**Проектування та розробка засобами Delphi 5 програмного забезпечення “Автоматизоване робоче місце менеджера»**

Курсовий проект Виконав: студент групи 4-ІС-59 Фатєєв Андрій Олександрович.

Український державний хіміко-технологічний університет.

Кафедра інформаційних технологій та кібернетики.

Дніпропетровськ

2002 р.

**Вступ.**

Перед тим, як ми поглибимося у вивчення автоматизованого робочого місця менеджера давайте з'ясуємо перше відразу ж виникле питання - хто ж такий менеджер?..

Управління з'явилося разом із людьми. Там де хоча б дві людини об'єднувалися в прагненні досягти якоїсь спільної мети, виникало завдання координації їхніх спільних дій, розв'язання якого хтось із них повинний був брати на себе. У цих умовах він ставав керівником, що управляє, а інший - його підлеглим, виконавцем.

На всіх етапах становлення суспільства проблема управління стояла досить гостро і багато людей намагалися вирішити її, але їхні труди носили розрізнений характер і не укладали узагальненої теорії.

І тільки в другій половині минулого століття після перемоги промислової революції на Заході ситуація різко змінилася. Ринкові відносини володіли всіма сферами життя суспільства. Як гриби після дощу зростали крупні фірми, що потребували великого числа керівників вищого і середнього рівнів, спроможних приймати грамотні раціональні рішення, що вміли працювати з великими масами людей , що були б вільні у своїх учинках. Тому від керуючих був потрібний високий професіоналізм, компетентність, уміння порівнювати свою діяльність з існуючими законами. У результаті з'являється група людей, що спеціально займаються управлінською діяльністю. Цим керівникам вже не потрібно тримати своїх підлеглих у покірності владною рукою. Головним завданням стає щоденне управління виробництвом із метою забезпечення найбільшого прибутку власникам фірми. Ці люди стали називатися менеджерами.

В теорії менеджменту менеджерами називають спеціалістів, що, використовуючи різноманітні методи і тактику управління, сприяють досягненню організацією або підприємством певних цілей.

Отож, менеджером є тип науково-практичного управління, тобто керівництва людьми і виробництвом, що дозволяє виконувати поставлені завдання найбільш гуманним і економічним шляхом. Оскільки завданням менеджменту є досягнення мети, важливо правильно її визначити. Якщо ми ставимо перед собою завдання удосконалювання організації, не уточнивши її цілей, то ризикуємо запропонувати кращі засоби виконання непотрібних функцій або кращі шляхи досягнення незадовільних кінцевих результатів. Без перебільшення можна сказати, що заподіяний при цьому збиток вимірюється мільярдами.

Мистецтво вибору і формулювання правильної мети, управління процесом її реалізації, включаючи оцінку досягнутого ступеня реалізації, відрізняють дійсно вмілого менеджера-керівника.

Неможливо прагнути управляти людьми, якщо і керівникам, і підлеглим цілі цілком не ясні. Управляти - значить спонукати інших до досягнення ясної мети, а не змушувати інших робити те, що вважаєш правильним.

Варто зауважити, що відповідно до простого американського визначення процесу управління “менеджер робить будь-що руками інших”.

Одне з основних завдань менеджера - створити спонукальні стимули для людини, при яких у неї виникає бажання працювати так, щоб сприяти досягненню цілей організації. Звідси випливає ще один принцип менеджера: управляти - значить вести до успіхів інших.

Відомий український економіст Валерій Терещенко підкреслював, що завдання керівника - диригувати, підібрати гарний персонал, розробити організаційну структуру підприємства, вибрати загальний напрям його діяльності, координувати всю роботу, боротися з непродуктивними витратами часу. Слід виділити ще одне з найважливіших завдань - вчитись інтенсивно працювати, поліпшувати якість роботи.

Отже, центральною фігурою в системі менеджменту будь-якої організації є менеджер-керівник, який керує власне організацією (підприємством), якимсь конкретним видом діяльності, функцією, підрозділом, службою, групою людей тощо. Американські дослідники Роберт Блейк та Джейн Моутон підкреслюють, що сильне та ефективне керівництво сприяє створенню атмосфери участі та колективної підтримки цілей діяльності організації, у якій її члени отримують стимул в усуненні перепон і досягненні максимальних результатів.

