи й проблеми.

Методи. Успішність й ефективність застосування ГІС багато в чому залежить від правильно складеного плану й правил роботи, які складаються у відповідності зі специфікою завдань і роботи кожної організації.

Як працює ГІС

ГІС зберігає інформацію про реальний світ у вигляді набору тематичних шарів, які об'єднані на основі географічного положення. Цей простий, але дуже гнучкий підхід довів свою цінність при рішенні різноманітних реальних завдань: для відстеження пересування транспортних засобів і матеріалів, детального відображення реальної обстановки й планованих заходів, моделювання глобальної циркуляції атмосфери.

2. ОСНОВИ СИСТЕМНОГО ПІДХОДУ В ПРОЕКТУВАННІ.

2.1. Принципи системного підходу в проектуванні

Проектування складних об'єктів засновано на застосуванні ідей і принципів, викладених у ряді теорій і підходів. Найбільш загальним підходом є системний підхід, ідеями якого пронизані різні методики проектування складних систем.

Для фахівця в області системотехніки ідеї й принципи системного підходу є очевидними й природними, однак їхнє дотримання й реалізація найчастіше зв'язані з певними труднощами, які обумовлені особливостями проектування. Як і більшість дорослих освічених людей, які правильно використовують рідну мову без залучення правил граматики, інженери використовують системний підхід без звертання до допомоги системного аналізу. Однак інтуїтивний підхід без застосування правил системного аналізу може виявитися недостатнім для вирішення усе більше складних завдань інженерної діяльності.

Основний загальний принцип системного підходу полягає в розгляді частин досліджуваного явища або складної системи з урахуванням їх взаємодії. Системний підхід містить у собі виявлення структури системи, типізацію зв'язків, визначення атрибутів, аналіз впливу зовнішнього середовища, формування моделі системи, дослідження моделі й, можливо, оптимізацію її структури й функціонування.

Системний підхід є базою для узагальнюючої дисципліни "теорія систем" (інша назва - "системний аналіз").

Теорія систем — дисципліна, у якій конкретизуються положення системного підходу; вона присвячена дослідженню й проектуванню складних економічних, соціальних, технічних систем, найчастіше слабоструктурованих. Характерними прикладами таких систем є виробничі системи. При проектуванні систем мета досягається в багатокрокових процесах прийняття рішень. Методи прийняття рішень часто виділяють у самостійну дисципліну, названу "Теорія прийняття рішень".

В техніці дисципліну, у якій досліджуються складні технічні системи, їхнє проектування, і аналогічну теорії систем, частіше називають системотехнікою. Предметом системотехніки є, по-перше, організація процесу створення, використання й розвитку технічних систем, по-друге, методи й принципи їхнього проектування й дослідження. У системотехніці важливо вміти сформулювати мету системи й організувати її розгляд з позицій поставлених цілей. Тоді можна відкинути зайві й малозначимі частини при проектуванні й моделюванні, перейти до постановки оптимізаційних завдань.

Системи автоматизованого проектування й керування ставляться до числа найбільш складних сучасних штучних систем. Їхнє проектування й супровід неможливі без системного підходу. Тому ідеї й положення системотехніки входять складовою частиною в дисципліни, присвячені вивченню сучасних автоматизованих систем і технологій їхнього створення й застосування.

Інтерпретація й конкретизація системного підходу мають місце в ряді відомих підходів з іншими назвами, які також можна розглядати як компоненти системотехніки.

Такими є **структурний, блочно-ієрархічний, об'єктно-орієнтований підходи**.

При **структурному підході**, який є різновидом системного, потрібно синтезувати варіанти системи з компонентів (блоків) і оцінювати варіанти при їхньому частковому переборі з попереднім прогнозуванням характеристик компонентів.

Блочно-ієрархічний підхід до проектування використовує ідеї декомпозиції складних описів об'єктів і відповідно засобів їхнього створення на ієрархічні рівні й аспекти, вводить поняття стилю проектування, встановлює зв'язок між параметрами сусідніх ієрархічних рівнів.

Ряд важливих структурних принципів, використаних при розробці інформаційних систем і насамперед їхнього програмного забезпечення (ПЗ), виражений у підході, який називається **об'єктно-орієнтованим проектуванням** (ООП). Такий підхід має наступні *переваги в рішенні проблем керування складністю й інтеграції ПЗ*:

- вносить у моделі додатків більшу структурну визначеність, розподіляючи наведені в додатку дані й процедури між класами об'єктів;
- скорочує обсяг специфікацій завдяки введенню в описи ієрархії об'єктів і відносин спадкування між властивостями об'єктів різних рівнів ієрархії;
- зменшує ймовірність перекручування даних внаслідок помилкових дій за рахунок обмеження доступу до певних категорій даних в об'єктах.

Опис у кожному класі об'єктів припустимих звертань до них і прийнятих форматів повідомлень полегшує узгодження й інтеграцію ПЗ.

Для всіх підходів до проектування складних систем *характерні також наступні особливості*:

- **Структуризація процесу** проектування, яка виражена декомпозицією проектних завдань і документації, виділенням стадій, етапів, проектних процедур. Ця структуризація є сутністю блочно-ієрархічного підходу до проектування.
- **Ітераційний характер** проектування.
- **Типізація й уніфікація** проектних рішень і засобів проектування.

У теорії систем і системотехніці уведений ряд термінів, серед них до базових потрібно віднести наступні поняття:

- **Система** — безліч елементів, що перебувають у відносинах і зв'язках між собою.
- **Елемент** — така частина системи, яку недоцільно піддавати при проектуванні подальшому членуванню.
- **Складна система** — система, яка характеризується великим числом елементів й, що найбільш важливо, великою кількістю взаємозв'язків елементів. Складність системи визначається також видом взаємозв'язків елементів, властивостями цілеспрямованості, цілісності, ієрархічності, багатоаспектності. Зрозуміло, що сучасні автоматизовані інформаційні системи й,

зокрема, системи автоматизованого проектування, є складними завдяки наявності в них перерахованих властивостей й ознак.

- **Підсистема** — частина системи (підмножина елементів й їхніх взаємозв'язків), що має властивості системи.
- **Надсистема** — система, стосовно якої розглянута система є підсистемою.
- **Структура** — відображення сукупності елементів системи і їхніх взаємозв'язків; поняття структури відрізняється від поняття самої системи також тим, що при описі структури беруть до уваги лише типи елементів і зв'язків без конкретизації значень їхніх параметрів.
- **Параметр** — величина, що виражає властивість або системи, або її частини, або середовища, які впливають на систему. Звичайно в моделях систем як параметри розглядають величини, що не змінюються в процесі дослідження системи. Параметри підрозділяють на зовнішні, внутрішні й вихідні, що виражають властивості елементів системи, самої системи, зовнішнього середовища відповідно. Вектори внутрішніх параметрів, вихідних параметрів і зовнішніх параметрів позначаються відповідно.
- **Фазова змінна** — величина, що характеризує енергетичне або інформаційне наповнення елемента або підсистеми.
- **Стан** — сукупність значень фазових змінних, зафіксованих в одній тимчасовій точці процесу функціонування.
- **Поводження** (динаміка) системи - зміна стану системи в процесі функціонування.
- **Система без післядії** — її поведження визначається завданням стану в цей момент і вектором зовнішніх впливів. У системах з післядією, крім того, потрібно знати передісторію поведження, тобто стану системи, в попередні миті.
- **Вектор змінних**, який характеризує стан (вектор змінних станів) — не надлишкова безліч фазових змінних, завдання значень яких у деякий момент часу повністю визначає поведження системи надалі (в автономних системах без післядії).
- **Простір станів** - безліч можливих значень вектора змінних станів.
- **Фазова траєкторія** — подання процесу (залежності) у вигляді послідовності крапок у просторі станів.

До характеристик складних систем часто відносять наступні поняття:

- **Цілеспрямованість** - властивість штучної системи, яка виражає призначення системи. Ця властивість необхідна для оцінки ефективності варіантів системи.
- **Цілісність** - властивість системи, яка характеризує взаємозв'язок елементів і наявність залежності вихідних параметрів від параметрів елементів, при цьому більшість вихідних параметрів не є простим повторенням або сумою параметрів елементів.
- **Ієрархічність** - властивість складної системи, що виражає можливість і доцільність її ієрархічного опису, тобто подання у вигляді декількох рівнів, між компонентами яких є відносини «ціле – частина».

Складовими частинами системотехніки є наступні основні розділи:

- ієрархічна структура систем, організація їхнього проектування;
- аналіз і моделювання систем;
- синтез й оптимізація систем.

Моделювання має два чітко помітних завдання:

1. створення моделей складних систем (в англomовному написанні - modeling);
2. аналіз властивостей систем на основі дослідження їхніх моделей (simulation).

Синтез також підрозділяють на два завдання:

1. синтез структури проєктованих систем (структурний синтез);
2. вибір чисельних значень параметрів елементів систем (параметричний синтез).

Ці завдання ставляться до області прийняття проектних рішень.

Моделювання й оптимізацію бажано виконувати з урахуванням статистичної природи систем. **Детермінованість** — лише окремий випадок. При проєктуванні характерна недостача достовірних вихідних даних, невизначеність умов прийняття рішень. Облік статистичного характеру даних при моделюванні значною мірою заснований на методі статистичних випробувань (методі Монте-Карло), а прийняття рішень - на використанні нечітких множин, експертних систем, еволюційних обчислень.

Приклад 1

Комп'ютер є складною системою завдяки наявності в ньому елементів, різноманітних зв'язків між елементами й підсистемами, властивостей цілеспрямованості, цілісності, ієрархічності. До підсистем комп'ютера належать: процесор (процесори), оперативна пам'ять, кеш-пам'ять, шини, пристрої вводу-виводу.

В якості надсистеми можуть виступати обчислювальна мережа, автоматизована й (або) організаційна система, до яких належить комп'ютер. Внутрішні параметри — часи виконання арифметичних операцій, читання (запису) у накопичувачах, пропускна здатність шин й ін. Вихідні параметри — продуктивність комп'ютера, ємність оперативної й зовнішньої пам'яті, собівартість, час наробітку на відмову й ін. Зовнішні параметри — напруга живлення мережі і його стабільність, температура навколишнього середовища й ін.

Приклад 2

Для двигуна внутрішнього згорання підсистемами є колінчатий вал, механізм газорозподілу, поршнева група, система змащення й охолодження. Внутрішні параметри - число циліндрів, обсяг камери згорання й ін. Вихідні параметри - потужність двигуна, КПД, витрата палива й ін. Зовнішні параметри - характеристики палива, температура повітря, навантаження на вихідному валу.

2.2. Рівні проектування

При використанні блочно-ієрархічного підходу до проектування подання проєктованої системи розподіляють на ієрархічні рівні. На верхньому рівні використовують найменш деталізоване подання, яке відбиває тільки самі загальні риси й особливості проєктованої системи. На наступних рівнях ступінь деталізації опису зростає, при цьому розглядають уже окремі блоки системи, але з урахуванням впливів його сусідів на кожний з них. Такий підхід дозволяє на кожному ієрархічному рівні формулювати завдання прийнятної складності, які

можна вирішити за допомогою наявних засобів проектування. Розбивка на рівні повинна бути такою, щоб документація на блок будь-якого рівня була доступна для огляду й сприймана однією людиною.

Інакше кажучи, блочно-ієрархічний підхід є декомпозиційним (його можна назвати також діакоптическим), який заснований на розбивці складного завдання великої розмірності на послідовно й (або) паралельно розв'язувані групи завдань малої розмірності, що істотно скорочує вимоги до використовуваних обчислювальних ресурсів або час вирішення завдань.

Можна говорити не тільки про ієрархічні рівні специфікацій, але й про **ієрархічні рівні проектування**, розуміючи під кожним з них сукупність специфікацій деякого ієрархічного рівня разом з постановками завдань, методами одержання описів і рішення виникаючих проектних завдань.

Список ієрархічних рівнів у кожному додатку може бути специфічним, але для більшості додатків характерно наступне *найбільш велике виділення рівнів*:

- **системний рівень**, на якому вирішують найбільш загальні завдання проектування систем, машин і процесів; результати проектування представляють у вигляді структурних схем, генеральних планів, схем розміщення встаткування, діаграм потоків даних і т.п.;
- **макрорівень**, на якому проектують окремі пристрої, вузли машин і приладів; результати надають у вигляді функціональних, принципівих і кінематичних схем, складальних креслень і т.п.;
- **мікрорівень**, на якому проектують окремі деталі й елементи машин і приладів.

У кожному додатку число виділених рівнів й їхніх найменувань можуть бути різними. Так, у радіоелектроніці мікрорівень часто називають компонентним, макрорівень — **схемотехнічним рівнем**. Між схемотехнічним і системним рівнями вводять рівень, який називається **функціонально-логічним рівнем**. В обчислювальній техніці системний рівень підрозділяють на рівні проектування ЕОМ (обчислювальних систем) і обчислювальних мереж. У машинобудуванні є рівні деталей, вузлів, машин, комплексів.

Залежно від послідовності рішення завдань ієрархічних рівнів розрізняють **спадне, висхідне й змішане проектування (стилі проектування)**.

Послідовність вирішення завдань від нижніх рівнів до верхнього характеризує **висхідне проектування**, зворотна послідовність приводить до **спадного проектування**, у змішаному стилі є елементи як висхідного, так і спадного проектування. У більшості випадків для складних систем віддають перевагу спадному проектуванню. Відзначимо однак, що при наявності заздалегідь спроектованих складених блоків (пристроїв) можна говорити про змішане проектування.

Невизначеність і нечіткість вихідних даних при спадному проектуванні (тому, що ще не спроектовані компоненти) або вихідних вимог при висхідному проектуванні (оскільки ТЗ є на всю систему, а не на її частині) обумовлюють необхідність прогнозування відсутніх даних з наступним їхнім уточненням, тобто послідовного наближення до остаточного рішення (ітераційність проектування).

Поряд з декомпозицією описів на ієрархічні рівні застосовують поділ подань про спроектовані об'єкти на аспекти.

Аспект опису (страта) — опис системи або її частин з деякої обговореної точки зору, обумовленої функціональним, фізичними або іншим типом відносин між властивостями й елементами.

Розрізняють аспекти функціональний, інформаційний, структурний і поведінковий (процесний). Функціональний опис відносять до функцій системи й найчастіше представляють його функціональними схемами.

Одержання функціональних описів часто називають **функціональним проектуванням**.

Інформаційний опис містить у собі основні поняття предметної області (сутності), словесне пояснення або числові значення характеристик (атрибутів) використаних об'єктів, а також опис зв'язків між цими поняттями й характеристиками.

Інформаційні моделі можна представляти графічно (графи, діаграми сутність-відношення), у вигляді таблиць або списків. Одержання інформаційних описів часто називають інформаційним проектуванням або стосовно до створення баз даних - інфологічним проектуванням.

Структурний опис належить до морфології системи, характеризує складові частини системи і їх міжз'єднання й може бути представлено структурними схемами, а також різного роду конструкторською документацією. Одержання конструкторської документації, тобто опис геометричних форм виробів, складу компонентів й їхнього просторового розміщення, називають **конструкторським проектуванням**.

Поведінковий опис характеризує процеси функціонування (алгоритми) системи й (або) технологічні процеси створення системи. Розробка алгоритмів і програмного забезпечення систем є предметом **алгоритмічного проектування**, а розробка технологічних процесів виготовлення виробів — предметом **технологічного проектування**.

Іноді аспекти описів зв'язують із **підсистемами**, функціонування яких засновано на різних фізичних процесах.

Відзначимо, що в загальному випадку виділення страт може бути неоднозначним. Так, крім зазначеного підходу, очевидна доцільність виділення таких аспектів, як функціональне (розробка принципів дії, структурних, функціональних, принципових схем), конструкторське (визначення форм і просторового розташування компонентів виробів), алгоритмічне (розробка алгоритмів і програмного забезпечення) і технологічне (розробка технологічних процесів) проектування систем. Прикладами страт у випадку **САПР** можуть служити також розглянуті далі види забезпечення автоматизованого проектування.

2.2.1.. Стадії проектування

Стадії проектування — найбільш великі частини проектування, як процесу, що розвивається в часі.

У загальному випадку виділяють стадії *науково-дослідних робіт* (НДР), ескізного проекту або дослідно-конструкторських робіт (ДКР), технічного, робочого проектів, випробувань пробних зразків або пробних партій. Стадію НДР іноді називають передпроектними дослідженнями або стадією технічної пропозиції. Зрозуміло, що під час переходу від стадії до стадії ступінь подробиці й старанність пророблення проекту зростають, і робочий проект уже повинен бу-

ти цілком достатнім для виготовлення пробних або серійних зразків. Близьким до визначення стадії, але менш чітко обговореним поняттям, є **поняття етапу проектування**. Проектування на початкових етапах, у процесі якого приймаються принципові проектні рішення по вигляду й принципам дії проєктованих пристроїв і систем, називають **концептуальним проектуванням**.

