ТРЕНІНГ-КУРС З ІНФОРМАЦІЙНОГО УПРАВЛІННЯ ПІДПРИЄМСТВАМИ ТА ПРОЄКТАМИ

Самостійна робота за змістовним модулем 2

2.1. Головні характеристики та властивості ППП

ППП – це програмний виріб, який є складовою прикладного програмного забезпечення. Поняття ППП використовується для визначення відособлених елементів ППЗ, поєднаних з метою більш ефективного використання ресурсів. За призначенням ППП – це комплекс взаємозв'язаних програм, що має спеціальну побудову, яка забезпечує значне підвищення продуктивності праці користувачів, що мають певну кваліфікацію, під час розв'язування певного класу задач.

Загальна характеристика ППП: пакет складається з певної кількості спільних програмних компонентів; призначений для розв'язування окремої задачі чи класу задач; забезпечує реалізацію вимог широкого кола користувачів до розв'язуваної задачі; документація, способи та засоби застосування пакета зорієнтовані на користувача, що має належний рівень кваліфікації у певній галузі діяльності.

За змістом ППП – це комплекс програм разом із документацією на його встановлення та експлуатацію, що має засоби настроювання і призначений для реалізації на ЕОМ певної кількості алгоритмів розв'язування певної задачі або класу задач і зорієнтований на певну категорію користувачів.

Головна властивість ППП – гнучкість, тобто можливість настроювання на певні особливості задачі, що розв'язується, та умови використання. Наявність засобів настроювання дає змогу застосовувати один раз розроблений ППП для розв'язування задачі або класу задач на різних об'єктах з урахуванням їхньої специфіки.

Ступінь гнучкості ППП визначається простотою та діапазоном настроювання пакета, а також якістю управління настроюванням і якістю управління процесом розв'язування задачі з використанням пакета. Простота та діапазон настроювання пов'язані з рівнем вимог до спеціальних знань користувача щодо застосування ЕОМ, принципів алгоритмізації та програмування під час застосування ППП, а також можливостями розв'язування задачі кількома методами чи способами, поданням вхідних даних та вихідної інформації у вигляді, що відповідає умовам користувача; застосуванням різних периферійних пристроїв. Необхідна якість управління настроюванням визначається кваліфікацією, досвідом та вмінням користувача. Для гнучкої системи характерне так зване адаптивне управління, тобто можливість пристосовуватися до змін зовнішнього та внутрішнього характеру. Отже, гнучка система має пристосовуватися до можливостей користувача: його рівня знань, вмінь, ступеня опанування системи. Якість управління процесом розв'язування задачі з використанням

пакета визначається можливістю оптимізації процесу обробки даних, тобто забезпеченням отримання якісної вихідної інформації з мінімальними витратами.

Рівень автоматизації програмування, що забезпечується пакетом, визначається ступенем участі користувача в організації процесів підготовки та виконання програм пакета.

Основним засобом спілкування між людиною та віртуальною машиною, що визначається ППП, є вхідна мова пакета. Тому і рівень автоматизації програмування, що забезпечується пакетом, визначається рівнем вхідної мови.

ППП – це особлива сукупність програм, здатна автоматично перебудовуватися на розв'язування нової задачі з певного класу; ППП перетворює систему користувача у віртуальну машину, яка в максимальній степені пристосована до розв'язання визначеного класу задач і можливостям користувача.

Ступінь гнучкості пакета значною мірою залежить від предметної сфери – множини задач, що розв'язуються з використанням пакета і мають сенс у тій сфері застосування, на яку його зорієнтовано. Будь-який процес, що реалізується ППП, можна віднести до однієї з груп: розв'язування функціональних задач, тобто задач, що забезпечують отримання нової інформації чи розширюють можливості операційної системи; розв'язування допоміжних, сервісних задач, які, як правило, призначені для полегшення використання пакета. Їхній зміст мало залежить від специфіки задач користувача. Поняття предметної сфери ППП тісно пов'язане з першою групою, оскільки кожний пакет зорієнтовано на певну галузь людської діяльності.

Але змістовний опис предметної сфери, зрозумілий і корисний користувачеві, не є достатнім для розробника пакета. Для нього важливі не тільки перелік даних та їхні характеристики, але й зв'язки між ними. Відтак, розробник ППП має побудувати певну модель предметної сфери. У ній є дані (змінні), що мають змістовну, зрозумілу користувачеві предметну назву, а також ідентифікатор, під яким ці дані фігурують у моделі предметної сфери. Сукупність властивостей (множина припустимих значень та набір припустимих операцій з ними) визначають тип даного.