У своїй роботі я хочу розглянути автоматизоване робоче місце менеджера як комплексне поняття, що включає в себе такий компонент як:

- програмне забезпечення для більш ефективної роботи менеджера.

Детальний розгляд цього компонента допомагає глибше розкрити і зрозуміти самий процес організації роботи менеджера. Висококваліфікований менеджер, що володіє відповідними особистими і діловими якостями може ефективно управляти організацією, оперувати вміло такими категоріями менеджменту, як лідерство, вплив і влада для досягнення поставлених цілей. Це сприяє правильній, збалансованій організації роботи менеджера, тим самим відкриваючи йому шлях до успіху в його діловій кар'єрі.

**1. Проектування програмного забезпечення автоматизованих інформаційних систем (АІС).**

**1.1 Мови проектування.**

Ефективність технологій проектування багато в чому визначається мовами проектування, що забезпечують спілкування фахівців-розроблювачів із засобами автоматизації їхньої праці. Уніфікація мов проектування дозволяє обмінюватися програмними чи засобами, скорочує витрати на освоєння мов і на технологічні засоби автоматизації їхнього використання.

У зв'язку з різноплановістю задач, розв'язуваних на різних технологічних етапах розробки, доцільна взаємозалежна система мов, що включає (у порядку спрощення проблемного орієнтування й ускладнення машинного орієнтування):

Мова керування задачами

Мова підготовки технологічних засобів

Мова специфікацій вимог

Алгоритмічна мова програмування

Макромова програмування

Автокоди (асемблери)

Мови налагодження: у статиці, у реальному часі

Головними вимогами, пропонованими до системи мов проектування, є: технологічність розробки ПС методом модального спадного проектування; одержання надійного ПС; мобільність ПС, тобто витривалість програмних компонентів як для різних об'єктних, так і технологічних ЕОМ; супроводжує ПС протягом усього життєвого циклу.

Вимоги містять у собі також простоту написання програм, пізнаванність їх, зручність спілкування користувача з технологічної ЕОМ у всіх режимах. Раціонально розмежовувати використання засобів мови на різних етапах проектування ПС між різними групами розроблювачів: системними програмістами, настроювачами кросів-систем на конкретні ЕОМ, розроблювачами функціональних програм.

Характеристика мов проектування:

Мовою керування завданнями забезпечуються всі етапи технології. Технологічні системи оснащуються монітором з мовою керування завданнями, у т.ч. керування базою даних у різних режимах. Цім досягаються витривалість технологічної системи й уніфікація керування її роботою. Мова керування завданнями, являє собою набір директив, що мають фіксований синтаксис. Для таких дій, як керування БД, набір директив стандартизований; для інших функціональних підсистем, набір директив визначається їхніми функціями. Елементами є діагностичні повідомлення про виявлені помилки.

Мова підготовки технологічних засобів доступний настроювачам ПС на середовище функціонування. У нього включається розділ, що представляє собою пакет опису загальних типів даних, їхніх атрибутів і машинно-залежних процедур. Мова визначає правила послідовності команд при реалізації операторів алгоритмічної мови чи макромови. Для алгоритмічної мови це можуть бути семантичні проблемно-орієнтовані мови, у яких використовуються деякі конструкції алгоритмічної базової мови, процедури й оператори розгалуження. Мова завдання форм вихідних документів і машинних носіїв визначає розташування інформації на текстових документах (лістінг програми, розподіл пам'яті й ін.) і машинних носіях.

Мова специфікації вимог призначений для оформлення рішень, прийнятих при структурному проектуванні ПС. На ньому специфікуються весь комплекс програм, групи програм і частки програми (процедури), а також пакети даних. У специфікаціях відбиваються основні характеристики програм, зв'язок їх між собою по керуванню й інформації, а також схема функціонування.

Мови програмування підтримують етап розробки програм. До програм ЕОМ пред'являються високі вимоги по ефективному використанню обчислювальних ресурсів. До цієї групи відносяться: алгоритмічні мови, макромови й автокоди.