Стадії (етапи) проектування підрозділяють на складові частини, які називають **проектними процедурами**. Прикладами проектних процедур може виступати підготовка деталіровочних креслень, аналіз кінематики, моделювання перехідного процесу, оптимізація параметрів й інші проектні завдання.

У свою чергу, проектні процедури можна розбити на більш дрібні компоненти, названі **проектними операціями**, наприклад, при аналізі міцності деталі сітковими методами операціями можуть бути побудова сітки, вибір або розрахунок зовнішніх впливів, власно моделювання полів напруг і деформацій, подання результатів моделювання в графічній і текстовій формах. Проектування зводиться до виконання деяких послідовностей проектних процедур — **маршрутів проектування**.

Прагнення скоротити тимчасові витрати на проектування призвело до розробки методик **паралельного проектування**, при якому паралельно в часі вирішуються завдання, зв'язані один з одним по вхідним і вихідним даним таким чином, що для рішення однієї з них потрібне знання результатів рішення іншого завдання. Оскільки ці результати до початку процедури паралельного проектування ще не отримані, у методиці паралельного проектування повинні бути зазначені способи завдання ще не певних значень параметрів. Прикладом паралельного проектування можуть служити паралельна розробка апаратних і програмних засобів обчислювальних систем або одночасна розробка літака й засобів його аеродромного обслуговування.

Іноді розробку технічного завдання на проектування називають **зовнішнім проектуванням**, а реалізацію ТЗ — **внутрішнім проектуванням**.

У ТЗ на проектування об'єкта вказують, принаймні, **що впливають дані**:

1. Призначення об'єкта;
2. Умови експлуатації. Поряд з якісними характеристиками (представленими у вербальній формі) є числові параметри, названі **зовнішніми параметрами**, для яких зазначені області припустимих значень. Приклади зовнішніх параметрів: температура навколишнього середовища, зовнішні сили, електричні напруги, навантаження й т.п.;
3. Вимоги до **вихідних параметрів**, тобто до величин, що характеризують властивості об'єкта, що цікавлять споживача. Ці вимоги виражені у вигляді **умов працездатності**.

Приклади умов працездатності:

- витрата палива на 100 км пробігу автомобіля < 8 л;
- коефіцієнт підсилення підсилювача на середніх частотах > 300 ; швидкодія процесора $> .$

3. ОСОБЛИВОСТІ Й ПРИНЦИПИ ПОБУДОВИ САПР (Автоматизація розрахунково-конструкторських робіт.)

Вирішення проблем **автоматизації завдань проектування** об'єктів за допомогою ІТ ґрунтується на системному підході, тобто на створенні й впровадженні САПР — систем автоматизованого проектування технічних об'єктів, які вирішують весь комплекс завдань від аналізу завдання до розробки повного обсягу конструкторської й технологічної документації. САПР являє собою великі організаційно-технічні системи, що складаються з комплексу засобів автоматизації проектування, взаємозалежного з підрозділами конкретної проектної організації.

3.1. Мета створення САПР

Ціль автоматизації - підвищити якість проектування, зменшити матеріальні витрати на нього, скоротити строки проектування й ліквідувати зріст кількості інженерно-технічних працівників, зайнятих проектуванням і конструюванням.

Науково обґрунтований розподіл функцій між людиною й ЕОМ має на увазі, що людина повинен вирішувати завдання, що носять творчий характер, а ЕОМ - завдання, рішення яких підлягає алгоритмізації.

Істотною відмінністю автоматизованого проектування від неавтоматизованого є можливість заміни фізичного моделювання, яке коштовне й займає багато часу, математичним моделюванням. При цьому слід мати на увазі одну найважливішу обставину: при проектуванні кількість варіантів незора. Тому не можна ставити завдання створення універсальної САПР.

Для створення САПР необхідно:

- удосконалити проектування на основі застосування математичних методів і засобів обчислювальної техніки;
- автоматизувати процеси пошуку, обробки й видачі інформації;
- використати методи оптимального й варіантного проектування; застосувати ефективні, які відбивають істотні особливості, математичні моделі проєктованих об'єктів;
- створювати банки даних, які містять систематизовані відомості довідкового характеру, необхідні для автоматизованого проектування об'єктів;
- підвищити якість оформлення проектної документації;
- підвищити творчу частку праці проєктувальників за рахунок автоматизації нетворчих робіт;
- уніфікувати й стандартизувати методи проектування;
- підготувати й перепідготувати фахівців;
- реалізувати взаємодію з автоматизованими системами різного рівня й призначення.

3.2. Склад САПР

САПР - система, що поєднує технічні засоби, математичне й програмне забезпечення, параметри й характеристики яких обирають із максимальним урахуванням особливостей завдань інженерного проектування й конструювання. У САПР забезпечується зручність використання програм за рахунок застосування спеціальних проблемно-орієнтованих мов і наявності інформаційно-довідкової бази.

Структурними складовими САПР є підсистеми, що володіють всіма власти-

востями систем і діють як самостійні системи. Це виділені по деяких ознаках **частини САПР**, які забезпечують виконання деяких закінчених проектних завдань із одержанням відповідних проектних рішень і проектних документів.

По призначенню підсистеми САПР розподіляють на два види: ті, що проектують й ті, що обслуговують.

*До тих систем, які **проектують**, відносяться підсистеми, які виконують проектні процедури й операції, наприклад:*

- підсистема компонування виробів;
- підсистема проектування складальних одиниць;
- підсистема проектування деталей;
- підсистема проектування схеми керування;
- підсистема технологічного проектування.

*До **обслуговуючих** відносяться підсистеми, які призначені для підтримки працездатності підсистем, які проектують, наприклад:*

- підсистема графічного відображення об'єктів проектування;
- підсистема документування;
- підсистема інформаційного пошуку й ін.

Залежно від відношення до об'єкта проектування розрізняють два види підсистем, які проектують:

- об'єктно-орієнтовані (об'єктні);
- об'єктно-незалежні (інваріантні).

*До **об'єктних підсистем** відносять підсистеми, які виконують одну або кілька проектних процедур або операцій, безпосередньо залежних від конкретного об'єкта проектування, наприклад:*

- підсистема проектування технологічних систем;
- підсистема моделювання динаміки, проектованої конструкції й ін.

*До **інваріантних підсистем** відносять підсистеми, які виконують уніфіковані проектні процедури й операції, наприклад:*

- підсистема розрахунків деталей виробів, машин;
- підсистема розрахунків режимів різання;
- підсистема розрахунку техніко-економічних показників й ін.

Процес проектування реалізується в підсистемах у вигляді певної послідовності проектних процедур й операцій. Проектна процедура відповідає частині проектної підсистеми, в результаті виконання якої приймається деяке проектне рішення. Вона складається з елементарних проектних операцій, має твердо встановлений порядок їхнього виконання й спрямована на досягнення локальної мети в процесі проектування. Під проектною операцією розуміють умовно виділену частину проектної процедури або елементарну дію, яку виконує конструктор у процесі проектування. Прикладами проектних процедур можуть виступати процедури розробки кінематичної або компонуєчої схеми верстата, технології обробки виробів і т.п., а прикладами проектних операцій - розрахунок припусків, рішення будь-якого рівняння й т.п.

Структурна єдність підсистем САПР забезпечується строгою регламентацією зв'язків між різними видами забезпечення, об'єднаних загальною для даної підсистеми цільовою функцією.

Розрізняють наступні види забезпечення:

- методичне забезпечення - документи, у яких відбиті склад, правила відбору й експлуатації засобів автоматизації проектування;
- лінгвістичне забезпечення - мови проектування, термінологія;
- математичне забезпечення - методи, математичні моделі, алгоритми;
- програмне забезпечення - документи з текстами програм, програми на машинних носіях й експлуатаційні документи;
- технічне забезпечення - пристрої обчислювальної й організаційної техніки, засоби передачі даних, вимірвальні й інші пристрої і їхні сполучення;
- інформаційне забезпечення - документи, які містять опис стандартних проектних процедур, типових проектних рішень, типових елементів, які комплектують вироби, матеріали й інші дані;
- організаційне забезпечення - положення й інструкції, накази, штатний розклад й інші документи, що регламентують організаційну структуру підрозділів й їхню взаємодію з комплексом засобів автоматизації проектування.

3.3. Основні принципи побудови САПР

Розробка САПР є великою науково-технічною проблемою, а її впровадження вимагає значних фінансових витрат. Накопичений досвід дозволяє виділити наступні *основні принципи побудови САПР*.

1. САПР — людино-машинна система. Всі створені системи проектування за допомогою ЕОМ є автоматизованими, важливу роль у них відіграє людина - інженер, що розробляє проект технічного засобу.

Людина в САПР повинна вирішувати, по-перше, всі завдання, які не формалізовані, по-друге, завдання, рішення яких людина здійснює на основі своїх евристичних здатностей більш ефективно, ніж сучасна ЕОМ на основі своїх обчислювальних можливостей. Чітка взаємодія людини й ЕОМ у процесі проектування - один із принципів побудови й експлуатації САПР.

2. САПР — ієрархічна система, що реалізує комплексний підхід до автоматизації всіх рівнів проектування. Ієрархія рівнів проектування відбивається в структурі спеціального програмного забезпечення САПР у вигляді ієрархії підсистем.

Слід особливо підкреслити доцільність забезпечення комплексного характеру САПР, тому що автоматизація проектування лише на одному з рівнів виявляється значно менш ефективною, чим повна автоматизація всіх рівнів. **Ієрархічна побудова** повинна бути не тільки у спеціальному програмному забезпеченні, але й в технічних засобів САПР, поділюваних на центральний обчислювальний комплекс й автоматизовані робочі місця проектувальників.

3. САПР — сукупність інформаційно-узгоджених підсистем. Цей дуже важливий принцип повинен відноситись не тільки до зв'язків між великими підсистемами, але й до зв'язків між більш дрібними частинами підсистем. Інформаційна узгодженість означає, що всі, або більшість можливих послідовностей завдань проектування обслуговуються інформаційно-погодженими програмами. Дві програми є інформаційно-погодженими, якщо всі ті дані, які являють собою об'єкт переробки в обох програмах, входять у числові масиви, не потребуючи змін при переході від однієї програми до іншої. Так, інформаційні зв'яз-

ки можуть проявлятися в тому, що результати рішення одного завдання будуть вхідними даними для іншого завдання. Якщо для узгодження програм потрібна істотна переробка загального масиву за участю людини, яка буде додавати відсутні параметри, вручну перекомпоновувати масив або змінювати числові значення окремих параметрів, то програми є інформаційно не погодженими. Ручне перекомпоновування масиву призводить до істотних тимчасових затримок, збільшення кількості помилок і тому зменшує попит на послуги САПР. Інформаційна непогодженість перетворює САПР на сукупність автономних програм, при цьому через неврахування в підсистемах багатьох факторів, оцінюваних в інших підсистемах, знижується якість проектних рішень.

4. САПР — відкрита система, яка розвивається. Існує, принаймні, дві вагомі причини, по яких САПР повинна бути системою, що змінюється в часі. По-перше, розробка настільки складного об'єкта, як САПР, забирає тривалий час, і економічно вигідно вводити в експлуатацію частини системи по ступені їхньої готовності. Уведений в експлуатацію базовий варіант системи надалі розширюється. По-друге, постійний прогрес техніки, проєктованих об'єктів, обчислювальної техніки й обчислювальної математики призводить до появи нових, удосконалених математичних моделей і програм, які повинні замінити старі, менш вдалі аналоги. Тому САПР повинна бути **відкритою системою**, тобто мати властивість зручного використання нових методів і засобів.

5. САПР — спеціалізована система з максимальним використанням уніфікованих модулів. Вимоги високої ефективності й універсальності, як правило, суперечливі. Стосовно до САПР це положення зберігає свою силу. Високої ефективності САПР, що виражається насамперед малими часовими й матеріальними витратами при рішенні проектних завдань, домагаються за рахунок спеціалізації систем. Зрозуміло, що при цьому збільшується кількість різних САПР. Щоб знизити витрати на розробку багатьох спеціалізованих САПР, доцільно будувати їх на основі максимального використання уніфікованих складових частин. Необхідною умовою уніфікації є пошук загальних рис і положень у моделюванні, аналізі й синтезі різнорідних технічних об'єктів. Безумовно, може бути сформульований і ряд інших принципів, що підкреслює багатосторонність і складність проблеми САПР.

3.4. Стадії створення САПР

Створення й розвиток САПР здійснюється самою проектною організацією із залученням (при необхідності) інших організацій-співвиконавців, у тому числі науково-дослідних інститутів і вищих навчальних закладів. Варто підкреслити, що створення САПР - складна й трудомістка робота, виконання якої під силу тільки великому висококваліфікованому колективу розробників.

*Процес створення САПР містить у собі вісім стадій: передпроектних досліджень: технічне завдання, технічна пропозиція, ескізний проєкт, технічний проєкт, робочий проєкт, виготовлення, налагодження й випробування, запровадження в дію. **Див. попередню Лекцію.***

Передпроектні дослідження проводяться для виявлення готовності конкретної проектною організацією до впровадження автоматизованих методів. Основу

цієї роботи становить **системне обстеження** об'єкта проектування й використаних в інженерній практиці **традиційних методів і прийомів** процесу проектування, а також обсягу технічної документації, яка розробляється в процесі проектування. Процес обстеження здійснюється головним чином опитуванням досвідчених проектувальників і конструкторів.

У результаті обстеження визначається необхідність й економічна ефективність створення автоматизованої системи. При цьому враховується обсяг проектно-конструкторських робіт, їхня періодичність, загальні витрати інженерної праці, можливість створення адекватного математичного опису й оптимізаційних процедур, необхідність підвищення якісних показників проектного виробу, скорочення строків проектування.

Технічне завдання (ТЗ) є вихідним документом для створення САПР і повинне містити найбільш повні **вихідні дані й вимоги**. Цей документ розробляє головний розроблювач системи. ТЗ на створення САПР повинне містити наступні основні розділи:

- «Найменування й область застосування», де вказують повне найменування системи й коротку характеристику області її застосування;
- «Підстава для створення», де вказують найменування директивних документів, на підставі яких створюється САПР;
- «Характеристика об'єктів проектування», де приводять відомості про призначення, склад, умови застосування об'єктів проектування;
- «Мета й призначення», де надається мета створення САПР, її призначення й критерій ефективності її функціонування;
- «Характеристика процесу проектування», де приводять загальний опис **процесу проектування**, вимоги до вхідної й вихідної інформації, а також вимоги до поділу проектних процедур (операцій), виконаних за допомогою неавтоматизованого й автоматизованого проектування;
- «Вимоги до САПР», де перелічують вимоги до САПР у цілому й до складу її підсистем, до застосування в складі САПР раніше створених підсистем і компонентів і т.п.;
- «Техніко-економічні показники», де оцінюють витрати на створення САПР, указують джерела одержання економії й очікувану ефективність від застосування САПР.

На стадіях технічної пропозиції, ескізного й робочого проектування вибираються й обґрунтовуються варіанти САПР, розробляються остаточні рішення. При цьому виконують наступні основні види робіт:

- виявлення процесу проектування (його алгоритм), тобто прийняття основних технічних рішень;
- розробка структури САПР й її взаємозв'язків з іншими системами (визначення складу проектних процедур й операції по підсистемах; уточнення складу підсистем і взаємозв'язків між ними; розробка схеми функціонування САПР в цілому);
- визначення складу методів, математичних моделей для проектних операцій і процедур; складу мов проектування; складу інформації (обсяг, способи її організації й види машинних носіїв інформації); складу загального, спеціа-

- лізованого загального й спеціального програмного забезпечення;
- формування складу технічних засобів (ЕОМ периферійні пристрої й інші елементи);
- прийняття рішень по математичному, інформаційному, програмному й технічному видах забезпечення по САПР у цілому й окремо по підсистемах;
- розрахунок техніко-економічних показників САПР.

Оформлення всієї документації, необхідної для створення й функціонування САПР, виконують на стадії робочого проектування.

На стадії виготовлення, налагодження й випробування роблять монтаж, налагодження й випробування комплексу технічних засобів автоматизації проектування, на тестових прикладах доводять програмне забезпечення й підготовлюють проектну організацію до запровадження в дію САПР.

Запровадження в дію системи здійснюють після пробного функціонування й приймальних випробувань у замовника.

