

# РОЗДІЛ 1. ОСНОВНІ ПОНЯТТЯ ІНТЕРФЕЙСІВ КОРИСТУВАЧА ТА ЗАСОБИ ЇХ ПРОЕКТУВАННЯ

## 1.1 Інтерфейс користувача

### 1.1.1 Визначення та суть інтерфейсу користувача

*Інтерфейс користувача (ІК)* - це сукупність засобів, за допомогою яких користувач взаємодіє з різними пристроями (з комп'ютером або побутовою технікою) або іншим складним інструментарієм (системою). Інтерфейс користувача - це такий різновид інтерфейсів, в якому з одного боку - людина, з іншого - машина (пристрій, програмне забезпечення) [1]. За визначенням Національного банку стандартизованих науково-технічних термінів, інтерфейс користувача - це комплекс апаратних і програмних засобів, що забезпечує взаємодію користувача з комп'ютером [2, 3].

ІК часто розуміють лише як зовнішній вигляд програмного забезпечення (ПЗ), але таке розуміння є надто вузьким, оскільки саме за допомогою інтерфейсу користувач сприймає програму в цілому та використовує її функціональність. ІК забезпечує підтримку прийняття рішень у визначеній предметній галузі та визначає порядок використання ПЗ і документації до нього. В дійсності, ІК об'єднує усі елементи і компоненти ПЗ, які здатні впливати на взаємодію користувача з програмним забезпеченням [1]. До таких елементів належать: набір задач, які користувач розв'язує за допомогою ПЗ; використовувана програмним забезпеченням метафора (наприклад, "робочий стіл" у операційній системі Windows); елементи управління ПЗ; навігація між блоками ПЗ; візуальний (і не тільки) дизайн вікон та екранних форм програми та інші складові (рис.1.1.1).

*Стиль інтерфейсу користувача* - це набір ознак, методів, прийомів діяльності, які характеризують індивідуальність інтерфейсу користувача, а також сукупність прийомів використання інструментів розроблення ПЗ.



Рис.1.1.1 - Складові інтерфейсу користувача

Процес проектування ІК – це складний, нелінійний, недетермінований і неортогональний процес [2]. Складність ІК обумовлюється рядом невизначеностей, які суттєво впливають на процес розроблення. Нелінійність проектування ІК полягає у відсутності фіксованого, впорядкованого і прямолінійного алгоритму від початку до кінця проектування. Процес проектування є невизначеним, оскільки не існує рівняння, за яким можна було б одержати однаковий результат при заданих однакових початкових умовах, більш того, одержати ідентичний результат практично неможливо. Інтерфейс користувача неортогональний у тому сенсі, що будь-який аспект проектного рішення може впливати на інші аспекти, до того ж результат цього впливу не завжди є позитивним та прийнятним.

### **1.1.2 Проблематика проектування інтерфейсу користувача**

Процес проектування сучасного ПЗ передбачає вирішення ряду задач, зокрема: зниження витрат на проектування, скорочення термінів проектування, покращення якості запропонованих рішень, забезпечення нескладного в освоєнні та використанні ПЗ, вивчення та впровадження нових технологій та засобів, досягнення кращих результатів в порівнянні з конкурентами.

Задоволеність користувача програмним продуктом або зручністю його використання в значній мірі визначається інтерфейсом користувача. Взагалі, задоволеність користувача – це функція невеликої кількості факторів (фактор - це причина, рушійна сила будь-якого процесу, яка визначає його характер або його окремі риси [4]):

- МОЖЛИВОСТЕЙ ІНТЕРФЕЙСУ КОРИСТУВАЧА
- ЧАСУ ВІДГУКУ
- НАДІЙНОСТІ
- пристосованості до інсталяції,
- інформаційної підтримки,
- пристосованості до супроводження
- інших факторів [5].

Виділені прописними літерами фактори відіграють найбільш важливу роль. Можливості ІК повинні повністю відображати функціональні можливості програми. Час відгуку ІК повинен бути мінімальним, щоб користувачу не доводилось довше, ніж потрібно, очікувати виконання заданої дії. Надійність ІК - це властивість зберігати в часі у встановлених межах значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування. Пристосованість до інсталяції відіграє суттєву роль у задоволеності користувача, оскільки користувач починає своє знайомство з програмним засобом та його інтерфейсом саме з процесу інсталяції програмного засобу. Інформаційна підтримка користувача - це навчальна та довідкова складові ПЗ, від яких залежить, наскільки швидко і легко користувач опанує новий програмний продукт. Супроводження ІК - це процес покращення, оптимізації та

усунення недоліків ІК після передачі програмного забезпечення в експлуатацію. Пристосованість ІК до супроводження важлива можливістю покращення ІК вже за участі користувачів. До інших факторів відносяться узгодженість, інтегрованість та вартість ІК, які впливають на задоволеність користувача інтерфейсом, а відтак і програмним продуктом в цілому.

Усі фактори задоволеності користувача та їх відносну важливість слід враховувати під час кожного етапу життєвого циклу програмного забезпечення ІК.

До труднощів проектування ІК слід віднести і той факт, що користувач не завжди може чітко висловити свої вимоги та побажання щодо програмного продукту та його інтерфейсу на етапі проектування, але є дуже категоричним щодо бажаності і небажаності тих чи інших властивостей на етапі введення ПЗ в експлуатацію.

Часто характеристики ІК щодо практичності, інтеграції та узгодженості не формулюються явно на етапі проектування ПЗ, а визначаються на рівні деяких очікувань, що призводить до невірного розуміння очікувань замовника проектувальниками ПЗ. Тому такі вимоги повинні визначатись явно, причому бути вимірюваними (мати кількісні характеристики), оскільки проектувальник ІК може не бачити і не розуміти видимі та зрозумілі замовникам і користувачам вимоги.

### **1.1.3 Проектування інтерфейсу з орієнтацією на користувача (User-Centered Design)**

В минулому розроблення ІК розвивалось лише шляхом еволюції технологій та систем, на базі яких розроблялось ПЗ. Такий підхід називають системно-керованою або технологічно-керованою розробкою. Побажання користувачів абсолютно не враховувались, їм надавались програмні функції з інтерфейсом, який розробники були в стані розробити.

Однією з головних причин створення невдалого програмного забезпечення є недостатнє залучення користувачів до проекту. Не менше значення мають і наслідки недостатньо активної участі користувачів у проекті, зокрема відсутність знань про

фактичних користувачів продукту та середовище використання розробленого ПЗ. Детальна інформація про користувачів та їх середовище допомагає встановити рамки, в яких повинно здійснюватись проектування ІК та забезпечуватись його практичність. Поряд з вимогами до ІК та його практичності інформація про користувачів та їх середовище допомагає колективу розробників виділити ті особливості продукту, яких потребують користувачі, що є важливим для вибору відповідних методів розроблення ІК та підходів до проектування спільного стилю додатків.

З початку 80-х років при розробленні ПЗ акцент було перенесено на користувача, причому користувачі залучались до розроблення. Однак їм відводилась пасивна роль: у них з'ясовували, які вимоги вони висувають до ПЗ і які задачі вони вирішуватимуть за його допомогою. Зараз більшість розробників дотримуються методології, котру називають "розробленням із залученням користувачів" та "розробленням, орієнтованим на користувачів, що навчаються". Новизна підходу полягає в тому, що користувачів розглядають як активних учасників самого процесу розроблення. Залучення користувачів сприяє підвищенню доступності інтерфейсу та програмного засобу, а також служить гарантією, що одержане ПЗ буде відповідати запитам та вимогам.

Розроблення, орієнтоване на користувачів, що навчаються, спрямоване на те, щоб в процесі вирішення своїх задач людина навчалась новим навичкам роботи з ПЗ, тобто на її інтелектуальний розвиток, тренування її уяви і одержання знань в різних галузях.

Розроблення, орієнтоване на користувача, базується на наступних керівних принципах:

- 1) розуміння потреб користувачів (рис.1.1.2) є рушійною силою усього проекту;
- 2) все, що користувачі бачать і до чого вони мають доступ, повинно проектуватись сумісними зусиллями;
- 3) інноваційний проект завжди є результатом інтенсивної роботи команди спеціалістів з різних галузей;
- 4) рішення щодо ІК повинні базуватись на зворотньому зв'язку з користувачами;

- 5) результати зворотнього зв'язку з користувачами повинні збиратись з заданою точністю, швидкістю та частотою;
- 6) зворотній зв'язок здійснюється як з потенційними, так і з фактичними користувачами;
- 7) розроблення, орієнтоване на користувача, повинне стандартизуватись і впроваджуватись;
- 8) розроблення, орієнтоване на користувача, повинне постійно вдосконалюватись.



Рис.1.1.2 – Необхідна при проектуванні ІК інформація про користувачів  
 Методи, за допомогою яких відбувається залучення користувачів до проекту, наведені у таблиці 1.1.1.

Таблиця 1.1.1 - Методи залучення користувачів до проекту

<b>Метод</b>	<b>Опис методу</b>
Метод спостереження	Спостереження за фактичними користувачами, які виконують реальну роботу
Метод опитування	Опитування всіх учасників проекту, які виконують комплекс задач або входять в робоче середовище користувача
Метод індивідуальних інтерв'ю	Неформальні інтерв'ю з фактичними користувачами
Метод групового інтерв'ю	Формальні або неформальні інтерв'ю з колективом або групою фактичних користувачів

Базові принципи проектування, орієнтованого на користувача [2]:

1) розуміння користувачів та їх задач, залучення користувачів на різних етапах життєвого циклу ПЗ;

2) визначення цілей, які можна виміряти; встановлення критеріїв успіху з точки зору користувачів та замовників;

3) проект повинен передбачати нову компетентність користувача, яка по відношенню до продукту включає пакування, маркетинг, навчання, надруковану інформацію, налагодження параметрів, інсталяцію, екранні форми (екранна форма - графічно відображена форма (локація) в програмному забезпеченні; цифрове зображення, одержане програмним забезпеченням за командою користувача), графіку, довідки, іншу експлуатційну підтримку, оновлення та деінсталяцію;

4) оцінювання та тестування ПЗ за участю фактичних користувачів для визначення, чи досягнуті цілі та які є проблеми;

5) ітеративний підхід – якщо цілі не досягнуті або існують проблеми, слід внести виправлення та провести повторну перевірку.

Різні підходи до проектування ПЗ описані у таблиці 1.1.2.

Таблиця 1.1.2 - Підходи до проектування ПЗ

№	Підхід до проектування	Опис підходу
1	Проектування "зовні-всередину" (outside-in)	Спрямований на інтерфейс та доступні користувачу властивості програмного продукту
2	Проектування "зсередини-назовні" (inside-out)	Починається з розгляду внутрішніх властивостей системи
3	Однократне проектування (без ітерацій)	Проектування без встановлення планованого обсягу робіт з конструювання продукту та ІК
4	Багатократне (ітераційне) проектування	Концентрується на ітераційній побудові ІК та його основних факторах практичності
5	Проектування з використанням теорії "великого вибуху" (big bang theory)	Спроба розробити "все або нічого", тобто ПЗ проектується і реалізується паралельно
6	Еволюційне проектування	Зосереджене на побудові продукту з покроковим нарощуванням і уточненням можливостей продукту

Будь-який процес розроблення ІК повинен бути ітераційним, оскільки вдалий інтерфейс неможливо одержати без періодичного повернення до попередніх етапів. Критерієм для завершення ітераційної розробки є той факт, що усі вимоги користувачів задоволені, а сам продукт відповідає запланованим цілям. Може здаватись, що ітераційний процес займає багато часу через багаторазові проходження етапів розробки. Однак початкові проходження етапів допомагають створити варіанти розробок та прототипів, які в наступних ітераціях зекономлять час на впровадження та тестування.

Отже, на сьогодні найбільш ефективним підходом розроблення ІК, орієнтованого на користувача, є еволюційний ітераційний підхід "зовні-всередину". Етапи проектування ІК,

орієнтованого на користувача, наведені у таблиці 1.1.3.

Таблиця 1.1.3 - Етапи проектування ІК, орієнтованого на користувача

<b>Етап</b>	<b>Опис етапу</b>
<b>(1)</b>	<b>(2)</b>
План	План створення продукту орієнтований на побудову ІК та забезпечення практичності; план враховує календарні терміни для кожного з етапів процесу розроблення ІК, визначає основні ризики, об'єднує всі можливі методи, встановлює цілі та критерії оцінювання ІК
Вимоги	На етапі встановлення вимог вирішуються задачі: опис профіля користувачів, постановка задач користувачів, оцінка поточного рівня практичності, аналіз можливостей ІК, аналіз тенденцій розвитку ІК
Концептуальне проектування	Концептуальний проект є сукупністю високорівневих описів, абстракцій та оглядової інформації, яка дає розробникам та кінцевим користувачам загальне уявлення про програмний продукт, його структуру та ІК
Проектування	Проект ІК є сукупністю характеристик програми, які сприймаються користувачем (вхідні сигнали, взаємодія користувача, відгук системи на вхідні сигнали та взаємодію користувача)
Прототипування	Створення прототипів та моделювання – ефективні засоби ранньої оцінки проекту; прототип – це матеріалізація побудованого проекту з використанням його передбачуваної платформи реалізації, включаючи обладнання, ОС, мови і засоби реалізації

(1)	(2)
Специфікація	Матеріалізація проекту програмного продукту в документальній формі, яка описує дії користувачів, а також вигляд та поведінку ПЗ в специфічних ситуаціях
Конструювання	Написання коду та автономне тестування
Оцінювання	Всі методи оцінювання пов'язані із залученням потенційних користувачів програмного продукту
Ітеративний підхід	Загальні критерії досягнення цілей створення ІК повинні бути чітко визначені, зрозумілі й прийняті замовниками та розробниками; досягнення поставлених цілей може вимагати багатократних ітерацій
Впровадження та підтримка	На етапі впровадження здійснюється: оцінка ПЗ за участю користувачів, які не залучались до розробки, пілотне тестування, виконання задач, які не оцінювались або не були передбачені під час проектування та розроблення

Для створення стилів ІК та успішного створення ПЗ потрібен висококваліфікований персонал, який володіє широким набором різноманітних спеціальних навичок в таких галузях: проектування ІК; технологія розробки ПЗ та ІК; тестування та оцінювання якості ПЗ та ІК; стандартизація ПЗ та ІК; інструментальні засоби реалізації та використання ІК в додатках; проектування та реалізація візуальних та графічних конструкцій; проектування та реалізація засобів навчання, керівництв, систем допомоги та електронної експлуатаційної підтримки; психологія та ергономіка, людська поведінка, сприйняття, навчання та пізнання; бізнес-планування; управління проектами. Отже, для успішного проектування ІК потрібен колектив розробників, кожен з яких є професіоналом в певній галузі і володіє наступними якостями: вміння працювати в команді, причому працювати як з розробниками, так і з комерційними структурами; вміння розуміти

користувачів, уточнювати і деталізувати нечітко сформульовані вимоги до ІК; навички визначення і погодження з користувачем кількісної оцінки практичності, інтегрованості та узгодженості ІК; вміння використовувати засоби і методи розроблення ПЗ; навички використання засобів і методів тестування ПЗ інтерфейсу користувача, розрахунку надійності та якості ПЗ інтерфейсу користувача; навички швидко і ефективно оцінювати і відновлювати проект та реалізацію.

Проблеми організаційного, групового та індивідуального характеру, які виникають при проектуванні ІК, відрізняються набагато більшою складністю, ніж технічні проблеми. 80% проблем пов'язано з людьми і лише 20% носять технічний характер [5].

Значення різних професійних навичок та участь у проекті відповідної кількості спеціалістів, які володіють кожним типом навичок, потребує ретельного планування для досягнення успішності проекту. Правильне визначення ролі розробника в колективі розробників дозволяє усунути невизначеності. Один спеціаліст може виконувати одну або декілька ролей.

