**Зміст**

Зміст

Вступ

1. Аналітичний розділ

2. Побудова інформаційно-математичної моделі задачі

3. Алгоритм задачі

4. Визначення структури даних

5. Розробка інтерфейсу програми з користувачем

6. Складання код програми

7. Реалізація проекту у візуальному середовищі

8. Аналіз програми

9. Тестування програми

10. Інструкція з експлуатації програми

11. Аналіз отриманих результатів

Висновок

Список літератури

Додаток A

## Вступ

В наш час будь–яка сфера життєдіяльності, що зв’язана з комп’ютерною та іншою високоінтелектуальною технікою стрімко розвивається, тому на підприємстві він є незамінним елементом роботи. Він спрошує виконання великої кількості письмової роботи, розрахунків та збереження інформації, також заощаджує час для пошуку потрібної інформації. На даний момент існує велика кількість різних програм які розроблені саме для того щоб вести облік. Тому багатьом підприємствам які ведуть облік матеріалів, необхідні саме такі програмами, що спрощують величезну кількість роботи. Непотрібно затрачати велику кількість часу на те щоб записати інформацію, просто необхідно відкрити програму і виконати необхідні дії. Існуючі програми можуть вести облік реалізованої продукції, список витрат, дати поставки товару та ще великої кількості інформації, яка подрібна для задоволення певних потреб підприємства. Програма, яка буде розроблена, представлятиме собою облік комп’ютерів в відділах на підприємства. Виконання даного курсового проекту стане фінальним етапом вивчення дисципліни "Програмування та алгоритмічні мови", та дозволить більш повно реалізувати потенціал студента в подальшому навчанні, створювати більш досконалі програмні продукти, та опанувати методи та структури високого рівня програмування з метою отримання найефективнішого використання ресурсів комп’ютера та створення таких додатків, які б полегшували роботу користувача та ефективно використовували його час.

## 1 Аналітичний розділ

Темою даного курсового проекту є: “Облік комп’ютерів та комплектуючих на підприємстві”. Аналізуючи вже існуючі подібні програми можна визначити основні і додаткові вимоги курсового проекту.

До основних вимог можна віднести:

- пошук за всіма комплектуючими комп’ютера;

- пошук по номеру комп’ютера;

- пошук працівника, за яким закріплений комп’ютер.

- сортування за вибраними критеріями;

До додаткових параметрів відноситься:

- зміна імені відділу;

- інвертування списку;

- переміщення комп’ютерів з одного відділу до іншого.

В даному курсовому проекті необхідно розробити програму обліку комп’ютерів. В ній необхідно створити пошук та сортування по різним критеріям, можливість додавання нових відділів та комплектацію комп’ютера, вилучати непотрібну інформацію. Передбачити можливість переміщення комп’ютерів з одного відділу до іншого, також змінювати назву відділу і редагувати комплектацію вже створених комп’ютерів. Усе це буде розроблятись в – Delphi. Середовище програмування Delphi дозволяє швидко і легко створювати програму у якій буде зручний та зрозумілий користувачу інтерфейс, а також легке та доступне управління програмою. Орієнтована мова програмування Delphi має свій набір компонентів, які допоможуть у реалізації та оформлені програми. Delphi має свій текстовий редактор та власний набір конкретних команд, які відрізняють його від інших мов програмування. В програмі обов’язково буде головне меню, в якому будуть представленні команди для роботи з програмою, за допомогою яких можна виконувати конкретні дії. Також буде довідка, для того щоб користувач міг детальніше розібратися з програмою, яка буде представлена перед ним, або просто звернутись до неї по допомогу, коли виникнуть проблеми, які не можна вирішити самостійно. Вагомим недоліком Delphi являється порівняно низька швидкість роботи, а також нераціональне використання ресурсів.

## 

## 2 Побудова інформаційно-математичної моделі задачі

Побудова інформаційної моделі задачі полягає у визначенні вхідних даних, способів їх зібрання та обробки, а також визначення вихідних даних. Для розв’язання будь-якої задачі необхідно скласти її математичну модель, тобто послідовність дій, які необхідно виконати, щоб отримати розв’язок задачі.

Постановка задачі: Скласти програму «Облік комп’ютерів та комплектуючих на підприємстві», в якій буде передбачено можливість створення нових відділів, нової комплектації комп’ютера, видаляти не потрібну інформацію, пошук потрібних даних та збереження результатів роботи у файл.

При створенні проекту дані будуть зберігатися в одному файлі.

Завантаження інформації відбуватиметься автоматично. Усі записи про комплектацію комп’ютера показуватимуться у таблиці для зручнішого відображення. Інформація про існуючі відділи буде відображена у списку. Пошук здійснюватиметься за допомогою введення потрібної інформації в поля для пошуку та буде відображена у вигляді таблиці. Вразі невірного введення інформації програма виведе повідомлення. Сортування комп’ютерів буде відбуватися шляхом вибору із списка параметра, за якими необхідно буде відсортувати дані. Результати відсортованих даних будуть показані в таблиці.

**3 Алгоритм задачі**

## Алгоритм – це чітка послідовність дій, які необхідно виконати для виконання задачі та отримання результату.

Блок-схема алгоритму – це графічне зображення його структури в якому кожний етап процесу оброблення даних подається у вигляді певної геометричної фігури.

Вхід

1

9

Вихід

і=1,n,1

4

Обирання відділу

3

Оновлення

списока

8

Так

i=j

6

Інформація про комп’ютер міняється місцями

7

Так

Ні

Ні

j=i+1,n,1

5

N=0

2



Рисунок 1 – Алгоритм сортування комп’ютерів за номером.

i=х,n,1

4

Вхід

1

Так

**I=Y**

6

Комп’ютер додається в таблицю

7

Вихід

8

Ні

Y=i+1,n,1

5

Очищення

таблиці

3

N=0

2

Рисунок 2 – Алгоритм пошуку комп’ютерів.

Ні

Вхід

1

Так

Вихід

9

Ні

Файл

існує?

2

Відкриття файлу для зчитування

3

Кінець файлу?

4

Зчитування відділу із всіма його комп’ютерами

5

Додавання назви відділу до списоку

6

Завантаження списоку комп’ютерів в таблицю

7

Закриття

файлу

8

Так

Рисунок 3 – Алгорим завантаження даних.

4 Визначення структури даних

При написанні коду програми необхідно буле використовувати безліч змінних різних типів даних: масиви, файлові змінні, записи, а також звичайні змінні.

Таблиця 1. Типи змінних та їх призначення.

|  |  |
| --- | --- |
| Тип | Призначення |
| Integer | Цілочисельний тип діапазоном від -2147483648 до 2147483647 |
| File of Department | Типізований файл |
| Boolean | Логічний тип, який має два значення: true або false |
| Word | Цілочисельний тип діапазоном від 0 до 65838 |
| String | Рядок символів діапазоном від 1 до 255 символів |
| Byte | Цілочисельний тип діапазоном від 0 до 255 |
| Record | Запис. Структорований тип даних |
| Array | Масив. Скінчений набір елементів одного типу |

Таблиця 2. Змінні, їх типи та призначення у програмі.

|  |  |  |  |
| --- | --- | --- | --- |
| Ідентифікатор | Тип змінних | Призначення | |
| SR | Byte | Номер по списку вибраного комп’ютера із таблиці. | |
| Count | Byte | Кількість відділів | |
| ADD | Boolean | При додаванні нового комп’ютера True, а при редагуванні False | |
| Editing | Boolean | Ознака редагування. | |
| Comps | Array | Масив до 100 комп’ютерів в одному відділі | |
| Dep | Array | Масив відділів | |
| VideoCard | | Record | | Дані про відеокарту |
| Processor | | Record | | Дані про процесор |
| MotherBoard | | Record | | Дані про материнську плату |
| RAM | | Record | | Дані про оперативну пам'ять |
| HDD | | Record | | Дані про жорсткий диск |
| Computer | | Record | | Дані про комп’ютер |
| Department | | Record | | Дані про відділ |
| Comps | | Record | | Дані про комп’ютер |

Таблиця 3. Структура записів.

| Назва запису | Поля запису | Тип полів | Призначення |
| --- | --- | --- | --- |
| VideoCard | Speed | Word | Швидкість |
| Capacity | Word | В розмір пам’яті Мб |
| Firma | String | Назва фірми |
| Monitor | Firma | String | Назва фірми |
| Width | Word | Ширина монітора в пікселях |
| Height | Word | Висота монітора в пікселях |
| Diagonal | Byte | Діагональ в дюймах |
| Processor | Firma | String | Назва фірми |
| Frequency | String | Частота процесора в ГГц |
| KernelCount | Byte | Кількість ядер процесора |
| MotherBoard | Firma | String | Назва фірми |
| Frequency | String | Частота в МГц |
| RAM | Firma | String | Назва фірми |
| TypeRam | String | Тип оперативної пам’яті |
| Capacity | Word | Об’єм пам’яті в Мб |
| Frequency | String | Частота в МГц |
| HDD | Firma | String | Назва фірми |
| TypeHDD | String | Тип жорсткого диску |
| Capacity | Word | Об’єм жорсткого диску в Гб |
| Cache | String | Об’єм кешу в Мб |
| Computer | Number | Word | Номер комп’ютера |
| USER | String | Працівник |
| MVC | VideoCard | Відеокарта |
| MM | Monitor | Монітор |
| MP | Processor | Процесор |
| MMB | MotherBoard | Материнська плата |
| MR | RAM | Оперативна пам’ять |
| MH | HDD | Вінчестер |
| Department | Name | String | Назва відділу |
| Director | String | Керівник відділу |
| CompsCount | Byte | Кількість комп’ютерів в відділі |

**5 Розробка інтерфейсу програми з користувачем**

Розробка даної програми буде відбуватись у середовищі програмування Delphi 6. Дана оболонка програмування 32-розрядна і використовується для створення користувацьких програм для Windows. Програма після її компілювання буде повністю відповідати потребностям операційній системі Windows.

Інтерфейс даної програми складається з загальної форми (рис. 4) на якій знаходяться компоненти:

StringGrid1(SG) – Таблиця для відображення даних про характеристики комп’ютера, його персональний номер та працівника який за ним працює.

GroupBox(GB) – Згрупована панель, на якій для зручності розміщуються потрібні елементи.

ComboBox1 – Список в якому розміщується існуючі відділи.

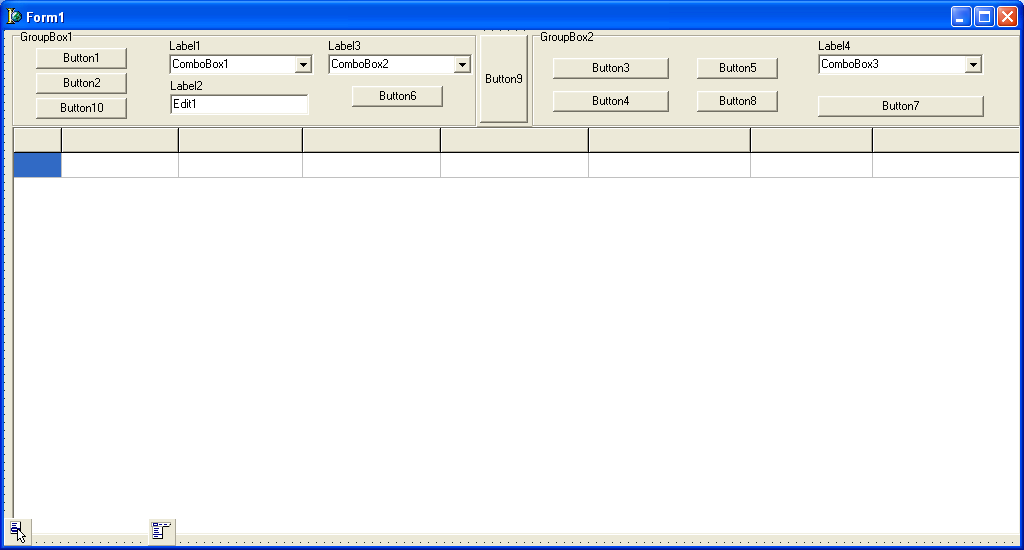


Рисунок 4 – Вікно головної програми.