Ім'я та тип є статичними атрибутами даного, а значення – динамічним. Між даними в моделі предметної сфери встановлюються зв'язки, або відношення, які визначаються змістом задач і утворюють постійну, незмінну структуру даних.

Сукупність даних у моделі предметної області становить інформаційну базу пакета.

Зв'язки даних, які встановлюють оброблювальні модулі пакета, реалізуються тільки за вказівкою користувача, після присвоєння (введення) значень окремим даним. Решта даних набувають своїх значень у результаті реалізації функцій модуля.

У [28] запропоновано модель предметної сфери (МПС), у якій об’єднано множину даних X, множину зв’язків між даними або структуру даних пакета R та множину функціональних зв’язків F, визначену модулями пакета:

$$\text{МПО} = \{X,\ R,\ F\}.$$

Для кожного пакета ці множини є скінченними.

Якщо протягом виконання пакета множини X, R, F лишаються незмінними (крім значень даних), то модель предметної сфери вважається статичною. Якщо користувач пакета має можливість протягом виконання пакета змінювати будь-яку множину з X, R, F (вилучати або додавати деякі її елементи),то модель предметної сфери називають динамічною.

Оскільки кожне дане характеризується трьома показниками (ім’я, тип, значення), то воно може належати до одного типу даних. Отже, множина X є об’єднанням підмножин однотипних даних, що не перетинаються:

$$X = \bigcup_{i=1}^{n} X_i \text{ ; якщо } i \neq j,\ X_i \cap X_j = \varnothing ,$$

де Xi – підмножина даних i-го типу; n – кількість типів даних;

$\varnothing$ – порожня множина.

Проте, в багатьох пакетах, особливо у ППП для розв’язування соціально-економічних задач, існують об’єднання даних різних типів у ієрархічні структури, причому кожна з них утворює окремий тип даних, або об’єднання даних одного типу в масиви. Ці об’єднання відіграють таку саму роль, що і скалярні дані, і так само утворюють підмножини даних.

Зв’язки між даними у таких об’єднаннях утворюють множину R у моделі предметної області пакета і встановлюються під час розробки його інформаційної бази. Дані, що не задовольняють умови зв’язків множини R, вважаються невизначеними, тому під час введення значень даних необхідно перевіряти їх на відповідність зв’язками і запобігати подальшим помилкам.

Функціональні зв’язки реалізуються в пакеті оброблювальними модулями, кожний з яких можна розглядати як деяку функцію fk, яка перетворює підмножину вхідних даних Xk (сфера визначення функції fk) у підмножину Yk (область значень функції fk):

$$Y_k = f_k(X_k),$$

де k = 1, …, m; m – кількість функцій (модулів) пакета.

Якщо функціональні зв’язки не змінюють значень вхідних даних, то

$$X_k \cap Y_k = \varnothing,$$

а якщо вони змінюють (повністю або частково) вхідні дані, то

$$X_k \cap Y_k \neq \varnothing.$$

Якщо модуль відбиває єдиний функціональний зв’язок між вхідними та вихідними даними, то він є підпрограмою без параметрів.

Якщо модуль, застосований до різних наборів вхідних даних, видає різні вихідні дані, тобто реалізує різні функціональні зв’язки, то він подається в пакеті у вигляді підпрограми з параметрами.

Якщо МПС побудовано правильно, то реалізація функціонального зв'язку з множини F не повинна руйнувати зв'язки між даними з множини R.

Стан обчислювального процесу і МПС змінюються під впливом визначальної інформації, яку задає користувач. Функціонування пакета відбивається на моделі предметної сфери як зміна стану моделі.

Нехай перед початком роботи з пакетом користувач установив початкові значення деяким змінним і модель опинилась у стані s0. Після виконання оброблювальних модулів f1, f2,…, fm модель пройде до станів s1, s2,…, sm . Можливі стани та переходи з одного стану до іншого можна подати у вигляді графа, вузли якого відповідають станам моделі, а дуги – модулям пакета, які забезпечують припустимі переходи.

Згідно з функціональним призначенням ППП – це функціонально завершений комплекс програм, зорієнтований на розв'язування певного логічно цілісного класу задач.