## 1.2 Еволюція інтерфейсів користувача

### 1.2.1 Інтерфейс командного рядка

Розвиток стилів та технологій ІК здійснюється паралельно еволюції операційних систем персональних комп'ютерів.

За кожним типом ІК розглянемо ряд властивостей (рис.1.2.1) [3]:

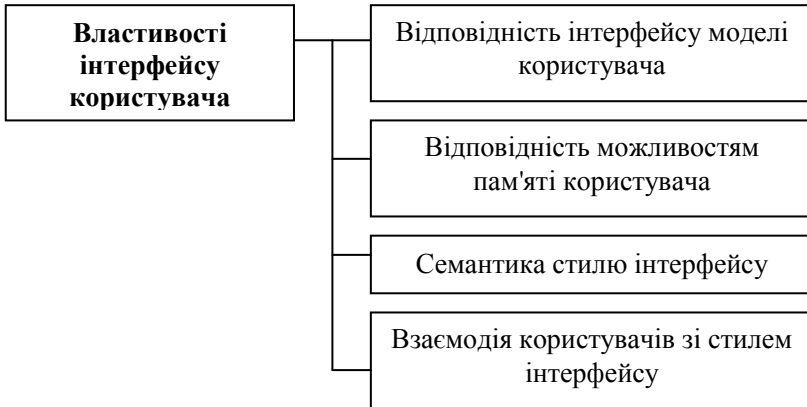


Рис.1.2.1 - Властивості інтерфейсу користувача

*Текстовий інтерфейс користувача* (TUI - Text User Interface або CUI - Character User Interface) - тип інтерфейсу користувача, який використовує при введенні-виведенні та представленні інформації виключно набір літерно-цифрових символів та символів псевдографіки [6]. Характеризується малими вимогами до ресурсів апаратури введення-виведення, пам'яті та високою швидкістю відображення інформації, тому широко використовувався на початковому етапі розвитку обчислювальної техніки. Його різновид - інтерфейс командного рядка - має окремі переваги в зручності перед графічним інтерфейсом. Тому програми, які базуються на текстовому інтерфейсі, мають деяке поширення до теперішнього часу, особливо в специфічних сферах та на обладнанні низької потужності.

### *Інтерфейс командного рядка (CLI - Command Line Interface)*

- це тип текстового інтерфейсу користувача, в якому інструкції комп'ютеру даються шляхом введення з клавіатури текстових рядків (команд), в UNIX-системах можливе застосування миші [7]. Інтерфейс командного рядка відомий також під назвою "консоль".

Інтерфейс командного рядка (ІКР) є оригінальним стилем взаємодії користувача з ЕОМ. Користувачі набирають запити або вказують дії формальною мовою, яка має власний словник, значення і синтаксис. Як правило, це набір команд операційної системи (ОС). Будь-яке ПЗ може використовувати ІКР для власних потреб. Однією з основних проблем ІКР є те, що він не захищає ні ОС, ні програму від помилкових та несанкціонованих дій користувача. Для роботи з ПЗ за допомогою ІКР користувачам потрібно знати, як працює комп'ютер і де знаходяться їхні дані.

Модель, застосовувана в ІКР, є моделлю програміста, а не користувача. Для багатьох користувачів засвоєння ІКР схоже на вивчення іноземної мови. Приклади ІКР наведені на рис.1.2.2: ліворуч показано ІКР, який використовується в операційній системі Windows для виконання команд MS DOS та запуску різних додатків; праворуч показано командний рядок MS DOS із запрошенням до введення команд.

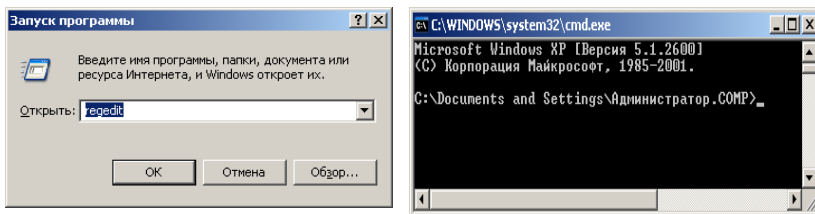


Рис.1.2.2 - Приклади інтерфейсу командного рядка

Проектувальники або професіонали в галузі програмування чи ОС здатні оцінити гнучкість та контрольованість ІКР.

У ІКР не дотримано одного з головних принципів розроблення ІК - зменшити навантаження на пам'ять користувача. Користувач повинен знати функції інтерфейсу та пам'ятати команди, які забезпечують доступ до цих функцій.

Стиль ІКР передбачає, що для введення команд користувачі використовують клавіатуру. Взаємодія користувачів з ІКР частіше передбачає операційно-об'єктно-орієнтований стиль синтаксису. Більшість команд являють собою дієслова або назви операцій, які користувачі планують виконувати. Спочатку користувачі набирають команду, далі назву об'єкта чи файла, над яким планують провести операцію. Подібний синтаксис команд забезпечує високоефективну взаємодію, хоча відсоток помилок при цьому також значний.

На рис.1.2.3 представлено ряд команд операційної системи MS DOS. Частина команд мають велику кількість опцій та параметрів, доступних лише досвідченим користувачам.

APPEND	EXE2BIN	PAUSE
ASSIGN	EXIT	PRINT
ATTRIB	EXPAND	PROMPT
BACKUP	FASTOPEN	QBASIC
BREAK	FC	RD
CALL	FDISK	RECOVER
CD	FIND	REM
CHCP	FOR	REN
CHDIR	FORMAT	RENAME
CHKDSK	GOTO	REPLACE
CLS	GRAFTABL	RESTORE
COMMAND	GRAPHICS	RMDIR
COMP	HELP	SET
COPY	IF	SETVER
CTTY	JOIN	SHARE
DATE	KEYB	SHIFT
DEBUG	LABEL	SORT
DEL	LH	SUBST
DIR	LOADFIX	SYS
DISKCOMP	LOADHIGH	TIME
DISKCOPY	MD	TREE
DOSKEY	MEM	TYPE
DOSSHELL	MIRROR	UNDELETE
ECHO	MKDIR	UNFORMAT
EDIT	MODE	VER
EDLIN	MORE	VERIFY
EMM386	NLSFUNC	VOL
ERASE	PATH	XCOPY

Рис.1.2.3 - Приклад команд MS DOS

У таблиці 1.2.1 представлено переваги та недоліки ІКР.

Таблиця 1.2.1 - Переваги та недоліки ІКР

<b>Переваги ІКР:</b>	<b>Недоліки ІКР:</b>
потужний і гнучкий вид взаємодії з ЕОМ для досвідчених користувачів	мала кількість або повна відсутність підказок та інструкцій
простота комбінування команд і параметрів	покращення функціональності інтерфейсу без додаткової документації є невидимим або невідомим для користувача
взаємодія, контрольована самими користувачами	потрібно роздруковане керівництво або електронна допомога
можливість використання скорочених назв команд	необхідне знання користувачами системи, програм і даних
використання мінімального обсягу машинної пам'яті	відсутній зворотній зв'язок та інформація про стан виконання задачі користувача
можливість використання разом з іншими ІК (наприклад, з графічним інтерфейсом)	необхідні навички набору тексту
	необхідно запам'ятовувати команди та синтаксис
	складність у вивченні
	відсутність змісту в назвах команд
	складність розуміння синтаксису команд
	неможливість налагоджування команд і синтаксису

## 1.2.2 Інтерфейс меню

*Меню* - інтерфейс користувача, який являє собою перелік можливих дій, відображуваних на екрані або у вікні для користувачів, з метою надання можливості вибору потрібних варіантів. Меню є важливою частиною ІК, вони дозволяють користувачам пересуватись програмним забезпеченням, надаючи можливість вибору потрібних пунктів або варіантів виконання операцій.

Меню бувають різними за формою, розмірами і стилем. У *повноекранних меню* (рис.1.2.4) доступні функції або задачі представлені у вигляді списків на екрані. Визначальною характеристикою більшості повноекранних меню є їх ієрархічна будова. При виконанні задачі користувачам необхідно рухатись по деревовидній структурі меню.

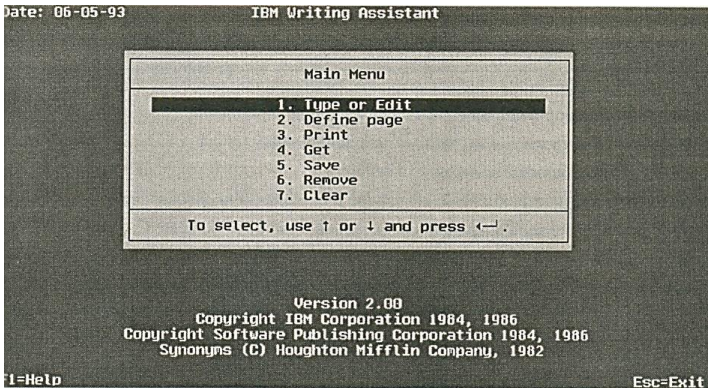


Рис.1.2.4 - Повноекранне меню

Більшість сучасних графічних програм використовують *панель меню*, розташовану в верхній частині екрану або вікна. Такий тип меню забезпечує швидкий доступ до різних функціональних можливостей ПЗ. Панель меню (рис.1.2.5) представляє собою динамічний список основного набору операцій або пунктів, які надають користувачу доступ до інших пунктів, представлених в окремому спадному меню. Панелі меню є невід'ємною частиною усіх основних графічних інтерфейсів

користувача і розроблених для цих інтерфейсів прикладних програм.

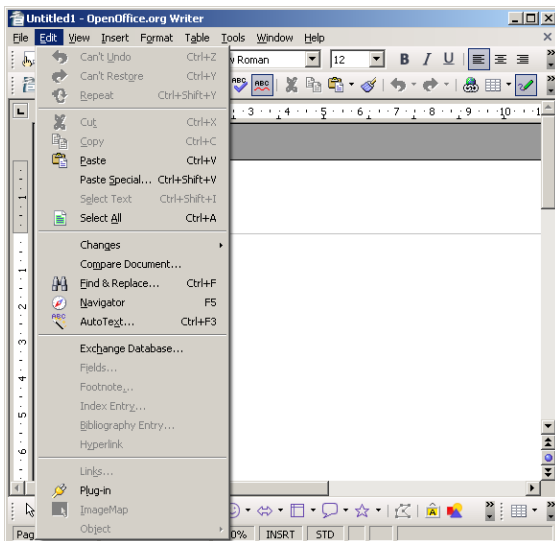


Рис.1.2.5 - Панель меню та підменю графічного ІК

Основною характеристикою меню є їх здатність динамічно змінюватись. Це забезпечує користувачів потрібними варіантами і маршрутами для виконання поточних задач або вибору потрібних об'єктів. Пункти меню можуть "затінюватись", якщо деякі операції на даний момент недоступні.

Іншими популярними типами меню є *панелі інструментів і палітри*. Панель інструментів (рис.1.2.6) представляє собою графічне меню операцій програми, інструментів і опцій, які користувачі можуть розташувати в будь-якому місці на екрані.

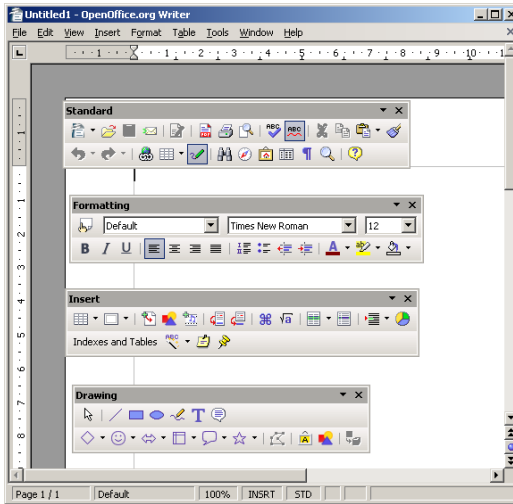


Рис.1.2.6 - Панелі інструментів текстового редактора Open Office Writer

Ще одним типом меню є *контекстне меню*. Його вміст залежить від контексту задач, які стоять перед користувачем. Такі меню можуть використовуватись для будь-якого типу додатків, як графічного, так і повноекранного інтерфейсу. Контекстні меню містять лише ті варіанти, які відповідають поточному або обраному пункту. Вони містять невеликий набір часто використовуваних дій, які можна знайти і на панелі меню. Так, наприклад, на рис.1.2.7 показано контекстне меню текстового редактора Open Office Writer застосовно до виділеного абзацу тексту.

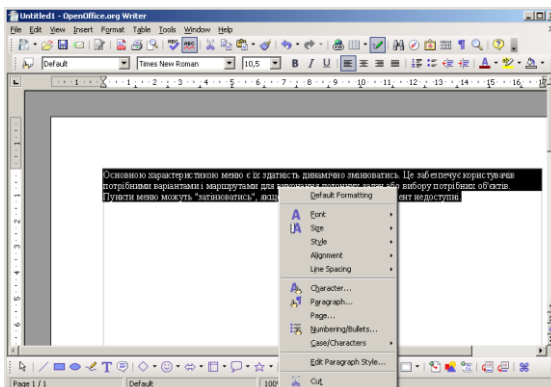


Рис.1.2.7 - Контекстне меню текстового редактора Open Office Writer

Переваги та недоліки меню як одного із засобів реалізації інтерфейсу користувача представлені в таблиці 1.2.2

Таблиця 1.2.2 - Переваги та недоліки меню

<b>Переваги</b>	<b>Недоліки</b>
користувачам не потрібно пам'ятати набір команд	невідповідність або неефективність для деяких задач
відмова від використання клавіатури зменшує кількість помилок	потрібна техніка навігації і вибору
навігація по ієрархії проста для нових користувачів	необов'язково робить інтерфейс зручним
скорочує час навчання користувача	змушує користувача пересуватись через декілька рівнів підменю
легко відстежувати відповіді і коригувати помилки	вимагає частого оновлення екрану
необхідний мінімальний обсяг друку керівництв користувача	користувач повинен розуміти принцип групування та ієрархії меню
може використовуватись сумісно з іншими типами інтерфейсів	користувач може "заблукати" в ієрархії меню
елементи меню та їх розташування можуть налагоджуватись користувачем	структура меню слідує за синтаксисом "об'єкт-дія"
гнучка система вибору команд (мнемоніки, миша, "гарячі" сполучення клавіш)	синтаксис меню повинен бути послідовним
підтримує тип пам'яті, орієнтований на впізнавання	використання режимів при роботі з меню та підменю відволікає увагу користувачів на стеження за системою

### 1.2.3 Графічний інтерфейс користувача

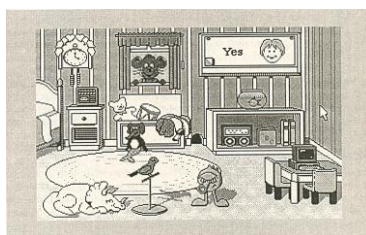
*Графічний інтерфейс користувача* (GUI - Graphical User Interface) - тип інтерфейсу користувача, в якому елементи інтерфейсу (меню, кнопки, значки, списки), представлені на екрані, виконані у вигляді графічних зображень [8].

На відміну від ІКР, в графічному інтерфейсі користувача (ГІК) забезпечено довільний доступ (за допомогою пристроїв введення) до всіх видимих екранних об'єктів (елементів інтерфейсу) та безпосереднє маніпулювання ними. Найчастіше елементи інтерфейсу в ГІК реалізовані на основі метафор та відображають свої призначення та властивості, що полегшує розуміння та засвоєння ПЗ недосвідченими користувачами.

Основною характеристикою графічного інтерфейсу користувача є використання ряду робочих елементів. ГІК (рис.1.2.8) є графічним представленням на екрані комп'ютера способу чи функції для інтерактивної взаємодії з програмами, об'єктами і даними. Він забезпечує користувачів необхідним інструментарієм і додатками, а не просто списком функцій.