ComboBox2 – Список сортування по відділу.

ComboBox3 – Список у якому розміщується вже введена інформація, за якими можна відсотувати конфігурацію комп’ютера.

Label1-Labe4 – Надпис. Підписує певний елемент на формі для кращого розуміння.

Button1 – Кнопка за допомогою якої створюється новий відділ.

Button2 – Кнопка знищення вже існуючого відділу і всіх його підгруп.

Edit1 – Поле введення. Відображається відповідальний за відділ.

Button6 – Кнопка яка перевертає список відсортованих даних.

Button10 – Кнопка для зміни назви відділу.

Button3 – Кнопка яка відповідає за створення нової комплектації комп’ютера.

Button4 – Кнопка для видалення існуючої конфігурації комп’ютера із відділу.

Button7 – Кнопка яка перевертає список відсортованих даних.

Button9 – Кнопка пошуку по вибраним критеріям.

Button8 – Кнопка для переміщення групи комп’ютерів в інший відділ.

MainMenu1 – Містить головне меню програми.

PopupMenu1 – Містить контекстне меню таблиці.

На формі 3 розміщуються компоненти які зображені на рисунку 5:

Label1, label2 – Надпис. Підпису спису.

ComboBox1, ComboBox2 – Список в якому міститься назва відділів.

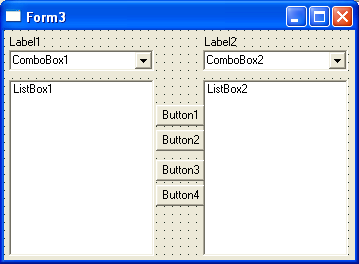


Рисунок 5 – Вікно переміщення комп’ютера з одного відділу у інший.

ListBox1- ListBox2 – Містить номера комп’ютерів вибраного відділу.

Button1- Button4 – Кнопки яка використовуються для перемішення з одного відділу до іншого.

На формі 2 (рис. 6) розміщуються такі компоненти:

Label1-Labe22 – Надпис. Підписує певний елемент на формі.

GroupBox1-GroupBox6 – Згрупована панель на якій розмішуються певні компоненти програми для зручнішого відображення.

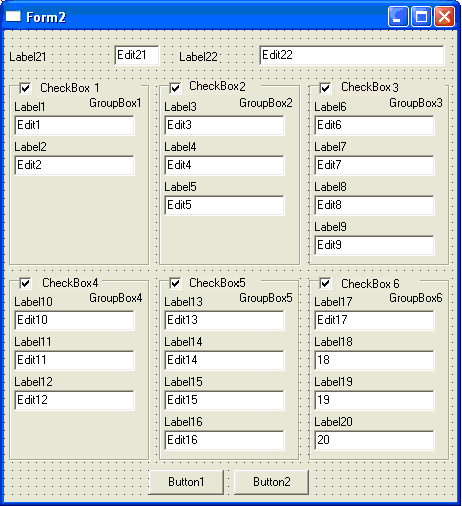


Рисунок 6 – Вікно утворення нової конфігурації комп’ютера.

ChekBox1-ChekBox6 – Прапорці. Відображає компонент комп’ютера який має знаходитися в комплект.

Edit1-Edit22 – Поле введення. Сюди вводяться дані такі як: номер комп’ютера, ім’я працівник, комплектуючі.

Button1- Кнопка підтвердження введення даних.

Button2 – Кнопка яка закриває вікно.

На формі 5 розміщуються компоненти які зображені на рисунку 7:

Edit1 – Поле введення. В нього вписується нова назва відділу.

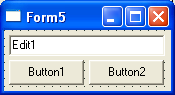


Рисунок 7 – Вікно для зміни імені відділу.

Button1- Кнопка підтвердження введення даних.

Button2 – Кнопка для закриття вікна.

Компоненти що знаходяться на формі 4 розмістити відповідно до рисунку 8.

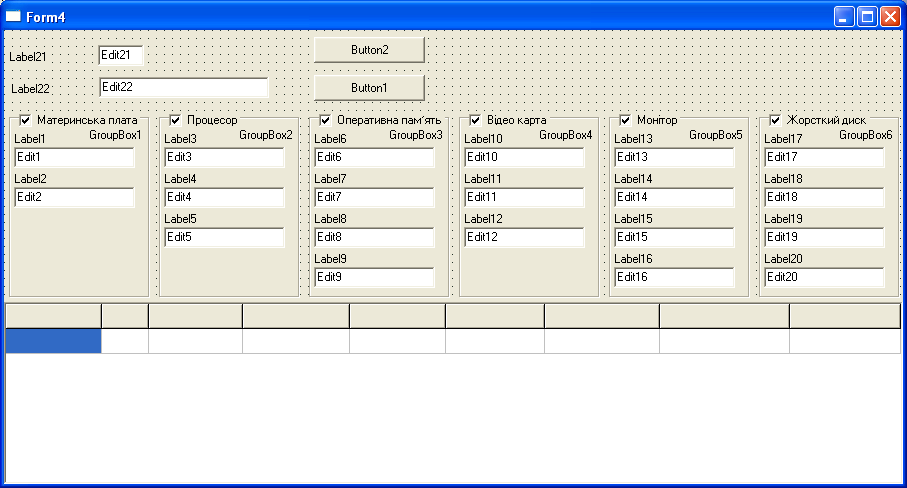


Рисунок 8 – Вікно пошуку потрібних даних.

На формі 4 містяться аналогічні компоненти з відповідними діями які знаходяться на формі 2, додатково: StringGrid1 – Таблиця. Потрібна для відображення даних про характеристики комп’ютера, його персональний номер та працівника який за ним працює.

На формі 6 розміщуються компоненти які зображені на рисунку 9:

TreeView1 – Дерево. Розміщуються розділи справки.

Memo1 – Багатостроковий текст.Відображається докладна інформація про вибраний відділ.

Button – Кнопка яка закриває вікно.

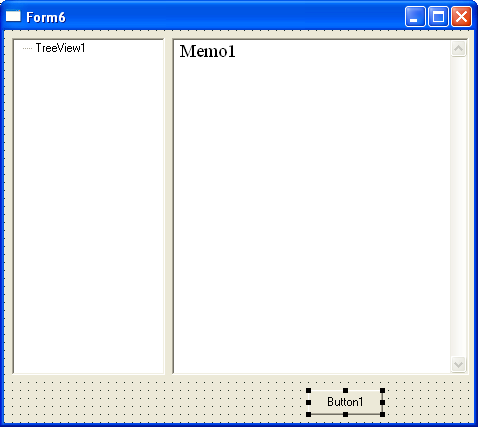


Рисунок 9 – Вікно допомоги користувача.

Форма 7 призначена для відображення інформації про автора (рис.10).

Іmage1 – Зображення. Відображає фотографію автора.

Label1-Label3 – Текст. Призначається для відображення інформації про автора.

Button1 – Кнопка яка закриває вікно.

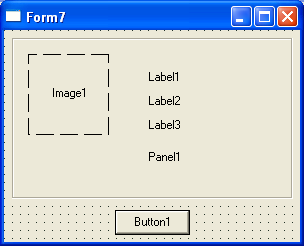


Рисунок 10 – вікно виведення інформації про автора.

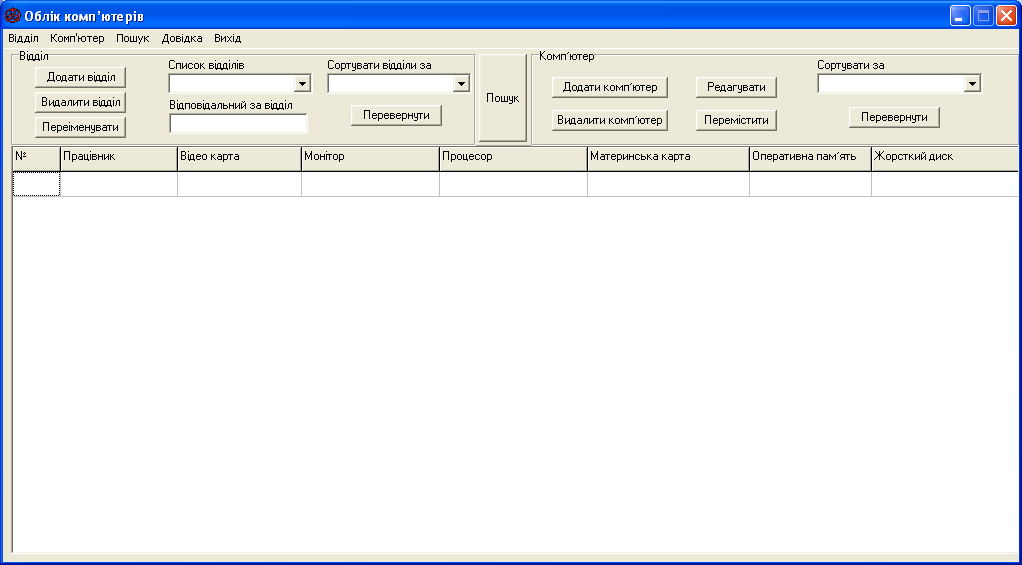


Рисунок 11 – Вікно готової програми.

**6 Складання коду програми**

Код програми – алгоритм дій, написаний на конкретній мові програмування. Цей проект написано з використанням Delphi 6.0, отже мова програмування Object Pascal.

В Delphi 6.0 вся програма складається з окремих модулів.

Створення нової комплектації комп’ютера відбувається таким чином (процедура TForm1.Button3Click):

if ((DEP[ComboBox1.ItemIndex+1].CompsCount>=100)or //Якщо комп’ютерів які знаходяться в одному відділі більше 100 тоді виходимо.

(ComboBox1.ItemIndex<0)) then Exit;

ADD:=True;// активізуємо функцію створення нового коп’ютера.

Form2.Show;// показуємо саму форму створення нового комп’ютера.

Form2.ClearEdit;// очищуємо поля.

Видалення відділу має такий вигля (процедура TForm1. DelDep):

if ((ComboBox1.ItemIndex>=0)and(ComboBox1.Items.Count>0)) then//Якщо відділ вибраний виводиться повідомлення підтвердження видалення.

if MessageDlg('Видалити відділ'+#13#10+'із всіма його комп’ютерами',

mtConfirmation,[mbOk,mbCancel],0)=mrOk then

DelDep(ComboBox1.ItemIndex+1);//Видаляємо відділ зі списку

Вибір відділу зі списку можна описати так (процедура ComboBox1Change):

if ComboBox1.ItemIndex>=0 then// якщо відділ вибраний тоді вивести

ShowDepartament(ComboBox1.ItemIndex+1); //всю інформацію про відділ на екран.

