

# ТРЕНІНГ-КУРС З ІНФОРМАЦІЙНОГО УПРАВЛІННЯ ПІДПРИЄМСТВАМИ ТА ПРОЄКТАМИ

## Самостійна робота за змістовним модулем 1

### 1.1. Призначення системи обробки даних та програмного забезпечення

Стрімке впровадження систем обробки даних і програмних засобів зумовлюється тим, що вони сприяють підвищенню продуктивності праці в усіх сферах людської діяльності завдяки використанню ЕОМ. Цю тезу покладено в основу визначення поняття «комп'ютеризація».

Комп'ютеризація – це процес розвитку індустрії комп'ютерних виробів і послуг та їх щонайширше використання в суспільстві: насичення виробництва, засобів транспорту й зв'язку, сфери управління, науки, освіти й побуту обчислювальною технікою.

Визначаючи перспективи комп'ютеризації, скористаємось загальною схемою актуальних напрямків розвитку систем обробки даних (СОД) (рис. 1.1).

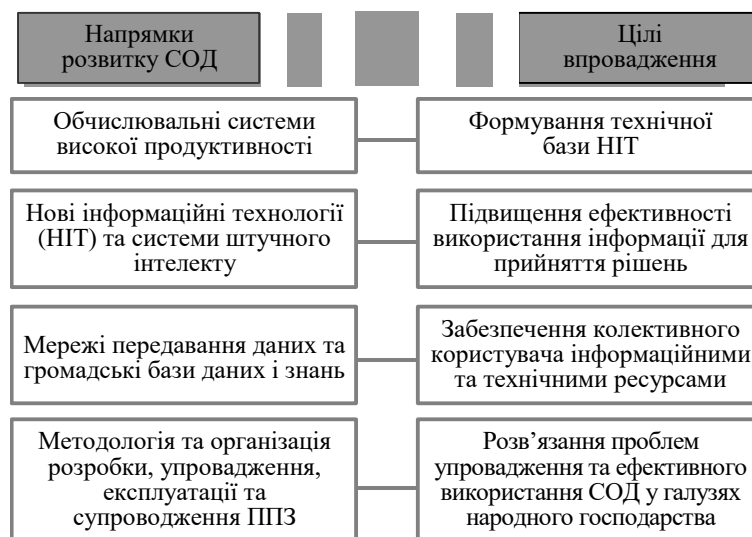


Рис. 1.1. Схема напрямків розвитку СОД

Призначення СОД – забезпечувати ввід даних, їх обробку за заданими алгоритмами та вивід результатів розв'язування у вигляді, що задовольняє потреби користувачів. Можливості інформаційного та технічного забезпечення не є предметом вивчення курсу.

Методи розробки програмного забезпечення – дисципліна, що ґрунтується на багатьох галузях знань. Так, для побудови алгоритмів вона використовує математичні методи, для оцінювання витрат і вибору ефективних рішень – методи інженерних і економічних розрахунків, для визначення вимог до системи обробки інформації, врахування різних втрат, організації праці виконавців, прогнозування – методи управління.

Можна виокремити такі функції ПЗ:

- ♦ керування технічними засобами, спрямоване на виконання всіх дій, потрібних для обробки даних;
- ♦ реалізація оптимального (з погляду кваліфікації користувача) рівня спілкування між людиною та обчислювальною системою.

## 1.2. Склад системи програмного забезпечення

Система програмного забезпечення (згідно з держстандартом) – це сукупність програм системи обробки даних і супровідних програмних документів, які дають змогу використовувати ЕОМ для розв’язування задач.

До складу системи програмного забезпечення належать: загальне (системне) ПЗ; прикладне (спеціальне) ПЗ [28] (рис. 1.4).