3.5. Відображення процесу проектування в програмне забезпечення САПР

Найважливішим питанням при створенні САПР після формалізації процесу проектування є питання відображення проектно-конструкторської діяльності інженера в програмному забезпеченні.

Взагалі процес проектування в САПР можна спрощено представити схемою, показаної на рис. 1. Ця схема відображує елементарний осередок проектно-конструкторського процесу, з ланцюжка яких складається реальний автоматизований процес. Всі системи проектування, створені за допомогою сучасних засобів обчислювальної техніки, є автоматизованими. Найважливішу роль у цих системах грає людина-інженер, яка розробляє проект нових технічних засобів. Людина в САПР вирішує всі неформалізовані проектні завдання й завдання планування робіт. Сучасна САПР є інструментом висококваліфікованого інженера-проектувальника, тому тісна взаємодія людини й ЕОМ у процесі проектування — один з найважливіших принципів побудови й експлуатації САПР.

Основним блоком у схемі процесу автоматизованого проектування (рис. 1) є **блок проектних рішень**. Залежно від повноти формалізації наших знань у конкретній предметній області проектне рішення може бути виконане автоматично або в інтерактивному режимі. На основі вхідних даних й обмежень (незалежні параметри проектування) блок змінює параметри (фактори рішення) до одержання гідних проектних рішень (залежних змінних).

Результати проектування повинні бути представлені у вигляді, зручному для сприйняття людиною, і містити інформацію, на основі якої інженер міг би висунути судження про результати проектування.

Якщо проектне рішення затверджується, то оформлюється необхідна вихідна документація; якщо необхідно корегування проекту, інженер, уточнюючи змінювані параметри, в інтерактивному режимі домагається потрібних результатів; коли ж проектно-конструкторський процес не призводить до бажаної мети, необхідно уточнити вхідні дані й обмеження



Рис. 1 – Схема процесу автоматизованого проектування

.Розгляд навіть такої спрощеної схеми процесу проектування дозволяє уточнити поділ функції між інженером й ЕОМ у САПР. Одержання варіантів проектних рішень й їхнє подання у вигляді, зручному для сприйняття людиною, **може бути покладене на ЕОМ у тій мері, у якій це дозволить зробити математичне забезпечення проектних процедур**. Але навіть при автоматичному одержанні варіантів проектних рішень за інженером залишаються найважливіші функції — **уведення вхідних даних** для проектування, **остаточна оцінка й затвердження проектних рішень**. В інтерактивному ж режимі проектування інженер безпосередньо бере участь в ході рішення завдань, впливаючи на вибір факторів рішення й уточнюючи незалежні змінні. Одержання вихідної документації відповідно до існуючих вимог є операцією рутинною й повинне виконуватися автоматично.

На підставі викладеного **модель програмного забезпечення** автоматизованої проектної процедури можна представити схемою, показаної на рис. 2.

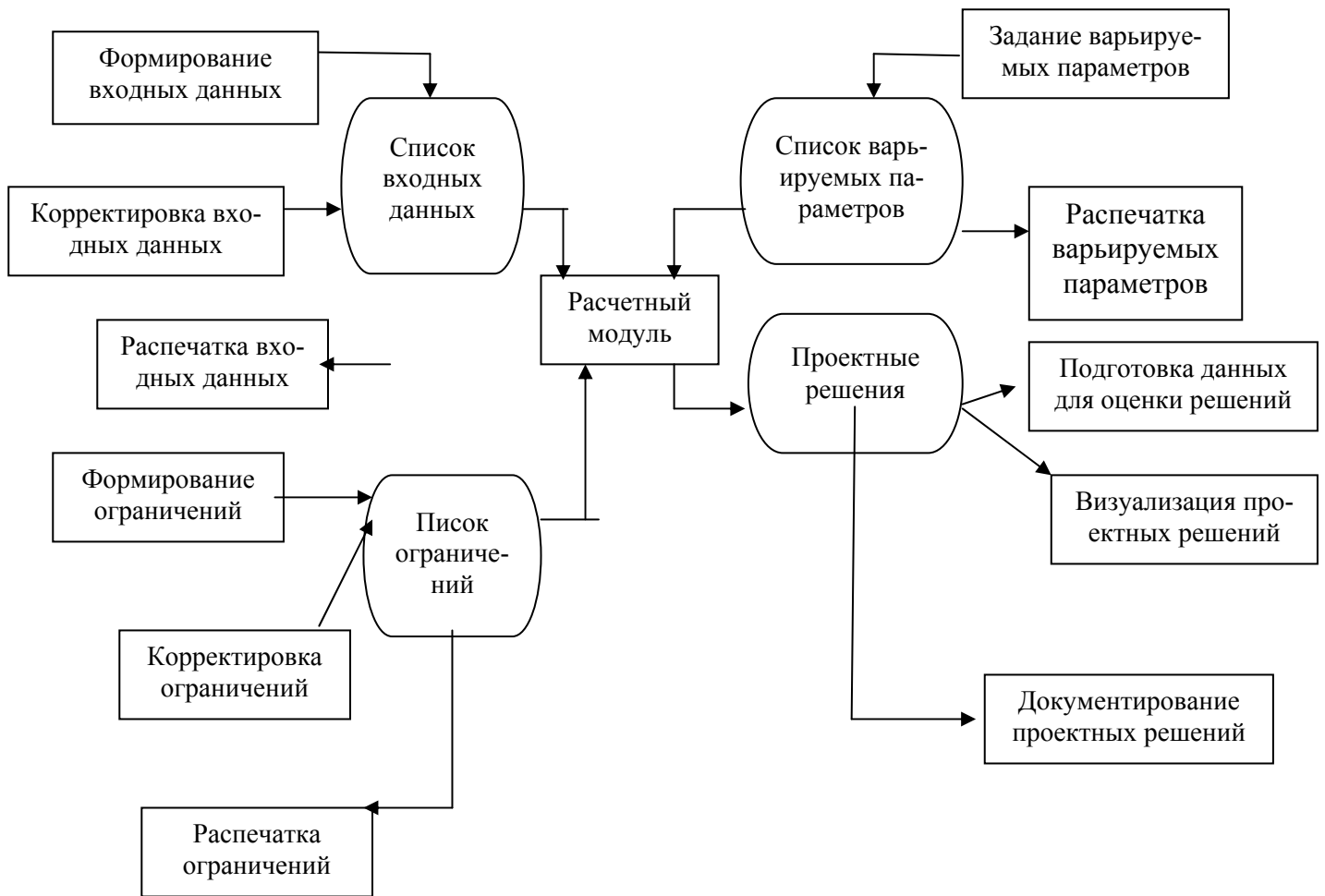


Рис. 2 – Модель программного забезпечення проектної процедури в САПР

Узагальнена модель програмного забезпечення проектної процедури в САПР має ряд складових і списки даних. В загальному вигляді кожна складова повинна реалізуватися своїм програмним модулем.

Призначення модуля формування вхідних даних складається в створенні списку цих даних для проектування та його контролю при уведенні в систему. Структура й формат списку вхідних даних залежать від змісту проектної процедури (розрахункового модуля). Необхідно передбачити існування декількох версій списку вхідних даних, які із заданими іменами зберігаються на ділянках магнітного диска. Структура списку даних визначається розроблювачем САПР, а формується він або в діалоговому режимі користувачем, або генерується автоматично попередніми проектними процедурами.

Програмний модуль корегування вхідних даних передбачає редагування (видалення, вставку й т.п.) списку, потреба в якому виникає через помилки користувача при уведенні даних, які виявляють при контролі, а також при необхідності їхнього уточнення в результаті аналізу й оцінки проектних рішень.

Створення й контроль списку змінюваних параметрів здійснюються програмними модулями їхнього завдання й роздруківки.

Розрахунковий модуль програмного забезпечення процесу проектування призначений для автоматичного виконання ЕОМ всіх тих операцій проектної процедури, які вдалося повністю формалізувати.

Одержані варіанти проектних рішень обробляються програмним модулем підготовки даних для оцінки рішень і передаються модулю візуалізації. Аналізуючи результати проектно-конструкторського процесу, інженер повинен мати можливість перегляду вихідних даних у вигляді таблиць, схем і креслень.

Припустимо існування декількох версій проектних рішень, які зберігаються на магнітному диску й можуть бути представлені в необхідному виді за допомогою програмного модуля документування проектних рішень.

Зв'язок між різними програмними модулями проектної процедури й взаємодія даної проектної процедури з іншими відбувається через загальну базу даних. Це дозволяє здійснювати інтерактивний автоматизований процес проектування зі збереженням безлічі різних версій, як вхідних даних, так і проектних рішень.

3.6. Специфіка інформаційного забезпечення САПР

До комплексу засобів автоматизованого проектування входить інформаційне забезпечення, що являє собою сукупність документів, які описують стандартні проектні процедури, типові проектні рішення, типові елементи й комплектуючі вироби, матеріали й інші дані, а також файли й блоки даних на машинних носіях із записом зазначених документів. Головною метою створення інформаційного забезпечення САПР є розробка інформаційної системи, яка дозволяє правильно й швидко вирішувати проектні завдання. Це може бути досягнуто своєчасною видачею джерелу запиту повної й достовірної інформації для виконання певної частини проектно-конструкторського процесу.

Головні вимоги до інформаційного забезпечення САПР наступні:

1. Наявність необхідної інформації для забезпечення як автоматизованих, так і ручних процесів проектування.

2. Можливість зберігання й пошуку інформації, яка являє результат ручних й автоматизованих процесів проектування.

3. Достатній обсяг сховищ інформації. Структура системи повинна допускати можливість нарощування ємності пам'яті разом з ростом обсягу інформації, яка підлягає зберіганню. Одночасно необхідно забезпечити компактність збереженої інформації й мінімальне зношування носіїв інформації.

4. Достатня швидкодія системи інформаційного забезпечення.

5. Можливість швидкого внесення змін і корегування інформації, доведення цих змін до споживача, а також одержання твердої копії документа.

При створенні інформаційного забезпечення САПР основна проблема полягає в перетворенні інформації, необхідної для виконання проектно-конструкторських робіт над певним класом об'єктів, у форму, прийнятну й найбільш раціональну для машинної обробки, і виводу інформації на ЕОМ у вигляді, зручному для сприйняття людиною.

4. СУЧАСНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ (САПР, CALS й ін.)

Автоматизація проектування здійснюється САПР. У САПР різних галузей промисловості прийнято виділяти системи *функціонального, конструкторського й технологічного проектування*.

Перші з них називають системами розрахунків й інженерного аналізу або системами CAE (Computer Aided Engineering).

Системи конструкторського проектування називають системами CAD (Computer Aided Design). Проектування технологічних процесів виконується в автоматизованих системах технологічної підготовки виробництва (АСТПП), що входять як складова частина в системи САМ (Computer Aided Manufacturing).

Для вирішення проблем спільного функціонування компонентів САПР різного призначення, координації роботи систем CAE/CAD/CAM, керування проектними даними й проектуванням розробляються системи, які одержали назву систем керування проектними даними **PDM** (Product Data Management). Системи PDM або входять до складу модулів конкретної САПР, або мають самостійне значення й можуть працювати разом з різними САПР.

На більшості етапів життєвого циклу, починаючи з визначення підприємств-постачальників вхідних матеріалів і компонентів і закінчуючи реалізацією продукції, потрібні послуги системи *керування ланцюгами поставок* — Supply Chain Management (SCM). Ланцюг поставок звичайно визначають як сукупність стадій збільшення доданої вартості продукції при її русі від компаній-постачальників до компаній-споживачів. Керування ланцюгом поставок визначає просування матеріального потоку з мінімальними витратами. При плануванні виробництва система SCM керує стратегією позиціонування продукції. Якщо час виробничого циклу менше часу очікування замовника на одержання готової продукції, то можна застосовувати стратегію "виготовлення на замовлення". Інакше доводиться використовувати стратегію "виготовлення на склад". При цьому в час виробничого циклу повинен входити час на розміщення й виконання замовлень на необхідні матеріали й комплектуючі на підприємствах-постачальниках.

Останнім часом зусилля багатьох компаній, які виробляють програмно-апаратні засоби автоматизованих систем, спрямовані на створення систем електронного бізнесу (E-commerce). Завдання, які розв'язуються системами E-commerce, вирішують не тільки до організації на сайтах Internet вітрин товарів і послуг. Вони поєднують у єдиному інформаційному просторі запити замовників і дані про можливості безлічі організацій, які спеціалізуються на наданні різних послуг і виконанні тих або інших процедур й операцій по проектуванню, виготовленню, поставкам замовлених виробів.

Проектування безпосередньо під замовлення дозволяє домогтися найкращих параметрів створеної продукції, а оптимальний вибір виконавців і ланцюжків поставок веде до мінімізації часу й вартості виконання замовлення. Координація роботи багатьох підприємств-партнерів з використанням технологій Intrenet пок-

ладається на системи E-commerce, названі *системами керування даними в інтегрованому інформаційному просторі* CPC (Collaborative Product Commerce)

Інформаційна підтримка етапу виробництва продукції здійснюється **автоматизованими системами керування підприємством (АСКП) і автоматизованими системами керування технологічними процесами (АСКТП)**.

До АСКП належать системи планування й керування підприємством ERP (Enterprise Resource Planning), планування виробництва й вимог до матеріалів MRP-2 (Manufacturing Requirement Planning) і згадані вище системи SCM. Найбільш розвинені системи ERP виконують різні бізнес-функції, пов'язані із плануванням виробництва, закупівлями, збутом продукції, аналізом перспектив маркетингу, керуванням фінансами, персоналом, складським господарством, обліком основних фондів і т.п. Системи MRP-2 орієнтовані, головним чином, на бізнесі-функції, безпосередньо пов'язані з виробництвом. У деяких випадках системи SCM й MRP-2 входять як підсистеми в ERP, останнім часом їх частіше розглядають як самостійні системи.

Проміжне положення між АСКП й АСКТП займає виробнича виконавча система MES (Manufacturing Execution Systems), призначена для вирішення оперативних завдань керування проектуванням, виробництвом і маркетингом.

До складу АСКТП входить система SCADA (Supervisory Control and Data Acquisition), яка виконує диспетчерські функції (збір й обробка даних про стан устаткування й технологічних процесів) і допомагає розробляти ПО для вбудованого встаткування. Для безпосереднього програмного керування технологічним устаткуванням використовують системи CNC (Computer Numerical Control) на базі контролерів (спеціалізованих комп'ютерів, названих промисловими), які вбудовані в технологічне встаткування із числовим програмним керуванням (ЧПК). Системи CNC називають також убудованими комп'ютерними системами.

На етапі реалізації продукції виконуються функції керування відносинами із замовниками й покупцями, проводиться аналіз ринкової ситуації, визначаються перспективи попиту на плановані вироби. Ці функції покладені на систему CRM.

Функції навчання обслуговуючого персоналу виконують інтерактивні електронні технічні керівництва IETM (Interactive Electronic Technical Manuals). З їхньою допомогою виконуються діагностичні операції, пошук компонентів, які відмовили, замовлення додаткових запасних деталей і деякі інші операції на етапі експлуатації систем.

Керування даними в єдиному інформаційному просторі протягом всіх етапів життєвого циклу виробів покладається на систему керування життєвим циклом продукції PLM (Product Lifecycle Management). Характерна риса **PLM** — забезпечення взаємодії різних автоматизованих систем багатьох підприємств, тобто технології PLM (включаючи технології CPC) є основою, яка інтегрує інформаційний простір, у якому функціонують САПР, ERP, PDM, SCM, CRM й інші автоматизовані системи багатьох підприємств.

4.1. Структура САПР

Як і будь-яка складна система, САПР складається з підсистем. Розрізняють підсистеми, які проектують й обслуговують.

Підсистеми, які проектують, безпосередньо виконують проектні процедури. Прикладами підсистем, що проектують, можуть служити підсистеми геометричного тривимірного моделювання механічних об'єктів, виготовлення конструкторської документації, схемотехнічного аналізу, трасування з'єднань у друкованих платах.

Підсистеми, які обслуговують, забезпечують функціонування підсистем, які проектують, їхню сукупність часто називають системним середовищем (або оболонкою) САПР. Типовими обслуговуючими підсистемами є:

- підсистеми керування проектними даними,
- **підсистеми розробки й супроводу програмного забезпечення CASE (Computer Aided Software Engineering)**,
- навчальні підсистеми для освоєння користувачами технологій, реалізованих у САПР.