Створення нового відділу відбувається таким чином (процедура TForm1. Button1Click):

if Count>29 then Exit;

for i:=0 to ComboBox1.Items.Count-1 do //Перевірка на наявність відділу

if ComboBox1.Text=ComboBox1.Items.Strings[i] then begin

ShowMessage('Такий відділ вже є!');//виводимо повідомлення що відділ вже існує.

Exit;//виходимо

end;

if ComboBox1.Text<>'' then begin//якщо список не дорівнює пустій назі тоді

INC(Count);//збільшуємо номерацію

…

ShowDepartament(ComboBox1.ItemIndex+1);// показуємо список відділів.

Editing:=True;//дозволяємо редагування.

Редагування інформації про комп’ютер має такий вигляд (процедура Button5Click):

if ((ComboBox1.ItemIndex<0)or(SR<1)or // якщо нічого не вибрано, тоді ми

(SR>DEP[ComboBox1.ItemIndex+1].CompsCount)) then Exit;//виходимо з процедури.

ADD:=False; // забороняємо створення нової інформації

Form2.Show;// показуємо форму 2

Form2.ClearEdit;//очищуємо поля

Лістинг програми наведений в додатку А.

**7 Реалізація проекту у візуальному середовищі**

Реалізація проекту у візуальному середовищі складається з таких етапів:

1. Запустити систему візуального об’єктно-орієнтовного програмування Delphi 6.
2. Створюється 7форм. При створенні форми вони мають стандартні назви.
3. Після створення 1-ї форми, потрібно її налаштувати згідно рисунка 4.
4. За допомогою палітри компонентів перетягується на форму Forma1 наступні компоненти: PopupMenu1, MainMenu1, Button1 – Button10, Label1 – Label4, ComboBox1, ComboBox2, ComboBox3, GroupBox1, GroupBox2, Edit1, SringGrid1. Налаштовуються ці компоненти для нормальної роботи програми.
5. На форму2 встановити такі компоненти: GroupBox1 – GroupBox6, Edit1 – Edit22, Label1 – Label22, Button1, Button2.
6. На форму3 встановити потрібні такі компоненти: ListBox1, ListBox2, Label1 – Label2, ComboBox1 – ComboBox2, Button1 – Button4.
7. На 4-ту форму нанести такі компоненти: GroupBox1 – GroupBox6, Edit1 – Edit22, Label1 – Label22, Button1, Button2, SringGrid1.
8. На 5-ту форму встановити такі компоненти як: Button1, Button2, Edit1.
9. На 6-ту форму перетягнути такі компоненти як: TreeView1, Memo1 та Button1.
10. На 7-му форму встановити такі компоненти як: Image1, три компоненти Label1 – Label3, Рanel1, Button1.
11. Розмістити усі компоненти як показано на рисунках 4 – 10.
12. Після подвійного натискання ввести код програми який приведений у додатку А.
13. Після виконання цих дій програму, у новій папці, потрібно відкомпілювати та закрити Delрhi.

**8 Аналіз програми**

Проаналізувавши програму можна помітити її недоліки та переваги. Головними недоліками програми, являється те, що програма бере свої дані тільки з одного файлу, тобто користувачу не дано можливості зберігати файл під іншою назвою, та зберігати його в будь-яке інше місце. В разі втручання (зміни імені власноруч), програма автоматично створить пустий файл, а попередні дані будуть не доступні.

Також до недоліків можна віднести те, що користувачу не надано можливості для зміни оформлення програми під свій погляд. Він лише може користуватись стандартним виглядом форми, що може не всіх задовольняти.

До переваг програми можна віднести те, що всі дії виконуються правильно. В програмі розроблений зручний інтерфейс. Усе дуже легко зрозуміти, навіть без використання довідки. Виведення даних на екран здійснюється у вигляді таблиці, що досить зручно. В основі програми лежить робота з типізованими файлами, масивами, масивами записів. При запуску програми перед користувачем з’явиться головне вікно де можливо виконати наступні дії:

– Створення нових записи.

– Видалення інформації..

– Сортувати за потрібними йому критеріями.

– Здійснити пошук за вибраними критеріями, покроково для досягнення потрібного результату.

– Переміщувати комп’ютери з одного відділу до іншого.

– При виході з програми можна зберігати данні, або відмовитись від їх збереження.

Також програма володіє довідковою інформацією, до якої користувач може звернутись в будь-який необхідний момент. Кожний об’єкт, який знаходиться на формі підписаний підказками. Це зроблено з тією метою, щоб користувач зміг швидше з орієнтуватися у програмі.

**9 Тестування програми**

Тестування програми – це один із способів перевірити програму на її правильну роботу. Воно допомагає виявити помилки та виправити їх. Також тестування допомагає виявити які необхідні мінімальні запити до комп’ютера.

Насамперед перевіряється чи відкривається і закривається програма. Перевіряється два способи закриття програми за допомогою пункту меню “Вихід”, або комбінації клавіш Alt+F4. Після кількох повторів не виявлено ніяких неполадок при проведені цих дій.

Потім перевірряється програма безпосередньо на готовому працюючому коді. При натиснені кнопки з надписом «Пошук» і введенням вже існуючого номера комп’ютера, шукаються дані. Цей експеримент пройшов успішно, запис знайдено. Але вже при наступному пошуку, в таблиці так і залишається попередній запис, а при пошуку нових даних старі не знищуються. Це можна віднести до збою в програмі. Перевіривши по новому код програми і переправивши деякі команди програма запрацювала нормально, бо при наступному запуску, та перевірці вже було все гаразд. Дані чітко виводилась у таблицю, та очищувались при виборі наступного пошуку. Також тестувалася програма на створення, редагування, видалення, сортування відділу та комплектації комп’ютера.

Дана програма також тестувалась на різних комп’ютерах з різними операційними системами класу Windows. Зокрема програма показала свою швидкодії та коректну роботу на операційній системі Windows XP Professional. Також на даних комп’ютерах було встановлене різне апаратне забезпечення, яке показано конкретні недоліки та не доопрацювання в програмі. В результаті цього тесту, було перевірено, як програма поводить себе на різних комп’ютерах. Вона довго завантажувалась на старих машина, не коректно працювала. Або відкривалась за декілька секунд, на операційних системах, де швидкодія процесора, та оперативної пам’яті досить висока. Тестування пройшло успішно, тому можна сказати що програма працює правельно.

**10 Інструкція з експлуатації програми**

Для нормальної роботи програми комп’ютер повинен відповідати таким умовам:

- Windows 9x/2000/ME/XP

- Pentium III 1000MGz

- 64 Mb Ram

- 32 Mb video card

- CD-ROM

- 2 МВ вільного місця

Для того щоб встановити програму, необхідно вставити диск з програмою, потім скопіювати повністю папку з програмою на комп’ютер, та запустити програму з назвою “ComputerCounter”.

Щоб запустити програму достатньо два рази клацнути на неї, програма знаходиться в папці.

Після запуску програми користувачу потрібно спочатку вибрати що саме він хоче зробити, можна шукати, видаляти, сортувати вже існуючі дані, чи почати створювати нові. Після того як користувач ввів правильні дані, йому потрібно натиснути кнопку пошуку. За допомогою неї здійснюється пошук вказаної інформації. Після чого шукану інформацію буде показано на екран. Якщо інформація яку потрібно знайти, буде введена не вірно то користувач буде повідомлений. Щоб створити новий відділ необхідно поставити курсор миші у поле де розміщенні вже існуючі відділи і ввести назву яку потрібно, після цього необхідно натиснути кнопку з надписом «Додати відділ», тоді у списку відділів з’явиться пустий відділ. Для того щоб змінити назву у вже існуючому відділі, необхідно для початку вибрати відділ для якого потрібно змінити ім’я, після чого натиснути кнопку з надписом «перейменувати», і зроблені дії вступлять в силу. При знищені відділу, необхідно просто вибрати необхідний відділ з відповідною назвою і натиснути кнопку на формі із надписом «Видалити відділ», після чого програма виведе повідомлення і підтвердження видалення. При натисканні кнопки «Ок», відділ із всією існуючою інформацією буде видалений.

11 Аналіз отриманих результатів

Аналіз отриманих результатів проводився після тестування програми. На кожному етапі створення відділу, назва додається до списку вже існуючих відділів, а при видаленні – програма виведе повідомлення на підтвердження повного видалення.

Виконаний проект повністю відповідає усім поставленим вимогам, тому з упевненістю можна сказати, що поставленої мети досягнуто.

В програмі можна створити, видалити, відредагувати, відсортувати та здійснити пошук по відділах та комп’ютерах.

Після введення нової назви відділу і її підтвердження, вона з’явиться у списку існуючих відділів. Після чого пересвідчитись в достовірності роботи програми.

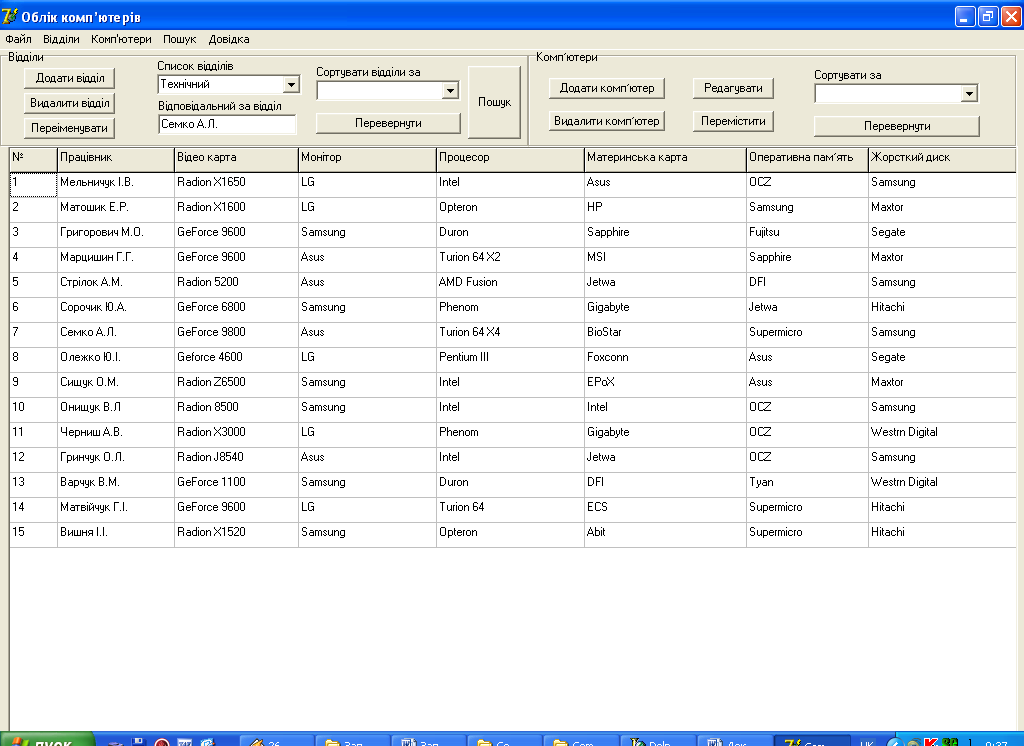


Рисунок 12 – Вікно програми перед початком роботи.