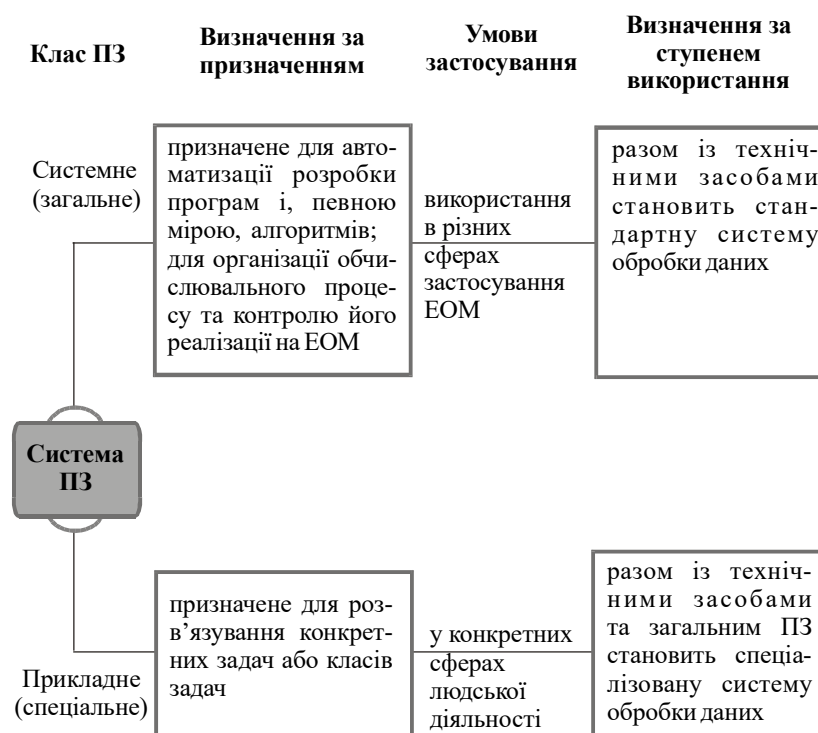


Рис. 1.2. Склад системи ПЗ

Зауважимо, що складові системного та прикладного ПЗ різні літературні джерела [22; 28] визначають майже однаково. Отже, урахувавши сучасний стан побудови та використання ПЗ зазначені складові можна подати схемою, зображеною на рис. 1.5.

### Характеристика складових загального ПЗ

Операційна система (ОС) – це комплекс керуючих програм, які забезпечують функціонування ЕОМ, включаючи планування та управління ресурсами ЕОМ, виконання завдань за запитами користувачів (виконання прикладних і обслуговувальних програм), управління вводом/виводом даних.

Основні ознаки, за якими різняться ОС:

- ♦ вимоги до процесора та пам’яті;
- ♦ вимоги до узгодженості обладнання;
- ♦ інтерфейс користувача (клавіатура; графічний інтерфейс);



Рис. 1.3. Складові системного та прикладного ПЗ

- ♦ програмна сумісність (будь-яка програма користується послугами ОС, а тому може працювати тільки під керуванням тієї ОС, яка забезпечує для неї ці послуги);
- ♦ режим обробки (одно- або багатозадачний: одночасно виконується кілька завдань, що, як правило, використовують різні ресурси ПК);
- ♦ рівень захисту даних (має лише Windows NT Workstation).

Програмні оболонки – це програми, що забезпечують найбільш зручний та наочний спосіб спілкування користувача з персональним комп'ютером під час виконання операцій керування файловою системою.

Виразним прикладом цього типу програм є файлові менеджери, які забезпечують виконання основних типових процедур маніпулювання файловою системою:

MS DOS – Norton Commander; PC Tools; PC Shell; Xtree Windows 3.1-Norton Desktop;

Windows 95 – Norton Navigator; Windows Commander.

До інструментальних систем належать засоби для розробки та налагодження програм, а саме: системи програмування та системи керування БД (СКБД).

Системи програмування містять мову програмування, транслятор, різні обслуговувальні програми для редагування текстів і налагодження програм.

Спинимось на різновидах систем програмування за ступенем алгоритмізації:

- процедурно-орієнтована – мова, що призначена для опису алгоритмів розв'язування задач у вигляді послідовності операторів, тобто для опису процедур розв'язування задач;
- проблемно-орієнтована – мова, за допомогою якої виконується опис постановки задачі (проблеми) та зазначаються вхідні дані. При цьому передбачається, що алгоритм розв'язування задачі буде побудовано

автоматично під час трансляції програми у послідовність машинних кодів (мова логічного програмування);

- об'єктно-орієнтована – система, що реалізує концепцію поєднання даних із програмами їх обробки (механізм приховування даних) та організації доступу до даних лише за допомогою певних інтерфейсів.

Різновиди систем програмування залежно від конструктивних особливостей ЕОМ:

машинонезалежна – мова, в якій не використовуються особливості конструкції конкретної ЕОМ, тобто її можна реалізувати на будь-якій моделі ЕОМ універсального призначення;

машинозалежна – мова, що призначена для реалізації на ЕОМ певного типу.

СКБД – це пакет прикладних програм (ППП), що містить сукупність мовних засобів і забезпечує централізоване зберігання та колективне використання БД.

Системні обслуговувальні програми призначені для виконання типових дій із носіями даних: ініціалізації дисків, перевірки дисків на наявність пошкоджених ділянок; оптимізації розміщення файлів на диску і т. ін. Системні обслуговувальні програми ще називаються зовнішніми командами операційної системи. Це, наприклад, програми форматування; резервування, динамічного ущільнення; оптимізації; обмеження доступу; архіватори.

Засоби контролю та діагностики призначено для перевірки справності окремих пристроїв машини та локалізації несправностей.

Програми окремого користувача – це програми для розв'язування специфічних задач певного користувача, тобто задач, які не виникають у інших користувачів і поки що не викликають у них зацікавленості; або це програми, в яких жорстко закладено особливості системи обробки даних конкретної організації, що унеможлиблює їх застосування в інших користувачів. Прикладом може бути програма утримання прибуткового податку, в якій програмно задано коефіцієнти для розрахунків (у разі зміни коефіцієнтів унаслідок інфляції програму не можна використовувати без додаткових зусиль програмістів). Інший приклад – програма оцінювання курсового проектування з дисципліни згідно з визначеними критеріями (доки таку саме методику не буде впроваджено для інших дисциплін).