Структурування САПР по різних аспектах обумовлює поява *видів забезпечення* САПР. Прийнято виділяти сім видів забезпечення:

- *технічне забезпечення* (ТЗ), яке містить різні апаратні засоби (ЕОМ, периферійні пристрої, мережне комутаційне встаткування, лінії зв'язку, вимірювальні засоби);
- *математичне забезпечення* (МЗ), яке поєднує математичні методи, моделі й алгоритми для виконання проектування;
- *програмне забезпечення* (ПЗ), яке є комп'ютерними програмами САПР;
- *інформаційне забезпечення* (ІЗ), яке складається з баз даних (БД), систем керування базами даних (СКБД), а також містить інші дані, які використовуються під час проектування;
- *лінгвістичне забезпечення* (ЛЗ), яке складається з мов спілкування між проектувальниками й ЕОМ, мов програмування й мов обміну даними між технічними засобами САПР;
- *методичне забезпечення* (МетЗ), яке містить різні методики проектування, іноді до МетО відносять також математичне забезпечення;
- *організаційне забезпечення* (ОЗ), яке містить штатні розклади, посадові інструкції й інші документи, які регламентують роботу проектного підприємства.

Відзначимо, що вся сукупність використаних при проектуванні даних називається інформаційним фондом САПР. Базою даних називають впорядковану сукупність даних, які відображують властивості об'єктів й їхній взаємозв'язок в деякій предметній області. Доступ до БД для читання, запису й модифікації даних здійснюється за допомогою СКБД, а сукупність БД і СКБД називають банком даних (БнД).

Класифікацію САПР здійснюють по ряду ознак, наприклад, по додатку, цільовому призначенню, масштабам (комплексності розв'язуваних завдань), характеру базової підсистеми - ядра САПР.

По додатках найбільш представницькими й широко використовуваними є наступні групи САПР.

1. САПР для застосування в галузях загального машинобудування. Їх часто називають *машинобудівними САПР* або MCAD (Mechanical CAD) системами.
2. *САПР в області радіоелектроніки*: системи ECAD (Electronic CAD) або EDA (Electronic Design Automation).
3. *САПР в області архітектури й будівництва*.

Крім того, відома велика кількість **спеціалізованих САПР**, або виділюваних у зазначених групах, або які являють самостійну галузь у класифікації. Прикладами таких систем є САПР великих інтегральних схем (БІС); САПР літальних апаратів; САПР електричних машин і т.п.

По цільовому призначенню розрізняють САПР або підсистеми САПР, які забезпечують різні аспекти (страти) проектування. Так, у складі MCAD з'являються САЕ/CAD/CAM системи.

За масштабами розрізняють окремі *програмно-методичні комплекси* (ПМК) САПР, наприклад, комплекс аналізу міцності механічних виробів відповідно до **методу кінцевих елементів** (МКЕ) або комплекс аналізу електронних схем; системи ПМК; системи з унікальною архітектурою не тільки програмного (software), але й технічного (hardware) забезпечень.

За характером базової підсистеми розрізняють наступні різновиди САПР.

1. САПР на базі підсистеми **машинної графіки й геометричного моделювання**. Ці САПР орієнтовані на додатки, де основною процедурою проектування є конструювання, тобто **визначення просторових форм і взаємного розташування об'єктів**. Тому до цієї групи систем ставиться більшість САПР в області машинобудування, побудованих на базі графічних ядер. У цей час широко використовуються уніфіковані графічні ядра, застосовані більш ніж в одній САПР, це ядра Parasolid фірми EDS Unigraphics й ACIS фірми Intergraph.
2. САПР на базі СКБД. Вони орієнтовані на додатки, у яких за порівняно нескладними математичними розрахунками переробляється великий обсяг даних. Такі САПР переважно зустрічаються в техніко-економічних додатках, наприклад, при проектуванні бізнес-планів, але мають місце також при проектуванні об'єктів, подібних до щитів керування в системах автоматики.
3. САПР на базі конкретного прикладного пакета. Фактично це автономно використані програмно-методичні комплекси, наприклад, імітаційного моделювання виробничих процесів, розрахунку міцності по методу кінцевих елементів, синтезу й аналізу систем автоматичного керування й т.п. Часто такі САПР відносять до систем САЕ. Прикладами можуть бути програми логічного проектування на базі мови VHDL, математичні пакети типу **MathCAD**.
4. Комплексні (інтегровані) САПР, які складаються із сукупності підсистем попередніх видів. Характерними прикладами комплексних САПР є САЕ/CAD/CAM-системи в машинобудуванні або САПР БІС. Так, САПР БІС містить у собі СКБД і підсистеми проектування компонентів, принципів, логічних і функціональних схем, топології кристалів, тестів для пе-

ревірки придатності виробів. Для керування настільки складними системами застосовують спеціалізовані системні середовища.

4.2. Введення в CALS-технології

CALS-технології визначають як технології комплексної комп'ютеризації сфер виробництва, ціль яким — уніфікація й стандартизація специфікацій промислової продукції на всіх етапах її життєвого циклу. Основні специфікації представлені проектною, технологічною, виробничою, маркетинговою, експлуатаційною документацією. В CALS-системах передбачені зберігання, обробка й передача інформації в комп'ютерних середовищах, оперативний доступ до даних у потрібний час й у потрібній місці з можливістю їхньої правильної інтерпретації.

Головне завдання створення й впровадження CALS-технологій — забезпечення однакового опису й інтерпретації даних, незалежно від місця й часу їхнього одержання в загальній системі, яка може мати масштаби аж до глобальної. **Структура проектної, технологічної й експлуатаційної документації, мови її подання повинні бути стандартизованими.** Тоді стає реальною успішна робота над загальним проектом різних колективів, розподілених у часі й просторі й які використовують різні CAE/CAD/CAM-системи. Та сама конструкторська документація може бути використана багаторазово в різних проектах, а та сама технологічна документація адаптована до різних виробничих умов, що дозволяє істотно скоротити й зменшити ціну загального циклу проектування й виробництва. Крім того, спрощується експлуатація систем.

Термінологія в області CALS ще остаточно не встановилася. Так, спочатку абревіатура CALS розшифровувалася як Computer Aided Logistics Systems, тобто автоматизована логістична підтримка. Оскільки під логістикою звичайно розуміють дисципліну, присвячену питанням постачання й керування запасами, а функції CALS набагато ширші й зв'язані з усіма етапами життєвого циклу промислових виробів, почали застосовувати більш відповідну предмету розшифровку CALS - Continuous Acquisition and Lifecycle Support. У російській мові поняттю CALS відповідають ІПв (Інформаційна Підтримка Виробів).

CALS-технології покликані виступати засобом, який інтегрує промислові автоматизовані системи в єдину багатофункціональну систему. Метою інтеграції автоматизованих систем проектування й керування є підвищення ефективності створення й використання складної техніки.

У чому виражається підвищення ефективності?

По-перше, підвищується якість виробів за рахунок більш повного обліку наявної інформації при проектуванні й прийнятті управлінських рішень. Так, обґрунтованість рішень, прийнятих в автоматизованій системі керування підприємством (АСКП), буде вищою, якщо особа, яка приймає рішення й відповідні програми АСКП мають оперативний доступ не тільки до бази даних АСКП, але й до баз даних інших автоматизованих систем (САПР, АСТПП й АСКТП) і, отже, можуть оптимізувати плани робіт, зміст заявок, розподіл виконавців, виділення фінансів і т.п. При цьому під оперативним доступом треба розуміти не просто можливість зчитування даних з баз даних (БД), але й легкість їхньої правильної інтерпретації, тобто погодженість за синтаксисом й семантикою із

протоколами, прийнятими в АСКП. Те ж потребують й інші системи, наприклад, технологічні підсистеми повинні з необхідністю сприймати й правильно інтерпретувати дані, які надходять від підсистем автоматизованого конструювання. Останнього не так легко домогтися, якщо основне підприємство й організації-суміжники працюють із різними автоматизованими системами.

По-друге, скорочуються матеріальні й тимчасові витрати на проектування й виготовлення. Застосування CALS дозволяє істотно скоротити обсяги проектних робіт, тому що описи раніше виконаних вдалих розробок компонентів і пристроїв, багатьох складних частин устаткування, машин і систем, що проектувалися раніше, зберігаються в базах даних мережних серверів, доступних будь-якому користувачеві технології CALS. Доступність знову ж забезпечується погодженістю форматів, способів, описів керування у різних частинах загальної інтегрованої системи. Крім того, з'являються більш широкі можливості для спеціалізації підприємств, аж до створення віртуальних підприємств, що також сприяє зниженню витрат.

По-третє, істотно знижуються витрати на експлуатацію завдяки реалізації функцій інтегрованої логістичної підтримки. Істотно полегшується рішення проблем ремонтпридатності, інтеграції продукції в різного роду системи й середовища, адаптації до мінливих умов експлуатації й т.п.

Ці переваги інтеграції даних досягаються застосуванням сучасних CALS-технологій. Промислові автоматизовані системи можуть працювати автономно, і в цей час так звичайне й відбувається. Однак ефективність автоматизації буде більш помітною, якщо дані, які генеруються в одній із систем будуть доступні в інших системах, оскільки прийняті в них рішення стануть більше обґрунтованими. Більш того, при уніфікації форматів і семантики даних стає можливим створення *віртуальних підприємств*, під якими розуміються об'єднання юридично незалежних підприємств, що здійснюють загальні проекти й виробництва на основі інформаційної взаємодії.

Щоб досягти належного рівня взаємодії промислових автоматизованих систем, потрібне створення єдиного інформаційного простору в рамках як окремих підприємств, так й, що більш важливо, у рамках об'єднання підприємств. Єдиний інформаційний простір забезпечується завдяки уніфікації як форми, так і змісту інформації про конкретні вироби на різних етапах їхнього життєвого циклу.

Уніфікація форми досягається використанням стандартних форматів і мов подання інформації в міжпрограмних обмінах і при документуванні.

Уніфікація змісту, яка розуміється як однозначно правильна інтерпретація даних про конкретний виріб на всіх етапах його життєвого циклу, забезпечується розробкою онтологій (метаописів) додатків, що закріплюють у прикладних протоколах CALS.

Уніфікація переліків і найменувань сутностей, атрибутів і відносин у певних предметних областях є основою для єдиного електронного опису виробу в CALS-просторі.

Розвиток CALS-технології стимулює утворення віртуальних виробництв, при яких процес створення специфікацій з інформацією для програмно керованого технологічного встаткування, достатньої для виготовлення виробу, може

бути розподілений у часі й просторі між багатьма організаційно автономними проектними студіями. Серед безсумнівних досягнень CALS-технології слід зазначити легкість поширення передових проектних рішень, можливість багаторазового відтворення частин проекту в нових розробках й ін.

Очікується, що успіх на ринку складної технічної продукції буде немислимий поза технологією CALS.

Проблематика CALS має ряд аспектів. За аналогією з аспектами автоматизованого проектування доцільно ці аспекти **називати видами забезпечення CALS** і виділяти лінгвістичне, інформаційне, програмне, математичне, методичне, технічне й організаційне забезпечення CALS.

До **лінгвістичного забезпечення** належать мови й формати даних про промислові вироби й процеси, використані для подання й обміну інформацією на етапах життєвого циклу виробів.

Інформаційне забезпечення становлять бази даних, які включають відомості про промислові вироби, використані різними системами в процесі проектування, виробництва, експлуатації й утилізації продукції. До складу інформаційного забезпечення входять також серії міжнародних і національних CALS стандартів і специфікацій.

Програмне забезпечення CALS представлене програмними комплексами, призначеними для підтримки єдиного інформаційного простору етапів життєвого циклу виробів. Це насамперед системи керування документами й документообігом, керування проектними даними (PDM), взаємодії підприємств у спільному електронному бізнесі (CPC), підготовки інтерактивних електронних технічних довідників і деякі інших.

Математичне забезпечення CALS включає методи й алгоритми створення й використання моделей взаємодії різних систем в CALS-технологіях. Серед цих методів, у першу чергу, слід назвати методи імітаційного моделювання складних систем, методи планування процесів і розподілу ресурсів.

Методичне забезпечення CALS подане методиками виконання таких процесів, як паралельне (сполучене) проектування й виробництво, структурування складних об'єктів, їх функціональне й інформаційне моделювання, об'єктно-орієнтоване проектування, створення онтологій додатків.

До технічного забезпечення CALS відносять апаратні засоби одержання, зберігання, обробки, візуалізації даних при інформаційному супроводі виробів. Взаємодія частин віртуальних підприємств, систем, що підтримують різні етапи життєвого циклу виробів, відбувається через лінії передачі даних і мережне комутуюче встаткування.

Нарешті, організаційне забезпечення CALS подане різного роду документами, сукупністю угод й інструкцій, які регламентують ролі й обов'язки учасників життєвого циклу промислових виробів.

5. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ САПР ТЕХНІЧНИХ ОБ'ЄКТІВ

Математичне забезпечення містить у собі моделі об'єктів, моделі завдань, методи й алгоритми проектних процедур. Проектування має на увазі облік су-перечливих вимог.

Його продуктами є моделі, що дозволяють нам зрозуміти структуру май-бутнього об'єкта, збалансувати вимоги й визначити схему реалізації.

Важливість побудови моделі. Моделювання поширене у всіх інженерних дисциплінах, у значній мірі через те, що воно реалізує принципи декомпозиції, абстракції й ієрархії. Кожна модель описує певну частину розглянутого об'єкта, а ми у свою чергу будуємо **нові моделі на базі старих**, у яких більш-менш упе-внені. Моделі дозволяють нам контролювати наші невдачі. Ми оцінюємо пово-дження кожної моделі у звичайних і незвичайних ситуаціях, а потім проводимо відповідні доробки, якщо нас щось не задовольняє.

5.1. Моделювання об'єктів і процесів

Поняття

Математична модель (у широкому сенсі - **модель завдання**) – образ(modus) проблеми, виражений математичною мовою. Для прикладних проблем (при по-становці Завдання), основою такого образу виступає запис:

$$U=F(x_1, x_2 \dots x_k; y_1, y_2 \dots y_l), \quad (1)$$

де U – показник корисності (**цільова функція**), зміст якого залежить від змісту проблеми; x_i ($i=1,2,\dots,k$) -керовані параметри, тобто ті, на значення яких може впливати керівник, проектувальник, менеджер і взагалі "особа, що приймає рі-шення" (ОПР); y_j ($j=1,2,\dots,l$) -некеровані параметри; **F** – **функція, що описує ві-дносини між вхідними в (1) величинами**. Назвемо набір з k чисел x - k -мірним вектором, а набір з l числа y - l -мірним вектором і позначимо їх таким чином:

$$X_k = (x_1, x_2 \dots x_k), \quad Y_l = (y_1, y_2 \dots y_l) \dots$$

Тепер вираження (1) стане коротшим:

$$U=F(X_k, Y_l) \quad (1')$$

Межі зміни керованих параметрів задають область їхніх припустимих значень, що описується нерівностями:

$$x_{i \min} \leq x_i \leq x_{i \max}, \quad i=1,2,\dots,k, \quad (2)$$

а іноді й **функціональними залежностями** між змінними x_i , про що будемо говорити нижче при розгляді прикладів. Тут же помітимо, що нерівності (2) і ці функціональні залежності входять до складу моделі завдання поряд з виражен-нями (1), (1'). Функція F може мати вигляд або формули, або алгоритму, за до-помогою якого, знаючи X й Y , можна обчислити U .

Приклади: див. лаб. раб.

5.2. Класифікація математичних моделей об'єктів $F(x,y)$ у САПР

Основні поняття.

! Визначення: Математичною моделлю реальної системи (об'єкта, процесу) називають сукупність співвідношень (формул, рівнянь, логічних виражень), які визначають характеристики стану системи залежно від її параметрів, початкових умов, вхідних сигналів (включаючи випадкові фактори) і часу.

.Модель - один з видів опису проєктованих об'єктів.

Вимоги до моделей: універсальності, адекватності, точності, економічності.

- **Ступінь універсальності:** повнота відображення в моделі властивостей реального об'єкта. Математична модель відображує лише деякі властивості об'єкта.
- **Точність** - ступінь збігу значень параметрів реального об'єкта й значень тих же параметрів реал. об'єкта. Наприклад, відносна похибка розрахунку:

$$e_i = (Y_m - Y_{\text{ист}}) / Y_{\text{ист}}$$

- **Адекватність** - здатність відображувати задані властивості об'єкта з погрішністю не вище заданої.
- **Економічність** – мінімізація витрат ресурсів часу й пам'яті на її реалізацію. Виходячи із цих вимог необхідно домагатися компромісу при побудові й використанні моделі .При цьому треба враховувати практичні вимоги , мети й завдання(розв'язувані з використанням конкретної моделі)

Класифікація.