Для переміщення комп’ютера з одного відділу в інший натиснемо:

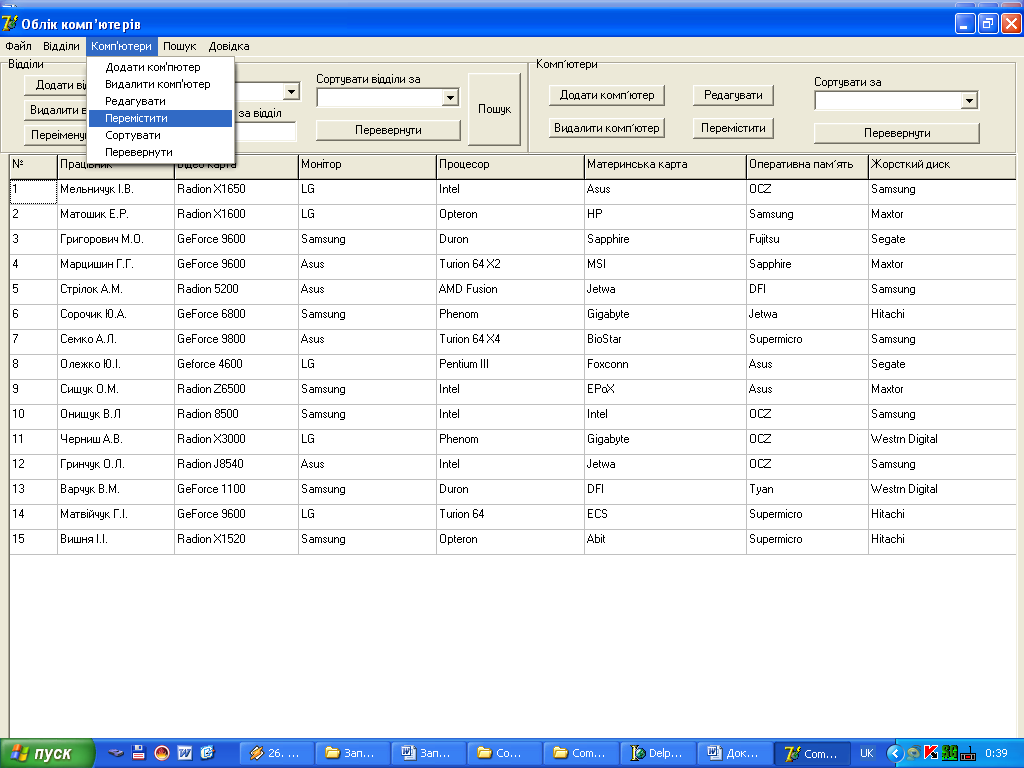


Рисунок 13 – Вибір команди із меню «Комп’ютери→перемістити».

Виберається відділ з якого переміщується (відділ1) у який (відділ2).

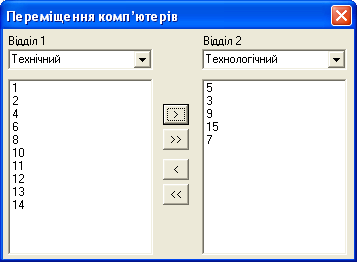
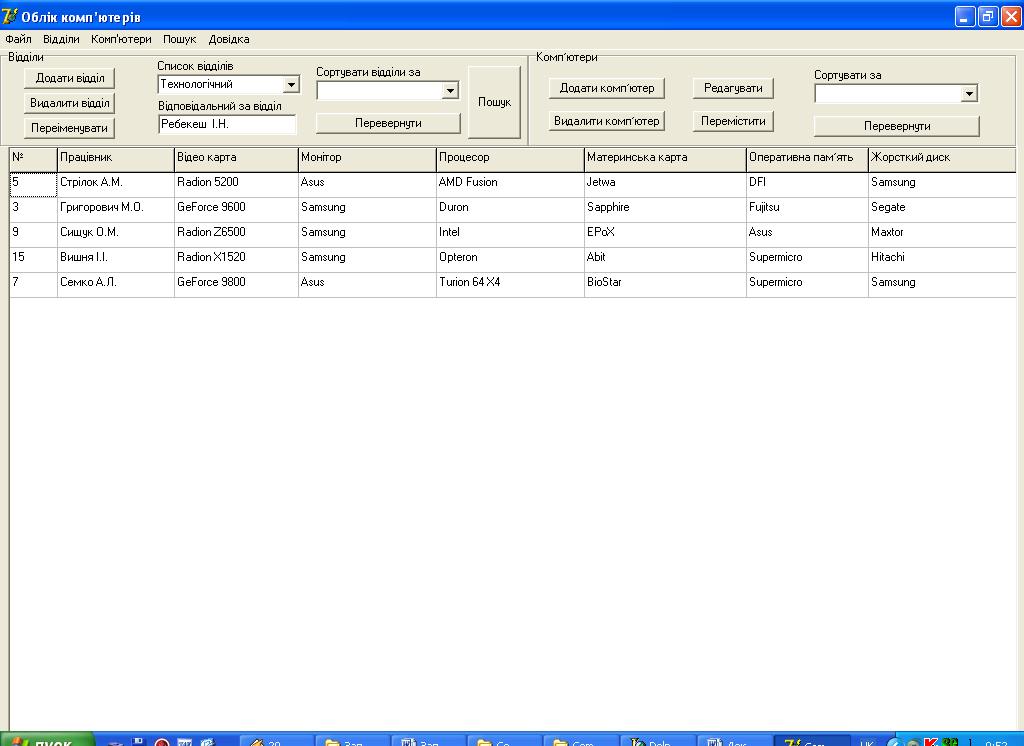


Рисунок 14 – Переміщення комп’ютерів з технічного в технологічний.



Рисинук 15 – Переміщена інформація.

##### Висновок

Під час виконання завдання курсової роботи, було розроблено програму, яка веде облік комп’ютерів на підприємстві, з можливість створення нової, зміни та видалення старої непотрібної інформації.

В процесі виконання курсової роботи, створена аналітична модель програми, процес аналізу завдання розглядається в аналітичному розділі, Створена інформаційно-математична модель програми, де було розроблено етапи роботи програми, і всі ці етапи розписані в інформаційно-математичній моделі, розроблений алгоритм програми, де описано в яких послідовностях повинні виконуватися частини програми, та з якими особливостями. Визначена структура даних, з’ясовані ті змінні, які необхідні для виконання поставленої задачі. В процесі роботи був розроблений інтерфейс користувача, який дозволяє легко та без особливих зусиль користуватися програмою. Складений код програми, відповідно до якого програма повинна виконувати всі поставлені перед нею завдання. Проведений аналіз програми, з’ясовані основні події роботи програми та процедури які їх опрацьовують, відбувалось тестування програми на наявність помилок та недоліків роботи, при тестуванні помилок та недоліків не виявлено, складена інструкція до використання програми та встановленні системні вимоги до персонального комп’ютера.

**Список літератури**

1. В. Фаронов – Система программирования Delphi;
2. Глинський Я.М – Turbo Pascal\Delphi 6 – видання;
3. Кэнту – Delphi для профисионалов;
4. Краснов – Графика в проектах Delphi;
5. Марков Є.П. – Програмированье в Delphi 7;
6. Мозговой М. -Занимательное программирование.
7. Григас Г. – Начала програмируванье. – М.,1987
8. Грогоно П. – Програмируванье на языке паскаль.
9. Спірідонов В.І. – Обчислювальна техніка і програмування.
10. Глинський Я.М. – Основи інформатики та обчислювальної техніки.
11. Абрамов С.А – Начала програмируванья на языке Паскаль.
12. Фаронов В.В. – Delphi 5 учебный курс.
13. Черняхівський В.В – Сучасна технологія візуального програмування.
14. Немнюгин – Turbo Pascal.
15. Климов – Pascal 7.0
16. Юрий Гурский – Трюки и эффекты в Delphi 7.
17. Род Стивенс – Delphi. Готовые алгоритмы.
18. Виктор Пестриков, Артур Маслобоев – Delphi на примерах.
19. Михаил Фленов – Программирование в Delphi глазами хакера.

**Додаток А**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, Menus, StdCtrls;

type

VideoCard=record //Відеокарта

Firma:String[255];

Speed:Word; //Швидкість

Capacity:Word; // В Мб

end;

Monitor=record //Монітор

Firma:String[255];

Width:Word; //Ширина в пікселях

Height:Word; //Висота в пікселях

Diagonal:Byte; //Діагональ в дюймах

end;

Processor=record //Процесор

Firma:String[255];

Frequency:String[5]; //Частота в ГГц

KernelCount:Byte; //Кількість ядер

end;

MotherBoard=record //Материнська плата

Firma:String[255];

Frequency:String[5]; //Частота в МГц

end;

RAM=record //Оперативна пам’ять

Firma:String[255];

TypeRam:String[255]; // DDR, DDR2, DDR3

Capacity:Word; //Об’єм в Мб

Frequency:String[5]; //Частота в МГц

end;

HDD=record //Вінчестер

Firma:String[255];

TypeHDD:String[255]; //ATA, SATA, SATA2...

Capacity:Word; //Об’єм в Гб

Cache:String[5]; //Об’єм кешу в Мб

end;

Computer=record //Комп’ютер

Number:Word; //Номер

USER:String[255]; //Працівник

MVC:VideoCard; //Відеокарта

MM:Monitor; //Монітор

MP:Processor; //Процесор

MMB:MotherBoard; //Материнська плата

MR:RAM; //Оперативна пам’ять

MH:HDD; //Вінчестер

end;

Department=record //Відділ

Name:String[255]; //Назва відділу

Director:String[255]; //Керівник відділу

CompsCount:Byte; //Кількість комп’ютерів в відділі

Comps:array[1..100] of Computer; //Масив до 100 комп’ютерів в одному відділі

end;

TForm1 = class(TForm)

SG: TStringGrid;

GroupBox1: TGroupBox;

ComboBox1: TComboBox;

Label1: TLabel;

Button1: TButton;

Button2: TButton;

Edit1: TEdit;

Label2: TLabel;

ComboBox2: TComboBox;

Label3: TLabel;

Button6: TButton;

Button10: TButton;

GroupBox2: TGroupBox;

Label4: TLabel;

Button3: TButton;

Button4: TButton;

Button5: TButton;

ComboBox3: TComboBox;

Button7: TButton;

Button8: TButton;

MainMenu1: TMainMenu;

N3: TMenuItem;

N4: TMenuItem;

N6: TMenuItem;

N7: TMenuItem;

N8: TMenuItem;

N9: TMenuItem;

N10: TMenuItem;

N11: TMenuItem;

N12: TMenuItem;

N13: TMenuItem;

N14: TMenuItem;

N15: TMenuItem;

N16: TMenuItem;

N17: TMenuItem;

N18: TMenuItem;

N2: TMenuItem;

N19: TMenuItem;

Button9: TButton;

N20: TMenuItem;

N21: TMenuItem;

PopupMenu1: TPopupMenu;

N22: TMenuItem;

N23: TMenuItem;

N24: TMenuItem;

N25: TMenuItem;