а) Робочий стіл ОС Windows



б) середовище програми Playroom

Рис. 1.2.8 - Приклади графічного інтерфейсу користувача

*Характеристики ГІК [9]:*

- 1) підтримує ідею сумісності між програмами;
- 2) користувачі можуть бачити графічні зображення і текст на екрані в тому вигляді, в якому вони будуть роздруковані;
- 3) слідує концепції інтерактивної взаємодії "об'єкт-дія";
- 4) дозволяє переміщувати інформацію між програмами;
- 5) надає можливість безпосереднього маніпулювання об'єктами та інформацією на екрані;

6) пропонує стандартні елементи інтерфейсу (меню та діалогові вікна);

7) забезпечує візуальне відображення інформації і об'єктів (іконки і вікна);

8) забезпечує візуальний зворотній зв'язок в процесі виконання користувачами дій та задач;

9) надає візуальне відображення дій користувача/системи, а також режимів (меню, палітри);

10) використовує графічні керуючі елементи, які дозволяють користувачам робити вибір та вводити дані;

11) надає користувачам можливість налагодити та персоналізувати інтерфейс та інтерактивні дії;

12) забезпечує гнучкість переходу від клавіатури до іншого пристрою введення даних.

Під час досліджень зручності застосування ГІК було окреслено коло задач, які найбільш часто розв'язуються користувачами різних рівнів підготовки: запуск програми, виклик підказки, відкриття файлу, збереження файлу, копіювання файлу, зміна кольору робочого столу, пошук файлу, запуск ще одного додатку, видалення файлу/папки, перейменування файлу/папки, вибір принтера, створення папки, перегляд черги завдань принтера, розміщення документу на сервері, закриття програми, відхилення видалення/відновлення файлу, перевірка стану ресурсів системи.

Для роботи з ГІК користувач повинен мати знання про:

1) апаратну і програмну конфігурацію комп'ютерної системи;

2) графічні об'єкти та методи роботи з ними;

3) основні елементи вікон, щоб маніпулювати вікнами та інформацією, яка в них міститься;

4) керуючі елементи ГІК (панель меню, панель інструментів, радіокнопка, командна кнопка, палітра інструментів, прапорець, смуга прокрутки і т.д.);

5) способи маніпулювання мишею.

В ГІК користувачі спочатку обирають об'єкт, а тоді дію, яку необхідно виконати над даним об'єктом.

Одним з недоліків ГІК є його орієнтація на додатки. При роботі з ГІК користувачі бачать інформацію на екрані, в них складається враження, що вони дійсно працюють з об'єктами, однак увага користувачів сконцентрована на додатках. Вони обирають додаток, вказують файл даних, який використовується, організовують додатки і дані на комп'ютерах у вигляді графічних древовидних структур, використовуючи диски і папки.

ГІК передбачає проблемно-орієнтований підхід - перш ніж виконувати операції з файлами, користувачі повинні запуснути програмне забезпечення, яке працює з даним типом файлів. Користувачі працюють з додатками у вікнах, які мають панель меню з варіантами маршрутів, відображуваними спадними меню. Такі меню містять варіанти дій і маршрутів, пов'язаних з об'єктами у вікні або додатком, який надає саме вікно.

Будь-який вдало розроблений ІК повинен знижувати навантаження на пам'ять користувача. ГІК здатні запропонувати наочні підказки та необхідні відомості, використовуючи комп'ютерні потужності для зберігання та пошуку інформації.

ГІК об'єднують три основних стилі інтерфейсу: командний рядок, меню та пряме маніпулювання (рис.1.2.9).

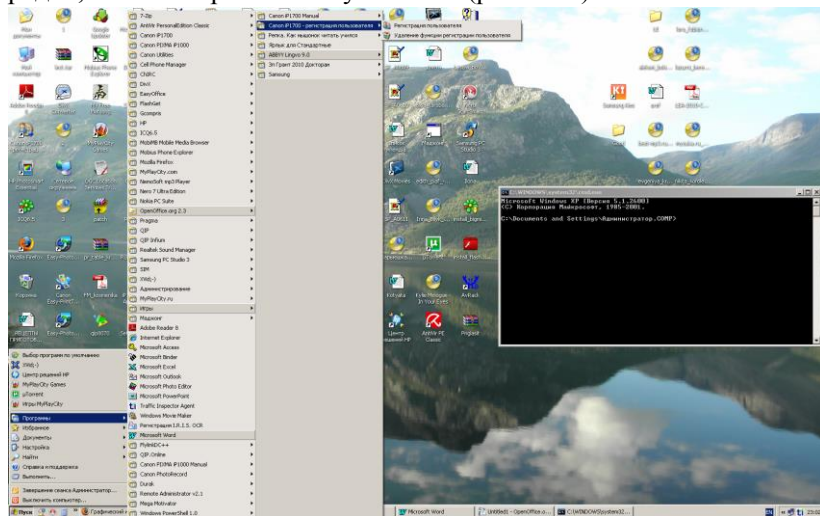


Рис.1.2.9 - Об'єднання командного рядка та меню в ГІК операційної системи Windows

Рядок статусу та поля інформації визначені в якості елементів вікна і використовуються для інформування користувачів про те, що відбувається всередині програми, тобто є візуальним зворотнім зв'язком.

Кожен керуючий елемент, іконка, колір ГІК повинні мати певне призначення. Семантичний зворотній зв'язок нагадує про змістовну характеристику об'єкта або виконання дії. Схожі елементи повинні поводитись аналогічним чином.

Методи інтерактивної взаємодії ГІК передбачають використання клавіатури або вказівних пристроїв для переміщення, вибору та маніпулювання інформацією на екрані.

До методів інтерактивної взаємодії ГІК належать [9]:

- 1) можливість використання клавіатури або миші в залежності від переваг користувача;
- 2) робота з різними типами меню;
- 3) два режими редагування: вставка або заміна;
- 4) технологія drag-and-drop;
- 5) буфер обміну.

Орієнтовані на користувача вдосконалення розробники будуть продовжувати впроваджувати в операційні системи та прикладне ПЗ по мірі зростання конкуренції серед розробників.

## 1.3 Стили інтерфейсу користувача. Моделі, методи розроблення та засоби проектування інтерфейсу користувача

### 1.3.1 Стили інтерфейсу користувача

На сьогодні існує ряд стилів ІК [5], які одержали поширення в індустрії програмних засобів (таблиця 1.3.1).

Таблиця 1.3.1 - Стили інтерфейсу користувача

Стиль ІК	Характеристики стилю
(1)	(2)
Графічний інтерфейс користувача (GUI – Graphical User Interface)	Стиль взаємодії “користувач-комп’ютер”, в якому застосовуються 4 елементи: вікна, піктограми, меню та покажчики. Іноді ГІК називають WIMP-інтерфейсом (Windows, Icons, Menus, Pointers). Найважливішими властивостями ГІК є можливість безпосереднього маніпулювання об’єктами, підтримка миші та інших вказівних пристроїв, використання графіки та наявність області для функцій і даних ПЗ. Кожному ПЗ властиві свої унікальні принципи побудови ГІК, причому всі додатки повинні бути витримані в цьому стилі. ГІК не гарантує більш високого рівня практичності. Належним чином спроектований ГІК-орієнтований додаток може перевершувати аналог з неграфічним інтерфейсом за рахунок підвищення ефективності роботи користувача та ступеня його задоволеності.
Web-інтерфейс користувача (WUI – Web User Interface)	Схожий на меню ієрархічної структури. Навігація виконується в рамках одного або декількох додатків з використанням текстових або візуальних гіперпосилань. Основні особливості WUI-додатків: інформація відображається в єдиному вікні (браузері), браузер забезпечує меню для Web-додатку; вибір дій обмежений; Web-сторінка має невисокий ступінь внутрішнього контролю над клієнтською областю для

(1)	(2)
	<p>відкриття спеціалізованих контекстних меню; створення спеціалізованих меню потребує додаткового програмування; функціональні можливості додатку повинні відображатись в методи для виклику команд; клієнтська область не містить традиційних піктограм; деякі додатки використовують графіку та анімацію в естетичних або навігаційних цілях, що несе в собі загрозу зовнішнього візуального шуму і збільшення часу відгуку при завантаженні та відкритті файлів; браузер і додатки забезпечують можливість відключення графіки; технологія drag-and-drop не підтримується, обмежені дії правої кнопки миші. До найбільш поширених компонентів WUI-інтерфейсів відносяться банери (заголовки), навігаційні панелі та візуальні або текстові гіперпосилання.</p>
<p>Інтерфейс користувача кишенькових персональних комп'ютерів (HUI - Hand-held User Interface)</p>	<p>Для введення даних застосовують “жестикаляційний” стиль з пером та сенсорним екраном. HUI-інтерфейс забезпечує деякі можливості ГІК, а саме: піктограми, меню і аналогічну поведінку курсора. У вікні пристрою одночасно відображається лише один об’єкт. Загальний стиль ІК для HUI-інтерфейсу називають SIMP-стилем (Screen, Icons, Menus, Pointers). Основні проблеми проектування HUI-інтерфейсу: спрощення вимог до користувача по введенню даних і взаємодії; використання обмеженої області дисплею.</p>
<p>Прикладний рівень ІК програмного забезпечення</p>	<p>Побудований на основі ІК програмний додаток в тій чи іншій мірі використовує компоненти стилю ІК або компоненти ІК прикладного рівня. Заснований на застосуванні ІК додаток здійснює виведення інформації та забезпечує взаємодію з користувачем за допомогою екрану та надає</p>

(1)	(2)
	<p>користувачу піктограми та курсор. Прикладний рівень додатку, заснованого на застосуванні ІК, містить всі функціональні можливості, які виходять за межі оформлення діалогових вікон та реалізовані із застосуванням різних стилів ІК. Можливості прикладного рівня включають наступні аспекти: концептуальне проектування; семантику об'єктів і операцій; наочне представлення та стиль поведінки об'єктів в межах клієнтської області; синтаксис об'єктів – формати даних, діапазони вхідних даних та послідовність зміни вікон; методи взаємодії та безпосереднього маніпулювання; унікальні дії; використання фізичиних пристроїв.</p>
<p>Об'єктно-орієнтовані ІК (ООІК)</p>	<p>Проектування ПЗ дає можливість надати користувачу інтерфейс, який має об'єктно-орієнтований стиль та/або об'єктно-орієнтовану внутрішню структуру (реалізацію). Більшість об'єктно-орієнтованих властивостей знаходять відображення в зовнішньому вигляді, поведінці, вимогах до взаємодії та функціональних можливостях ІК. Об'єктно-орієнтований прикладний ІК повинен мати наступні властивості: забезпечувати пряме маніпулювання, забезпечувати пряме введення даних, забезпечувати контекстну залежність інтерфейсу від об'єктів.</p>

### 1.3.2 Принципи побудови "дружнього" інтерфейсу користувача

Поняття "дружнього" інтерфейсу користувача (User-friendly Interface) розшифровується як "інтерактивний програмний інтерфейс, який забезпечує природний для користувача режим взаємодії з обчислювальною машиною" [10, 11]. Відтоді, як ІК став

одним з найважливіших елементів ПЗ, його якість та "дружність" хвилює кожного розробника і проектувальника.

Думка користувача про те, що деякий ІК якісний, тобто кращий за інші, залежить від невеликої кількості характеристик, які стають очевидними після знайомства з будь-яким додатком незалежно від стилю ІК. Якісний ІК повинен базуватись на 5 головних принципах, які позначають аббревіатурою **SAPCO** [9] (рис.1.3.1)

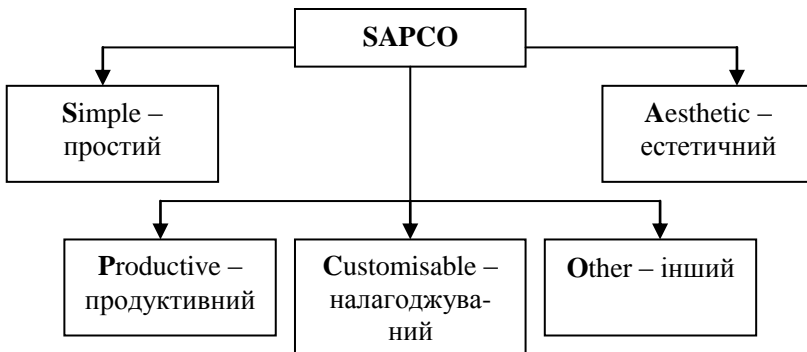


Рис.1.3.1 - Принципи SAPCO проектування якісного ІК

*Простий.* Програмні об'єкти забезпечують підвищення продуктивності і нарощення можливостей ПЗ без внесення додаткової складності в ІК. На початкових етапах розробки стилю ІК широко використовується мінімалізм і розбиття ПЗ на множину слабозв'язаних рівнів. Якісний прикладний ІК не потребує довідника або діалогової довідкової системи, щоб почати виконувати задачі за його допомогою. Якісний прикладний ІК зрозумілий на інтуїтивному рівні (користувачу потрібне лише пояснення, як досягти результату).

*Естетичний.* Програмні об'єкти повинні мати естетичну та ергономічну привабливість, при цьому широко використовується графічний дизайн та візуалізація. Якісний прикладний ІК повинен не викликати негативних емоцій у користувача, використовувати метафори реального світу користувача та максимально візуалізувати інформацію.

*Продуктивний.* Використання програмних продуктів вимагає мінімальних зусиль для вирішення задач користувача. ІК реалізується таким чином, щоб уникнути складної ієрархічності вікон та/або екранів, а також зайвих дій з клавіатурою та мишею.

*Налагоджуваний.* Програмне забезпечення доступне в різних формах для задоволення індивідуальних потреб. Програмні об'єкти ІК повинні мати можливість налагодження (налагоджуваність). Якісний прикладний ІК дозволяє користувачу обрати метод взаємодії і методи макетування та доступу для оптимізації потреб користувачів.

### **1.3.3 Моделі інтерфейсу користувача**

Основними моделями ІК є ментальна модель та моделі користувача, програміста, проектувальника [9].

*Ментальна, або концептуальна модель* - відповідає на питання "як користувач розуміє і взаємодіє з ПЗ". ІВМ константує: "Ментальна модель необов'язково точно відображає ситуацію та її компоненти на даний момент часу. Вона допомагає передбачити, що відбудеться далі, служить основою для розуміння, аналізу і прийняття рішень" [12].

Ментальні моделі дозволяють користувачам [9]:

- передбачити (або позначити невидимі) події;
- знайти причини позначених подій;
- визначити необхідні дії для здійснення потрібних змін;
- використовувати їх як мнемонічні пристрої для запам'ятовування подій та зв'язків (відношень);
- забезпечити розуміння аналогічних пристроїв.

В багатьох відношеннях концептуальна модель ІК - це фактично його архітектура. Послідовні рівні проектних рішень дозволяють нарощувати ступінь деталізації зовнішнього вигляду, поведінки, інформаційної підтримки та взаємодії користувачів на більш пізніх етапах проектування. Концептуальне проектування, яке стосується концептуальних моделей, стилю ІК та навігаційних структур, не обмежується деталями зовнішнього вигляду, поведінки, взаємодії і на функціональних можливостях продукту. Перший вірний крок важливий для збільшення шансів на

досягнення цілей проекту у відношенні ІК, практичності, узгодженості, інформаційної підтримки, часу відгуку, надійності та інших проектних факторів.

Альтернативні концептуальні рішення формуються, оцінюються і піддаються ітеративному перегляду, а також швидкому нарощенню при невеликих витратах - на концептуальному рівні. Однак помилка в концептуальних проектних рішеннях, не виявлена на ранніх проектних етапах розробки, передається на наступний етап і потенційно впливає на множини екранних форм, взаємодій, підтримку користувачів, програмний код, навчання, тестування, план-графік та вартість розробки. Вартість виправлення такої помилки буде тим вищою, чим довше помилка залишається невиявленою.

Концептуальна модель являє собою попередній ескіз, який показує основні функції ІК, візуальні, інформаційні та інші характеристики проекту продукту. Архітектура ІК - це розподіл функцій та програмний інтерфейс між ПЗ та користувачем.