Алгоритмічні мови при конкретному застосуванні є підмножиною базової мови. Основними властивостями алгоритмічних мов є: типізація мови, можливість визначення нових типів даних, у т.ч. індексованих, комбінованих і посилальних типів із вказівкою обмежень на область значень, можливість семантичного контролю застосування даних різних типів; структурованість програмних компонентів і даних, строге визначення структурних операторів; наявність пакетів, що містять описи глобальних даних, типів і процедур; наявність задач, що забезпечують опис рівнобіжного виконання програм; забезпечення роздільної компіляції приватних програм і пакетів даних. Наявність елементів мови, що набудовуються, (процедур, операцій) прив'язки до конкретного ЕОМ і т.д.

Макромови (машинно-залежні алгоритмічні мови) використовуються для запису програм із застосуванням операторів, що найбільше адекватно відбивають дії груп команд конкретної ЕОМ (арифметики з присвоюванням, порівняння з переходом, організації циклу і перемикача й ін.). До складу макромови входять оператори, що відповідають структурним операторам алгоритмічної мови.

Автокоди (асемблери), у які включаються макрозасоби (системні і структурні макрокоманди), що забезпечують інтерфейс між програмами, записаними на мовах більш високих рівнів, а також структуризацію програм.

Мови, використовувані на етапі налагодження програм забезпечують проведення контролю результатів роботи програми по різним вихідним даним. Цей тип включає: мова налагодження в статиці, що дає можливість задавати вказівки про режими налагодження, вихідні дані і склад вихідних результатів, мова комплексного динамічного налагодження.

**1.2 Етапи розробки програм.**

Етап розробки програм включає:

методичні документи, що містять правила:

запису програм на мовах програмування;

організації взаємодії програм;

розміщення різних частин програми в пам'яті реалізуючої ЕОМ;

специфікації вимог на програмні модулі, що дозволяє визначити структуру, функції модуля і його зв'язок з іншими модулями ПС, специфікація модуля містить:

заголовок, що доцільно записувати у тому ж виді, як він прийнятий для мов програмування, тобто включати в нього ім'я модуля, імена і типи формальних параметрів і коментар;

паспорт модуля, що містить опис усіх вхідних і вихідних глобальних даних, викликаємих модулів; сюди ж включаються дані про мову програмування й орієнтовані значення часу виконання й обсяг модуля;

функції модуля;

специфікації вимог на глобальні модулі даних складаються одночасно зі специфікаціями на програмні модулі;

вони містять глобальні перемінні, об'єднані в чи структури глобальні константи (форми аналогічні, але в специфікаціях на модулі даних відсутній розділ функціональної схеми і містяться тільки описи даних чи значення констант);

програми на мовах програмування розробляються у відповідності зі специфікаціями з застосуванням методів структурного програмування.

Після виконання процедур запису програми в бібліотеку про проводиться контроль вихідного тексту для виявлення помилок, зв'язаних з порушенням правил розробки програм.

1.3 Методи контролю програм.

Контроль складається в перевірці вхідної програми на відповідність деяким формальним правилам; він підрозділяється на лексичний, синтаксичний і семантичний.

Логічний контроль тексту вирішує задачі виявлення символів, що не належать до алфавіту вхідної мови і груп виділених символів, що не належать до системних символів і символів конкретного представлення мови. Звичайно лексичний контроль сполучається з кодуванням символів у внутрішнє представлення транслятора.

Синтаксичний контроль перевіряє вхідний текст на відповідність синтаксису мови, заданій у його формальному описі. Одним з методів є контроль бінарних відносин, тобто виявлення таких пар символів, що у мові поруч записані бути не можуть.

Семантичний контроль перевіряє правильність застосування конструкцій у конкретному записі. При програмному методі перевірка правильності застосування конструкцій виробляється семантичними програмами, логіка яких заснована на неформалізованих правилах синтаксису і семантики мови.

Методи розміщення перемінних використовуються з застосуванням щільного упакування в пам'яті багаторозрядних ЕОМ. Використовуються такі способи розміщення при який апаратна вибірка реалізується мінімальним числом чи команд за найменший час.

Серед перемінних, використовуваних у програмі, виділяються вхідні і вихідні параметри, якими обмінюються програми, що безпосередньо викликають один одного. Обмін інформацією з вхідних і вихідних параметрів найкраще реалізується при наявності магазина, у який параметри завантажуються і зчитуються в порядку, зазначеному в списку формальних параметрів у заголовку програми. При відсутності магазина вводиться угода про розміщення параметрів у робочому полі.