N26: TMenuItem;

N27: TMenuItem;

procedure Save;

procedure Load;

procedure FormCreate(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure ShowComp(DepIndex,Index,Row:Byte);

procedure ShowDepartament(Index:Byte);

procedure DelDep(Index:Byte);

function DelComp(DepIndex,Index:Byte):Boolean;

function MoveCompToDep(DepIndex,Index,NewDep:Byte):Boolean;

function MoveAllCompToDep(DepIndex,NewDep:Byte):Boolean;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure ComboBox1KeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure ComboBox1Change(Sender: TObject);

procedure Edit1Change(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure SGSelectCell(Sender: TObject; ACol, ARow: Integer;

var CanSelect: Boolean);

procedure SGDblClick(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure SGKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

procedure RefreshDep;

procedure SortDepByName;

procedure SortDepByCount;

procedure ComboBox2Change(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure SortByNumber;

procedure SortByUserName;

procedure SortByMMB;

procedure SortByMP;

procedure SortByMR;

procedure SortByMVC;

procedure SortByMM;

procedure SortByMH;

procedure SortByFreqProc;

procedure SortByRAMCap;

procedure SortByHDDCap;

procedure SortByDiagonal;

procedure ComboBox3Change(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure Button9Click(Sender: TObject);

procedure Button10Click(Sender: TObject);

procedure N12Click(Sender: TObject);

procedure N11Click(Sender: TObject);

procedure N10Click(Sender: TObject);

procedure N7Click(Sender: TObject);

procedure N18Click(Sender: TObject);

procedure N17Click(Sender: TObject);

procedure N16Click(Sender: TObject);

procedure N15Click(Sender: TObject);

procedure N21Click(Sender: TObject);

procedure N20Click(Sender: TObject);

procedure N14Click(Sender: TObject);

procedure N9Click(Sender: TObject);

procedure N8Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N22Click(Sender: TObject);

procedure N23Click(Sender: TObject);

procedure N25Click(Sender: TObject);

procedure N24Click(Sender: TObject);

procedure N27Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure N19Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

F:File of Department; //Файл запису

Dep:array[1..30] of Department; //Масив відділів

Count:Byte=0; //Кількість відділів

ADD:Boolean=True; //При додавані нового комп’ютера True, а при редагувані False

SR:Byte=0; //Select Row, номер по списку вибраного комп’ютера із таблиці

Editing:Boolean=False; //Ознака редагування

implementation

uses Unit2, Unit3, Unit4, Unit5, Unit6, Unit7;

{$R \*.dfm}

procedure TForm1.Save; //Збереження в файл

var i:Byte;

FileName:String;

begin

FileName:=ExtractFilePath(Application.ExeName)+'Computers'; //Генерація ім’я файлу

AssignFile(F,FileName); ReWrite(F);

for i:=1 to Count do

Write(F,Dep[i]);

CloseFile(F);

end;

procedure TForm1.Load; //Завантаження з файлу

var FileName:String;

begin

FileName:=ExtractFilePath(Application.ExeName)+'Computers'; //Генерація ім’я файлу

if not FileExists(FileName) then Exit; //Перевірка на наявність файлу

AssignFile(F,FileName); Reset(F);

Count:=0;

while not eof(F) do begin

INC(Count);

Read(F,DEP[Count]);

ComboBox1.Items.Add(DEP[Count].Name); //Додаємо назву відділу в список

end;

ComboBox1.ItemIndex:=0;

ShowDepartament(1);

CloseFile(F);

end;

procedure TForm1.FormCreate(Sender: TObject);

var

l: DWORD;

begin

l := GetWindowLong(Self.Handle, GWL\_STYLE);

l := l and not (WS\_MAXIMIZEBOX);

l := SetWindowLong(Self.Handle, GWL\_STYLE, l);

SG.Cells[0,0]:='№';

SG.Cells[1,0]:='Працівник';

SG.Cells[2,0]:='Відео карта';

SG.Cells[3,0]:='Монітор';

SG.Cells[4,0]:='Процесор';

SG.Cells[5,0]:='Материнська карта';

SG.Cells[6,0]:='Оперативна пам’ять';

SG.Cells[7,0]:='Жорсткий диск';

Load;

end;

procedure TForm1.Button3Click(Sender: TObject);

begin

Form2.Caption:='Додати комп`ютер';

if ((DEP[ComboBox1.ItemIndex+1].CompsCount>=100)or

(ComboBox1.ItemIndex<0)) then Exit;

ADD:=True;

Form2.Show;

Form2.ClearEdit;

end;

procedure TForm1.ShowComp(DepIndex,Index,Row:Byte); //Показати комп’ютер в таблиці

begin

if ((Row<1)or(Index<1)or(Index>DEP[DepIndex].CompsCount)) then Exit;

if Row>=SG.RowCount then SG.RowCount:=Row+1;

SG.Rows[Row].Clear;

SG.Cells[0,Row]:=IntToStr(DEP[DepIndex].Comps[Index].Number);

SG.Cells[1,Row]:=DEP[DepIndex].Comps[Index].USER;

SG.Cells[2,Row]:=DEP[DepIndex].Comps[Index].MVC.Firma;

SG.Cells[3,Row]:=DEP[DepIndex].Comps[Index].MM.Firma;

SG.Cells[4,Row]:=DEP[DepIndex].Comps[Index].MP.Firma;

SG.Cells[5,Row]:=DEP[DepIndex].Comps[Index].MMB.Firma;

SG.Cells[6,Row]:=DEP[DepIndex].Comps[Index].MR.Firma;

SG.Cells[7,Row]:=DEP[DepIndex].Comps[Index].MH.Firma;

end;

procedure TForm1.ShowDepartament(Index:Byte); //Вивести всю інф про відділ на екран

var i:Word;

begin

if ((Index>Count)or(Index<1)) then begin

Edit1.Clear;

SG.RowCount:=2;

SG.Rows[1].Clear;

Exit;

end;

for i:=0 to Count-1 do

if ComboBox1.Items.Strings[i]=DEP[Index].Name then begin

ComboBox1.ItemIndex:=i; Break;

end;

if DEP[Index].CompsCount>0 then

SG.RowCount:=DEP[Index].CompsCount+1

else begin

SG.RowCount:=2;

SG.Rows[1].Clear;

end;

Edit1.Text:=DEP[Index].Director;

for i:=1 to DEP[Index].CompsCount do

ShowComp(Index,i,i);

end;

procedure TForm1.DelDep(Index:Byte); //Видалення відділу

var i:Byte;

begin

if ((Count<1)or(Index<1)) then Exit;

for i:=Index to Count-1 do

Dep[i]:=Dep[i+1];

DEC(Count);

ComboBox1.Items.Delete(Index-1);

ComboBox1.ItemIndex:=0;

ShowDepartament(1);

Editing:=True;

end;

function TForm1.DelComp(DepIndex,Index:Byte):Boolean; //Видалення комп’ютера

var i:Byte;

begin

Result:=False;

if ((DepIndex<1)or(DepIndex>Count)) then Exit;

if ((Index<1)or(Index>DEP[DepIndex].CompsCount)) then Exit;

for i:=Index to DEP[DepIndex].CompsCount-1 do

DEP[DepIndex].Comps[i]:=DEP[DepIndex].Comps[i+1];

DEC(DEP[DepIndex].CompsCount);

ShowDepartament(DepIndex);

Result:=True;

Editing:=True;

end;

//Перенесення одного комп’ютера з одного відділу в інший

function TForm1.MoveCompToDep(DepIndex,Index,NewDep:Byte):Boolean;

begin

Result:=False;

if DepIndex=NewDep then Exit;

if ((DepIndex<1)or(DepIndex>Count))or

((NewDep<1)or(NewDep>Count)) then Exit;

if ((Index<1)or(Index>DEP[DepIndex].CompsCount))or

(DEP[NewDep].CompsCount=100) then Exit;

INC(DEP[NewDep].CompsCount);

DEP[NewDep].Comps[DEP[NewDep].CompsCount]:=DEP[DepIndex].Comps[Index];

DelComp(DepIndex,Index);

Result:=True;

Editing:=True;

end;

//Перенесення всіх комп’ютерів з одного відділу в інший

function TForm1.MoveAllCompToDep(DepIndex,NewDep:Byte):Boolean;

var i,C:Byte;

begin

Result:=False;

if DepIndex=NewDep then Exit;

if ((DepIndex<1)or(DepIndex>Count))or

((NewDep<1)or(NewDep>Count)) then Exit;

if ((DEP[NewDep].CompsCount+DEP[DepIndex].CompsCount)>100) then Exit;

C:=DEP[DepIndex].CompsCount;

for i:=1 to C do begin

INC(DEP[NewDep].CompsCount);

DEP[NewDep].Comps[DEP[NewDep].CompsCount]:=DEP[DepIndex].Comps[i];

DelComp(DepIndex,1);

end;

Result:=True;

Editing:=True;

end;

procedure TForm1.Button1Click(Sender: TObject); //Додавання нового відділу

var i:Byte;

begin

if Count>29 then Exit;

for i:=0 to ComboBox1.Items.Count-1 do //Перевірка на наявність відділу

if ComboBox1.Text=ComboBox1.Items.Strings[i] then begin

ShowMessage('Такий відділ вже є!');

Exit;

end;

if ComboBox1.Text<>'' then begin

INC(Count);

Dep[Count].Name:=ComboBox1.Text;

Dep[Count].CompsCount:=0;

ComboBox1.Items.Add(ComboBox1.Text);

ComboBox1.ItemIndex:=ComboBox1.Items.Count-1;

ShowDepartament(ComboBox1.ItemIndex+1);

Editing:=True;

end;

end;

procedure TForm1.Button2Click(Sender: TObject); //Видалення відділу

begin

if ((ComboBox1.ItemIndex>=0)and(ComboBox1.Items.Count>0)) then

if MessageDlg('Видалити відділ'+#13#10+'із всіма його комп’ютерами',

mtConfirmation,[mbOk,mbCancel],0)=mrOk then

DelDep(ComboBox1.ItemIndex+1);

end;

procedure TForm1.ComboBox1KeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if Key=13 then Button1.OnClick(Button1); //Додавання відділу

end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);

begin

if Editing then

if MessageDlg('Зберегти зміни?',mtConfirmation,[mbNo,mbYes],0)=mrYes then

Save;

end;

procedure TForm1.ComboBox1Change(Sender: TObject); //Вибір відділу

begin

if ComboBox1.ItemIndex>=0 then

ShowDepartament(ComboBox1.ItemIndex+1);

end;

procedure TForm1.Edit1Change(Sender: TObject); //Зміна директора відділу

begin

if ComboBox1.ItemIndex>=0 then

DEP[ComboBox1.ItemIndex+1].Director:=Edit1.Text;

end;

procedure TForm1.Button5Click(Sender: TObject); //Редагування інф про комп’ютер

begin

Form2.Caption:='Редагувати комп`ютер';

if ((ComboBox1.ItemIndex<0)or(SR<1)or

(SR>DEP[ComboBox1.ItemIndex+1].CompsCount)) then Exit;

ADD:=False;

Form2.Show;

Form2.ClearEdit;