ППП можна класифікувати або за сферами застосування та класами розв'язуваних задач, або за орієнтацією на певний метод чи процедуру обробки.

2.2. Класифікація ППП за сферами застосування та класами задач

За сферами застосування та класами розв'язуваних задач пакети прикладних програм можна класифікувати так:

- ППП, що розширюють можливості операційних систем;
- ППП спеціального застосування;
- ППП для розв'язування інженерних і науково-технічних задач;
- ППП для розв'язування економічних задач; інтегровані ППП.

ППП, що розширюють можливості операційних систем, призначені забезпечувати нестандартні режими експлуатації обчислювальних засобів чи функціонування спеціальних технічних засобів. До них належать ППП, що забезпечують роботу багатокористувацьких систем; роботу з віддаленими абонентами; роботу в інтерактивному режимі; роботу в реальному масштабі часу; реалізацію спеціальної організації файлів і доступу у них до даних (системи керування базами даних).

ППП спеціального застосування призначені для розв'язування прикладних задач і реалізації математичних методів, які є спільними для багатьох сфер застосування ЕОМ. До цього типу пакетів можна віднести ППП, що реалізують методи теорії масового обслуговування, статистичні методи обробки та аналізу даних, методи математичного програмування; методи імітаційного моделювання, прогнозування та розпізнавання образів, ділових ігор, рекурсивні та ітераційні методи, методи обробки символьної інформації; текстові редактори, табличні процесо-ри і т. ін.

ППП для розв'язування інженерних і науково-технічних задач – це, як правило, набір підпрограм, на базі яких користувач може побудувати програми розв'язування власних задач. Найчастіше ці пакети

використовуються разом із процедурно-орієнтованими системами програмування.

ППП для розв'язування економічних задач мають певну специфіку, зумовлену особливістю та складністю структур соціально-економічної інформації, різницею у кваліфікації користувачів (що потребує використання відповідних інтерфейсів), різноманітністю задач і способів застосування результатів їх розв'язування, використанням генераторів звітів.

Інтегровані ППП – програмний продукт, до складу якого входять функціонально відмінні один від одного компоненти, що здатні взаємодіяти між собою передаванням інформації та поєднані єдиним уніфікованим інтерфейсом користувача. Основні компоненти інтегрованих систем: текстовий редактор, табличний процесор, система керування БД, графічний редактор, засоби обміну даними з віддаленими абонентами.

Інтегровані пакети можна розглядати як засіб для створення так званих автоматизованих робочих місць (АРМ).

Автоматизація управління виробництвом використовує концепцію розподіленої обробки економічної інформації, коли кожна з планово-управлінських функцій автоматизується за допомогою ПЕОМ, установленого безпосередньо на робочому місці відповідного спеціаліста. Пакети програм, що реалізують ці функції, разом з інформаційними ресурсами, технічними та організаційно-технологічними засобами утворюють автоматизоване робоче місце.

Серед ППП, що поширюють можливості ОС, особливе місце посідають системи керування базами даних.

Основні поняття баз даних і системи керування БД (СКБД)

Концепція БД грунтується на відокремленні процесів формування, введення та вибирання даних від процесу безпосереднього розв'язування проблемних задач.

Відмінність від підходу, пов'язаного з використанням процедурних мов програмування, існує у двох аспектах: 1) організація БД; 2) організація керування даними.

БД – це, по суті, пойменована сукупність даних, що відбиває стан об'єктів реального світу та відношень між ними в розглядуваній предметній сфері. Отже, БД є моделлю предметної сфери.

За структурою БД – це сукупність даних, організованих за певними правилами, що передбачають спільні принципи опису, збереження та маніпулювання даними незалежно від прикладних програм.

Атрибут (реквізит) – найменша одиниця даних.

Ключові атрибути (поля) – елементи, за якими можна впорядкувати певну сукупність даних.

Запис – одиниця обміну даними між програмою та зовнішньою пам'яттю.

Файл – пойменована сукупність усіх записів даного типу; найменша пойменована одиниця організації даних на зовнішньому носії.

Набір файлів – сукупність пов'язаних (тобто таких, що за певної умови та за визначеними ознаками можуть бути поєднані в одній групі) файлів.