Математичні моделі в першу чергу визначаються предметною областю проєктування :

- фізичним и математичним описом законів и умов функціонування об'єкту;
- середовищем функціонування та засобами взаємодії об'єкта з середовищем;
- структурою об'єкта, елементною базою, засобами організації структури об'єкта;
- параметрами, які змінюються або налаштовуються.

ТИПИ:

- Структурні, функціональні.
- Макрорівневі, макрорівневі.
- Аналітичні, алгоритмічні, імітаційні.
- Теоретичні , емпіричні .

У моделях розрізняють інформацію (дані) трьох типів:

- дані про елементи моделювання;
- дані про властивості об'єкту й елементів;
- дані про відношення між елементами та об'єктом.

Абстрагування об'єктів проводять як по глибині структурування (ієрархічна система, система елементів, нероздільний), так і по ступені абстрагування елементів і властивостей об'єкта (структурний, логічний і кількісний рівні).

1. **На структурному рівні** моделюється структура об'єкта на найнижчому рівні його структурування, коли математична модель надана у вигляді безлі-

чі властивостей і параметрів елементів, які описані через функціональні зв'язки, співвідношення порядку, суміжності, з'єднання. *Розрізняють такі рівні :*

- **топологічні** - склад і взаємозв'язок елементів об'єкта
- **геометричні** – (форми деталей , рівняння поверхонь і ліній
- **каркасні** -

При цьому використовується апарат теорії множин і графів, теорії масового обслуговування й т.д.

2 На логічному рівні моделювання кожній безлічі, булевій матриці бінарних співвідношень або структурному графові відповідають набори логічних співвідношень і змінних, які відображують причинно-наслідкові зв'язки. При проектуванні використовується апарат математичної логіки.

3 На кількісному рівні кожному елементу безлічі, булевій матриці або логічній змінній ставиться в співвідношення алгебраїчна або інша кількісна змінна, а логічні співвідношення переходять у кількісні співвідношення: рівняння, нерівності й ін.

Моделювання на **кількісному рівні** відображує функціональні, енергетичні, речовинні й просторові зв'язки. Ці зв'язки звично описуються часовими-просторово-тимчасовими співвідношеннями й визначаються через звичайні диференціальні рівняння або рівняння в частинних похідних.

Якщо об'єкт складається із системи елементів, зв'язок між якими описується тільки одною змінною (часом), то вживаються моделі із зосередженими параметрами. Елементи об'єкта кількісно описуються **компонентними рівняннями**. (Наприклад, закони Кирхгофа)

Розділяють **мікромоделі** (які відображують фізичні процеси в безперервному часі й просторі – дифф. рівняння в частинних похідних) і **макромоделі**, коли не береться до уваги внутрішня структура об'єктів, які моделюються, або елементів. - як правило звичайні диффер. рівняння).

Для сталих режимів - використовуються системи алгебраїчних рівнянь

Функціональні моделі розподіляються на аналітичні й алгоритмічні

Аналітичні – являють явні вираження вихідних параметрів як функцію вхідних і внутрішніх параметрів, тобто

$$Y = F(X, Q)$$

Такі моделі одержувати досить складно (як правило, при спрощенні й ін. обмеженнях).

Основна частина розрахунків у САПР(по обсязі й витратам) виконується на кількісному рівні моделювання, де враховуються всі існуючі нелінійні співвідношення.

Тому доцільно автоматизувати там, де тільки можливо, процедури формування **аналітичної моделі** об'єкта (його елементів) на обраному рівні структурування й абстрагування.

Як правило, у програмах аналізу й синтезу в САПР одержання моделей проєктованих об'єктів забезпечується реалізацією ММ елементів і методів формування ММ систем - об'єктів.

Методика одержання (побудови) математичних моделей)

Наступні етапи:

1. Вибір властивостей об'єкта, які підлягають відображенню в моделі.
2. Збір вихідної інформації.
3. Синтез структури ММ, тобто вибір **структури** математичних співвідношень.
4. Розрахунок числових **значень параметрів** ММ: Це завдання мінімізації похибки моделі заданої структури.
5. Оцінка точності й адекватності ММ
6. Якщо **отримали незадовільний результат**: вибір нової структури, і знову оцінка параметрів.

Перетворення ММ:

1. Спрощення
2. Лінеаризація (тобто одержання лінійних систем рівнянь)
3. Інші види

Ефективність мат. забезпечення САПР залежить від реалізації чисельних методів рішення багатьох математичних завдань.

Для побудови моделей використовують два принципи: **дедуктивний** (від загального до частки) і **індуктивний** (від часткового до загального).

При **першому підході** розглядається окремий випадок загальновідомої фундаментальної моделі, що пристосовується до умов об'єкта, який моделюємо, з урахуванням конкретних обставин.

Другий спосіб припускає висування гіпотез, декомпозицію складного об'єкта, аналіз, а потім синтез. Тут широко використовується подоба, пошук аналогій, умовивід з метою формування яких-небудь закономірностей у вигляді припущень про поведінку системи.

Технологія моделювання жадає від дослідника вміння коректно формулювати проблеми й завдання, прогнозувати результати, проводити розумні оцінки, виділяти головних і другорядних факторів для побудови моделей, знаходити аналогії й виражати їх мовою математики.

Отже, висновки

Основу гарної моделі становить знання предмета: потрібно досконально розібратися в суті проблеми, для якого розробляється модель. Вираження (1) і (1') показують, що для цього потрібно:

1. Чітко сформулювати показник корисності U (цільову функцію або критерій якості), який повинен досить щільно відбивати **мети** об'єкта, який моделюється.

2. Визначити фактори (параметри моделі), якими ми хочемо й можемо управляти (компоненти вектора X), можливі межі їхньої зміни й функціональні обмеження, якщо такі існують

Зазвичай, практичний досвід показує, які параметри доступні керуванню, а які - ні. Для повної прозорості в цьому питанні корисно намалювати схему впливу параметрів на показник корисності та один на одного. Вона дозволить уникнути характерної помилки - залучення до складу керованих параметра, який сам є функцією іншого керованого параметра. Нічого, крім ускладнення моделі, а іноді й обчислювальних труднощів, це не принесе.

3. Визначити склад і величини **некерованих параметрів**, які впливають на показник корисності (компоненти вектора Y). Тут розроблювача моделі очікують найбільші труднощі: з одного боку, хочеться відбити в моделі максимальну кількість таких параметрів, оскільки (особливо при відсутності досвіду) здається, що чим більше їх буде уведено в модель, тим вона буде точніша; з іншого боку, завжди присутнє бажання **зробити модель як можна простішою**, зрозумілішою, а для цього кількість параметрів потрібно всіляко скорочувати. Перефразовуючи відоме виречення, скажемо, що ви повинні *навчитися враховувати головне, відкидати другорядне й постійно молити Бога дати вам мудрість відрізнити перше від другого*.

На жаль, запропонувати тут якісь рецепти неможливо: уміння приходить із досвідом!

Лише в найпростіших випадках, коли параметр входить доданком до складу суми, можна порівнювати його з іншими доданками, і якщо виявиться, що цей параметр у багато разів менше найменшого з них, то їм можна зневажити.

4. Визначити вид функції F . Ця операція часто носить формальний характер, оскільки вид функції визначається природою показника корисності. У багатьох завданнях ця функція - лінійна форма, тобто сума, у яку параметри входять із постійними множниками - коефіцієнтами, роль яких виконують некеровані параметри.

Однак зустрічаються й більше складні залежності. І, нарешті, як ми вже відзначали, ця функція взагалі може не мати явного виду, а існувати у вигляді алгоритму, за яким пишеться обчислювальна програма. Головна проблема тут полягає в тому, щоб функція F правильно відбивала відносини між всіма компонентами моделі.

Якщо фактори, що істотно впливають на показник корисності, враховані, і функція F правильно описує відносини між ними, то можна сподіватися, що модель адекватна об'єкту, для якого вона розроблялася.

Поняття адекватності припускає, що результати моделювання з достатньою точністю відповідають результатам реальної роботи об'єкта. Як перекоонатися в адекватності моделі? Для цього існує кілька прийомів, які можна звести до двох ситуацій.

1). Якщо ви маєте у своєму розпорядженні дані про роботу об'єкта, який ви моделюєте (або аналогічного йому), у попередній вашому дослідженню період, ви можете витягти із цих даних реальні значення параметрів, що входять у модель, обчислити показник корисності й зрівняти результат, який отримали, з його фактичним значенням. Якщо відхилення не перевищує величини, яка влаштовує вас, (на... 15%), модель **визнається адекватною**. У протилежному випадку починається пошук й усунення причин розбіжностей. Ця процедура триває доти, доки необхідна згода не буде отримана, після чого модель можна використовувати для планування майбутніх дій.

2). Якщо даних про попередню роботу об'єкта у вас немає (створюється новий об'єкт), і добути відомості про роботу аналогічного об'єкта неможливо, доводиться діяти методом оцінки межових значень. Сутність його полягає в тому, що деяким параметрам приписуються крайні значення, наприклад нульові

або, навпаки, надзвичайно великі. У цьому випадку, як правило, можна логічними міркуваннями, а іноді аналітичними викладеннями пророчити результат моделювання. Якщо розрахунок за допомогою моделі приводить до того ж або близького результату, то модель визнається адекватною. У протилежному випадку - див. останні речення п.1.

Домігшись адекватності, сміливо приступайте до експериментів з розробленою моделлю. У ході цих експериментів ви не тільки одержите корисні результати, але й незабаром відчуєте, що ваші знання й розуміння процесу її функціонування заглиблюються, тобто модель служить тренажером ваших здатностей. Справді, сидячи за комп'ютером, ви можете, нічим не ризикуючи, не роблячи ніяких витрат (крім власного часу), ставити самі сміливі випробування, оцінювати їхні результати, вибираючи стратегію керування.

6. ОПТИМІЗАЦІЯ В САПР - ТЕХНОЛОГІЇ МСАД

6.1. Основи теорії

Під завданням оптимізації будемо розуміти знаходження екстремуму (мінімуму або максимуму) функції однієї або декількох дійсних змінних. До рішення оптимізаційних завдань зводяться завдання пошуку коренів нелінійних рівнянь і систем, апроксимації функцій й ін.

Нехай дана дійсна функція n дійсних змінних $F(\mathbf{x})$, $\mathbf{x} = (x_1, \dots, x_n)$, певна в замкнутій області $S \in R^n$. Потрібно знайти

$$\min_{\mathbf{x} \in S} F(\mathbf{x}) \quad \left(\max_{\mathbf{x} \in S} F(\mathbf{x}) \right) \quad (1)$$

І точку $\mathbf{x}^* \in S$, у якій він досягається.

Відмітимо, що пошук максимуму функції $F(\mathbf{x})$ еквівалентний пошуку мінімуму функції $-F(\mathbf{x})$. Тому далі сконцентруємо увагу саме на завданні мінімізації.

Множина S виражає обмеження завдання. Наприклад, у багатьох фізичних завданнях потрібно, щоб змінні були ненегативні: $S = \{\mathbf{x}: x_i \geq 0\}$. Якщо S збігається з R^n , то говорять про завдання *безумовної оптимізації*, у протилежному випадку говорять про *умовну оптимізацію*.

Визначимо також наступні два терміни. Будемо говорити, що $F(\mathbf{x})$ має в точці $\mathbf{x}^* \in S$ *глобальний мінімум*, якщо $F(\mathbf{x}) \geq F(\mathbf{x}^*)$ для будь-якого $\mathbf{x} \in S$. Аналогічно, $F(\mathbf{x})$ має в точці $\mathbf{x}^* \in S$ *локальний мінімум*, якщо $F(\mathbf{x}) \geq F(\mathbf{x}^*)$ для всіх $\mathbf{x} \in S \cap D(\mathbf{x}^*)$, де $D(\mathbf{x}^*)$ – деяка околиця крапки \mathbf{x}^* .

Якщо локальний мінімум досягається у внутрішній точці \mathbf{x}^* області S і функція $F(\mathbf{x})$ диференціюється в цій точці, то \mathbf{x}^* є критичною точкою, тобто виконуються умови:

$$\frac{\partial F}{\partial x_i}(\mathbf{x}^*) = 0, \quad i = 1, \dots, n. \quad (2)$$

Зворотнє невірне: критична точка не зобов'язана бути екстремальною. Наприклад, в одновимірному випадку для функції $y = x^3$ точка $x=0$ є критичною, але не є крапкою екстремума. У двовимірному випадку для функції

$F(x, y) = x^2 - y^2$ критичної є точка $(0,0)$. Вона також не реалізує екстремум, оскільки одномірна функція $F(x,0)$ має мінімум при $x = 0$, а одномірна функція $F(0, y)$ – максимум при $y = 0$.

Якщо ж локальний мінімум досягається на межі області S , то точка мінімуму не зобов'язана бути критичною. Наприклад, функція $f(x) = x$ досягає на відрізку $[1,2]$ мінімум у точці $x^*=1$, але $f'(1) = 1 \neq 0$.

У дослідженні операцій завдання оптимізації називають завданнями математичного програмування.



6.2. Види завдань математичного програмування.

Загальне завдання математичного програмування розбивають на завдання, назви яких визначаються виглядом функції, яка оптимізується і функцій, які входять в умови-обмеження, типом змінних й алгоритмом рішення:

- 1 лінійного програмування – функції лінійні;
- 2 нелінійного програмування - хоча б одна з функцій нелінійні;
- 3 квадратичного програмування - $f(x)$ є квадратичною функцією, обмеження лінійні (більш точно, $F(x)$ повинна бути квазівизначеною квадратичною формою);
- 4 сепарабельного програмування - $f(x)$ являє собою суму функцій, різних для кожної змінної, умови - обмеження - можуть бути як лінійними, так і нелінійними (але всі недиагональні елементи матриці, що складає із других часток похідних будь-якої функції завдання, дорівнюють нулю);
- 5 целочисельного (лінійного й нелінійного) програмування - координати шуканого вектора є тільки цілими числами;
- 6 випукле програмування – функція $f(x)$ - випукла, $g_i(x)$ - увігнуті, тобто. розглядають випуклі функції на випуклих множинах і т.п.

У якості оптимізаційних завдань виступають різні завдання проектування й керування. У випадку одного проектного параметру ($n=1$) цільова функція є функцією однієї змінної і її можна надати у вигляді плоского графіка. При $n=2$ - її графіком є поверхня.

В одному завданні може бути кілька цільових функцій (так зване багатокритеріальні завдання).

Наприклад, при проектуванні виробів одночасно необхідно забезпечити максимальну надійність виробу, його максимальну продуктивність, мінімальну матеріалоемність при виготовленні. Може статися, що деякі функції не сумісні, у таких випадках необхідно встановлювати пріоритет тієї або іншої функції.

6.3. Оптимізація в середовищі MCAD: Функції Minimize й Maximize.

У завданнях аналізу й побудови математичних моделей й інших допомагають вбудовані функції **Minimize** й **Maximize**, які з'явилися в системах MathCAD починаючи з 8 версії.

Ці функції належать до категорії вбудованих функцій і реалізують *процедуру пошуку екстремуму* функцій $F(\mathbf{a})$ багатьох змінних a_i як при наявності, так і при відсутності обмежень на комбінації останніх. Функції $F(\mathbf{a})$ у завданнях оптимізації можуть бути як лінійними, так і нелінійними (наприклад, квадратичними). Тому при використанні вбудованих функцій **Minimize** й **Maximize** передбачений вибір методу оптимізації (наприклад, метод сполучених градієнтів, метод Ньютона для нелінійних функцій $F(\mathbf{a})$), для чого необхідно натиснути *праву кнопку миші* при наведенні курсору на **Minimize** або **Maximize**.

Пошукові процедури широко використовуються в завданнях побудови математичних моделей у проектуванні.

Зрозуміло, що можливість практичного застосування вбудованих функцій **Minimize** й **Maximize** не виключає можливість розробки спеціальних програм пошуку екстремуму, якщо буде така потреба.

З теорії методів пошуку екстремумів відомо, що якість одержаних у результаті пошуку рішень великим ступенем залежить від характеру ліній рівнів. У тому випадку, коли поверхня рівнів має характер витягнутого яру, можуть виникнути проблеми досягнення точного рішення. На жаль, більшість завдань визначення параметрів апроксимуючих виражень належить саме до цього типу.