На основі аналізу характеристик користувачів, вимог, задач та ділових потреб можна представити в явній і наочній формі основну ідею проекту, організаційні принципи та структуру, а також типові поведінкові моделі ПЗ по відношенню до візуалізації та взаємодії.

Склад концептуальної моделі представлений на рис.1.3.2.

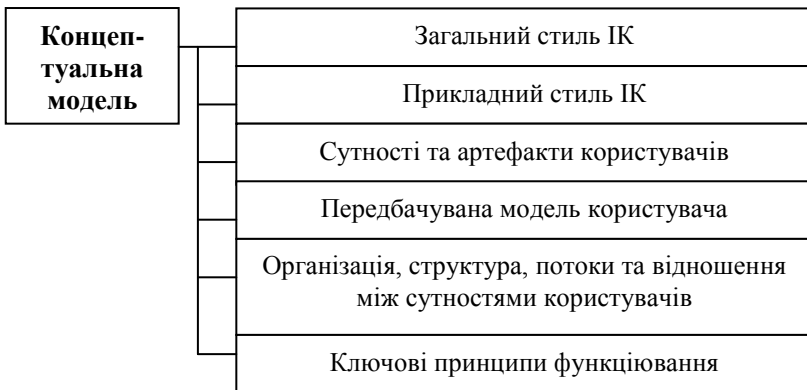


Рис.1.3.2 - Компоненти концептуальної моделі

Основна задача, пов'язана з концептуальним проектуванням, полягає в тому, щоб встановити, які функції покладаються на користувача, а які - на ПЗ. Результат цієї задачі визначається наступними аспектами:

- 1) задачі, які підлягають виконанню;
- 2) користувачі, які виконують задачі;
- 3) сильні сторони та обмеження можливостей користувачів;
- 4) сильні сторони та обмеження апаратного забезпечення, операційної системи, ПЗ, засобів ІК та стилів взаємодії, які належать задачі;
- 5) перспективні характеристики ПЗ, які покращують, нарощують або автоматизують окремі аспекти розв'язку задач користувачів;
- 6) перспективний шлях, який веде до досягнення заданих часових та вартісних обмежень;
- 7) перспективи виконання вимог по відношенню до ІК та практичності.

*Модель користувача* (МК) - сприйняття роботи програмного забезпечення користувачем. Це погляд користувача на події, які відбуваються в ПЗ. МК є ефективним способом вивчення та прогнозування поведінки ПЗ. Невдалий дизайн інтерфейсу викликає у користувачів сумніви в правильності своїх дій. Єдиний спосіб визначення вигляду МК – поспілкуватись з користувачем та подивитись, як він працює. Рекомендується 5 способів збирання інформації про користувачів:

- 1) аналіз їхніх задач;
- 2) інтерв'ю зі справжніми (фактичними) та потенційними користувачами;
- 3) відвідування місць їх роботи;
- 4) відгуки клієнтів;
- 5) тести на придатність ІК.

Спілкуватись потрібно безпосередньо з користувачами, а не з їх менеджерами та керівництвом.

Склад МК представлено на рис.1.3.3.

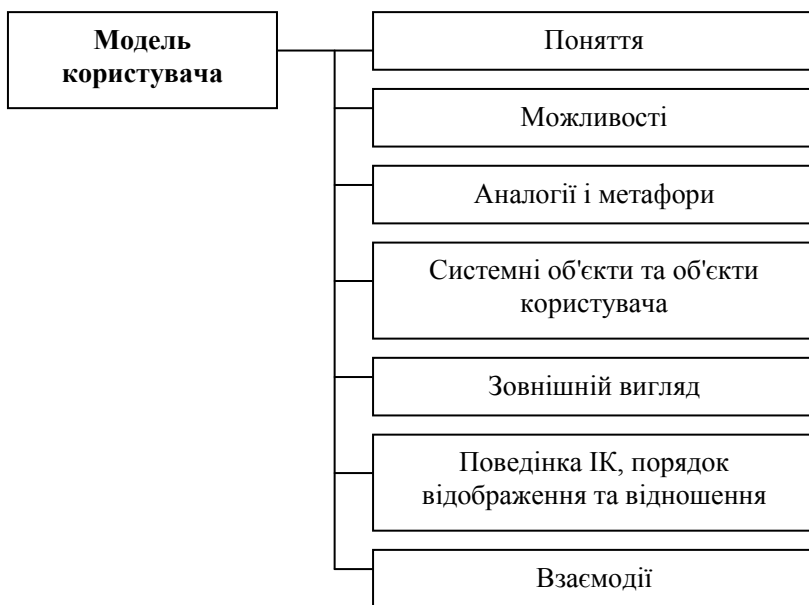


Рис.1.3.3 - Компоненти моделі користувача

*Модель програміста* – це функціональна специфікація програмного продукту. Вона легка для відображення, оскільки може бути формально і недвозначно описана.

*Модель проектувальника* – це дещо середнє між моделлю користувача та моделлю програміста. Проектувальник дізнається про ідеї, побажання користувача, поєднує їх зі своїми навичками і матеріалами та проектує ПЗ, яке повинно задовольняти потреби користувача. Модель проектувальника описує об'єкти користувача і техніку маніпулювання ними.

### **1.3.4 Методи, засоби та інструменти розроблення інтерфейсу користувача**

Недостатня участь користувачів в розробленні програмних продуктів є причиною підвищеного проектного ризику. Без участі користувачів колектив розробників не зможе одержати відповіді на наступні питання [5]:

- 1) хто є користувачами ПЗ?;
- 2) які задачі стоять перед користувачем?;
- 3) які вимоги або переваги користувачів?;
- 4) наскільки користувачі готові/не готові приймати участь в функціональному та інформаційному проектуванні та проектуванні ІК?;
- 5) якими будуть переваги користувачів у випадку неминучості компромісу?

Традиційними методами залучення користувачів до розроблення ПЗ є їх участь у проекті на етапах аналізу вимог та його оцінки. Однак існує ряд простих і ефективних методів залучення користувачів на більш ранніх етапах, а то й протягом усього життєвого циклу розроблення ПЗ.

*На етапі планування* користувачі можуть допомогти у вирішенні проблем: складання уявлення про кваліфікацію і навички співробітників та керівників; потреби у додатковому залученні користувачів до проекту; вказання додаткових джерел вхідної інформації; вказання вимог користувачів, даних і процедур; точка зору користувачів на пріоритети та вимоги до ПЗ.

*На етапі аналізу вимог* користувачі допомагають у: визначенні профілів користувача, ділового контексту, потоків задач, стереотипів робочої поведінки; виявленні найбільш часто виконуваних задач і функцій; виявленні проблем з використовуваними платформами та операційними системами; вдосконаленні та зміні бізнес-процесу; формуванні послідовності екранних форм та кроків діалогів; розташуванні функціональних кнопок та даних на екрані; виявленні переваг по відношенню до графічних представлень та термінології та впливів при керуванні змінами.

*На етапі проектування* користувачі можуть приймати участь у розробці таких проектних рішень: поведінка, інтеграція і узгодженість робочого середовища ІК; концептуальні моделі, термінологія і піктограми; послідовності екранних форм – кроки/вміст, графічний стиль, компоновання; всі деталі, що стосуються ІК, а також загальний досвід користувачів.

На етапі реалізації користувачі допомагають при вирішенні наступних задач: чіткість представлення графічних елементів ІК на пристроях відображення; кількість елементів в списках вибору; кількість елементів в списку, який повертає операція пошуку; початковий час відгуку при виклику зображення; непередбачувана поведінка програмного забезпечення; незвичайні формати друку.

Інструментарій розробника інтерфейсу користувача представлено в таблиці 1.3.2.

Таблиця 1.3.2 - Інструментарій розробника ІК

Інструмент	Характеристика інструменту
(1)	(2)
Графіка	<p><i>Графічне представлення інформації</i> надзвичайно важливе. Кожен рядок, керуючий елемент, блок, частина тексту, колір і рисунок, які з'являються на екрані, впливають на користувача як окремо, так і в комплексі з усім іншим.</p> <p>При роботі з <i>графічним представленням інформації</i> слід розглядати 2 фактори: графічна перевага — складається з комплексних ідей, зв'язаних з ясністю, точністю та ефективністю; графічна цілісність — представляє собою використання графічних засобів для точного відображення даних.</p>
Колір	<p>Колір є найменш використовуваним елементом при розробленні ІК. Колір повинен застосовуватись акуратно, адже люди мають різноманітні фізіологічні, психологічні, культурні та емоційні реакції на кольори. Часто колір використовується лише в якості декоративного елемента. Це обмежує його здатність відображати значущу інформацію в інтерфейсі. Один з основних принципів використання кольору — не нашкодити. Використання кольору в ІК може викликати труднощі у людей, які мають деякі проблеми із зором (кольорова сліпота — дальтонізм). Колір часто застосовується для якісного та кількісного</p>

(1)	(2)
	<p>виділення. Колір є засобом фокусування уваги: статистика показує, що колір на 32% ефективніший для фокусування уваги і на 25% - для виклику реакції користувача [13]. Операційні системи (ОС) пропонують стандартні кольорові схеми і палітри, які слід використовувати при розробленні інтерфейсу. Знання теорії сприйняття та особливостей зору призвело до створення важливих фундаментальних рекомендацій щодо використання кольору незалежно від засобу представлення інформації [14-16]. Дані рекомендації застосовні до використання кольору в друкованих матеріалах, а також на комп'ютерних дисплеях.</p>
<p>Звук та анімація</p>	<p>Питання використання звуку в якості зворотнього зв'язку є дискусійним. Більшість сучасного комп'ютерного ПЗ використовує звуковий зворотній зв'язок — короткі «біпи», які сигналізують про помилку чи невірно обраний варіант. Але навіть такі незначні звуки можуть набридати. Програмне забезпечення має бути контрольованим: користувач повинен мати можливість регулювати гучність або зовсім вимикати звук. Рекомендації щодо використання звуку, як і кольору, в більшому ступені засновані на особливостях людського сприйняття та здатності до навчання, ніж на особливостях ПЗ [17].</p> <p>Під <i>анімацією</i> розуміється зміна в часі візуального представлення графічного елемента. Наприклад, при переміщенні миші по різних областях екрану її вказівник змінює свою форму для відображення типу дії, яку можна виконати над об'єктом або в даній області. Іконки і вказівники з анімацією теж можуть бути корисними. Аналогічно звуку, головна перевага анімації полягає саме в розважальності дії. Курсор з анімацією спрощує його пошук на екрані,</p>

(1)	(2)
	особливо це важливо для невеликих комп'ютерних записників з малим екраном. Анімація може застосовуватись для виділення важливих іконок, відображення стану певного об'єкту і навіть пояснення його поведінки.
Термінологія та міжнародне проектування	Розробники та користувачі не завжди спілкуються однією мовою. Ось чому розробники повинні визначитись з термінологією і нести відповідальність за кожне слово, яке з'являється на екрані. Потрібно бути особливо уважними при організації інформації в багатомовних інтерфейсах.
Керуючі елементи інтерфейсу	Оптимальність керуючих елементів залежить від типу даних, кількості доступних даних, вигляду представлення іншої інформації на екрані, а також способу взаємодії користувачів з керуючими елементами екрану. Одним з ключових факторів, який слід враховувати при виборі керуючих елементів, є масштабованість керуючих елементів та ІК.
Пряме маніпулювання	Інтерфейси стають все більш об'єктно-орієнтованими, графічними і візуальними, тому перевага часто надається прямому маніпулюванню drag-and-drop. Проблема цього методу полягає в тому, що відсутня візуальна вказівка на те, що об'єкти можуть переміщуватись.
Розробка і схеми вікон	Компонування та розроблення елементів вікна — мистецтво і наука одночасно. Колір, шрифт, розмір, тип керуючих елементів, їх розмір, орієнтація, питання симетричності, виділення та багато інших факторів — все це впливає на кінцевий вигляд вікон [18].

## **1.4 Людський фактор, ергономіка, психологія та соціологія в проектуванні інтерфейсу користувача**

### **1.4.1 Людський фактор**

Під людським фактором розуміється вивчення людини та використання одержаної інформації для проектування засобів, задач і середовища ПЗ з метою забезпечення продуктивної та комфортної роботи [5].

Проектування ІК пов'язане з діяльністю двох категорій людей - розробників та користувачів. Розробники ІК, в свою чергу, теж є користувачами іншого прикладного ПЗ. Отже, існують можливості вдосконалити ІК як для розробників, так і для користувачів. Розробники можуть використовувати більш досконалі методи і засоби проектування програмного забезпечення ІК, а користувачі матимуть можливості одержати всі переваги готового ПЗ.

Характеристики комп'ютерних засобів введення, обробки і виведення інформації істотно відрізняються, але ще більш відмінності існують в характеристиках введення, обробки і виведення інформації користувачами. Розробники повинні розуміти, що користувачі мають як сильні сторони, так і обмеження і слабкості, пов'язані з віком, станом здоров'я, пам'яттю та іншими факторами. Отже, слід проектувати ІК, враховуючи слабкі сторони користувачів, лише тоді можна одержати ефективне ПЗ.

Недостатність уваги до людського фактору призводить до створення непродуктивного, важкого у вивченні та у використанні ПЗ.

### **1.4.2 Роль і місце ергономіки, психології та соціології при проектуванні інтерфейсу користувача**

Ергономіка вивчає взаємозв'язки між людиною та її роботою [5]. Цей термін використовується як синонім дослідження людського фактора.

Зручність роботи не надає такої користі бізнесу, як продуктивність роботи, але продуктивність підвищуватиметься, якщо користувачу буде зручно працювати. Для вимірювання

продуктивності роботи користувача використовуються метрики (наприклад, час, необхідний для виконання задачі без помилок). Вимірювання комфортності можливе лише в термінах рівня задоволеності користувача та його думок з приводу легкості використання ПЗ.

В контексті проектування ІК мета застосування ергономічних принципів полягає в підвищенні ефективності та задоволеності користувача інтерфейсом. Фактори, які належать до апаратного забезпечення, сконцентровані на фізичних та антропометричних характеристиках та особливостях людей. При проектуванні ПЗ слід брати до уваги ряд притаманних людині властивостей нефізичного характеру та враховувати психологічні характеристики людини. Ергономіка та людські фактори ПЗ призначені для опису результатів вивчення інформації про користувача та застосування цієї інформації при проектуванні ПЗ з метою підвищення ефективності та задоволеності користувачів.

Існує два основних принципи ергономіки щодо ПЗ [5]:

- 1) навчання і набуття практики людиною з метою пристосування до існуючого ПЗ;
- 2) проектування ПЗ, яке відповідає можливостям його користувачів.

Для проектування інтерфейсу, орієнтованого на користувача, використовується другий принцип ергономіки, тому що саме інтерфейс повинен налагоджуватись на потреби користувачів.

Незалежно від того, є користувач розробником ПЗ чи він працює з прикладним ПЗ, необхідно детально дослідити особливості візуального сприйняття та опрацювання ним інформації. На рис. 1.4.1 показана спрощена модель опрацювання інформації користувачем персонального комп'ютера (ПК) [5].

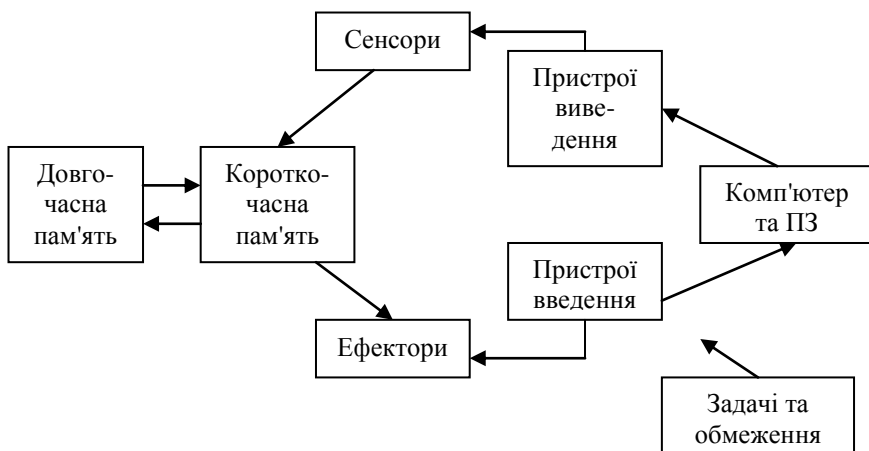


Рис.1.4.1 - Модель опрацювання інформації користувачем ПК

Основними характеристиками процесу людського пізнання, яким повинен приділяти увагу проєктувальник ПЗ, є:

- відчуття та сприйняття;
- навчання та запам'ятовування;
- увага та мова.