Form2.RecToEdit;

end;

procedure TForm1.SGSelectCell(Sender: TObject; ACol, ARow: Integer;

var CanSelect: Boolean);

begin

SR:=ARow; //Робимо вибраний комп’ютер поточним

end;

procedure TForm1.SGDblClick(Sender: TObject);

begin

Button5.OnClick(Button5); //Редагування інф про комп’ютер

end;

procedure TForm1.Button4Click(Sender: TObject); //Видалення комп’ютера

var CurD:Byte;

begin

CurD:=ComboBox1.ItemIndex+1;

if ((ComboBox1.Items.Count<1)or(CurD<1)) then Exit;

if ((SR<1)or(SR>DEP[CurD].CompsCount)) then Exit;

if MessageDlg('Видалити комп’ютер',mtConfirmation,[mbOk,mbCancel],0)=mrOk then

DelComp(CurD,SR);

end;

procedure TForm1.SGKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if Key=46 then //Видалення комп’ютера натиском клавіші Del

Button4.OnClick(Button4);

end;

procedure TForm1.RefreshDep; //Оновити список відділів

var i:Byte;

begin

ComboBox1.Clear;

for i:=1 to Count do

ComboBox1.Items.Add(DEP[i].Name);

if ComboBox1.Items.Count>0 then

ComboBox1.ItemIndex:=0;

ShowDepartament(1);

end;

procedure TForm1.SortDepByName; //Сортувати відділи за назвою

var i,j:Byte;

D:Department;

begin

for i:=1 to Count-1 do

for j:=i+1 to Count do

if DEP[i].Name<DEP[j].Name then begin

D:=DEP[i];

DEP[i]:=DEP[j];

DEP[j]:=D;

end;

RefreshDep;

end;

procedure TForm1.SortDepByCount; //Сортувати відділи за кількістю копм’ютерів

var i,j:Byte;

D:Department;

begin

for i:=1 to Count-1 do

for j:=i+1 to Count do

if DEP[i].CompsCount<DEP[j].CompsCount then begin

D:=DEP[i];

DEP[i]:=DEP[j];

DEP[j]:=D;

end;

RefreshDep;

end;

procedure TForm1.ComboBox2Change(Sender: TObject);

begin

//Сортування відділів за вибраним критерієм

if ComboBox1.Items.Count>0 then

case ComboBox2.ItemIndex of

0 : SortDepByName;

1 : SortDepByCount;

end;

end;

procedure TForm1.Button6Click(Sender: TObject); //Інвертувати список відділів

var i,L:Byte;

D:Department;

begin

if Count<=1 then Exit;

L:=Trunc(Count/2);

for i:=1 to L do begin

D:=DEP[i];

DEP[i]:=DEP[Count-i+1];

DEP[Count-i+1]:=D;

end;

RefreshDep;

end;

procedure TForm1.SortByNumber; //Сортування за номером

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].Number>DEP[Cur].Comps[j].Number then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByUserName; //Сортування за ім’ям працівника

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].USER>DEP[Cur].Comps[j].USER then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByMMB; //Сортування за виробником материнської плати

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].MMB.Firma>DEP[Cur].Comps[j].MMB.Firma then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByMP; //Сортування за виробником процесора

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].MP.Firma>DEP[Cur].Comps[j].MP.Firma then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByMR; //Сортування за виробником ОП

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].MR.Firma>DEP[Cur].Comps[j].MR.Firma then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByMVC; //Сортування за виробником відеокарти

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].MVC.Firma>DEP[Cur].Comps[j].MVC.Firma then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByMM; //Сортування за виробником монітору

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].MM.Firma>DEP[Cur].Comps[j].MM.Firma then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByMH; //Сортування за виробником вінчестера

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].MH.Firma>DEP[Cur].Comps[j].MH.Firma then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByFreqProc; //Сортування за частотою процесора

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].MP.Frequency>DEP[Cur].Comps[j].MP.Frequency then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByRAMCap; //Сортування за об’ємом ОП

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].MR.Capacity>DEP[Cur].Comps[j].MR.Capacity then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByHDDCap; //Сортування за об’ємом вінчестера

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].MH.Capacity>DEP[Cur].Comps[j].MH.Capacity then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.SortByDiagonal; //Сортування за розміром діагоналі монітору

var i,j,C,Cur:Byte;

D:Computer;

begin

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

C:=DEP[Cur].CompsCount;

for i:=1 to C-1 do

for j:=i+1 to C do

if DEP[Cur].Comps[i].MM.Diagonal>DEP[Cur].Comps[j].MM.Diagonal then begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[j];

DEP[Cur].Comps[j]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.ComboBox3Change(Sender: TObject);

begin

//Сортування комп’ютерів за вибраним критерієм

case ComboBox3.ItemIndex of

0 : SortByNumber;

1 : SortByUserName;

2 : SortByMMB;

3 : SortByMP;

4 : SortByMR;

5 : SortByMVC;

6 : SortByMM;

7 : SortByMH;

8 : SortByFreqProc;

9 : SortByRAMCap;

10 : SortByHDDCap;

11 : SortByDiagonal;

end;

end;

procedure TForm1.Button7Click(Sender: TObject);

var i,L,Cur:Byte;

D:Computer;

begin

//Інвертувати список комп’ютерів в вибраному відділі

if Count<=1 then Exit;

Cur:=ComboBox1.ItemIndex+1;

if ((Cur<1)or(Cur>Count)) then Exit;

L:=Trunc(DEP[Cur].CompsCount/2);

for i:=1 to L do begin

D:=DEP[Cur].Comps[i];

DEP[Cur].Comps[i]:=DEP[Cur].Comps[DEP[Cur].CompsCount-i+1];

DEP[Cur].Comps[DEP[Cur].CompsCount-i+1]:=D;

end;

ShowDepartament(Cur);

end;

procedure TForm1.Button8Click(Sender: TObject);

begin

if Count>0 then

Form3.Show; //Показати вікно перенесення комп’ютерів

end;

procedure TForm1.Button9Click(Sender: TObject);

begin

Form4.Show;

end;

procedure TForm1.Button10Click(Sender: TObject);

begin

if ((ComboBox1.ItemIndex>=0)and(ComboBox1.Items.Count>0)) then

Form5.Show;

end;

procedure TForm1.N12Click(Sender: TObject);

begin

Button1.Click;

end;

procedure TForm1.N11Click(Sender: TObject);

begin

Button2.Click;

end;

procedure TForm1.N10Click(Sender: TObject);

begin

Button10.Click;

end;

procedure TForm1.N7Click(Sender: TObject);

begin

Button6.Click;

end;

procedure TForm1.N18Click(Sender: TObject);

begin

Button3.Click;

end;

procedure TForm1.N17Click(Sender: TObject);

begin

Button4.Click;

end;

procedure TForm1.N16Click(Sender: TObject);

begin

Button5.Click;

end;

procedure TForm1.N15Click(Sender: TObject);

begin

Button8.Click;

end;

procedure TForm1.N21Click(Sender: TObject);

begin

Button7.Click;

end;

procedure TForm1.N20Click(Sender: TObject);

begin

Button9.Click;

end;

procedure TForm1.N14Click(Sender: TObject);

begin

ComboBox3.SetFocus;

ComboBox3.DroppedDown := True

end;

procedure TForm1.N9Click(Sender: TObject);

begin

ComboBox1.SetFocus;

ComboBox1.DroppedDown := True

end;

procedure TForm1.N8Click(Sender: TObject);

begin

ComboBox2.SetFocus;

ComboBox2.DroppedDown := True

end;

procedure TForm1.N4Click(Sender: TObject);

begin

if Editing then

if MessageDlg('Зберегти зміни?',mtConfirmation,[mbNo,mbYes],0)=mrYes then

Save;

Halt;

end;

procedure TForm1.N22Click(Sender: TObject);

begin

Button3.Click;

end;

procedure TForm1.N23Click(Sender: TObject);

begin

Button4.Click;

end;

procedure TForm1.N25Click(Sender: TObject);

begin

Button5.Click;

end;

procedure TForm1.N24Click(Sender: TObject);

begin

Button8.Click;

end;

procedure TForm1.N27Click(Sender: TObject);

begin

Button9.Click;

end;

procedure TForm1.N2Click(Sender: TObject);

begin

Form6.ShowModal;

end;

procedure TForm1.N19Click(Sender: TObject);

begin

Form7.ShowModal;

end;

end.

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls;

type

TForm2 = class(TForm)

GB1: TGroupBox;

CB1: TCheckBox;

Edit1: TEdit;

Label1: TLabel;

Edit2: TEdit;

Label2: TLabel;

GB2: TGroupBox;

Label3: TLabel;

Label4: TLabel;

Edit3: TEdit;

Edit4: TEdit;

CB2: TCheckBox;

Label5: TLabel;

Edit5: TEdit;

GB3: TGroupBox;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Edit6: TEdit;

Edit7: TEdit;

Edit8: TEdit;

CB3: TCheckBox;

Label9: TLabel;

Edit9: TEdit;

GB4: TGroupBox;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Edit10: TEdit;

Edit11: TEdit;

Edit12: TEdit;

CB4: TCheckBox;

GB5: TGroupBox;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Edit13: TEdit;

Edit14: TEdit;

Edit15: TEdit;

CB5: TCheckBox;

Label16: TLabel;

Edit16: TEdit;

GB6: TGroupBox;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Edit17: TEdit;

Edit18: TEdit;

Edit19: TEdit;

Edit20: TEdit;

CB6: TCheckBox;

Button1: TButton;

Button2: TButton;

Edit21: TEdit;

Label21: TLabel;

Edit22: TEdit;

Label22: TLabel;

procedure ClearEdit;

procedure RecToEdit;

procedure CB1Click(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure CB2Click(Sender: TObject);

procedure CB3Click(Sender: TObject);

procedure CB4Click(Sender: TObject);

procedure CB5Click(Sender: TObject);

procedure CB6Click(Sender: TObject);

function Verification:Boolean;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Edit2KeyPress(Sender: TObject; var Key: Char);

procedure Edit5KeyPress(Sender: TObject; var Key: Char);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses Unit1, Unit4;

{$R \*.dfm}

function STI(S:String):Integer;

var E:Integer;

begin

//Ця функція альтернативна StrToInt, тільки при помилці результат = 0

Val(S,Result,E);

if E<>0 then Result:=0;

end;

procedure TForm2.ClearEdit; //Очищення всіх полів

begin

CB1.Checked:=True; CB2.Checked:=True;

CB3.Checked:=True; CB4.Checked:=True;

CB5.Checked:=True; CB6.Checked:=True;

Edit1.Clear; Edit2.Clear; Edit3.Clear;

Edit4.Clear; Edit5.Clear; Edit6.Clear;

Edit7.Clear; Edit8.Clear; Edit9.Clear;

Edit10.Clear; Edit11.Clear; Edit12.Clear;

Edit13.Clear; Edit14.Clear; Edit15.Clear;

Edit16.Clear; Edit17.Clear; Edit18.Clear;

Edit19.Clear; Edit20.Clear; Edit21.Clear;

Edit22.Clear;

end;

procedure TForm2.RecToEdit; //Показ інф з запису в полях редагування

var CurD:Byte;

begin

CurD:=Form1.ComboBox1.ItemIndex+1;

if CurD<1 then Exit;

Edit1.Text:=DEP[CurD].Comps[SR].MMB.Firma;