**1.4 Методи оптимізації програм.**

Для одержання при трансляції програм з малим коефіцієнтом розширення (тобто ефективно використовують пам'ять і продуктивність реалізуючих ЕОМ), необхідно здійснювати оптимізацію програм з використанням наступних методів: уведення у вхідну мову засобів, що дозволяють програмісту здійснювати найбільш ефективний запис програм чи давати транслятору вказівки про методи оптимізації; введення обмежень на використання погано програмувальних конструкцій вхідної мови для конкретних ЕОМ чи складання інструкцій програмісту по застосуванню мови з метою одержання оптимальних програм, зокрема по включенню послідовності операторів мови більш низького рівня; включення оптимізаційних блоків у транслятори. Автоматичні машинно-незалежні методи оптимізації включають: локальні, проведені в межах оператора (лінійної ділянки програми; глобальні, потребуючі побудови графа програми й організації його перегляду по тим чи іншим ознакам, іменам перемінних).

Комплексування містить у собі організацію взаємодії за інформацією і керуванням при складанні комплексу програм з окремих програмних модулів, розроблювальних незалежно.

Комплексування за інформацією здійснюється через глобальну пам'ять перемінних і констант із єдиною ідентифікацією величин у всіх програмах.

У загальному випадку комплексування приводить до необхідності перетрансляції чи частини всіх програм після присвоєння їм нових початкових адрес, що є дуже трудомістким процесом, що вимагає контролю правильності результатів цієї процедури. Щоб уникнути перетрансляції, застосовуються абсолютні і відносні методи.

При абсолютних методах коректування програм виробляється з використанням вставок. У програму вставляються команди переходу до вставки, що містить команди зміни програми і команду повернення.

Відносні методи застосовуються з записом інформації в пам'ять команд, що істотно спрощує проблему взаємного розташування програм. Конкретно це втілено в методах завантаження об'єктних програм з редагуванням зв'язків. Редагуванню піддаються команди внутрішніх і зовнішніх передач чи керування команди, що використовують глобальні перемінні і мітки.

Автономна пересунення програм може бути доповнена взаємної пересуненням. Для цього передача керування на інші програми здійснюється через масиви виклику, що містять початкові адреси програм (у програмі використовуються команди передачі керування по вмісту чи пам'яті регістрів), або команди передачі керування на входи програм (у програмі використовуються команди безпосередньої передачі керування). При зміні взаємного розташування програм змінюється тільки вміст масивів виклику, а в самих програмах не виробляється редагування зовнішніх зв'язків.

Найбільш ощадливим методом взаємодії програм по керуванню є метод безпосередньої передачі керування на вхід викликуваної програми. У цьому випадку при зміні взаємного розташування програм потрібно обов'язкове редагування команд передачі керування викликуваній чи програмі адресних констант.

**1.5 Методи завантаження модулів.**

Методи завантаження істотно впливають на час реалізації завантаження, що має особливо велике значення при роботі в діалоговому режимі. Після коректування програми в загальному випадку потрібно зробити перезавантаження всіх програм з редагуванням внутрішніх і зовнішніх зв'язків.

Завантаження модулів може відбуватися по декількох стратегіях:

у порядку їхнього збереження в бібліотеці;

у послідовності заздалегідь привласнених їм номерів;

відповідно до ієрархії підпорядкованості.

1.6 Засоби автоматизації розробки програм.

Завдання трансляції полягає в аналізі розробленого програмістом вхідного тексту програми, записаного мовою програмування, його контролі і перетворенні у вихідний текст, яким може бути або програма для реалізуючої ЕОМ, або проміжна мова. Система трансляційних засобів складається з взаємозалежних трансляторів з кожної мови програмування. Така система дозволяє виключити дублювання функцій і компонент трансляторів з мов нижніх рівнів у трансляторах з мов верхнього рівня. Для одержання малого розширення програм вимагаються транслятори з алгоритмічних мов програмування, що мають багатопросмотровую структуру.