Користувач одержує інформацію шляхом зорового та слухового сприйняття. Зорова та слухова інформація, яка передається користувачу прикладним ПЗ, повинна знаходитись в діапазоні, що ефективно сприймається. З сенсорних буферів дані передаються в короткочасну пам'ять, де включаються процеси розпізнавання образів, спрямовані на усунення невизначеності, визначення контексту, ідентифікацію та інтерпретацію образів.

*Сенсорні буфери* в системі опрацювання інформації людиною тимчасово зберігають фіксовані образи, обсяг яких достатній для того, щоб користувач їх міг аналізувати. Хоча запам'ятовувана інформація відрізняється високим рівнем деталізації, вона зберігається протягом дуже короткого проміжку часу. Існує декілька типів сенсорної пам'яті – по одному для кожного з органів чуттів. Іконічна пам'ять використовується для зберігання зорової інформації, а звуконаслідувальна пам'ять

зберігає звукові дані. Образ представляє собою зоровий запис, який зберігає просторові відношення, а звуковіддзеркалення зберігає часові відношення. Інформація в кожному з буферів швидко руйнується.

Частина пам'яті людини, яка зберігає обмежений обсяг інформації, що належить до поточного часу, називається *короткотерміною пам'яттю*. Інформація, яка зберігається в пам'яті цього типу, відрізняється наступними особливостями:

- 1) запам'ятовується автоматично;
- 2) для відтворення потрібні дуже незначні зусилля;
- 3) обсяг порівняно невеликий ( $7 \pm 2$  елемента);
- 4) утримується повторенням;
- 5) відтворюється відносно повільно (в порівнянні зі швидкістю вхідних сигналів);
- 6) легко втрачається при відвертанні або недостатності уваги.

Інформація, яка належить до колишнього досвіду, міститься в *довготерміновій пам'яті* і відрізняється наступними особливостями:

- 1) запам'ятовування та відтворення вимагають деяких зусиль;
- 2) залежить від індивідуальної інтерпретації;
- 3) відрізняється величезним обсягом (до мільярда елементів);
- 4) організована ієрархічно;
- 5) відносно постійна у часі;
- 6) усвідомлюється швидше, ніж згадується.

Частина пам'яті людини, яка зберігає історію підходів до вирішення поточних проблем, зберігає тимчасові результати та відображає зміни в планах на майбутнє, називається *проміжною пам'яттю*.

Існують *методи обробки інформації*, спрямовані на навчання, увагу та прийняття рішень. *Навчання* – це процес, який приймає різні форми (запам'ятовування, засвоєння понять та правил, моторні навички для автоматичних процедур, що повторюються). Люди навчаються, спираючись на вже наявні в них

знання. *Увага* передбачає свідому обізнаність та опрацювання інформації, відрізняється винахідливістю, може бути сконцентрованою або розсередженою. Пошук інформації, оцінка альтернатив і результатів, вибір дій є процесом *прийняття рішень*.

Люди, відповідальні за проектування програмних продуктів, повинні бути уважними до фізичних вад користувачів. До 5% від загального числа усіх користувачів мають фізичні вади. Стосовно до підтримки користувачів, яким необхідно забезпечити їх особливі фізичні потреби, використовується термін *доступність*. Фактор доступності слід розуміти як можливість більш якісної підтримки користувачів і уваги до їх потреб.

До найбільш очевидних соціальних факторів, які слід враховувати при розробленні інтерфейсів, належить тиск, пов'язаний з наступними аспектами виконання роботи: прозорість у відношенні використання робочого часу; допущені промахи і помилки, їх наслідки; рівень підтримки з боку ПЗ групових задач, інформування про помилки; мотивація, конкуренція і т.і.

Соціальні фактори не слід ігнорувати, питання участі, комунікації, кооперації, інтересів та цілей користувачів необхідно обов'язково враховувати при проектуванні ІК.

*Рекомендації щодо людського, ергономічного, психологічного та соціального факторів для проектувальників інтерфейсу користувача:*

- 1) використовувати ергономічні стилі при проектуванні ІК;
- 2) пам'ятати про потреби та переваги користувачів;
- 3) орієнтуватись на знання, які вже мають користувачі, не перевантажувати їх надлишком інформації;
- 4) заохочувати користувачів до навчання;
- 5) не перевантажувати короткочасну пам'ять;
- 6) використовувати довгочасну пам'ять, котра базується на вже надбаному досвіді;
- 7) враховувати можливість помилкових дій користувача;
- 8) автоматизувати рутинні процедури;
- 9) підтримувати процес прийняття рішень;

10) використовувати параметри, прийняті в системі за замовчуванням, і відповідні засоби налагодження;

11) використовувати системні можливості забезпечення доступності.

## 1.5 Правила проектування інтерфейсу користувача

### 1.5.1 Правило 1: "дати контроль користувачу"

Провідний інженер з розроблення програмного забезпечення та інтерфейсів користувача компанії Cisco Systems Трейсі Леонард сказав: "Золоте правило проектувальника: "Ніколи не робити іншим того, що не подобається тобі". Згадайте, що вам не подобається в ПЗ, яким ви користуєтесь, і не робіть того ж в програмі, над якою працюєте" [9].

Однією з актуальних вимог до сучасного ПЗ є необхідність налаштування інтерфейсів до вимог та потреб користувачів. Кожен позитивний досвід спілкування користувача з ПЗ дозволяє йому підвищувати рівень компетентності та навички.

Основними принципами розробки інтерфейсу користувача є [9]:

- контроль інтерфейсу користувачем;
- зменшення навантаження на пам'ять користувача;
- послідовність інтерфейсу.

Хансен подає наступний список принципів проектування інтерфейсу [9]:

- знати користувача;
- скоротити об'єми інформації для запам'ятовування;
- оптимізувати операції;
- позбутися помилок.

Ці принципи застосовні до усього програмного забезпечення, в усіх типах та стилях інтерфейсів. Трагування цих принципів залежить від операційної системи, складових інтерфейсу користувача та його задач. Моделі користувача теж різні і впливають на те, яким чином будуть застосовуватись вказані принципи. На деяких етапах розробки проекту може постати питання: "Що відбудеться далі?". Відповідь повинна бути: "Те, чого захоче користувач".

Розглянемо основні правила проектування інтерфейсу користувача.

*Правило 1: "дати контроль користувачу".* Основне правило проектування ІК – дати користувачу контроль над системою. Проста аналогія – водій автомобіля та пасажир потягу. Рішення – їхати власним авто або потягом – повинен приймати саме пасажир (в нашому випадку користувач). Розробник інтерфейсу повинен виділити в якості основного такий принцип проектування, при якому продукт буде задовільняти всім вимогам. Досвідчені проектувальники дозволяють користувачам вирішувати деякі задачі на власний розсуд. Так, наприклад, європейські архітектори по завершенні будівництва складного комплексу будівель повинні прокласти між ними доріжки для пішоходів. На майданчиках між будинками ставлять таблички: “Ходіть, будь-ласка, по траві”. Через деякий час будівельники повертаються і лише тоді, згідно волевиявленню населення, асфальтують протоптані доріжки. Це прекрасний приклад надання можливості користувачу контролю над ситуацією і проектування інтерфейсу, який би відповідав потребам та побажанням користувача.

*Основними положеннями надання контролю користувачу над інтерфейсом є безрежимність, гнучкість, можливість переривання, корисність, поблажливість, можливість орієнтування, доступність, спрощення користування, пристосовуваність, інтерактивність.*

*Безрежимність* - вибір та використання режимів розважливо. Режими (певні умови роботи, діяльності) - атрибут багатьох програмних інтерфейсів, але застосовувати їх потрібно лише при необхідності.

Існує ряд інтерфейсів, які занадто часто перемикаються між режимами. На екрані з’являється діалогове вікно режиму, і користувач стає обмеженим в діях не лише в даній програмі, але й не має можливості переходу в інші програми.

Інтерфейс може працювати в двох режимах, які в багатьох випадках необхідні, однак позбавляють користувача самостійності:

1) режим користувача – обмежує режим роботи і дозволяє користувачу виконувати лише певні дії в певному місці програми. В наш час проектувальники інтерфейсів все частіше обходяться без

перемикання режимів. Можливо, оптимальним варіантом є показаних в форматі, який відповідає рівню доступу користувача;

2) системний режим – використовується достатньо рідко. Якщо система знаходиться в цьому режимі, йому дозволено працювати лише з поточною програмою. При розробленні повідомлень та інформації про допомогу в такому режимі слід пам'ятати про те, наскільки незручний системний режим для користувача.

При виборі режимів важливо слідувати принципу негайного візуального оберненого зв'язку. Користувач повинен бути постійно впевнений в тому, що він знаходиться в потрібному режимі. Режими інтерфейсу повинні бути настільки природними, щоб користувачу було комфортно працювати з ними.

*Гнучкість* - забезпечує користувачу можливість вибору, наприклад, можливість роботи з клавіатурою передбачає використання клавіатури замість миші. Панелі інструментів створені для прискорення роботи при використанні миші, однак при роботі з клавіатурою до них важко дістатись – для подібних випадків передбачені підменю. Більшість користувачів мають звичку працювати як з клавіатурою, так і з мишею.

*Можливість переривання* - дозвіл користувачу перемикає увагу між різними програмними засобами. Програмні інтерфейси повинні бути спроектовані так, щоб користувач міг в будь-яку хвилину перерватись або зберегти результати роботи. Забезпечення контролю користувача над програмою та його підтримка – ось головні принципи розроблення.

Не потрібно змушувати користувачів закінчувати виконання початих послідовностей дій. Вони повинні мати вибір – анулювати або зберегти дані і повернутись туди, де вони перервались.

*Корисність* - демонстрація повідомлень, які допомагатимуть користувачу в роботі. У інтерфейсі потрібно використовувати зрозумілі для користувача терміни. Даний принцип застосовний не лише для повідомлень, але й для усіх текстів на екрані: запрошень, інструкцій, заголовків, написів на кнопках. Усі текстові аспекти інтерфейсу повинні розроблятися

спеціалістами, котрі мають знання з орфографії та лексики. При написанні системної та програмної документації, а також повідомлень слід обрати вірний тон. Невдалі термінологія та тон призводять до того, що користувачі звинувачуватимуть себе у помилках, що виникають при використанні ПЗ.

*Поблажливість* - створення умов для негайних та зворотніх дій, а також зворотнього зв'язку. Недостатність зворотнього зв'язку в більшості програмних продуктів змушує користувача витратити багато сил на виконання поставленої задачі. Наприклад, у інтерфейсі командного рядка в MS DOS файли видаляються командою DEL. Виконання цієї команди потрібно перевіряти командою DIR, тобто зворотнього зв'язку з командою DEL користувач не має.

Кожен програмний продукт повинен мати функції відхилення (скасування) та повторення дії. Необхідно інформувати користувача, якщо дана дія не може бути відхилена, і, по можливості, дозволити йому альтернативну дію.

*Можливість орієнтування* - забезпечення доступу користувачу до будь-якої частини інтерфейсу. Користувач повинен мати можливість вільно орієнтуватись в інтерфейсі, мати доступ до будь-якої його частини, можливість пересуватись вперед і назад по нисхідній та висхідній структурі інтерфейсу, отримувати зручні контекстні підказки там, де вони потрібні. Наприклад, панель задач Windows показує, які програми виконуються, і надає користувачу доступ до них.

Панелі інструментів, меню, палітри та інші елементи інтерфейсу операційних систем, набори програмних продуктів та різні утиліти - все це розроблено, щоб допомогти користувачам орієнтуватись в операційній системі та в просторі жорсткого диску. Системні та програмні "помічники" та "асистенти" теж пропонують допомогу для орієнтації в програмних функціях або задачах.

*Доступність* - налагодження системи на користувачів з різним рівнем підготовки. Деякі програми пропонують спеціальні інтерфейси, які допомагають обрати рівень складності взаємодії. Наприклад, панель задач та підменю програм можуть бути

стандартними або деталізованими в залежності від вибору користувача та типу виконуваної задачі. Для досвідчених користувачів потрібно передбачити швидкий доступ до функцій ПЗ.

*Спрощення користування* - створення зрозумілого, "дружнього" ІК. ІК – "міфічна" частина програмного продукту. При вдалому проектуванні користувачі не помічають інтерфейсу. Якщо інтерфейс розроблений невдало, користувачам доведеться докласти чимало зусиль для ефективного використання програмного продукту. "Прозорість" інтерфейсу забезпечується тим, що користувачу надається можливість користуватись об'єктами, відмінними від системних команд.

*Приспосовуваність* - надання можливості користувачу налагоджувати інтерфейс за своїм смаком. Користувач повинен мати можливість налагодження: способу представлення інформації на свій вибір (кольори, шрифти, розташування елементів), поведінки інтерфейсу (дії за замовчуванням, макроси, кнопки) та інтерфейсних функцій (натискання кнопок чи клавіш, сполучення клавіш для швидкого вибору команд, мнемоніка, розташування кнопок миші для виконання команд).

*Інтерактивність* - дозвіл користувачу маніпулювати об'єктами інтерфейсу. Всюди, де це можливо, потрібно дозволяти користувачу безпосередньо взаємодіяти з об'єктами на екрані в природньому, натуральному стилі. Користувачі повинні комфортно почуватись при виконанні операцій та знати про передбачуваний результат.

### **1.5.2. Правило 2: "зменшити навантаження на пам'ять користувача"**

*Положеннями зменшення навантаження на пам'ять користувача* є запам'ятовування, розпізнавання, інформування, терпимість, швидкість, інтуїтивність, перенос, контекст, організація.

*Запам'ятовування* - не слід завантажувати короточасну пам'ять. Короточасна пам'ять допомагає нам зберігати інформацію на протязі невеликого проміжку часу. Користувачі

часто працюють над декількома задачами одночасно, тому не варто ІК завантажувати короточасну пам'ять користувача в моменти перемикань між ними. Цей принцип проектування часто порушується, що змушує застосовувати зовнішні “засоби” для зберігання інформації (папір, калькулятор).

Функції програм (виділення останньої дії та її повтор) та дії з використанням буферу обміну дозволяють користувачам маніпулювати частинами інформації, необхідними в багатьох місцях, а також всередині задач. Оптимальним є варіант, коли програма в потрібному місці може автоматично зберегти та передати дані, коли користувач зайнятий виконанням інших задач.

*Розпізнавання* - потрібно опиратись на розпізнавання, а не на повторення. Підтримка інтерфейсом довгочасної пам'яті передбачає розпізнавання інформації користувачем, перш ніж він згадає її. Легше обрати якийсь об'єкт зі списку, ніж згадувати його правильну назву для введення в порожній рядок. Онлайнова допомога, повідомлення, поради щодо користування інструментами, система контекстної допомоги і т.і. допомагають користувачу обирати інформацію, а не згадувати її.

Слід передбачити списки і меню, що містять об'єкти чи документи, які можна обирати, не змушуючи користувачів вводити інформацію в командному рядку без підтримки системи.

*Інформування* - наявність візуальних заставок. Необхідний аспект будь-якого графічного ІК та об'єктно-орієнтованого ІК полягає в тому, що користувачі повинні знати, в якому місці ПЗ вони знаходяться, які дії виконують наразі і які дії можуть виконати в подальшому.