Edit2.Text:=DEP[CurD].Comps[SR].MMB.Frequency;

Edit3.Text:=DEP[CurD].Comps[SR].MP.Firma;

Edit4.Text:=DEP[CurD].Comps[SR].MP.Frequency;

Edit5.Text:=IntToStr(DEP[CurD].Comps[SR].MP.KernelCount);

Edit6.Text:=DEP[CurD].Comps[SR].MR.Firma;

Edit7.Text:=DEP[CurD].Comps[SR].MR.TypeRam;

Edit8.Text:=IntToStr(DEP[CurD].Comps[SR].MR.Capacity);

Edit9.Text:=DEP[CurD].Comps[SR].MR.Frequency;

Edit10.Text:=DEP[CurD].Comps[SR].MVC.Firma;

Edit11.Text:=IntToStr(DEP[CurD].Comps[SR].MVC.Speed);

Edit12.Text:=IntToStr(DEP[CurD].Comps[SR].MVC.Capacity);

Edit13.Text:=DEP[CurD].Comps[SR].MM.Firma;

Edit14.Text:=IntToStr(DEP[CurD].Comps[SR].MM.Width);

Edit15.Text:=IntToStr(DEP[CurD].Comps[SR].MM.Height);

Edit16.Text:=IntToStr(DEP[CurD].Comps[SR].MM.Diagonal);

Edit17.Text:=DEP[CurD].Comps[SR].MH.Firma;

Edit18.Text:=DEP[CurD].Comps[SR].MH.TypeHDD;

Edit19.Text:=IntToStr(DEP[CurD].Comps[SR].MH.Capacity);

Edit20.Text:=DEP[CurD].Comps[SR].MH.Cache;

Edit21.Text:=IntToStr(DEP[CurD].Comps[SR].Number);

Edit22.Text:=DEP[CurD].Comps[SR].USER;

end;

procedure TForm2.CB1Click(Sender: TObject);

begin

GB1.Enabled:=CB1.Checked;

end;

procedure TForm2.FormShow(Sender: TObject);

begin

Form1.Enabled:=False;

if OP then

Form4.Enabled:=False;

end;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);

begin

if not OP then

Form1.Enabled:=True

else

Form4.Enabled:=True;

end;

procedure TForm2.CB2Click(Sender: TObject);

begin

GB2.Enabled:=CB2.Checked;

end;

procedure TForm2.CB3Click(Sender: TObject);

begin

GB3.Enabled:=CB3.Checked;

end;

procedure TForm2.CB4Click(Sender: TObject);

begin

GB4.Enabled:=CB4.Checked;

end;

procedure TForm2.CB5Click(Sender: TObject);

begin

GB5.Enabled:=CB5.Checked;

end;

procedure TForm2.CB6Click(Sender: TObject);

begin

GB6.Enabled:=CB6.Checked;

end;

function TForm2.Verification:Boolean;

var Cur,C,i,j:Byte;

begin

Result:=True;

if Edit21.Text='' then begin //Якщо не введено номер комп’ютера

ShowMessage('Введіть номер комп’ютера'); Result:=False; Exit;

end;

Cur:=Form1.ComboBox1.ItemIndex+1;

if ADD then

C:=DEP[Cur].CompsCount+1

else C:=SR;

for j:=1 to Count do //Перевірка номера на наявність

for i:=1 to DEP[j].CompsCount do begin

if ((i=C)and(j=Cur)) then Break;

if StrToInt(Edit21.Text)=DEP[j].Comps[i].Number then begin

ShowMessage('Комп’ютер з таким номером вже є!'); Result:=False; Exit;

end;

end;

if Edit22.Text='' then begin //Якщо не введено номер комп’ютера

ShowMessage('Введіть ПІБ працівника'); Result:=False; Exit;

end;

if CB1.Checked then

if ((Edit1.Text='')or(Edit2.Text='')) then begin //Якщо не введено всіх даних про материнську плату

ShowMessage('Заповніть всі поля материнської плати'); Result:=False; Exit;

end;

if CB2.Checked then

if ((Edit3.Text='')or(Edit4.Text='')or

(Edit5.Text='')) then begin //Якщо не введено всіх даних про процесор

ShowMessage('Заповніть всі поля процесора'); Result:=False; Exit;

end;

if CB3.Checked then

if ((Edit6.Text='')or(Edit7.Text='')or

(Edit8.Text='')or(Edit9.Text='')) then begin //Якщо не введено всіх даних про ОП

ShowMessage('Заповніть всі поля оперативної пам’яті'); Result:=False; Exit;

end;

if CB4.Checked then

if ((Edit10.Text='')or(Edit11.Text='')or

(Edit12.Text='')) then begin //Якщо не введено всіх даних про відеокарту

ShowMessage('Заповніть всі поля відеокарти'); Result:=False; Exit;

end;

if CB5.Checked then

if ((Edit13.Text='')or(Edit14.Text='')or

(Edit15.Text='')or(Edit16.Text='')) then begin //Якщо не введено всіх даних про монітор

ShowMessage('Заповніть всі поля монітора'); Result:=False; Exit;

end;

if CB6.Checked then

if ((Edit17.Text='')or(Edit18.Text='')or

(Edit19.Text='')or(Edit20.Text='')) then begin //Якщо не введено всіх даних про HDD

ShowMessage('Заповніть всі поля жорсткого диска'); Result:=False; Exit;

end;

end;

procedure TForm2.Button1Click(Sender: TObject); //Додати, або змінити інф про комп

var Cur,C:Byte;

begin

if not Verification then Exit;

Cur:=Form1.ComboBox1.ItemIndex+1;

if ADD then begin //Якщо додати комп’ютер тоді збільшуємо кількість комп’ютерів

INC(DEP[Cur].CompsCount);

C:=DEP[Cur].CompsCount;

end else C:=SR;

//Перенесення даних в запис

DEP[Cur].Comps[C].Number:=StrToInt(Edit21.Text);

DEP[Cur].Comps[C].USER:=Edit22.Text;

if CB4.Checked then begin

DEP[Cur].Comps[C].MVC.Firma:=Edit10.Text;

DEP[Cur].Comps[C].MVC.Speed:=STI(Edit11.Text);

DEP[Cur].Comps[C].MVC.Capacity:=STI(Edit12.Text);

end;

if CB5.Checked then begin

DEP[Cur].Comps[C].MM.Firma:=Edit13.Text;

DEP[Cur].Comps[C].MM.Width:=STI(Edit14.Text);

DEP[Cur].Comps[C].MM.Height:=STI(Edit15.Text);

DEP[Cur].Comps[C].MM.Diagonal:=STI(Edit16.Text);

end;

if CB2.Checked then begin

DEP[Cur].Comps[C].MP.Firma:=Edit3.Text;

DEP[Cur].Comps[C].MP.Frequency:=Edit4.Text;

DEP[Cur].Comps[C].MP.KernelCount:=STI(Edit5.Text);

end;

if CB1.Checked then begin

DEP[Cur].Comps[C].MMB.Firma:=Edit1.Text;

DEP[Cur].Comps[C].MMB.Frequency:=Edit2.Text;

end;

if CB3.Checked then begin

DEP[Cur].Comps[C].MR.Firma:=Edit6.Text;

DEP[Cur].Comps[C].MR.TypeRam:=Edit7.Text;

DEP[Cur].Comps[C].MR.Capacity:=STI(Edit8.Text);

DEP[Cur].Comps[C].MR.Frequency:=Edit9.Text;

end;

if CB6.Checked then begin

DEP[Cur].Comps[C].MH.Firma:=Edit17.Text;

DEP[Cur].Comps[C].MH.TypeHDD:=Edit18.Text;

DEP[Cur].Comps[C].MH.Capacity:=STI(Edit19.Text);

DEP[Cur].Comps[C].MH.Cache:=Edit20.Text;

end;

Form1.ShowDepartament(Cur);

Editing:=True;

Close;

end;

procedure TForm2.Button2Click(Sender: TObject);

begin

Close;

end;

procedure TForm2.Edit2KeyPress(Sender: TObject; var Key: Char);

begin

if not (Key in ['0'..'9','.',#8]) then Key:=#0;

end;

procedure TForm2.Edit5KeyPress(Sender: TObject; var Key: Char);

begin

if not (Key in ['0'..'9',#8]) then Key:=#0;

end;

end.

unit Unit3;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls;

type

TForm3 = class(TForm)

ComboBox1: TComboBox;

ComboBox2: TComboBox;

LB1: TListBox;

LB2: TListBox;

Label1: TLabel;

Label2: TLabel;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

procedure FormShow(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure ComboBox1Change(Sender: TObject);

procedure ComboBox2Change(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

uses Unit1;

{$R \*.dfm}

procedure TForm3.FormShow(Sender: TObject);

begin

Form1.Enabled:=False; //Оновлення списку відділів для переміщення

ComboBox1.Items:=Form1.ComboBox1.Items;

ComboBox2.Items:=Form1.ComboBox1.Items;

ComboBox1.ItemIndex:=0;

ComboBox2.ItemIndex:=0;

ComboBox1.OnChange(ComboBox1);

ComboBox2.OnChange(ComboBox2);

end;

procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Form1.Enabled:=True;

end;

procedure TForm3.ComboBox1Change(Sender: TObject);

var i,Cur:Byte;

begin

//Показуємо всі комп’ютери з вибраного відділу

LB1.Clear;

Cur:=ComboBox1.ItemIndex+1;

for i:=1 to DEP[Cur].CompsCount do

LB1.Items.Add(IntToStr(DEP[Cur].Comps[i].Number));

end;

procedure TForm3.ComboBox2Change(Sender: TObject);

var i,Cur:Byte;

begin

//Показуємо всі комп’ютери з вибраного відділу

LB2.Clear;

Cur:=ComboBox2.ItemIndex+1;

for i:=1 to DEP[Cur].CompsCount do

LB2.Items.Add(IntToStr(DEP[Cur].Comps[i].Number));

end;

procedure TForm3.Button1Click(Sender: TObject);

begin

//Переміщення одного комп’ютера

if Form1.MoveCompToDep(ComboBox1.ItemIndex+1,LB1.ItemIndex+1,

ComboBox2.ItemIndex+1) then begin

LB2.Items.Add(LB1.Items.Strings[LB1.ItemIndex]);

LB1.Items.Delete(LB1.ItemIndex);

end;

end;

procedure TForm3.Button2Click(Sender: TObject);

begin

//Переміщення всіх комп’ютерів

if Form1.MoveAllCompToDep(ComboBox1.ItemIndex+1,ComboBox2.ItemIndex+1) then begin

LB2.Items.AddStrings(LB1.Items);

LB1.Clear;

end;

end;

procedure TForm3.Button3Click(Sender: TObject);

begin

//Переміщення одного комп’ютера

if Form1.MoveCompToDep(ComboBox2.ItemIndex+1,LB2.ItemIndex+1,

ComboBox1.ItemIndex+1) then begin

LB1.Items.Add(LB2.Items.Strings[LB2.ItemIndex]);

LB2.Items.Delete(LB2.ItemIndex);

end;

end;

procedure TForm3.Button4Click(Sender: TObject);

begin

//Переміщення всіх комп’ютерів

if Form1.MoveAllCompToDep(ComboBox2.ItemIndex+1,ComboBox1.ItemIndex+1) then begin