Завершуючи аналіз можливості застосування пошукових процедур і відповідних вбудованих функцій MathCAD при побудові моделей, відзначимо основні правила, які допомагають досягти якісного рішення:

1. Вихідна характеристика об'єкта повинна бути задана на інтервалі часу, який гарантує інформативність вихідних даних; завдання цієї характеристики на малому інтервалі часу, коли ще не виявилися всі особливості її зміни, не гарантують досягнення прийнятної точності;

2. Використання додаткової інформації сприяє скороченню часу рішення й підвищенню точності результату;

3. Аналіз тривимірного зображення фрагментів поверхонь рівнів і варіація початкового наближення дозволяє уникнути локальних мінімумів.

7. ПОНЯТТЯ ПРО БАЗИ ДАНИХ ТА СИСТЕМИ КЕРУВАННЯ НИМИ.

7.1. Класифікація баз даних. Основні засоби обробки даних.

Бази даних є одним з головних компонентів сучасних інформаційних систем.

Інформаційна система — це взаємозалежна сукупність засобів, методів і персоналу, використаних для зберігання, обробки й видачі інформації.

Мета будь-якої інформаційної системи — обробка інформації конкретної предметної області.

Під *предметною областю* розуміється сукупність зв'язаних між собою функцій, завдань керування в деякій області діяльності підприємства, за допомогою яких досягається виконання поставленої мети.

База даних — це інформаційні структури, які містять взаємозалежні дані про реальні об'єкти.

Особливостями такої сукупності даних є:

- досить великі обсяги інформації;
- максимально можлива компактність зберігання даних;
- можливість отримання з бази даних різноманітної і в певній предметній області;
- зручні для користувача вид і форма отриманої інформації;
- висока швидкість доступу до даних;
- надійність зберігання інформації й можливість надання санкціонованого доступу до даних для окремих користувачів;
- зручність і простота конструювання користувачем запитів, форм і звітів для вибірки даних. Створення бази даних, її підтримка й забезпечення доступу користувачів до неї здійснюється за допомогою спеціального програмного інструмента - системи керування базами даних.

Система керування базами даних (СКБД) — це програмне забезпечення для створення й редагування баз даних, перегляду й пошуку інформації в них.

За технологією обробки бази даних розподіляються на *централізовані* й *розподілені*. Централізована база даних зберігається в пам'яті однієї машини. Розподілена база даних складається з декількох частин, збережених на декількох машинах обчислювальної мережі. Робота з такою базою здійснюється за допомогою системи керування розподіленою базою даних - СКРБД.

Централізовані бази даних за способом доступу розподіляються на:

- бази даних з локальним доступом (дані й процедури їхньої обробки зберігаються на одній машині);
- бази даних з вилученим (мережним) доступом. СКБД із вилученим доступом можуть бути побудовані з використанням архітектури файл-сервер і клієнт-сервер.

Архітектура файл-сервер. Принцип організації: одна машина виділена в якості центральної (сервер файлів), на ній зберігається централізована БД. Інші машини мережі виконують функції робочих станцій. Файли бази даних відповідно до користувальницьких запитів робочих станцій передаються на ці станції

й там обробляються. Продуктивність такої системи падає, якщо потрібно інтенсивний одночасний доступ до одних даних.

Архітектура клієнт-сервер. Принцип організації: центральна машина (сервер бази даних) зберігає централізовану БД і процедури обробки. Клієнт надсилає запит, він оброблюється сервером, і дані, отримані за запитом, передаються клієнтові.

7.2. Основні види моделей БД

Основою бази даних є модель даних. Інформаційно-логічна (міфологічна) модель предметної області відбиває предметну область у вигляді сукупності інформаційних об'єктів й їхніх структурних зв'язків.

Інформаційний об'єкт — це опис деякої сутності (явища, реального об'єкта, процесу) у вигляді сукупності логічно зв'язаних реквізитів. Наприклад, інформаційний об'єкт Студент описує деяку сутність - студент. Реквізитний склад цього інформаційного об'єкта, тобто його структура, це буде: № групи, ФІО, № залікової книжки, дата народження. Інформаційний об'єкт має безліч реалізацій - екземплярів.

Бази даних також можуть бути ієрархічними, мережними й реляційними.

Ієрархічна (деревоподібна) модель даних являє собою ієрархію елементів, названих вузлами. Вузол - це сукупність атрибутів даних, які описують інформаційний об'єкт. На самому верхньому рівні є тільки один вузол - корінь. Кожен вузол крім кореня зв'язаний тільки з одним вузлом на більш високому рівні, названому вихідним вузлом для даного вузла. Кожен вузол може бути пов'язаний з одним або декількома вузлами більш низького рівня, названими породженими (підлеглими). Вузли, що не мають породжених, називаються листами. Кількість дерев у базі даних визначається числом кореневих записів. До кожного запису бази даних існує тільки один шлях.

Мережна модель також ґрунтується на поняттях вузол, рівень, зв'язок. Мережна модель даних - це модель, у якій породжений вузол може мати більше одного вихідного вузла. У мережній структурі будь-який елемент будь-якого рівня може бути пов'язаний з будь-яким іншим елементом.

Реляційна модель бази даних складається з одного або декількох файлів, кожний з яких відповідає одній таблиці.

Основна мета проектування бази даних — це скорочення надмірності збережених даних, а отже, економія обсягу використаної пам'яті, зменшення витрат на багаторазові операції відновлення надлишкових копій й усунення можливості виникнення протиріч через зберігання в різних місцях відомостей про один і той самий об'єкт. При створенні баз даних варто дотримуватися методології нормалізації відносин.

Процес проектування інформаційних систем є досить складним завданням. Він починається з побудови інфологічної моделі даних, тобто ідентифікації сутностей. Потім необхідно виконати наступні кроки процедури проектування даталогічної моделі, тобто міфологічна модель повинна бути '': відображена в комп'ютеро-орієнтовану даталогічну модель, «зрозумілу» СКБД.

➤ Представити предметну область у вигляді сукупності окремих незалежних друг від друга об'єктів, кожний з яких буде описаний своєю таблицею.

- Для кожної таблиці визначити ключові поля; установити зв'язок між таблицями; для кожного зв'язку визначити тип.
- Розробити структуру кожної таблиці: перелік полів, їхні типи й властивості.
- Заповнити таблиці даними.
- Розробити необхідні запити до БД, вхідні й вихідні форми й звіти.
- Передбачити можливість автоматизації часто виконуваних дій шляхом створення макросів і програмних модулів.

7.3. Реляційні бази даних та їх особливості. Види зв'язку між реляційними базами.

Реляційна база даних — це сукупність взаємозалежних таблиць, кожна з яких містить інформацію про об'єкти певного типу. Рядок таблиці містить дані про один об'єкт (наприклад, товар, клієнт), а стовпці таблиці описують різні характеристики цих об'єктів - атрибутів (наприклад, найменування, код товару, відомості про клієнта). Записи, тобто рядки таблиці, мають однакову структуру - вони складаються з полів, що зберігають атрибути об'єкта. Кожне поле, тобто стовпець, описує тільки одну характеристику об'єкта й має строго певний тип даних. Всі записи мають ті самі поля, тільки в них відображуються різні інформаційні властивості об'єкта.

У реляційній базі даних кожна таблиця повинна мати первинний ключ - поле або комбінацію полів, які єдиним чином ідентифікують кожен рядок таблиці. Якщо ключ складається з декількох полів, він називається складовим. Ключ повинен бути унікальним й однозначно визначати запис. За значенням ключа можна відшукати єдиний запис. Ключі служать також для впорядкування інформації в БД.

Таблиці реляційної БД повинні відповідати вимогам нормалізації відносин. Нормалізація відносин - це формальний апарат обмежень на формування таблиць, що дозволяє усунути дублювання, забезпечує несуперечність збережених у базі даних, зменшує витрати праці на ведення бази даних.

Нехай створена таблиця Студент, що містить наступні поля:

- № групи,
- ФІО,
- № залікової книжки,
- дата народження,
- назва спеціальності,
- назва факультету.

Така організація зберігання інформації буде мати ряд недоліків:

- дублювання інформації (найменування спеціальності й факультету повторюються для кожного студента), отже, збільшиться обсяг БД;
- процедура відновлення інформації в таблиці стає труднішою через необхідність редагування кожного запису таблиці.

Нормалізація таблиць призначена для усунення цих недоліків. Є три нормальні форми відносин.

Перша нормальна форма. Реляційна таблиця наведена до першої нормальної форми тоді й тільки тоді, коли жоден з її рядків не містить у будь-якому

своєму полі більше одного значення й жодне з її ключових полів не порожньо. Так, якщо з таблиці Студент потрібно одержувати відомості по імені студента, то поле ФІО варто розбити на частині Прізвище, Ім'я, По батькові.

Друга нормальна форма. таблиця задана в другій нормальній формі, якщо вона задовольняє вимогам першої нормальної форми й всі її поля, що не входять у первинний ключ, зв'язані повною функціональною залежністю з первинним ключем.

Щоб привести таблицю до другої нормальної форми, необхідно визначити функціональну залежність полів. Функціональна залежність полів - це залежність, при якій в екземплярі інформаційного об'єкта певному значенню ключового реквізиту відповідає тільки одне значення описового реквізиту.

Третя нормальна форма. Таблиця перебуває в третій нормальній формі, якщо вона задовольняє вимогам другої нормальної форми, жодне з її неключових полів не залежить функціонально від будь-якого іншого неключового поля. Наприклад, у таблиці Студент (№ групи, ФІО, № залікової книжки, Дата народження, Староста) три поля - № залікової книжки, № групи, Староста перебувають у транзитивній залежності. № групи залежить від № залікової книжки, а Староста залежить від № групи. Для усунення транзитивної залежності необхідно частину полів таблиці Студент перенести в іншу таблицю Група. Таблиці приймуть наступний вид: Студент (№ групи, ФІО, № залікової книжки, Дата народження), Група (№ групи, Староста).

Над реляційними таблицями можливі наступні операції:

- Об'єднання таблиць із однакою структурою. Результат - загальна таблиця: спочатку перша, потім друга (конкатенація).
- Перетинання таблиць із однакою структурою. Результат - вибираються ті записи, які перебувають в обох таблицях.
- Вирахування таблиць із однакою структурою. Результат - вибираються ті записи, яких немає у від'ємнику.
- Вибірка (горизонтальна підмножина). Результат - вибираються записи, що відповідають певним умовам.
- Проекція (вертикальна підмножина). Результат - відношення, яке містить частина полів з вихідних таблиць.
- Декартово - добуток двох таблиць Записи результуючої таблиці виходять шляхом об'єднання кожного запису першої таблиці з кожним записом іншої таблиці.

Реляційні таблиці можуть бути зв'язані одна з одною, отже, дані можуть витягатися одночасно з декількох таблиць. Таблиці зв'язуються між собою для того, щоб в остаточному підсумку зменшити обсяг БД. Зв'язок кожної пари таблиць забезпечується при наявності в них однакових стовпців.

Існують наступні типи інформаційних зв'язків:

- один-до-одного;
- один-до-багатьох;
- багато – до - багатьох.

Зв'язок один-до-одного припускає, що одному атрибуту першої таблиці відповідає тільки один атрибут другої таблиці й навпаки.

Зв'язок один- до- багатьох припускає, що одному атрибуту першої таблиці відповідає кілька атрибутів другої таблиці.

Зв'язок багато - до - багатьох припускає, що одному атрибуту першої таблиці відповідає кілька атрибутів другої таблиці й навпаки.

7.4. СКБД Access та їхня структура

Таблиця — це об'єкт БД, який служить для уведення й зберігання інформації. Таблиця складається із записів (рядків), які містять інформацію, що зберігається в ній, і полів (стовпців), які утворюють структуру бази даних.

Для кожного поля задається ім'я поля, тип даних, перелік властивостей, опис.

Ім'я поля повинне бути унікальним, містити не більше 64 символів, включаючи пробіли, причому ім'я поля не повинне починатися із пробілу. Імена полів можуть містити будь-які символи, за винятком (.); (!); ('); (I); (J).

Тип даних визначає вид і діапазон значень, які можуть утримуватися в даному полі. Microsoft Access пропонує наступні типи даних.

- *Текстові* — тип даних, використовуваний для зберігання символьних або числових даних, не потребуючих обчислень. Поле даного типу може містити до 255 символів.
- *Числовий* — тип даних для зберігання числових значень, які використовуються у математичних розрахунках. Дані цього типу можуть утримуватися в 1, 2, 4, 8 байтах залежно від значення поля Розмір поля.
- *Поле Метод* — спеціальний тип даних для зберігання більших обсягів інформації, по обсязі перевищуючих 255 символів. Це поле може містити до 65535 символів.
- *Дата / час* — тип даних для зберігання календарних дат і поточного часу.
- *Грошовий тип* — тип даних для зберігання грошових значень, точність подання яких коливається від 1 до 4 знаків після коми. Ціла частина може містити до 15 десяткових знаків. Довжина поля 8 байт.
- *Лічильник* — спеціальний тип даних для унікальних натуральних чисел, що володіє властивістю автоматичного нарощування при кожному новому записі. Довжина поля 4 байти.
- *Логічний тип* — спеціальний тип даних, що має тільки два значення, які інтерпретуються як Так/Ні, Істина/Неправда, Вкл/Викл. Довжина поля 1 байт.
- *Поле об'єкта OLE* — містить посилання на OLE-об'єкт, наприклад документи Microsoft Word, Microsoft Excel, звук, зображення й інші види об'єктів. Обмежено дисковим простором.
- *Гіперпосилання* — спеціальне поле, призначене для зберігання гіперпосилання, за допомогою якого можна посилатися на довільний фрагмент даних усередині файлу або сторінки на тім же комп'ютері, у локальній мережі або Інтернеті. Довжина до 2048 символів.

При виборі типу даних для кожного поля відкривається відповідний набір властивостей, за допомогою якого користувач визначає вид і функціональні характеристики даного поля.

Найбільш важливими є наступні властивості.

- *Розмір поля* — визначає максимальну довжину даних, які можуть утримуватися в даному полі.
- *Формат поля* — визначає спосіб формування даних у даному полі таблиці.
- *Число десяткових знаків* — визначає кількість знаків після коми для полів числового й грошового типу.
- *Маска уведення* — визначає форму уведення даних у поле й полегшує контроль за вводять символами, що.
- *Підпис* — задає назви полів таблиці, які виводяться в різних режимах.
- *Значення за замовчуванням* — указує значення, вводи автоматично й поле при створенні нового запису.
- *Умова призначення* — обмеження, що накладає на дані, що вводять у дане поле.
- *Повідомлення про помилку* — це текст повідомлення, що буде виводитися у випадку порушення умови призначення.
- *Обов'язкове поле* — визначає, чи є уведення даних у це поле обов'язковим.
- *Індексоване поле* — визначає, чи є дане поле індексованим. Індекс прискорює всі операції, пов'язані із сортуванням, угрупованням або пошуком записів за значенням.

Бажано, щоб кожна таблиця мала ключове поле. Ключ однозначно визначає кожен запис у таблиці; повторювані значення ключа не допускаються. Зв'язуватися можуть тільки таблиці, що мають ключові поля.

Для визначення ключа виділяється, як правило, одне поле, і на панелі інструментів натискається кнопка Ключ із зображенням ключа (логотип Access). Однак у ситуації, коли дані одного поля не можуть бути унікальними для кожного запису, можна призначити ключовими два або більше поля.

Ключові поля таблиці індексуються автоматично.

При уведенні даних у полі здійснюється автоматична перевірка типу полів даних. Крім того, можна встановити додаткові правила перевірки даних, які вводять, у властивості поля Умова на значення, що дозволяє забезпечити уведення коректних даних у поле запису. Перевірка значення здійснюється при переході до іншого поля або іншого запису. Якщо уведене значення не відповідає заданій умові, то на екран виводиться повідомлення про помилку.

7.5. Операції з таблицями. Фільтрація даних.

Access дозволяє робити редагування полів і записів таблиці. Перейменування, додавання, видалення полів можливі як у режимі Конструктора, так й у режимі Таблиці. Уведення даних у таблицю і їхнє редагування ведеться в режимі Таблиці. Існують три операції із записами: додавання, видалення, зміна.

Дані, які зберігаються в таблицях бази даних, повинні мати прості й доступні засоби пошуку потрібної інформації. Для цього Access пропонує такі засоби, як сортування, фільтрація й пошук даних.

Сортування використовується для зручності знаходження даних. За замовчуванням, коли таблиця відкривається в режимі Таблиці, вона впорядкована за значенням ключового поля. Сортування записів можна робити як по одному, так і по декількох полях. Для цього потрібно встановити курсор на будь-який

рядок відповідного стовпця або виділити кілька стовпців і виконати команду Сортування по зростанню (зменшенню) меню Запису або скористатися відповідними кнопками інструментальної панелі.

Для того щоб повернутися до первісного порядку сортування, потрібно виконати команду Видалити фільтр меню Запису.