Візуальна інформація служить нагадуванням для користувачів. Коли користувачі працюють в якомусь режимі або з мишею, це повинно відображатись на екрані за допомогою відповідної індикації. Форма курсора може змінюватись для вказання поточного режиму або дії, а індикатор – вмикатись і вимикатись. Візуальна інформативність продукту полягає в наявності візуальних підказок інтерфейсу, які інформують користувача про його поточні дії.

*Терпимість* - треба передбачити параметри за замовчуванням, команди відхилення (скасування) та повторення дії. Інтерфейс повинен надавати користувачу можливість зміни установок та налаштувань ПЗ, а також можливість повернення до установок за замовчуванням. Користувачі за своїми перевагами можуть змінити колір, шрифт, властивості прикладної програми або операційної системи, не задумуючись про їх початковий варіант.

При розробленні інтерфейсу потрібно передбачити багаторівневі системи відхилення та повторення команд.

*Швидкість* - слід передбачити "швидкі" шляхи проходження інтерфейсу. Крім комбінованого використання миші та клавіатури існує ряд інших можливостей прискорити роботу з програмою. "Тарячі" клавіші та ярлики зменшують навантаження на пам'ять користувача та доводять виконання операцій до автоматизму.

Є 2 основних способи встановлення ярликів: прискорюючі та мнемонічні. Мнемонічні (або доступні) ярлики - це одиночні буквенно-цифрові символи, які встановлюють курсор на потрібний об'єкт та дозволяють зробити вибір. Вони використовують різні меню (панель, підменю, контекстні) та списки. Мнемонічні символи повинні бути унікальними для кожного роду дій. Наприклад, типове меню вікна використовує стандартні клавіші: F - для файлу, E - для редагування, V - для перегляду, H - для виклику довідкової системи. Наступний рівень меню (підменю) використовують свої налагодження мнемонічних клавіш: N - новий документ, O - відкрити, C - закрити, S - зберегти. Мнемонічні символи прискорюють рух і вибір потрібного меню або списку.

Прискорюючі ярлики (або клавіші швидкого доступу) – це клавіші або комбінації клавіш, які користувачі повинні натиснути для здійснення якої-небудь дії. Наприклад, Ctrl+P – прискорюючий ярлик для друку.

Щойно користувачі добре засвоять програмний продукт, вони починають відчувати потребу в прискорюючих ярликах. Не слід ігнорувати цю необхідність, однак при проектуванні потрібно слідувати стандартам щодо призначення клавіш та їх комбінацій,

інакше користувачі робитимуть помилки, використовуючи звичну комбінацію клавіш, яка виконуватиме непередбачувану дію.

*Інтуїтивність* - потрібно активізувати синтаксис дій з об'єктами. Об'єктно-орієнтований інтерфейс надає ряд переваг від використання об'єктно-орієнтованого синтаксису взаємодії. Користувачам не треба запам'ятовувати, яку дію можна виконати в певний момент часу для об'єкта. При виборі об'єкта користувач бачить у меню дії, які можуть бути виконані над об'єктом. Недопустимі дії виділяються сірим кольором.

Об'єктно-орієнтований синтаксис дозволяє користувачу зрозуміти взаємозв'язок між об'єктами та діями в програмному продукті.

*Перенос* - в інтерфейсі слід використовувати метафори (переноси, які використовують назву об'єкту одного класу для опису об'єкту іншого класу [22]) реального світу (рис.1.5.1), які дозволяють користувачу переносити свої знання з реального світу у світ комп'ютерів. Наприклад, для програми Калькулятор не варто розробляти новий інтерфейс, він повинен співпадати з інтерфейсом звичайного калькулятора, який добре засвоїли користувачі. Однак при виборі та використанні метафор для інтерфейсу слід бути обережними. Обираючи метафору, потрібно зафіксувати її та слідувати їй весь час. Якщо виявиться, що метафора не відповідає своєму призначенню в усьому інтерфейсі, з'явиться необхідність вибору нової метафори.

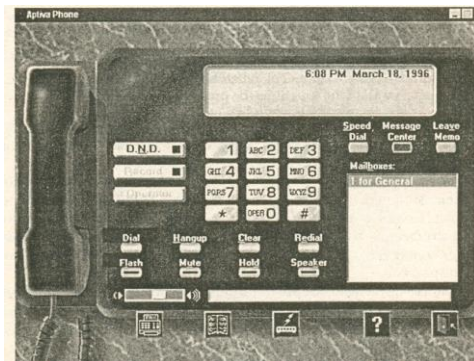


Рис.1.5.1 - Метафора реального світу в ІК телефонній комп'ютерній системі

*Контекст* - слід застосовувати розкриття та пояснення понять і дій. Користувачі не повинні вважати можливості ПЗ більшими чи іншими, ніж вони є насправді. На кожному рівні інтерфейсу не варто показувати абсолютно всі функції ПЗ, а лише ті, в яких є потреба саме на даному рівні. Деякі програмні продукти надають різні меню для користувачів. Спочатку можна обрати просте меню для щоденних, звичайних потреб. Пізніше, по мірі засвоєння продукту або при виникненні потреби в більш складних властивостях програми, користувачі зможуть обрати більш складне меню. Це приклад того, як користувач одержує контроль над задачею та програмою. Нові властивості інтерфейсів (Майстри та Порадники) розкривають та пояснюють поняття та дії при вирішенні задач. Майстри допомагають користувачу виконувати задачу крок за кроком, кожен з яких є зрозумілим.

Розробник має забезпечити користувачу легкий доступ до часто використовуваних функцій та дій. Можна приховати частину властивостей та функцій і дозволити користувачу викликати їх по мірі необхідності. Не потрібно намагатись відобразити всю інформацію в головному вікні, для цього можна використати вторинні вікна.

*Організація* - потрібно збільшити візуальну ясність інтерфейсу. Принципи візуального проектування дають можливість полегшити сприйняття інформації шляхом використання групувань об'єктів в меню або списки; нумерації об'єктів; заголовків та запрошень. На рис.1.5.2 показано макет вікна з погано (А) та добре (Б) організованою інформацією. На рис.1.5.2 (Б) інформація групована, використано заголовки, меню, а на рис.1.5.2 (А) інформація абсолютно не систематизована, що зменшує її читабельність та сприйняття.

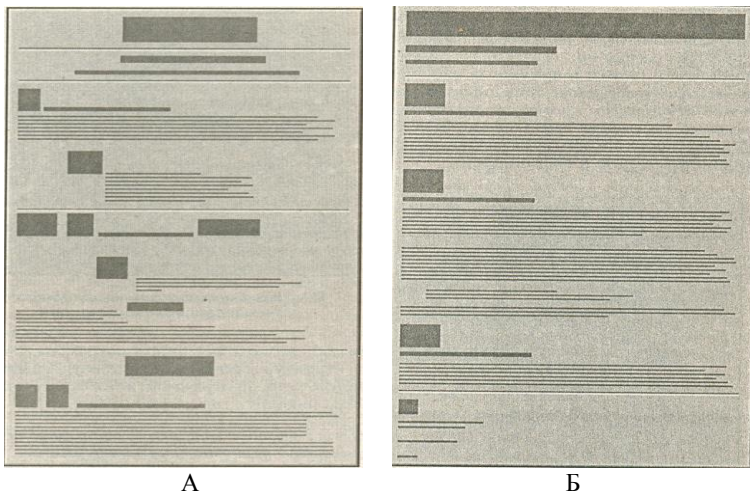


Рис.1.5.2 - Макет вікна програми з погано (А) та добре (Б) організованою інформацією

Деякі програми одночасно надають надто багато інформації на екрані, що викликає відчуття хаосу. Інформація має подаватись у порядку, зрозумілому користувачам, тобто “форма повинна відповідати призначенню”.

### **1.5.3. Правило 3: "зробити інтерфейс сумісним"**

Сумісність – ключовий аспект інтерфейсу. Однією з основних переваг сумісності є те, що користувачі можуть перенести свої знання та навички зі старої програми, якою вони користувались раніше, у нову.

Наприклад, при створенні інформаційного повідомлення бажано присвоїти йому ідентифікаційний номер, який надалі слід видавати разом з повідомленням. Це дозволить користувачу зрозуміти, що схожі повідомлення з’являються кожен раз в певній ситуації, незалежно від того, в якому місці інтерфейсу він знаходиться.

*Положеннями розроблення сумісного інтерфейсу є:* послідовність інтерфейсу, досвід, прогнозування, відношення, передбачуваність.

*Послідовність* інтерфейсу - проектування послідовного, узгодженого інтерфейсу. Користувачі повинні мати опорні точки інтерфейсу. Це заголовки вікон, навігаційні карти та деревовидні структури. Користувачу також потрібна можливість завершити поставлену задачу без зміни середовища роботи або перемикання між стилями введення інформації. Візуальна допомога підкаже користувачу, що відбудеться при завершенні задачі.

*Досвід* - спільна сумісність всіх програм. Один з головних аспектів в розробці інтерфейсу – можливість навчання користувача головним концепціям системи та програмного забезпечення, що можуть застосовуватись у нових ситуаціях та інших програмах. Сумісність - одна з головних властивостей ІК, яка проявляється на трьох рівнях: подання інформації, поведінка програми та техніка взаємодії.

Сумісність в поданні інформації передбачає, що користувачі можуть сприймати інформацію та об'єкти в схожому логічному, візуальному та фізичному вигляді в усьому програмному продукті. Стиль представлення інформації не повинен змінюватись без видимих причин.

Сумісність в поведінці програми передбачає, що об'єкт є однаковим всюди. Поведінка контрольних інструментів, таких як кнопки, списки і меню, не змінюється всередині програми або у різних програмах одного розробника. Користувачі мають бути впевнені в послідовній поведінці об'єктів інтерфейсу.

Сумісність техніки взаємодії також дуже важлива. Однакові сполучення “швидких” клавіш та способи роботи з мишею повинні працювати в схожих за призначенням програмах. Мнемонічні схеми клавіатури теж не повинні змінюватися. Користувачі очікують аналогічних результатів при взаємодії однаковими методами з різними об'єктами.

Коли схожі об'єкти поводяться по-різному у схожих ситуаціях, у користувачів відбувається негативний перенос знань.

Це гальмує вивчення інтерфейсу програми і призводить до втрати користувачем впевненості у своїх силах.

*Прогнозування* - збереження результатів взаємодії. Якщо користувач, виконуючи одні й ті ж дії, одержує різні результати, він більш схильний звинувачувати у цьому себе, ніж програмне забезпечення.

Сумісність кроків і дій має витримуватись в усьому програмному продукті. Процеси реєстрації та навігації мають бути послідовними. Стандартні елементи інтерфейсу повинні поводитись однаково. Якщо результати можуть відрізнятись від того, що очікує користувач, слід інформувати його перед виконанням дії, а також надати йому опції виконання, скасування або здійснення іншої дії.

*Відношення* - естетична привабливість та цілісність. Деякі з сучасних програмних продуктів виглядають так, ніби вони розроблені різними людьми, які жодного разу не спілкувались між собою. Сумніви користувачів щодо цілісності ПЗ виникають, якщо в різних його частинах виявляється непослідовність кольорів, шрифтів, іконок та інших складових інтерфейсу.

Приємний для сприйняття інтерфейс не приховує недолік функціональності програмного забезпечення.

*Передбачуваність* - заохочення вивчення. Задача проєктувальників ІК - створити "дружній" інтерфейс, який заохочуватиме користувача досліджувати його складові і властивості без страху зробити щось невірне.

## **Вправи з розв'язками**

**Вправа 1.** Вкажіть характеристики користувача, які є найбільш важливими для ІК?

Найбільш важливими для ІК є наступні характеристики:

- 1) попередні знання та досвід, використовувані в галузі апаратного та програмного забезпечення;
- 2) попередні знання та досвід, які стосуються підтримки прикладних задач;
- 3) фізичні та індивідуальні характеристики;
- 4) фізичні та соціальні особливості робочого середовища.

**Вправа 2.** Наведіть приклади необхідних проектних рішень інтерфейсу з орієнтацією на користувача.

Необхідні проектні рішення:

- 1) інтерфейс для новачків, випадкових користувачів та спеціалістів;
- 2) поступове розкриття всіх можливостей;
- 3) декілька методів підтримки продуктивності;
- 4) можливості вибору користувацьких переваг;
- 5) множинні представлення.

**Вправа 3.** Які навички, на вашу думку, повинна мати команда розробників ІК?

Команда розробників ІК повинна мати навички в наступних галузях:

- 1) проблемного аналізу (пошук та формулювання проблеми, постановка задачі);
- 2) програмування;
- 3) розроблення ІК;
- 4) графічного проектування;
- 5) написання технічної документації;
- 6) тестування ПЗ та ІК на зручність застосування.

**Вправа 4.** Написати програму мовою програмування Visual C++ для реалізації наступного завдання для графічного інтерфейсу користувача GDI: створити і обрати шрифт з заданими

характеристиками. В лівому нижньому куті вікна додатку 10 разів вивести один і той же текст з різними кольорами тексту.

### Текст програми:

```
#include "Functions.h"
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
HINSTANCE hInstance;
char szClass[]="TextOutClass";
// створення вікна
INT WINAPI WinMain(HINSTANCE hInstance, HINSTANCE,
PSTR, INT nCmdShow)
{MSG msg; HWND hwnd; ::hInstance=hInstance;
if (!RegClass(WndProc, szClass, COLOR_WINDOW))
return FALSE;
hwnd=CreateWindow (szClass, "Виведення тексту",
WS_OVERLAPPEDWINDOW|WS_VISIBLE,
CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
CW_USEDEFAULT, 0, 0, hInstance, NULL);
if (!hwnd) return FALSE;
while (GetMessage(&msg, NULL, 0, 0))
DispatchMessage(&msg);
return(int)msg.wParam; }
//робота з текстом, який виводиться у вікні
LRESULT CALLBACK WndProc(HWND hwnd, UINT msg,
WPARAM wParam, LPARAM lParam)
{char szFont[]="Будь-який текст";
static short cyClient;
switch (msg){
// визначення розміру шрифту
case WM_SIZE:
{cyClient=HIWORD(lParam); return 0;}}
// визначення характеристик шрифту та кольору текста
case WM_PAINT:
{PAINTSTRUCT ps;
HDC hdc=BeginPaint(hwnd, &ps);
static LOGFONT lf;
lf.lfCharSet=DEFAULT_CHARSET;
lf.lfPitchAndFamily=DEFAULT_PITCH;
strcpy(lf.lfFaceName, "TimesNewRoman");
lf.lfHeight=20;
```

```

lf.lfWeight=FW_BOLD;
for (int i=0;i<10;i++)
{lf.lfOrientation=lf.lfEscapement=i*100;
HFONT hNFont=CreateFontIndirect(&lf);
HFONT hOFont=(HFONT)SelectObject(hdc,hNFont);
SetTextColor(hdc, RGB(i*15, i*20, i*25));
SetBkColor(hdc, RGB(255-i*15, 255-i*20,255-i*25));
TextOut(hdc, 0, cyClient-30, szFont, strlen(szFont));
SelectObject(hdc, hOFont); DeleteObject(hNFont); }
EndPaint(hwnd, &ps);
return 0; }
case WM_DESTROY:
{PostQuitMessage(0); return 0;}
}
return DefWindowProc(hwnd, msg, wParam, lParam);}

```

**Вправа 5.** Написати програму мовою програмування Visual C++ для реалізації завдання: головне меню містить розділ "Файл" з рядками "Створити", "Відкрити" і "Вихід". При виборі рядків "Створити" або "Відкрити" створити вікно, яке містить розділ меню "Фігури" зі списком імен геометричних фігур, при виборі імені відобразити фігуру з таким іменем та позначити ім'я будь-якою позначкою. При повторному виборі імені видалити з форми фігуру і видалити позначку.