Прикладні програми характеризуються більшою універсальністю, оскільки відповідають вимогам багатьох користувачів до тієї задачі, яку вони реалізують. Цим зумовлюється їх значне поширення. Якщо взяти, скажімо, ту саму програму утримання прибуткового податку, то незалежність від рівня коефіцієнтів за рахунок використання таблиці податків робить її універсальною для впровадження.

Пакети прикладних програм – це особлива форма прикладної програми. Як правило, це багатофункціональний комплекс взаємозв'язаних програм, який має певні правила побудови та мовні засоби керування. ППП є засобом підвищення праці не тільки кінцевих користувачів, а й розробників програмного забезпечення. Наприклад, пакет моделювання дискретних систем

має спеціальні мовні засоби описування моделі системи, робота якої аналізується за допомогою набору статистичних характеристик, що їх автоматично формує пакет.

### 1.3. Поняття програмного виробу

Суспільний поділ праці зумовлює перетворення сфери виробництва програмних засобів на складову суспільного матеріального виробництва.

Аналізуючи види програм, доходимо висновку, що програмні засоби мають властивості, притаманні товару (рис. 1.6), а отже, цілком правомірним є термін «програмний виріб», що означає програму на носії даних, яка є продуктом промислового виробництва [8].

Програмний виріб – це універсальне програмне забезпечення, призначене для широкого кола користувачів, яке має розроблятися, підтримуватися в роботоздатному стані й модифікуватися протягом тривалого часу.

Умови застосування програмних виробів висувають вимоги до виробництва, які суттєво відрізняються від процесу розробки програм для власного споживання (див. рис. 1.6). Уже недостатньо, щоб програма якось працювала. Ідеться про забезпечення якісно нового рівня зручності щодо керування виконанням програми, підготовки початкових даних, оформлення та обробки результатів. Потрібно так налагодити програму, щоб користувач не дуже часто стикався з помилками, і передбачити захист від хибних дій користувача; забезпечити захист програм і даних; так документувати програму, щоб користувач мав змогу без додаткових консультацій зрозуміти правила роботи з нею і відрізнити свої помилки від помилок програми або збоїв ЕОМ, а також виправляти власні помилки.

Існує кілька підходів до проектування програмного забезпечення, але найбільш поширеними нині є структурний, або функціонально-модульний, та об'єктно-орієнтований. Вони різняться способом декомпозиції складних програмних систем. У першому випадку структура системи описується як ієрархічна система інформаційно пов'язаних модулів, що реалізують її функції. У другому використовується об'єктна декомпозиція, коли статична структура системи описується в термінах об'єктів і зв'язків між ними, а її поведіння – у термінах обміну повідомленнями.

Практично всі поширені методи структурного підходу базуються на таких основних принципах [9]:

- «поділяй і владарюй»;
- ієрархічне впорядкування – організація системи у вигляді деревоподібної структури з додаванням нових деталей на кожному рівні;
- абстрагування – виокремлення істотних аспектів системи і відкидання неістотних;
- несуперечливість – узгодженість елементів системи;
- структурованість даних.

Структурний підхід використовує дві групи засобів, що описують функціональну структуру системи та відношення між даними, кожній з яких відповідають певні моделі:

- ◆ DFD – діаграма потоків даних;
- ◆ SADT (метод структурного аналізу та проектування) – моделі та функціональні діаграми;
- ◆ ERD – діаграми «сутність-зв'язок», найбільш популярні в CASE-засобах.

На кожному етапі життєвого циклу програмного забезпечення використовуються і відповідно інтерпретуються свої діаграми.

На етапах аналізу вимог до ПЗ і відповідного визначення специфікацій використовуються SADT-моделі.

Моделі ERD застосовуються для опису даних на концептуальному рівні, не залежному від засобів СКБД.

На етапі проектування, для опису структури проекрованої системи ПЗ використовуються моделі DFD і ERD, які можуть уточнюватись, доповнюватись новими конструкціями для подання даних на логічному рівні.

Отже, сутність структурного підходу до розробки ПЗ полягає в його декомпозиції на автоматизовані функції та підфункції, які, у свою чергу, можуть розбиватися на нові функції, і т.д.

Проте кожна підфункція може містити тільки ті елементи, які належать функції (підфункції) попереднього рівня – «батьківського», причому вона повинна мати всі елементи, що є на «батьківському» рівні, який разом з його інтерфейсами забезпечує контекст підфункції нижчого рівня. Нічого до неї не можна додати або з неї вилучити. У цьому полягає і головна вада структурного підходу: процеси (функції) і дані існують окремо в моделі програмної системи, проектування ведеться від процесів до даних, тобто структури даних перебувають на другому плані. Крім того, такий підхід застосовний тільки у проектуванні «згори донизу», тобто у спадному проектуванні.