Пошук і заміна даних передбачені в режимі Таблиці. Вона дає можливість пошуку й заміни даних за зразком або в контекстному меню, або у всій таблиці. Для цього потрібно виконати команду Знайти меню Виправлення або клацнути мишею по кнопці Знайти на панелі інструментів. З'явиться діалогове вікно Пошук і Заміна, що має дві відповідні вкладки для пошуку й заміни даних. Потім потрібно ввести в поля відповідні значення й виконати зазначені дії.

Фільтрація — це спосіб швидкого відбору записів у режимі таблиці за заданим критерієм. Зробити відбір записів можна за допомогою наступних команд:

- Фільтр по виділеному,
- Звичайний фільтр;
- поле Фільтр для;
- Розширений фільтр.

Фільтр по виділеному фрагменту використовується для того, щоб знайти в таблиці значення, яке містить ті записи, які потрібно відібрати. Для цього варто виділити в полі запис, який містить умову відбору, і виконати команду Фільтр - Фільтр по виділеному меню Запису.

Звичайний фільтр використовується для відбору записів за значеннями декількох полів. Для виконання даного фільтра використовується команда Фільтр - Змінити фільтр меню Запису.

Поле Фільтр для використовується в тому випадку, коли не вдається швидко знайти зразок відбору в стовпцях таблиці й потрібно ввести конкретне шукане значення або вираження, результат якого буде використовуватися на полі, для якого потрібно вказати умови відбору. Викликається це поле щигликом правої миші на полі, для якого потрібно вказати умови відбору, і вибором з контекстного меню відповідної команди.

Розширений фільтр використовується для створення складних фільтрів, тому що дозволяє сформулювати критерій відбору у вікні команди Розширений фільтр, зв'язуючи умови відбору операціями логічних функцій «й» та «або». Для виконання даного фільтра використовується команда Фільтр - Розширений фільтр меню Запису.

Для того щоб зручно було працювати з усіма даними, які зберігаються в реляційних таблицях, між ними повинні бути встановлені зв'язки. Існують наступні правила для встановлення зв'язків між таблицями:

- таблиці повинні містити поля з однаковими записами;
- типи даних, довжини полів, а головне - значення обов'язково повинні збігатися.

Зв'язки між таблицями встановлюються на *Схемі даних*.

Схема даних будується відповідно до інформаційно-логічної моделі даних. При побудові схеми даних Access автоматично вибирає тип зв'язку по обраному полю таблиць.

Одна зі зв'язаних таблиць є головною (базовою), друга - підлеглою. Access дозволяє встановити зв'язки наступних типів:

- зв'язок один-до-одному, при якому одному запису з головної таблиці відповідає один запис із підлеглої таблиці;
- зв'язок один-до-багатьох, при якому одному запису з головної таблиці відповідає декілька записів з підлеглої таблиці.

Щоб викликати вікно *Схема даних*, необхідно виконати однойменну команду меню *Сервіс* або натиснути відповідну кнопку інструментальної панелі.

Додати таблицю у вікно *Схема даних* можна, якщо викликати правою кнопкою миші контекстне меню й вибрати пункт *Додати таблицю* або виконати команду *Зв'язок - Додати таблицю*, а також використовуючи відповідну кнопку інструментальної панелі.

У вікні *Схема даних* для установки зв'язку між таблицями варто перетягнути поле, по якому встановлюється зв'язок, з базової таблиці в поле підлеглої. У вікні *Зв'язку*, яке з'явилося, можна задати параметри для встановлюваного зв'язку:

- забезпечення цілісності зв'язку;
- каскадне відновлення полів;
- каскадне видалення зв'язаних полів.

Цілісність даних означає систему правил, які використовуються для підтримки зв'язків між записами у зв'язаних таблицях, а також забезпечують захист зв'язаних даних від випадкового видалення або зміни.

7.6. Запити до бази даних. Види запитів та технологія їх створення.

Запити — це об'єкт бази даних, який служить для витягу даних з таблиць і надання їх користувачеві в зручному вигляді.

Особливість запитів полягає в тому, що вони черпають дані з базових таблиць і створюють на їхній основі тимчасову таблицю. Застосування запитів дозволяє уникнути дублювання даних у таблицях і забезпечує максимальну гнучкість при пошуку й відображенні даних у базі даних.

Access дозволяє створювати запити двох типів: *QBE-запити*, *SQL-запити*.

QBE-запити (Query By Example) — запити, які будуються за допомогою конструктора запитів, що являє собою графічний інструмент для створення запитів за зразком.

SQL-запити — запити, які будуються за допомогою уніфікованого набору інструкцій SQL (Structured Query Language — структурована мова запитів).

Всі запити діляться на дві групи: *запити-вибірки*, *запити-дії*.

- *Запити-вибірки* здійснюють вибірку даних з таблиць відповідно до заданих умов. До цієї групи запитів ставляться наступні.
- *Запит до зв'язаних таблиць* — дозволяє робити вибірку даних зі зв'язаних таблиць.
- *Перехресний запит* — відображує підсумкові дані з угрупованням їх по горизонталі й вертикалі, виводячи результати їхньої обробки у вигляді таблиць.

- *Запит з параметром* — дозволяє користувачеві задати критерій відбору, увівши потрібний параметр при виклику запиту.
- *Запит з полем, яке обчислюється*, — дозволяє розрахувати дані на основі інших полів з того ж рядка запити.
- *Запит із критерієм пошуку* — дозволяє робити відбір записів відповідно до заданого критерію пошуку.
- *Запит із підсумками* — робить математичні обчислення й видає результат.
- Запити-дії дозволяють модифікувати дані в таблицях: видаляти, обновлювати, додавати записи. До цієї групи запитів ставляться наступні.
- Запити на створення таблиці створюють таблиці на підставі даних, які містяться в результуючій множині запити.
- Запити на додавання записів дозволяють додавати в таблицю записи, створені запитом.
- Запити на відновлення змінюють значення існуючих полів відповідно до заданого критерію.
- Запити на видалення видаляють записи з однієї або декількох таблиць одночасно.

В Access можна створювати запити за допомогою Майстра запитів і за допомогою Конструктора.

Для створення запитів за допомогою Майстра є наступні майстри:

- Майстер створення простого запити;
- Майстер створення перехресного запити;
- Майстер створення запитів на пошук повторюваних записів;
- Майстер створення запитів на пошук записів без підлеглих записів.

При створенні запити за допомогою Майстра робиться покрокове виконання дій у діалоговому вікні Майстра запитів.

Конструктор запитів дозволяє створювати нові й змінювати існуючі запити, тому він є основним способом при створенні запитів QBE.

При створенні запитів у режимі Конструктора відкривається спеціальний бланк, названий бланком запитів за зразком. Цей бланк складається із двох областей. У верхній частині відображується структура таблиць, до яких адресований запит, а нижня частина являє собою таблицю опису запити, що має 5 основних рядків. Кількість стовпців у запиті визначається кількістю обраних полів, які будуть використані в запиті.

Призначення рядків у бланку Конструктора запитів:

- 1-я рядок – Поле - призначене для уведення імені поля запити й позначення поля - джерела даних або вираження, яке буде обчислюватися;
- 2-я рядок - Ім'я таблиці - призначене для завдання імені таблиці - джерела даних;
- 3-я рядок – Сортування - призначене для вказівки порядку сортування в даному полі. Може приймати наступні значення: по зростанню, по убутанню, відсутній;
- 4-я рядок - Вивід на екран - містить прапорець, що вказує на те, чи буде виводитися відповідне значення поля;
- 5-я рядок - Умова відбору - містить критерій, по якому буде відбуватися відбір записів у результуючу таблицю.

Запити можуть бути створені на основі однієї або декількох таблиць. Багато-табличні запити дозволяють одержати інформацію з декількох попередньо зв'язаних між собою таблиць.

7.7. Створення форм. Їх види, властивості та структура.

Форма — це об'єкт БД, призначений для уведення й відображення інформації. Форми дозволяють виконати перевірку коректності даних при уведенні, проводити обчислення, забезпечують доступ до даних у зв'язаних таблицях за допомогою підлеглих форм.

Робота з формами може відбуватися в трьох режимах: у режимі Форми, у режимі Таблиці, у режимі Конструктора. Вибрати режим роботи можна за допомогою кнопки Вид панелі інструментів Конструктор форм або за допомогою команди меню Вид.

У режимах Форми й Таблиці можна здійснювати додавання, видалення й редагування записів у таблиці або в запиті, який є джерелом даних для форм.

У режимі Конструктора можна робити зміну зовнішнього вигляду форми, додавання й видалення елементів керування, розробку.

Види форм. В Access можна створити форми наступних видів:

- форма в стовпець або повноекранна форма;
- стрічкова форма;
- таблична форма;
- форма головна/підлегла;
- зведена таблиця;
- форма-діаграма.

Форма в стовпець являє собою сукупність певним чином розташованих полів уведення з відповідними їм мітками й елементами керування. Найчастіше ця форма використовується для уведення й редагування даних.

Стрічкова форма служить для відображення полів групи записів. Поля не обов'язково розташовуються у вигляді таблиці, однак для одного поля приділяється стовпець, а мітки поля розташовуються як заголовки стовпців.

Таблична форма відображує дані в режимі таблиці.

Форма головна/підлегла являє собою сукупність форми в стовпець і табличної. Її має сенс створювати при роботі зі зв'язаними таблицями, у яких установлений зв'язок типу один-до-багатьох.

Форма Зведена таблиця виконується майстром створення зведених таблиць Excel на основі таблиць і запитів Access (майстер зведених таблиць є об'єктом, впровадженим в Access, щоб використати його в Access, необхідно встановити Excel). Зведена таблиця являє собою перехресну таблицю даних, у якій підсумкові дані розташовуються на перетинанні рядків і стовпців з поточними значеннями параметрів.

Форма з діаграмою. В Access у форму можна вставити діаграму, створену Microsoft Graph. Graph є впровадженим OLE-додатком і може бути запущений з Access. Із впровадженою діаграмою можна працювати так само, як і з будь-яким об'єктом OLE.

Структури форми. Будь-яка форма може включати наступні розділи:

- заголовок форми - визначає верхню частину форми й може містити текст, графіку й інші елементи керування;
- верхній колонтитул - розділ відображується тільки в режимі попереднього перегляду й звичайно містить заголовки стовпців;
- область даних - визначає основну частину форми, що містить поля, отримані із джерела даних;
- нижній колонтитул - розділ відображується тільки в режимі попереднього перегляду в нижній частині екранної сторінки й звичайно містить номер сторінки, дату й т.д.;
- примітка форми - відображується унизу останньої екранної сторінки форми. Форма може містити усі розділи або деякі з них.

Як і будь-який об'єкт бази даних, форма має властивості. Значення цих властивостей для всієї форми, її розділів або елементів керування задаються у вікнах властивостей відповідних об'єктів. Для відображення на екрані вікна властивостей потрібно натиснути кнопку Властивості на панелі інструментів Конструктор форм.

Вікно властивостей виділеного об'єкта містить наступні вкладки:

- *Макет* — за допомогою цих властивостей задається макет форми;
- *Дані* — за допомогою цих властивостей задається джерело даних;
- *Події* — містить перелік властивостей, пов'язаних з об'єктом;
- *Інші* — перелік інших властивостей;
- *Усе* — перелік всіх властивостей.

Основні властивості форми:

- підпис - дозволяє задати назву форми, яка буде виводитися в області заголовка;
- режим за замовчуванням - визначає режим відкриття форми (простої, стрічкової, таблична форми);
- припустимі режими - властивість, що визначає, чи можна за допомогою команд меню Вид переходити з режиму форми в режим конструктора;
- властивості смуги прокручування, область виділення, кнопки переходу, розділові лінії, кнопка віконного меню, розмірів вікна, кнопка закриття, кнопка контекстної довідки, тип границі - визначають, чи будуть виводитися ці елементи у вікно форми;
- властивості дозволити додавання, дозволити видалення, дозволити зміни - визначають, чи можна користувачеві редагувати дані через форму. Ці властивості можуть приймати значення Так/Ні;
- уведення даних - визначає режим відкриття форми й приймає значення Так/Ні. Режим Так - відкриття форми тільки для додавання нових записів. Режим Ні - перегляд існуючих записів і додавання нових;
- блокування записів - визначають способи блокування записів у режимі роботи з базою даних багатьох користувачів.

Для створення форм в Access використовуються наступні види.

- Автоформа - автоматизований засіб для створення форм трьох стандартних типів: у стовпець, стрічкова, таблична. При цьому у форму вставляються всі поля джерела даних. ,
- Майстер форм - програмний засіб, що дозволяє створювати структуру одного із трьох стандартних типів форми в режимі діалогу з розроблювачем форми. При цьому у форму уставляються обрані користувачем поля із джерела даних.
- Конструктор форм - дозволяє конструювати форму користувачем у вікні конструктора форм.

Самим зручним способом створення нової форми є наступна технологія: форма створюється з використанням автоформи або майстром форм, а потім допрацьовується в режимі конструктора.

Джерелом дані форми є одна або кілька зв'язаних таблиць й/або запитів.

Елементом керування називають кожний об'єкт форми або звіту, який служить для виводу даних на екран, оформлення або виконання макрокоманд. Елементи керування можуть бути зв'язаними, такими, які обчислюють, або вільними.

Зв'язаний (приєднаний) елемент керування приєднаний до поля базової таблиці або запиту. При уведенні значення у зв'язаний елемент керування поле таблиці поточного запису автоматично обновлюється. Поле таблиці є джерелом даних зв'язаного елемента керування.

Елемент керування, який обчислюється, створюється на основі виражень. У вираженнях можуть використовуватися дані полів таблиці або запиту, дані іншого елемента керування форми або звіту й функції.

Вільні елементи керування призначені для виводу на екран даних, ліній, прямокутників і малюнків. Вільні елементи керування називають також змінними або змінними пам'яті.

Всі елементи керування можуть бути додані у форму або звіт за допомогою панелі інструментів елементів керування, яка з'являється при роботі з формою або звітом.

7.8. Звіти, їх створення, призначення та використання.

Звіт — це об'єкт бази даних, який призначений для виводу інформації з баз даних, насамперед на принтер. Звіти дозволяють вибрати з баз даних потрібну користувачеві інформацію, оформити її у вигляді документа, перед виводом на печатку переглянути на екрані. Джерелом даних для звіту може служити таблиця або запит. Крім даних, отриманих з таблиць, у звіті можуть відображатися поля, які обчислюються, наприклад, підсумкові суми.

Нижче наведені види звітів.

- *Одноколонний звіт (у стовпець)* — довгий стовпець тексту, що містить напису полів, їхнього значення із всіх записів таблиці або запиту.
- *Багатоколонний звіт* — створюється зі звіту в один стовпчик і дозволяє вивести дані звіту в трохи колонок.
- *Табличний звіт* — звіт, який має табличну форму.
- *Звіт з угрупованням даних і підведенням підсумків* — створюється з табличного звіту об'єднанням даних у групи з підрахунком підсумків.
- *Перехресний звіт* — будується на основі перехресних запитів і містить підсумкові дані.
- *Складений звіт* — звіт, який має складну структуру та включає один або кілька звітів.
- *Звіт, отриманий злиттям документів з Word* (складовий документ).
- *Поштові наліпки* — спеціальний тип багатоколонного звіту, призначений для печатки імен й адрес груп.
- *Звіт по зв'язаних таблицях* — звіт, який дозволяє виводити дані з декількох таблиць, що мають зв'язок один-до-багатьох.
- **Структура звіту.** Звіт може складатися з наступних розділів.
- **Заголовок звіту** - відображується тільки на першій сторінці й визначає висоту області заголовка звіту, що може містити текст, графіку й інші елементи керування.
- **Верхній колонтитул** - виводиться на верху кожної сторінки; як правило, містить заголовки стовпців.
- **Заголовок групи** - друкується перед першим записом, що входить у групу; звичайно містить підсумкові дані по записах, що входить у групу.
- **Область даних** - призначена для відображення записів джерела даних звіту.
- **Область примітки групи** - відображується після обробки останнього запису групи; звичайно містить підсумкові дані по записах, що входять у групу.

- Нижній колонтитул - виводиться внизу кожної сторінки; містить, наприклад, номер сторінки звіту, дату печатки звіту й т.д.
- Розділ приміток - відображується тільки внизу останньої сторінки; сюди поміщають поля з підсумковими значеннями по всіх записах, включеним у звіт. Звіт може бути створений за допомогою Майстра звітів або в режимі Конструктора. Звичайно використовують обидва способи. Майстер звітів дозволяє прискорити процес створення звіту, робота в ньому робиться в покроковому режимі в діалозі з користувачем. Доробити створений майстром звіт можна в режимі Конструктора.