Текст програми:

```

#include "Functions.h"
#define FILE_OPEN 1001
#define FILE_SAVE 1002
#define FILE_QUIT 1003
#define EDIT_FIND 2001
#define EDIT_REPLC 2002

//створення вікна
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
HINSTANCE hInstance;
char szClass[]="RemoveMenu";
INT WINAPI WinMain(HINSTANCE hInstance, HINSTANCE,
PSTR, INT nCmdShow)

```

```

{ MSG msg; HWND hwnd; ::hInstance=hInstance;
if (!RegClass(WndProc, szClass, COLOR_WINDOW)) return
FALSE;
hwnd=CreateWindow (szClass, "Додаток зі статичним
меню",
WS_OVERLAPPEDWINDOW|WS_VISIBLE, CW_USEDEFAULT,
CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
0, 0, hInstance, NULL);
if (!hwnd) return FALSE;
while (GetMessage(&msg, NULL, 0, 0))
{TranslateMessage(&msg);
DispatchMessage(&msg);}
return(int)msg.wParam;}
//створення меню
BOOL CreateMenuItem(HMENU hMenu, char *str,
UINT uIns, UINT uCom,
HMENU hSubMenu, BOOL flag, UINT fType)
{MENUITEMINFO mii;
mii.cbSize=sizeof(MENUITEMINFO);
mii.fMask=MIIM_STATE | MIIM_TYPE | MIIM_SUBMENU |
MIIM_ID;
mii.fType=fType; mii.fState=MFS_ENABLED;
mii.dwTypeData=str; mii.cch=sizeof(str);
mii.wID=uCom; mii.hSubMenu=hSubMenu;
return InsertMenuItem(hMenu, uIns, flag, &mii);}
LRESULT CALLBACK WndProc(HWND hwnd, UINT msg, WPARAM
wParam, LPARAM lParam)
{static HMENU hMainMenu, hFileMenu, hEditMenu;
static BOOL fFileOpened;
switch(msg)
{ case WM_CREATE:
{hMainMenu=CreateMenu();
// Створюємо тимчасове меню для розділу "Файли"
hFileMenu=CreatePopupMenu();
int i=0; //Ініціалізація позиції в меню hFileMenu
CreateMenuItem(hFileMenu, "&Відкрити", i++,
FILE_OPEN, NULL, FALSE, MFT_STRING);
CreateMenuItem(hFileMenu, "&Зберегти", i++,
FILE_SAVE, NULL, FALSE, MFT_STRING);
CreateMenuItem(hFileMenu, NULL, i++, 0,

```

```

NULL, FALSE, MFT_SEPARATOR);
CreateMenuItem(hFileMenu, "&Вихід", i++, FILE_QUIT,
NULL, FALSE, MFT_STRING);
//Створюємо тимчасове меню для розділу "Правка"
hEditMenu=CreatePopupMenu();
i=0; //Ініціалізація позиції в меню hEditMenu
CreateMenuItem(hEditMenu, "&Знайти", i++, EDIT_FIND,
NULL, FALSE, MFT_STRING);
CreateMenuItem(hEditMenu,          "&Замінити",          i++,
EDIT_REPLC, NULL, FALSE, MFT_STRING);
//Підключаємо тимчасові меню до головного меню
i=0; //Ініціалізація позиції в меню hMainMenu
CreateMenuItem(hMainMenu, "&Файл", i++, 0,
hFileMenu, FALSE, MFT_STRING);
SetMenu(hwnd, hMainMenu);
DrawMenuBar(hwnd);
fFileOpened=FALSE; return 0; }
case WM_COMMAND:
{switch (LOWORD (wParam))
{case FILE_OPEN:
{if (!fFileOpened)
{CreateMenuItem (hMainMenu, "&Правка", 1, 0,
hEditMenu, FALSE, MFT_STRING);
DrawMenuBar(hwnd); fFileOpened=TRUE; }
return 0; }
case FILE_SAVE:
{if (fFileOpened)
{RemoveMenuItem (hMainMenu, 1, MF_BYPOSITION);
DrawMenuBar(hwnd); fFileOpened=FALSE; }
return 0; }
case EDIT_FIND:
{MessageBox(hwnd,      "Команда   EDIT_FIND",      "Меню",
MB_OK);
return 0; }
case EDIT_REPLC:
{MessageBox(hwnd,      "Команда   EDIT_REPLC",      "Меню",
MB_OK);
return 0; }
case FILE_QUIT:
{DestroyWindow(hwnd); return 0;}

```

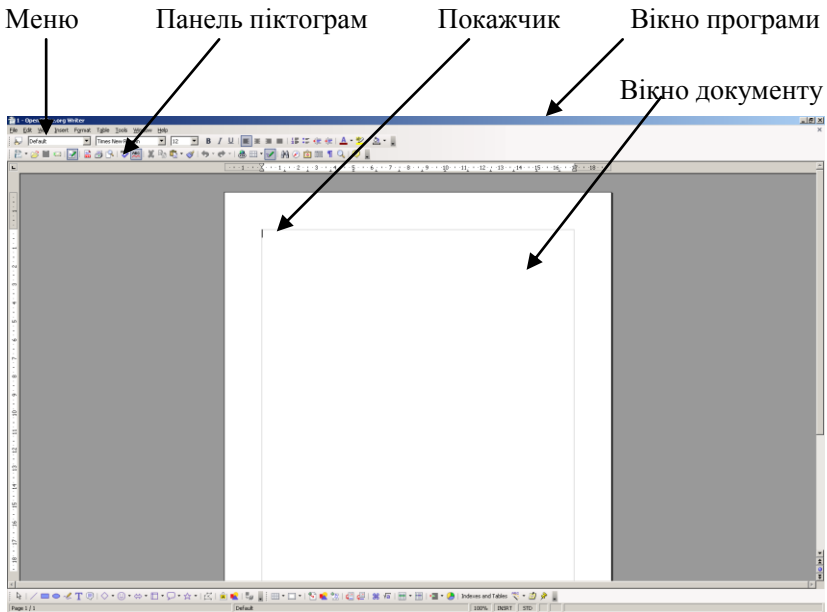
```

}
return 0; }
case WM_DESTROY:
{if (!fFileOpened) DestroyMenu(hEditMenu);
PostQuitMessage(0); return 0;}
}
return DefWindowProc (hwnd, msg, wParam, lParam);}

```

**Вправа 6.** Вкажіть такі елементи ГІК, як вікно, піктограми, меню та покажчик для будь-якого ГІК на ваш вибір.

Покажемо задані елементи ГІК на прикладі інтерфейсу текстового редактора Open Office Write.



**Вправа 7.** Сформуйте основні рекомендації по розробленню інтерфейсу користувача та компоунанню вікон

Основні рекомендації по розробленню інтерфейсу користувача і компоунанню вікна, запропоновані Уілкінсом [19], мають наступний вигляд:

- 1) асиметрія = активний стан, симетрія = стан спокою;
- 2) ніколи не слід ставити помірність понад усе;
- 3) слід уважно стежити за послідовністю та узгодженістю інтерфейсу користувача;
- 4) слід пам'ятати, що узгоджений інтерфейс ПЗ означає кращу впроваджуваність ПЗ;
- 5) слід чітко вирівнювати керуючі елементи;
- 6) якщо потрібно розташовувати деякі елементи за межами вирівнювання, то слід робити це логічно;
- 7) слід створювати екранні об'єкти за зразком об'єктів з реального фізичного світу;
- 8) абсолютна симетричність ускладнює читання інформації з екрану;
- 9) елементи, які не відповідають шаблону, виділяються з загальної маси;
- 10) елементи однакового розміру та кольору сприймаються як приналежні до однієї групи.

**Вправа 8.** Одній групі користувачів запропонували запам'ятати розташування керуючих елементів інтерфейсу та принципи роботи з ними шляхом чотириразового показу цих елементів та принципів роботи з ними, а іншій групі - шляхом дворазового показу та дворазової спроби роботи, які чергувалися. Виявилось, що повнота запам'ятовування у користувачів цих підослідних груп неоднакова. За годину в учасників однієї з груп матеріал утримувався пам'яттю на 75%, в учасників іншої - на 52%, а за десять днів - відповідно на 72% і 25%. До якої дослідної групи належать числові дані експериментів, які характеризують більш повне запам'ятовування?

Очевидно, що числові дані експериментів, які характеризують більш повне запам'ятовування належать до учасників другої групи, які не тільки спостерігали за керуючими елементами інтерфейсу та принципами роботи з ними, але й самі намагались попрацювати з цими елементами.

**Вправа 9.** Які із нижченаведених здібностей знадобляться: в математичній діяльності; в технічній творчості (проектуванні інтерфейсів користувача)?

Здібності:

- 1) здібність відчувати музичну виразність;
- 2) здатність до комбінування просторових образів, просторова уява;
- 3) запам'ятовування загальних схем міркувань, доведень, висновків, узагальнених способів вирішення питань;
- 4) цілісність сприйняття, вміння бачити окремі частини предмету в їх співвідношенні з іншими частинами;
- 5) повнота, яскравість уявлення наочних образів;
- 6) здібність до представлення сполучень музичних звуків;
- 7) вміння знаходити раціональний підхід до практичних задач з врахуванням властивостей та можливостей інструментарію;
- 8) комплексний підхід до вирішення питання;
- 9) вміння оцінювати співвідношення пропорцій та розмірів;
- 10) здатність до образного уявлення за словесним описом;
- 11) точність сприйняття кольорових відтінків;
- 12) сприйняття мовних особливостей;
- 13) здатність розрізняти звуки по висоті звучання;
- 14) послідовність, обґрунтованість, логічність міркувань;
- 15) точне запам'ятовування зорових вражень;
- 16) емоційна чуйність, здатність уявити себе на місці іншої людини;
- 17) вміння свої думки та відчуття передавати за допомогою наочних образів;
- 18) здатність до аналізу та синтезу матеріалу.

Здібності 3, 8, 14, 18 необхідні для математичної діяльності, а здібності 2, 4, 5, 7, 9, 10, 11, 12, 15, 16, 17 повинні бути притаманні проектувальнику інтерфейсів користувача. Здібності 1, 6, 13 не властиві жодному наведеному виду діяльності, вони необхідні для музичної творчості.

**Вправа 10.** Наведіть приклади використання режимів роботи в ІК, а також приклади для режиму додатку та для системного режиму.

Прикладом використання режимів роботи є палітра інструментів для малювання в графічному редакторі або вікно друку з багатьма можливими варіантами друку (режимами).

Режим додатку ілюструє приклад, коли у режимі перегляду документа чи бази даних немає можливості додавати, видаляти або модифікувати записи.

Системний режим ілюструє ситуація, коли в принтері закінчився папір і з'явилося відповідне повідомлення. У деяких програмах не можна продовжувати роботу, доки користувач не забезпечить наявність паперу в принтері.

**Вправа 11.** Наведіть приклади вдалого та невдалого повідомлення з врахуванням принципів корисності.

Повідомлення типу "Пароль надто короткий. Він повинен бути не менше 15605 байт" можливо й інформативне, але воно не корисне для користувача, оскільки той не знає, скільки символів містить 15605 байт. Таке повідомлення порушує принцип "прозорості".

Повідомлення типу "Пароль надто короткий. Він повинен містити не менше 6 символів" є інформативним, корисним та вдалим.

### **Контрольні питання**

- 1) Дайте визначення користувацького інтерфейсу.
- 2) Перерахуйте складові користувацького інтерфейсу.
- 3) Від чого, на вашу думку, залежить задоволеність користувачів користувацьким інтерфейсом?
- 4) На яких керуючих принципах заснована розробка інтерфейсу, орієнтованого на користувача?
- 5) Які фактори слід брати до уваги при вивченні користувачів з точки зору розроблюваного для них ІК?
- 6) Які методи залучення користувачів до проекту ви знаєте?
- 7) Які ви знаєте базові принципи проектування, орієнтованого на користувача?
- 8) Назвіть етапи проектування інтерфейсу, орієнтованого на користувача.
- 9) Якими якостями повинні володіти розробники інтерфейсу, орієнтованого на користувача?
- 10) Назвіть головну мету колективу розробників інтерфейсу, орієнтованого на користувача.
- 11) Які фактори інтерфейсу користувача слід розглядати за кожним типом ІК?
- 12) Що представляє собою інтерфейс командного рядка?
- 13) Наведіть приклади прикладних програм та операційних систем, які використовують інтерфейс командного рядка.
- 14) Назвіть переваги та недоліки ІКР.
- 15) Що представляє собою інтерфейс меню?
- 16) Перерахуйте типи меню за формою, розмірами та стилем.
- 17) Наведіть приклади програм та операційних систем, які використовують повноекранне меню, панель меню, панелі інструментів, контекстне меню.
- 18) Назвіть переваги та недоліки інтерфейсу меню.
- 19) Що представляє собою графічний інтерфейс користувача? Наведіть приклади програм та операційних систем, які використовують ГІК.

- 20) Перерахуйте характеристики ГПК. Назвіть необхідні знання для роботи з ГПК.
- 21) Перерахуйте та охарактеризуйте основні стилі інтерфейсу користувача.
- 22) Назвіть та розшифруйте принципи проектування "дружнього" ІК.
- 23) В чому полягає суть ментальної (концептуальної) моделі ІК?
- 24) В чому полягає суть моделі користувача для ІК?
- 25) В чому полягає суть моделі програміста ІК?
- 26) В чому полягає суть моделі проектувальника ІК?
- 27) Які ви знаєте методи сумісної розробки ІК на етапах планування, аналізу вимог, проектування та реалізації?
- 28) Перерахуйте засоби проектування ІК.
- 29) В чому суть графічного відображення інформації та візуального представлення інформації в ІК?
- 30) Як повинен використовуватись колір в ІК?
- 31) Що, на вашу думку, розуміється під людським фактором проектування ІК?
- 32) З якими категоріями людей пов'язане проектування ІК? Чому?
- 33) Охарактеризуйте роль і місце ергономіки, психології та соціології при проектуванні ІК.
- 34) Назвіть основні принципи ергономіки.
- 35) Опишіть схему опрацювання інформації людиною.
- 36) Яким характеристикам процесу людського пізнання повинен приділяти увагу проектувальник ІК?
- 37) У чому суть фактора доступності з точки зору проектування ІК?
- 38) З чим пов'язані соціальні проблеми в галузі проектування ІК?
- 39) В чому суть таких методів оброблення інформації, як навчання та прийняття рішень?
- 40) Чим, на вашу думку, відрізняється короткочасна пам'ять людини від довгочасної пам'яті?
- 41) Назвіть основні принципи проектування ІК.

42) Які положення проектування ІК дозволяють надати контроль користувачу?

43) Які положення проектування ІК дозволяють зменшити навантаження на пам'ять користувача?

44) Які положення проектування ІК забезпечують сумісність інтерфейсу?

45) Опишіть "запам'ятовування" як положення, яке дозволяє зменшити навантаження на пам'ять користувача.

46) Опишіть "швидкість" як положення, яке дозволяє зменшити навантаження на пам'ять користувача.

47) Опишіть "досвід" як положення, яке дозволяє зробити інтерфейс сумісним.

48) Опишіть "прогнозування" як положення, яке дозволяє зробити інтерфейс сумісним.

49) Опишіть "поблажливість" як положення, яке дозволяє надати контроль користувачу.

50) Опишіть "інтерактивність" як принцип, який дозволяє надати контроль користувачу.

### Тестові питання

1) Як, на вашу думку, слід розуміти поняття інтерфейсу користувача?

1. Як зовнішній вигляд програми;
2. Як представлення елементів і компонентів програми;
3. Як зовнішній вигляд програми та об'єднання і представлення всіх елементів і компонентів програми, які впливають на взаємодію користувача з програмою;
4. Як сукупність пристроїв та технологій введення-виведення, використовуваних у ПЗ;
5. Інша відповідь.

2) Функцією від яких факторів є функція задоволеності користувача інтерфейсом? (виберіть найповнішу відповідь)

1. Від часу відгуку та пристосованості до інсталяції;
2. Від часу відгуку та можливостей ІК;
3. Від часу відгуку, можливостей ІК, надійності, пристосованості до інсталяції, інформаційної підтримки, пристосованості до супроводження;
4. Від можливостей ІК, надійності, пристосованості до інсталяції, інформаційної підтримки, пристосованості до супроводження;
5. Інша відповідь.