Транслятор з алгоритмічної мови вирішує наступні задачі:

лексичний контроль проводиться при першому перегляді тексту, при цьому виконується ряд інших функцій транслятора, зв'язаних з виділеними в процесі синтаксичного аналізу конструкціями мови, - складання списку глобальних перемінних, міток і ін.;

розподіл пам'яті локальних перемінних програмних модулів виробляється транслятором по фіксованій для реалізуючої ЕОМ логіку з використанням типу, довжини і розмірності масиву, зазначених в описі перемінних; організація локальної пам'яті значною мірою визначається структурою ПЕОМ і залежить від способів адресації, наявності апаратної вибірки перемінного числа бітів, системи модифікації і т.д.;

семантичний контроль використання величин у різних конструкціях програми здійснюється після того, як визначені характеристики усіх величин, застосовуваних у програмі (локальних і глобальних перемінних, констант);

оптимізація програм проводиться по тексту вхідною мовою чи проміжною мовою, структура якого пристосована для рішення даної задачі;

генератор команд формує машинну команду з її складових, інформація про які отримана в результаті трансляції програми; будь-яка машинна команда може бути представлена як сукупність полів: код операції, операнд, база, індексація, ознаки типу адресації, ознаки умов;

завантажник, використовуючи редактор зв'язків, робить комплексування або всі об'єктні модулі, що містяться в бібліотеці, або частини модулів, що підлягають налагодженню.

Результатом трансляції програм є модуль. Модулі складаються з керуючої й інформаційної частин.

Керуюча частина містить дані, що використовуються завантажником для редагування і завантаження програми (початкова адреса трансляції, словники переміщень і зовнішніх імен, різні ознаки, що характеризують структуру програми і методи завантаження).

Інформаційна частина містить програму модуля в кодах реалізуючої ЕОМ у формі, необхідної для роботи завантажника.

Існують наступні типи модулів: абсолютний - інформаційна частина модуля набудована на те місце пам'яті, де він буде виповнюватися; об'єктний - результат трансляції програми, у керуючій частині містить інформацію, по якій її інформаційна частина редагується по внутрішніх і зовнішніх зв'язках; завантажувальний - результат об'єднання декількох об'єктних модулів в один модуль, готовий до виконання.

**2. Опис програми “Price for Print”**

**2.1 Короткий опис підприємства де буде експлуатуватися програма.**

Дана програма була розроблена спеціально для підприємства “Cross Print”, на якому я проходив виробничу практику. Це підприємство займається випуском друкарської продукції (брошур, буклетів, книг). Швидко зростаючі потреби підприємства і кількість клієнтів змушує його впроваджувати в процес своєї роботи нові технології такі як розробка автоматизованих систем, автоматизованих робочих місць (АРМ) із прикладним програмним забезпеченням.

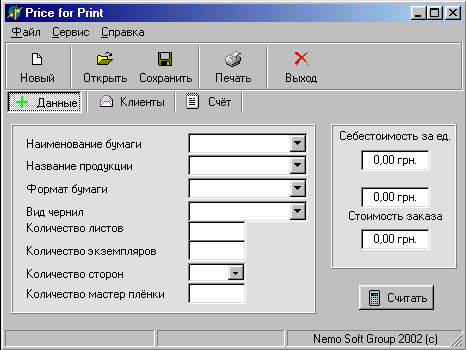
Програма “Price for Print” покликана прискорити роботу менеджера з продажу підприємства “Cross Print” тобто фактично автоматизувати його роботу.

2.2 Інтерфейс програми.

“Price for Print” призначена для підрахунку ціни за надані друковані послуги.

Програма складається з трьох форм:

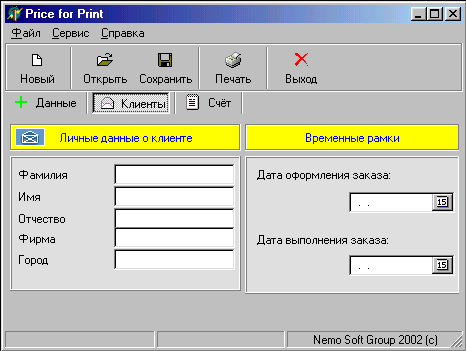
Головна форма “Замовлення”



Мал. 1 Головна форма, сторінка 1 “Замовлення”

На головній формі розташовані три сторінки на мал.1 представлена 1 сторінка “Замовлення”. На цій сторінці менеджер (користувач) вводить дані по яких будить розраховуватися ціна замовлення.