Розділ Область даних може містити поля, які обчислюються, призначені для відображення у звітах значень виражень на основі вихідних даних.

У режимі Конструктора доступні такі властивості звіту, як Сортування й угруповання даних, які розташовані на панелі інструментів конструктора звітів і форм. У діалоговому вікні Сортування й угруповання можна визначити поле або вираження, по якому будуть згруповані дані, а також згруповані дані можна відсортувати по зростанню або убутанню. ACCESS дозволяє групувати дані двома способами - по категоріях і по діапазону значень, що може бути як числовим, так й алфавітним.

7.9. Макроси та їх конструювання

Макросом називається набір з однієї або більше макрокоманд, які виконують певні операції. Макроси використовуються для автоматизації часто виконуваних завдань (наприклад, відкриття форми, виконання запиту).

Виконання макросу можливо:

- з вікна бази даних: на вкладці Макроси слід натиснути кнопку Запуск;
- при виникненні певної події деякого об'єкта, наприклад, натискання кнопки. У цьому випадку макрос виконується автоматично.

Макрос є зручним засобом виконання простих завдань, таких як відкриття або закриття форм, вивід на екран і приховання панелей інструментів, запуск звітів і т.д.

Макросу привласнюється задане користувачем ім'я. Якщо макросу привласнити стандартне ім'я Autoexec, то такий макрос буде виконуватися автоматично відразу ж після завантаження Access.

Макрос може складатися з однієї або декількох макрокоманд. Процедура створення макросу в основному складається у виборі макрокоманд і визначенні їхніх параметрів. Для створення макросу треба:

1. У вікні бази даних на вкладці Макроси натиснути кнопку Створити.

2. На панелі макрокоманд в осередку стовпця Макрокоманда вибрати зі списку, який розкривається, ім'я макрокоманди.
3. В осередок стовпця Примітка ввести текст коментарю до макрокоманди (необов'язково).
4. У нижній половині вікна вказати аргументи макрокоманди.

Якщо макрос повинен складатися з декількох макрокоманд, варто перейти на наступний рядок і повторити вищевказані дії.

Можна використати технологію переміщення об'єктів для швидкого створення макросу. Для цього треба у вікні бази даних вибрати об'єкт і перемістити його за допомогою миші в осередок макрокоманди у вікні макросу. Наприклад, для того щоб створити макрос, що відкриває форму, варто перемістити за допомогою миші форму в осередок макрокоманди.

Наявні в Access-макрокоманди можна класифікувати за призначенням таким чином:

- макрокоманди переміщення за даними;
- макрокоманди виконання макросу, процедури або запиту;
- макрокоманди переривання виконання;
- макрокоманди експорту-імпорту;
- макрокоманди роботи з об'єктами;
- інші макрокоманди.

Макрокоманди виконуються в порядку їхнього розташування в бланку. Однак Access дозволяє задати умови виконання певних макрокоманд. Умова задається в стовпці Умова (для виводу стовпця слід виконати команду Умова меню Вид) і являє собою умовне вираження. Умовне вираження можна ввести із клавіатури або задати за допомогою будівника виражень. В осередках стовпця Умова неможна використовувати вираження .SQL.

При запуску макросу перевіряється значення кожного умовного вираження. Якщо це вираження істинно, виконується макрокоманда, яка перебуває в тому ж рядку, і всі наступні макрокоманди, у яких в осередках стовпця Умова втримується позначкою (...). Якщо умовне вираження є неправдою, Access проігнорує ці макрокоманди й перейде до найближчого рядка, у якому утримується нова умова або осередок умов є порожнім.

Access дозволяє поєднувати групу родинних макросів в один макрос. У цьому випадку макет макросу (вікно макросу в режимі конструктора) повинен містити стовпець Ім'я макросу. Щоб його вивести, слід виконати команду Імена

макросів меню Вид. Для запуску макросу із групи макросів слід вказати: *Ім'я Групи макросів. Ім'я Макросу.*

8. КОМП'ЮТЕРНА ГРАФІКА. ПРОГРАМИ CORELDRAW ТА AUTOCAD

Програма CorelDRAW має дивну універсальність і міць, будучи рівною мірою корисною й у промисловому дизайні, і в розробці рекламної продукції, і в підготовці публікацій, і в створенні зображень для веб-сторінок. CorelDRAW міцно втримує місце в ряду світових лідерів серед програм для роботи з векторною графікою, а по багатьом параметрам і перевершуючи всі інші, і в неї - величезна армія користувачів-професіоналів, які вважають CorelDRAW своїм основним робочим інструментом.

Система Auto CAD призначена в першу чергу для двовимірного малювання й випуску за її допомогою проектної документації всіляких галузей знань. Великою перевагою системи AutoCAD є можливість наступного формування електронного архіву креслень. Кожний зі створених файлів малюнків легко редагується, що дозволяє швидко одержувати аналоги по кресленнях-прототипах. Як засоби захисту від несанкціонованого доступу у файлах малюнків, створених системою AutoCAD 2006, передбачені паролі й електронні цифрові підписи. Крім цього, остаточні варіанти документів можна публікувати й передавати замовникам у форматах, які не редагуються, тобто DWF або PDF. Однак невірно було б вважати AutoCAD інструментом тільки для двовимірної роботи. Система дозволяє виконувати досить складні тривимірні побудови в будь-якій площині простору й відображувати їх на різних видових екранах з різних точок зору. Механізм простору аркуша й видових екранів надає можливість розробляти креслення з видами й проекціями тривимірних об'єктів, побудованих у просторі моделі. У системі AutoCAD по одній моделі можна одержати кілька аркушів креслярського документа.

8.1. Растрові й векторні зображення

Всі зображення, з якими працюють програми машинної графіки, розподіляються на два класи: *растрові* й *векторні*.

У термінології машинної графіки (галузі практичної інформатики, що займається автоматизацією побудови зображень й їхньої обробки за допомогою комп'ютерів) *растрові* або *крапкове* зображення прийнято називати масивом *пікселей* — однакових за розміром й формою плоских геометричних фігур (найчастіше — квадратів або кіл), розташованих у вузлах регулярної (тобто однакової форми і розміру) сітки. Для кожного пікселя тим або іншим способом задаються кольори (звичайно кольори кодуються числами фіксованої розрядності). Подання растрового зображення в пам'яті комп'ютера - це масив відомостей про кольори всіх пікселей, упорядкований певним чином (наприклад, по рядках, як у телевізійному зображенні).

Найбільш близьким аналогом такого явища віртуального світу, як піксельне зображення, у реальному світі є мозаїка. Піксельне зображення складається

ся з рівномірно розташованих на площині елементів однакового розміру й форми (пікселей), подібно мозаїці, що складається зі шматочків кольорового скла - смальти. При дотриманні певних умов (головні з них - невеликі розміри фрагментів смальти й досить велика відстань глядача від поверхні зображення) окремі шматочки смальти, що становлять мозаїчне зображення, не видні - око глядача сприймає зображення як єдине ціле.

Векторним зображенням у комп'ютерній графіці прийнято називати сукупність більш складних і різноманітних геометричних об'єктів. Номенклатура таких об'єктів може бути більш-менш широкою, але, зазвичай, до неї залучаються найпростіші геометричні фігури (кола, еліпси, прямокутники, багатокутники, відрізки прямих і дуги кривих ліній). Найважливіша особливість векторної графіки полягає в тому, що для кожного об'єкта (або, як ми будемо більш точно говорити далі, *класу геометричних об'єктів*) визначаються *керуючі параметри*, які конкретизують його зовнішній вигляд. Наприклад, для окружності такими керуючими параметрами є діаметр, кольори, тип і товщина лінії, а також кольори внутрішньої області.

8.2. Об'єктно-орієнтовний підхід в Corel Draw.

CorelDRAW являє собою *інтегрований об'єктно-орієнтований* пакет програм для роботи з *векторною графікою*. Спробуємо розібратися, що означають ці терміни.

Слова «інтегрований пакет» означають, що CorelDRAW X3 являє собою не окрему програму, орієнтовану на рішення будь-якої однієї чітко поставленої задачі, а сукупність програм (пакет), орієнтованих на рішення множини різних задач, які виникають при роботі користувача в певній прикладній області, а саме - в області ілюстративної графіки.

Інтегрованість пакета слід розуміти в тому так, що присутні в ньому програми можуть легко обмінюватися даними або послідовно виконувати різні дії над тими самими даними. Так досягається багатofункціональність пакету, можливості різних програм поєднуються, інтегруються в єдине ціле, яке являє собою щось більше, ніж сума складових частин.

Термін «*об'єктно-орієнтований*» варто розуміти так, що всі операції в процесі створення й зміни зображень користувач виконує не із зображенням у цілому й не з його дрібними, атомарними частками (пікселями зображення), а з *об'єктами* — семантично навантаженими елементами зображення. Почавши зі стандартних об'єктів (кіл, прямокутників, текстів і т.д.), користувач може будувати складені об'єкти (наприклад, значок у розглянутому вище прикладі) і маніпулювати ними як єдиним цілим. Таким чином, зображення стає ієрархічною структурою, на самому верху якої перебуває векторне зображення в цілому, а аж унизу - стандартні об'єкти.

Друга особливість об'єктної орієнтації пакета полягає в тому, що кожному стандартному класу об'єктів ставиться у відповідність унікальна сукупність *керуючих параметрів*, або *атрибутів класу*. Якщо ми говоримо про прямо косокутний висотою 200 мм і шириною 300 мм, залитий синіми кольорами, обведений жовтою лінією шириною 3 пункти, із центром, розташованим на відстані

150 мм по вертикалі й 250 мм по горизонталі від лівого нижнього кута сторінки з кутом нахилу довгої сторони до горизонталі, що становить 32° , ми маємо справу з екземпляром класу — об'єктом, для якого зафіксовані значення керуючих параметрів.

Третьою особливістю об'єктної орієнтації пакета є те, що для кожного стандартного класу об'єктів визначений перелік *стандартних операцій*. Наприклад, можна розвертати, масштабувати описаний вище прямокутник, закруглювати його кути, перетворювати його в об'єкт іншого класу - замкнуту криву[3].

8.3. Опис AutoCAD.

Програма AutoCAD призначена насамперед для створення креслень. Креслення, створені з її допомогою, повинні задовольняти стандартам, установленим з давніх давин для своєрідних видів креслярської продукції, які традиційно виконувалися вручну й найчастіше створюються так і понині. Капіталовкладення в технологію AutoCAD значно вище, ніж витрати на олівці й папір; та й цикл навчання більш тривалий: ви повинні опанувати й комп'ютер, і саму систему AutoCAD.

Нижче наведені вирішальні доводи на користь застосування AutoCAD замість олівця й паперу.

Точність. Якщо AutoCAD правильно настроєний, то малювати лінії, окружності й інші фігури з точними розмірами набагато простіше в AutoCAD, ніж на папері.

Простота внесення змін. Креслення набагато легше виправляти на екрані комп'ютера, чим на папері. У всякому разі після змін, зроблених в AutoCAD, на кресленні не залишається ворсу й пилу від ластику й плям від олівцевого графіту.

Продуктивність. За допомогою AutoCAD креслярські роботи виконуються значно швидше, ніж вручну, особливо в тих випадках, коли одне креслення містить кілька однакових фрагментів (наприклад, креслення багатопверхового будинку). З іншого боку, для виконання подібних маніпуляцій в AutoCAD необхідний відповідний рівень підготовки й деякий досвід роботи. Тому не дивуйтеся, якщо робота в AutoCAD піде не так швидко, як хотілося б, незважаючи на те, що ви досвідчений кресляр і досягли досконалості, орудуючи олівцем на папері.

AutoCAD стала колискою цілої індустрії програмно-технічних продуктів. Компанія Autodesk допомогла цьому процесу, поряд з безмірним числом інших досягнень, створивши ряд програмних інтерфейсів до AutoCAD, які стали використовуватися незалежними компаніями для розробки власних додатків. Одні із цих продуктів одержали визнання, інші перебувають у безвісності, але суть у тому, що Autodesk зірко стежить за можливостями розширення AutoCAD. Деякі додатки, створені незалежними компаніями, настільки гарні, що Autodesk здобуває їх, а потім випускає в складі чергової версії AutoCAD. Кожен новий програмний інтерфейс стимулює видання книг, появу навчальних курсів, розробку додаткових програм, навіть виникнення нових ідей і підходів. У підсумку, якщо порівнювати "закінчені продукти", тобто не тільки програмне ядро, таке як

AutoCAD, але й всі доповнення, розширення, навчальні курси, книги й т.д., то із системою AutoCAD просто дивно комусь змагатися.

ІНФОРМАЦІЙНО-МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

ДЖЕРЕЛА, ІНТЕРНЕТ – АДРЕСИ	
1	2
1	Прохорский Г. В. Информационные технологии в архитектуре и строительстве. Издательство: КноРус Описание: Учебное пособие – 2010.- 264с
2	Справочник по САПР / А.П. Будя, А.Е. Кононюк, Г.И. Куценко и др. Под ред. В.И. Скурихина. Киев: Техника, 1988
3	Електронний посібник по системі MathCad./ CD ROM
4	Норенков И.П. Основы автоматизированного проектирования- М.: МГТУ им. Н.Э.Баумана, - 1999. – 424с.
5	Електронний посібник по AUTOCAD/ CD ROM.
6	Бидасюк Ю. М. Mathsoft® MathCAD 12. Самоучитель, с ил.; 2005, М.-Диалектика.. -224 с
7	Курбатова Е. А. MATLAB 7. Самоучитель, с ил.; 2005, ; М. -Диалектика. - 256 с
8	Давид Фрей. AutoCad 2002 -2004 на примерах. К. : Юниор - 2004
9	Шуляк И. В. AutoCAD 2009 для проектировщиков, с ил.; 2008- Диалектика. - 960 с
10	Майкл Грох, Джозеф Стокман, Гэвин Пауэлл. Microsoft Office. Access 2007. Библия пользователя. М. – Диалектика-2007.- 768с.
11	Джонс Дж. К. Методы проектирования. Пер. с англ. 2-е изд., доп. М.: Мир, 1986.
12	Дьяконов В. П. Mathcad 11/12/13 в математике. Справочник. М: - Изд-во: Радио и связь. -2006. – 958с.
13	Жарков Н.В. AutoCAD 2008: официальная русская версия. Эффективный самоучитель. М. Наука и техника. 2008. - 592 с
14	Шевченко Н.А. Access 2003. Искусство создания базы данных. НТ Пресс. - 2005.- 160с.

Продовження табл.

15	Сайт АСAD: [Електрон. ресурс]. – Режим доступу: http://www.acad.du.ua
16	Сайт SAPR: [Електрон. ресурс]. – Режим доступу: http://www.sapr.ru
17	Цифровий репозиторій ХНАМГ // www.ksame.ua
18	Ковтанюк Ю.С. Corel Draw 12: офіційна російська версія. Керівництво користувача. Київ: «Мк-пресс», 2006.
19	Ткачев Д.А. AutoCAD 2007:самовчитель. Спб: «Питер», 2007
20	Миронов Д., Corel Draw X3: навчальний курс. Спб: «Питер», 2006
21	Зоммер В., AutoCAD 2008: Керівництво кресляра, конструктора, архітектора. М: «Біном», 2008
Методичне забезпечення	
1.	Гринчак М.В., Шаповалов А.Л "Інформаційні технології в будівництві і бази даних " Конспект лекцій. Електронний варіант, 2009. 64с.

НАВЧАЛЬНЕ ВИДАННЯ

ГРИНЧАК Микола Васильович
ШАПОВАЛОВ Анатолій Леонідович,
КУЗЬМИЧОВА Катерина Володимирівна

Конспект лекцій
з дисципліни
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В
БУДІВНИЦТВІ І БАЗИ ДАНИХ»
(для студентів 3 курсу денної і заочної форм навчання
напряму підготовки 6.060101 «Будівництво»
спеціальності «Промислове і цивільне будівництво»)

Відповідальна за випуск *К. В. Кузьмичова*

За авторською редакцією

Комп'ютерне верстання *К. А. Алексанян*

План 2010, поз. 146Л

Підп. до друку 05.05.10
Друк на різнографі.
Зам. №

Формат 60x84/16
Ум. друк. арк. 3,8
Тираж 50 пр.

Видавець і виготовлювач:
Харківська національна академія міського господарства,
вул. Революції, 12, Харків, 61002
Електронна адреса: rectorat@ksame.kharkov.ua
Свідоцтво суб'єкта видавничої справи:
ДК № 4064 від 12. 05. 2011 р.