3) Який метод залучення користувача в проект передбачає спостереження за фактичними користувачами, які працюють із запропонованим ПЗ в даний момент часу?

1. Метод спостереження;
2. Метод опитування;
3. Метод індивідуальних інтерв'ю;
4. Метод групових інтерв'ю;
5. Інша відповідь.

4) Який процес проектування спрямований на ІК та властивості програмного продукту, доступні користувачу?

1. Проектування "зовні-всередину";
2. Проектування "зсередини-назовні";
3. Однократне проектування;
4. Проектування за теорією "великого вибуху".

5) Який, з вашої точки зору, підхід до проектування слід обирати для розробки інтерфейсу, орієнтованого на користувача?

1. Однократне проектування "ззовні-всередину" за типом "великого вибуху";
2. Еволюційне ітераційне проектування "зсередини-назовні";
3. Еволюційне ітераційне проектування "ззовні-всередину";
4. Еволюційне однократне проектування "зсередини-назовні"
5. Інша відповідь.

6) Як називається стиль взаємодії користувача з ЕОМ, при якому користувачі набирають запити або вказують дії формальною мовою?

1. Інтерфейс меню;
2. Графічний інтерфейс користувача;
3. Об'єктно-орієнтований інтерфейс користувача;
4. Інтерфейс командного рядка;
5. Інша відповідь.

7) Як називається стиль взаємодії користувача з ЕОМ, який представляє перелік можливих дій, відображуваних на екрані або в вікні для користувачів, з метою вибору потрібних варіантів?

1. Інтерфейс меню;
2. Графічний інтерфейс користувача;
3. Об'єктно-орієнтований інтерфейс користувача;
4. Інтерфейс командного рядка;
5. Інша відповідь.

8) Як можна обґрунтувати назву "контекстне меню"?

1. Вміст такого меню залежить від контексту задач і властивостей, притаманних даному об'єкту;
2. Вміст такого меню залежить від стилю інтерфейсу користувача;
3. Вміст такого меню залежить від інтерфейсу, з яким працюють користувачі;
4. Вміст такого меню залежить від операційної системи, з якою працюють користувачі.

9) Як називається стиль взаємодії користувача з ЕОМ, при якому на екрані відображається графічне представлення інтерактивної взаємодії з програмами, об'єктами і даними?

1. Інтерфейс меню;
2. Графічний інтерфейс користувача;
3. Об'єктно-орієнтований інтерфейс користувача;
4. Інтерфейс командного рядка;
5. Інша відповідь.

10) В якій послідовності працюють користувачі ГІК?

1. Дія-об'єкт;
2. Об'єкт-дія;
3. Інтерфейс-дія;
4. Дія-інтерфейс;
5. Інша відповідь.

11) Як називається ІК, в якому застосовуються такі елементи, як: вікна, піктограми, меню, покажчики?

1. Графічний ІК або WIMP-інтерфейс;
2. Web-інтерфейс користувача;
3. Інтерфейс користувача кишенькових пристроїв або HUI-інтерфейс;
4. Графічний ІК або SIMP-інтерфейс
5. Інша відповідь.

12) Що таке банер як елемент WEB-інтерфейсу користувача?

1. Візуальний заголовок, відображуваний в верхній частині Web-сторінки;
2. Список варіантів вибору гіперпосилань, які забезпечують доступ до інформації;
3. Варіант вибору, який відображає наступну сторінку інформації або пересуває фокус відображення на іншу область тієї ж сторінки;
4. Область пристрою відображення, використовувана для представлення взаємодії з об'єктами, інформації про об'єкти або для виконання дій над об'єктом;
5. Інша відповідь.

13) Що мається на увазі під принципом естетичності (aesthetic) для "дружнього" ІК?

1. ІК не повинен потребувати довідника або діалогової довідкової системи, щоб почати виконувати за його допомогою прості задачі користувачів;
2. Інтерфейс повинен справляти гарне враження, мати естетичну або ергономічну привабливість;
3. ІК повинен виконувати роботу швидше і не гірше, ніж його аналоги;
4. ІК повинен дозволяти користувачу обрати метод взаємодії так, як і методи макетування та доступу для оптимізації користувацьких потреб;
5. Інша відповідь.

14) Що таке модель користувача ІК?

1. Внутрішнє відображення того, як користувач розуміє і взаємодіє з ПЗ; це відображення фізичної системи або ПЗ;
2. Модель, яка відображає спосіб визначення дизайну інтерфейсу в діалозі з користувачем;
3. Функціональна специфікація програмного продукту;
4. Модель, яка описує об'єкти, з якими працює користувач, і техніку маніпулювання ними;
5. Інша відповідь.

15) Засоби проектування ІК - це: (виберіть найповнішу відповідь)

1. Програмне забезпечення, апаратне забезпечення, матеріали, інструментальні засоби розробки;
2. Програмне забезпечення, апаратне забезпечення, обладнання, матеріали, шаблони проектування/прототипування/оцінки, інструментальні засоби розробки;
3. Програмне забезпечення, апаратне забезпечення, обладнання, шаблони проектування/прототипування/оцінки, інструментальні засоби розробки;

4. Програмне забезпечення, апаратне забезпечення, обладнання, матеріали, шаблони проектування/прототипування/оцінки, інструментальні засоби розробки, впровадження;
  5. Інша відповідь.
- 16) Основними принципами ергономіки є: (вибрати найповнішу відповідь)
1. Навчання і набуття практики людиною з метою пристосування до існуючого ПЗ;
  2. Проектування ПЗ, яке відповідає можливостям людей;
  3. Навчання і набуття практики людиною з метою пристосування до існуючого ПЗ; проектування ПЗ, яке відповідає можливостям людей;
  4. Навчання і набуття практики людиною з метою пристосування до існуючого ПЗ; проектування ПЗ, яке відповідає можливостям людей; пристосування ПЗ до техніки;
  5. Інша відповідь.
- 17) Що таке короткотривала пам'ять людини?
1. Частина пам'яті людини, яка зберігає обмежений обсяг інформації, що належить до поточного часу;
  2. Буфер, який тимчасово зберігає фіксовані образи, обсяг яких достатній для того, щоб користувач міг їх аналізувати;
  3. Частина пам'яті, яка зберігає інформацію, що належить до колишнього досвіду людини;
  4. Частина пам'яті, яка зберігає історію підходів до вирішення поточних проблем, тимчасові результати та відображає зміни в планах на майбутнє;
  5. Інша відповідь.
- 18) Процес навчання - це:
1. Складний процес, який приймає різні форми - запам'ятовування, засвоєння понять та правил, моторні навички для автоматичних процедур, що повторюються;

2. Свідома обізнаність та опрацювання інформації, яка відрізняється винахідливістю, сконцентрованістю або розсередженістю;
3. Пошук інформації, оцінка альтернатив і результатів, вибір спрямування дій;
4. Можливість більш якісної підтримки користувачів і уваги до їхніх потреб;
5. Інша відповідь.

19) Процес прийняття рішень - це:

1. Складний процес, який приймає різні форми - запам'ятовування, засвоєння понять та правил, моторні навички для автоматичних процедур, що повторюються;
2. Свідома обізнаність та обробка інформації, яка відрізняється винахідливістю, сконцентрованістю або розсередженістю;
3. Пошук інформації, оцінка альтернатив і результатів, вибір спрямування дій;
4. Можливість більш якісної підтримки користувачів і уваги до їхніх потреб;
5. Інша відповідь.

20) Доступність ІК - це:

1. Складний процес, який приймає різні форми - запам'ятовування, засвоєння понять та правил, моторні навички для автоматичних процедур, що повторюються;
2. Свідома обізнаність та обробка інформації, яка відрізняється винахідливістю, сконцентрованістю або розсередженістю;
3. Пошук інформації, оцінка альтернатив і результатів, вибір спрямування дій;
4. Можливість більш якісної підтримки користувачів і уваги до їхніх потреб;
5. Інша відповідь.

21) Що таке безрежимність з точки зору положень, за якими користувачу дається контроль над системою?

1. Розважливе використання режимів;
2. Надання користувачу можливості обирати;

3. Дозвіл користувачу сфокусувати увагу;
4. Демонстрація повідомлень, які допомагатимуть в роботі користувачу
5. Інша відповідь.

22) Що таке запам'ятовування з точки зору положень, які дозволяють зменшити навантаження на пам'ять користувача?

1. Не перевантаження короткочасної пам'яті користувача;
2. Орієнтація на розпізнавання, а не на повторення;
3. Необхідність представлення візуальних заставок;
4. Необхідність передбачення установок за замовчуванням, команд відхилення та повторення дії;
5. Інша відповідь.

23) Що таке прогнозування з точки зору положень створення сумісного інтерфейсу?

1. Проектування послідовного інтерфейсу;
2. Спільна сумісність усіх програм;
3. Збереження результатів взаємодії;
4. Естетична привабливість та цілісність;
5. Інша відповідь.

24) Вкажіть принципи розроблення ІК?

1. Контроль інтерфейсу, зменшення навантаження на пам'ять користувача; послідовність ІК;
2. Скорочення запам'ятовування, оптимізація операцій, відсутність помилок;
3. Інша відповідь.

25) Що таке інтерактивність з точки зору положень, за якими користувачу дається контроль над системою?

1. Надання можливості користувачу налагоджувати інтерфейс за своїм смаком;
2. Дозвіл користувачу маніпулювати об'єктами інтерфейсу;
3. Налагодження можливостей системи на користувачів з різним рівнем підготовки;
4. Створення більш зрозумілого ІК;
5. Інша відповідь.

### **Завдання для самоперевірки**

1) Опишіть функціональні можливості ІК середовища програмування FreePascal.

2) Складіть бажані вимоги до ІК для програми керування принтером.

3) Опишіть інформацію про трьох студентів вашої групи з точки зору потенційних користувачів розроблюваного вами ПЗ та його інтерфейсу.

4) Складіть перелік важливих питань для методу індивідуальних інтерв'ю.

5) Розробіть концептуальну модель для інтерфейсів користувача програми керування принтером.

6) Напишіть програму мовою програмування C++ для реалізації задачі: головне меню містить рядки "Невидимий курсор", "Звичайний курсор" і "Вихід". Створити плаваюче меню з такими ж рядками. Щоб курсор став невидимим, викликати функцію ShowCursor(0), видимим - ShowCursor(1).

7) Напишіть програму мовою програмування C++ для реалізації задачі: головне меню містить розділ "Рисунок" з назвами чотирьох геометричних фігур. Після вибору фігури відобразити її в певній частині вікна. При натисканні правої кнопки миші над будь-якою відображеною фігурою на місці натискання відобразити плаваюче меню з відповідними обраній фігурі командами.

8) Написати програму мовою програмування C++ з використанням GDI для реалізації задачі: від лівого верхнього до нижнього правого кута робочої області вікна вивести текст таким чином, щоб він повністю заповнював робочу область вікна при будь-яких змінах розмірів вікна. Використовувати шрифт Times New Roman Cyr.

9) Напишіть програму мовою програмування C++ з використанням GDI для реалізації задачі: в центрі вікна намалювати мішень з 10 полів і в кожному полі вивести його значення (від 1 на краю до 10 в центрі). При будь-яких змінах розмірів вікна мішень повинна повністю відобразитись у вікні. Поля розфарбувати випадково обраними різними кольорами.

10) Напишіть програму мовою програмування C++ з використанням GDI для реалізації задачі: в робочій області вікна пензлем за замовчуванням в метричній системі координат намалювати еліпс.

11) Вкажіть такі елементи Web-інтерфейсу користувача, як банер, навігаційна панель та гіперпосилання для будь-якого Web-інтерфейсу на ваш вибір.

12) На прикладі обраного вами інтерфейсу вкажіть, які принципи SAPCO в ньому присутні. Оцініть, чи є "дружнім" та якісним цей інтерфейс.

13) Складіть модель користувача для будь-якого текстового редактора на ваш вибір.

14) Складіть модель програміста для програми, яка формує типовий файл (прізвище та ім'я, день і місяць народження) про довільну кількість ваших друзів і, використовуючи сформований типовий файл, виводить прізвища тих друзів, які народились влітку.

15) Наведіть приклади вдалих та невдалих інтерфейсів користувача з точки зору використання в них кольору, звуку та анімації.

16) Вкажіть умови, які потрібні для виникнення та підтримки мимовільної, а які - довільної уваги [20]?

1. Постановка питань та розв'язання невеликих задач протягом певного відрізка часу;

2. Особливості діючих подразників: їх новизна, абсолютна та відносна сила, контраст між ними, зміна в подразниках;

3. Усвідомлення поточних результатів діяльності в формі внутрішнього словесного звіту;

4. Найкращий розпорядок діяльності, створення звичних умов для роботи;

5. Використання певних потреб, з задоволенням яких пов'язаний навчальний матеріал;

6. Використання певних установок та психічних станів;

7. Постановка істотно значущих цілей та задач діяльності.

17) Два користувачі, сидячи за своїми комп'ютерами, успішно працювали з інтерфейсом програми. В цей час третій

користувач працював з тим же інтерфейсом, але часто помилявся. Досвідчений фахівець, який проводив тренінг по засвоєнню цього інтерфейсу, пропонував двом першим користувачам виправляти помилки третього користувача. При цьому перший користувач швидко підходив і одразу виправляв недоліки третього користувача. Другий же користувач не міг відразу відірватись від своєї роботи, а, підходячи до третього користувача, дивився на екран його монітору "незрячим" поглядом. Коли ж тренер звертався до нього з одним-двома питаннями, користувач швидко й добре виправляв помилки третього користувача. Які індивідуальні особливості уваги проявляються у першого та другого користувачів?

18) Який процес пам'яті (запам'ятовування, збереження, впізнавання, відтворення, пригадування, згадування, забування [21]) проявляється в описаних діях: через тиждень після роботи з інтерфейсом користувач розповідає іншим користувачам про цей інтерфейс та принципи роботи з ним і відновлює 70% матеріалу; опитаний через місяць по тому ж питанню, він відновлює лише 45% опрацьованого ним матеріалу?

19) Оберіть, які специфічні особливості пам'яті та її проявів відповідають особливостям людської пам'яті, а які - особливостям пам'яті комп'ютера:

1. Весь обсяг матеріалу ніколи не запам'ятовується повністю;

2. Інформація відображається лише за умови, коли в запам'ятовуючому пристрої відбулись зміни під впливом інформації, що надходить;

3. Процеси сприйняття, збереження та обробки інформації здійснюються за допомогою електрохімічних змін в білкових структурах;

4. Весь обсяг матеріалу може бути запам'ятований повністю;

5. Сигнали інформації є зовнішніми та випадковими по відношенню до запам'ятовуючого пристрою;

6. Процеси сприйняття, збереження та обробки інформації здійснюються за допомогою процесів в електронних пристроях;

7. Процеси сприйняття, збереження та обробки інформації носять вибірковий характер;

8. Запам'ятовується нестандартизована інформація;

9. Запам'ятовується жорстко стандартизована інформація.

20) Досліджувався процес запам'ятовування принципів роботи з інтерфейсом у певної групи користувачів. В одному випадку тренер кількаразово розповідав про принципи роботи з інтерфейсом та показував їх; в другому випадку користувачам пропонувався план та опис вимог, за якими проектувався інтерфейс, а потім тренер один раз розповідав про принципи роботи з інтерфейсом та показував їх; в третьому випадку тренер один раз розповідав про принципи роботи з інтерфейсом та показував їх, а потім пропонував скласти функціональний опис інтерфейсу користувачам. В якому випадку, на вашу думку, запам'ятовування було найбільш ефективним, а в якому - найменш ефективним? Чому?

21) Наведіть приклади вдалого та невдалого рішення для інтерфейсу з точки зору принципів гнучкості, поблажливості, розпізнавання, інтуїтивності, передбачуваності.