Набір файлів утворює БД, якщо додержано певних вимог. Ідеться про те, що файли, поєднані в БД, мають бути:

1) інтегрованими – характеризуватися відсутністю або невеликою кількістю даних;

2) взаємозв'язаними – забезпечувати повну та несуперечливу інформацію про предметну сферу;

3) інформаційно-орієнтованими – містити такі відомості про об'єкти предметної сфери, в яких зацікавлений користувач;

4) незалежними – існувати незалежно від функцій, що їх вони підтримують.

Керування БД спрямоване на забезпечення розподілу даних, збереження їх незалежності, збільшення доступності, зменшення надмірності.

СКБД – це пакет прикладних програм, що містить сукупність мовних і програмних засобів, призначених для створення, ведення та спільного використання БД багатьма користувачами.

Основні вимоги до СКБД:

1) ненадмірність даних;

2) незалежність даних;

3) цілісність даних;

4) авторизація доступу.

Розглянемо кожну з цих вимог докладніше.

1. СКБД містять засоби, що дають змогу зменшити дублювання даних в інформаційній базі. Основним засобом досягнення ненадмірності є їх структуризація.

Зміст структуризації даних у базі: інформація, що є спільною для множини примірників даних, може бути розміщена в одному примірнику (запису) окремої групи даних.

2. У традиційних системах обробки даних, що базуються на процедурних мовах програмування, опис даних виконується безпосередньо у прикладних програмах, а тому будь-яка зміна опису даних призводить до необхідності модифікації прикладних програм. Під час використання СКБД описи даних зберігаються окремо, незалежно від прикладних програм. Завдяки цьому уможливлюється модифікація описів даних, після виконання якої раніше написані прикладні програми виконуються успішно, без будь-якої зміни.

3. Існують поняття фізичної та логічної цілісності даних.

Фізична цілісність – це захист інформації від ненавмисного руйнування та збоїв обладнання. Забезпечується, як правило, спеціальними програмними засобами.

Логічна цілісність – це захист даних від неузгоджених або помилкових дій користувача.

4. За умов авторизованого (санкціонованого) доступу до БД передбачається введення засобів захисту, що забезпечує обмеження доступу до інформації або заборону доступу до даних небажаних користувачів.

Організація даних

Моделі даних, на яких базується СУБД:

- ієрархічна (деревоподібна);
- сіткова;
- реляційна (Visual FoxPro).

Підвищення продуктивності прикладних програм, що працюють у середовищі СКБД, можна досягти за рахунок впорядкування даних та встановлення відношень між об'єктами БД.

Упорядкування даних можна здійснити:

1) сортуванням даних за ключем або ключами;

2) індексацією даних.

Реляційний підхід до побудови бази даних дає змогу усунути усі недоліки, притаманні ієрархічним і сітковим моделям даних, і забезпечує такі переваги:

1. Простоту. Використання двовимірних таблиць – найпростіший засіб роботи з базами даних для непідготовленого чи не дуже досвідченого користувача.

2. Гнучкість. Полягає в тому, що відношення можна «розрізати» чи «склеювати» за допомогою операцій проекції і об'єднання, що дозволяє користувачам діставати різноманітні файли в потрібній формі.

3. Точність. Напрямки зв'язку, які звичайно присутні у структурах баз, можуть бути пропущені. Відношення за своєю природою мають конкретніше значення і піддаються математично точним методам маніпулювання з використанням таких засобів, як алгебра відношень та обчислення відношень.

4. Реляційне зображення дає чітку картину взаємозв'язків реквізитів із різних відношень чи обчислень.

5. Реляційна база даних забезпечує незалежність даних, оскільки припускає можливість збільшення бази даних без зміни прикладних програм. Незалежність даних простіше забезпечити в нормалізованій базі, ніж у випадку ієрархічних – мережоподібних – структур. Реляційна база даних виглядає зрозумілішою, тоді як логічна схема бази даних під час зображення зв'язків залишається зрозумілою доти, доки кількість стрілок невелика. Зростання бази даних призводить до збільшення кількості зв'язків, що неможливо чітко відобразити на схемі з напрямленими зв'язками. А проте, використовуючи нормалізовані структури, можна задовольнити всі вимоги зростання і модифікації бази.

6. Таємність. Контроль таємності спрощується, оскільки для кожного відношення може бути задано правомірність доступу.

7. Простоту впровадження. Фізичне розміщення плоских файлів набагато простіше, ніж розміщення дерево- та мережоподібних структур.