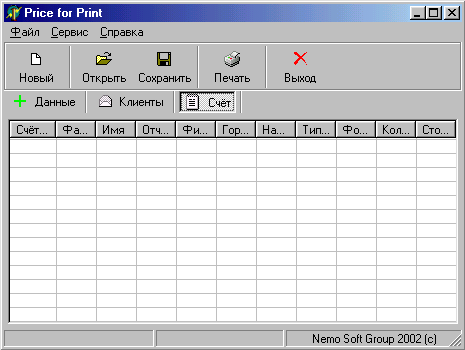
Головна форма “Клієнти”



Мал. 2 Головна форма, сторінка 2 “Клієнти”

На другій сторінці “Клієнти” менеджер (користувач) вводить дані про клієнта, що зробив замовлення і часові рамки (дата оформлення замовлення, дата виконання замовлення).

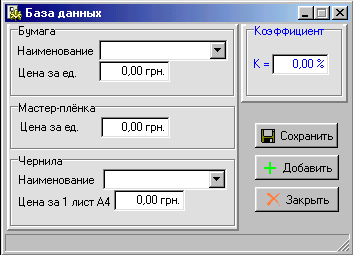
Головна форма “Рахунок”



Мал. 3 Головна форма, сторінка 3 “Рахунок”

На сторінці три “Рахунок” видається звітна інформація з вартості замовлення, про замовника, про найменування продукції і матеріалів.

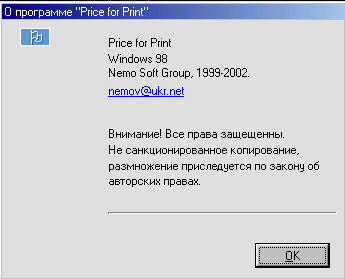
Друга форма “База даних”



Мал. 4 Форма 2 “База даних”

В другий формі “База даних” можна зробити запис у базу, даних які використовуються при обчисленні ціни замовлення.

Третя форма “Про програму”



Мал. 5 Форма 3 “Про програму”

Остання третя форма представляє менеджеру (користувачу) інформацію про розробника програми і попереджає про захист усіх прав.

3. Розрахунки.

3.1 Розрахунок собівартості однієї одиниці продукції.

Для того щоб одержати усю вартість замовлення необхідно знайти спочатку собівартість однієї одиниці продукції. У мене в програмі собівартість визначається по наступній формулі:



“Собівартість” = [ (“Вартість папера”\*“Кількість аркушів”+“Вартість чорнила”\*“Кількість аркушів”\*“Кількість сторін”+“Вартість майстер плівки”\*“Кількість майстер плівки”)/“Кількість екземплярів” ]

3.2 Розрахунок вартості замовлення.

Головним вихідним параметром у даній програмі є “Вартість замовлення” до його розрахунку варто підходити дуже ретельно тому що найменша помилка в розрахунках може привести до великих фінансових утрат. Розрахункова формула не дуже складна от її вид:



“Вартість замовлення” = [ “Собівартість”\*“Коефіцієнт корекції”\*“Кількість екземплярів” ]

“Коефіцієнт коректності” - це коефіцієнт який містить у собі різні виробничі витрати (брак, витрати на роботу верстата і т.і.).

**Аналіз результатів**

Вихідні результати програми цілком задовольняють потреби менеджера, підвищують ефективність і швидкість його праці.

Нижче приведені графіки роботи менеджера:



Проаналізувавши ці графіки можна зробити висновок що ефективність роботи менеджера підвищилася в де кілька разів. Це значно підвищує потенціал підприємства в цілому.

**Список литературы**

Кирилов В.В. Основи проектування реляційних баз даних: Навч. посібник. – Спб.: ІТМО, 1994. – 90 с.

Сєргєєв О.П. Borland Delphi. – К.: ДЦКСТ, 1998. – 83 с.

Фаронов В. В. Delphi 5 керівництво програміста. – М.: “Нолидж”, 2001. – 880 с.

Рогач І.Ф., Сендзюк М.А., Антонюк В.А. Інформаційні системи у фінансово-кредитних установах: Навч. посібник. – К.: КНЕУ, 1999. – 261 с.