LB1.Items.AddStrings(LB2.Items);

LB2.Clear;

end;

end;

end.

unit Unit4;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, StdCtrls, Unit1;

type

TForm4 = class(TForm)

Label21: TLabel;

Label22: TLabel;

GB1: TGroupBox;

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Edit2: TEdit;

CB1: TCheckBox;

GB2: TGroupBox;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

CB2: TCheckBox;

GB3: TGroupBox;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Edit6: TEdit;

Edit7: TEdit;

Edit8: TEdit;

Edit9: TEdit;

CB3: TCheckBox;

GB4: TGroupBox;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Edit10: TEdit;

Edit11: TEdit;

Edit12: TEdit;

CB4: TCheckBox;

GB5: TGroupBox;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

Edit13: TEdit;

Edit14: TEdit;

Edit15: TEdit;

Edit16: TEdit;

CB5: TCheckBox;

GB6: TGroupBox;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Edit17: TEdit;

Edit18: TEdit;

Edit19: TEdit;

Edit20: TEdit;

CB6: TCheckBox;

Edit21: TEdit;

Edit22: TEdit;

SG: TStringGrid;

Button1: TButton;

Button2: TButton;

procedure DeleteRow(Row: Longint);

procedure FormShow(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormCreate(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure ShowComp(DepIndex,Index,Row:Byte);

function NN(D,C:Byte):Boolean;

function MB(D,C:Byte):Boolean;

function MP(D,C:Byte):Boolean;

function MR(D,C:Byte):Boolean;

function MVC(D,C:Byte):Boolean;

function MM(D,C:Byte):Boolean;

function MH(D,C:Byte):Boolean;

function Find:Boolean;

procedure Button1Click(Sender: TObject);

function GetIndex(Row:Integer):TPoint;

procedure SGSelectCell(Sender: TObject; ACol, ARow: Integer;

var CanSelect: Boolean);

procedure SGDblClick(Sender: TObject);

procedure SGKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

OP:Boolean=False; //Якщо вікно відкрите тоді True

implementation

{$R \*.dfm}

type

TCSGrid = class(TStringGrid)

private

public

procedure MoveRow(FromIndex, ToIndex: Longint);

end;

procedure TCSGrid.MoveRow(FromIndex, ToIndex: Longint);

begin

RowMoved(FromIndex, ToIndex); { Захищений метод TStringGrid }

end;

procedure TForm4.DeleteRow(Row: Longint); //Процедура видалення рядків

begin

if ((SG.RowCount=2)and(Row>0)) then SG.Rows[1].Clear;

if ((Row>0)and(SG.RowCount>2)) then

with TCSGrid(SG) do

if (RowCount > 0) and (Row < RowCount) then

begin

if (Row < RowCount - 1) then

MoveRow(Row, RowCount - 1);

Rows[RowCount - 1].Clear;

RowCount := RowCount - 1;

end;

end;

procedure TForm4.FormShow(Sender: TObject);

begin

OP:=True;

Form1.Enabled:=False;

Button2.OnClick(Button2);

end;

procedure TForm4.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Form1.Enabled:=True;

OP:=False;

end;

procedure TForm4.FormCreate(Sender: TObject);

begin

SG.Cells[0,0]:='Відділ';

SG.Cells[1,0]:='№';

SG.Cells[2,0]:='Працівник';

SG.Cells[3,0]:='Відео карта';

SG.Cells[4,0]:='Монітор';

SG.Cells[5,0]:='Процесор';

SG.Cells[6,0]:='Материнська карта';

SG.Cells[7,0]:='Оперативна пам’ять';

SG.Cells[8,0]:='Жорсткий диск';

end;

procedure TForm4.Button2Click(Sender: TObject); //Очищення всіх полів

begin

CB1.Checked:=True; CB2.Checked:=True;

CB3.Checked:=True; CB4.Checked:=True;

CB5.Checked:=True; CB6.Checked:=True;

Edit1.Clear; Edit2.Clear; Edit3.Clear;

Edit4.Clear; Edit5.Clear; Edit6.Clear;

Edit7.Clear; Edit8.Clear; Edit9.Clear;

Edit10.Clear; Edit11.Clear; Edit12.Clear;

Edit13.Clear; Edit14.Clear; Edit15.Clear;

Edit16.Clear; Edit17.Clear; Edit18.Clear;

Edit19.Clear; Edit20.Clear; Edit21.Clear;

Edit22.Clear;

SG.RowCount:=2;

SG.Rows[1].Clear;

end;

procedure TForm4.ShowComp(DepIndex,Index,Row:Byte); //Показати комп’ютер в таблиці

begin

if ((Row<1)or(Index<1)or(Index>DEP[DepIndex].CompsCount)) then Exit;

if Row>=SG.RowCount then SG.RowCount:=Row+1;

SG.Rows[Row].Clear;

SG.Cells[0,Row]:=DEP[DepIndex].Name;

SG.Cells[1,Row]:=IntToStr(DEP[DepIndex].Comps[Index].Number);

SG.Cells[2,Row]:=DEP[DepIndex].Comps[Index].USER;

SG.Cells[3,Row]:=DEP[DepIndex].Comps[Index].MVC.Firma;

SG.Cells[4,Row]:=DEP[DepIndex].Comps[Index].MM.Firma;

SG.Cells[5,Row]:=DEP[DepIndex].Comps[Index].MP.Firma;

SG.Cells[6,Row]:=DEP[DepIndex].Comps[Index].MMB.Firma;

SG.Cells[7,Row]:=DEP[DepIndex].Comps[Index].MR.Firma;

SG.Cells[8,Row]:=DEP[DepIndex].Comps[Index].MH.Firma;

end;

function CMP(S1,S2:String):Boolean; //Порівняння 2-ох рядків

begin

Result:=((S1='')or(S1=S2));

end;

function TForm4.NN(D,C:Byte):Boolean; //Пошук по номеру і імені працівника

begin

Result:=False;

if ((Edit21.Text='')and(Edit22.Text='')) then Exit;

if CMP(Edit21.Text,IntToStr(DEP[D].Comps[C].Number))and

CMP(Edit22.Text,DEP[D].Comps[C].USER) then Result:=True;

end;

function TForm4.MB(D,C:Byte):Boolean; //Пошук по материнській платі

begin

Result:=False;

if ((Edit1.Text='')and(Edit2.Text='')) then Exit;

if CB1.Checked then

if CMP(Edit1.Text,DEP[D].Comps[C].MMB.Firma)and

CMP(Edit2.Text,DEP[D].Comps[C].MMB.Frequency) then Result:=True;

end;

function TForm4.MP(D,C:Byte):Boolean; //Пошук по процесору

begin

Result:=False;

if ((Edit3.Text='')and(Edit4.Text='')and(Edit5.Text='')) then Exit;

if CB2.Checked then

if CMP(Edit3.Text,DEP[D].Comps[C].MP.Firma)and

CMP(Edit4.Text,DEP[D].Comps[C].MP.Frequency)and

CMP(Edit5.Text,IntToStr(DEP[D].Comps[C].MP.KernelCount)) then Result:=True;

end;

function TForm4.MR(D,C:Byte):Boolean; //Пошук по ОП

begin

Result:=False;

if ((Edit6.Text='')and(Edit7.Text='')and(Edit8.Text='')and(Edit9.Text='')) then Exit;

if CB3.Checked then

if CMP(Edit6.Text,DEP[D].Comps[C].MR.Firma)and

CMP(Edit7.Text,DEP[D].Comps[C].MR.TypeRam)and

CMP(Edit8.Text,IntToStr(DEP[D].Comps[C].MR.Capacity))and

CMP(Edit9.Text,DEP[D].Comps[C].MR.Frequency) then Result:=True;

end;

function TForm4.MVC(D,C:Byte):Boolean; //Пошук по відеокарті

begin

Result:=False;

if ((Edit10.Text='')and(Edit11.Text='')and(Edit12.Text='')) then Exit;

if CB4.Checked then

if CMP(Edit10.Text,DEP[D].Comps[C].MVC.Firma)and

CMP(Edit11.Text,IntToStr(DEP[D].Comps[C].MVC.Speed))and

CMP(Edit12.Text,IntToStr(DEP[D].Comps[C].MVC.Capacity)) then Result:=True;

end;

function TForm4.MM(D,C:Byte):Boolean; //Пошук по монітору

begin

Result:=False;

if ((Edit13.Text='')and(Edit14.Text='')and(Edit15.Text='')and

(Edit16.Text='')) then Exit;

if CB5.Checked then

if CMP(Edit13.Text,DEP[D].Comps[C].MM.Firma)and

CMP(Edit14.Text,IntToStr(DEP[D].Comps[C].MM.Width))and

CMP(Edit15.Text,IntToStr(DEP[D].Comps[C].MM.Height))and

CMP(Edit16.Text,IntToStr(DEP[D].Comps[C].MM.Diagonal)) then Result:=True;

end;

function TForm4.MH(D,C:Byte):Boolean; //Пошук по вінчестеру

begin

Result:=False;

if ((Edit17.Text='')and(Edit18.Text='')and(Edit19.Text='')and

(Edit20.Text='')) then Exit;

if CB5.Checked then

if CMP(Edit17.Text,DEP[D].Comps[C].MH.Firma)and

CMP(Edit18.Text,DEP[D].Comps[C].MH.TypeHDD)and

CMP(Edit19.Text,IntToStr(DEP[D].Comps[C].MH.Capacity))and

CMP(Edit20.Text,DEP[D].Comps[C].MH.Cache) then Result:=True;

end;

function TForm4.Find:Boolean; //Пошук

var X,Y,C:Byte;

begin

Result:=False;

SG.RowCount:=2;

SG.Rows[1].Clear;

C:=0;

for X:=1 to Count do

for Y:=1 to DEP[X].CompsCount do

if NN(X,Y) or MB(X,Y) or MP(X,Y) or MR(X,Y) or

MVC(X,Y) or MM(X,Y) or MH(X,Y) then begin

INC(C);

ShowComp(X,Y,C);

end;

if C>0 then Result:=True;

end;

procedure TForm4.Button1Click(Sender: TObject);

begin

if not Find then ShowMessage('Нічого не знайдено!');

end;

function TForm4.GetIndex(Row:Integer):TPoint;

var X,Y:Byte;

begin

Result.X:=0; //Номер відділу

Result.Y:=0; //Номер комп’ютера

if SG.Cells[1,Row]='' then Exit;

for X:=1 to Count do

for Y:=1 to DEP[X].CompsCount do

if DEP[X].Comps[Y].Number=StrToInt(SG.Cells[1,Row]) then begin

Result.X:=X; Result.Y:=Y; Exit;

end;

end;

procedure TForm4.SGSelectCell(Sender: TObject; ACol, ARow: Integer;

var CanSelect: Boolean);

var A:TPoint;

begin

A:=GetIndex(ARow);

if A.X<1 then Exit;

Form1.ComboBox1.ItemIndex:=A.X-1;

Form1.ShowDepartament(A.X);

SR:=A.Y; //Робимо вибраний комп’ютер поточним

end;

procedure TForm4.SGDblClick(Sender: TObject);

begin

Form1.Button5.OnClick(Form1.Button5); //Редагуємо інф про вибраний комп’ютер

end;

procedure TForm4.SGKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

var CurD:Byte;

begin

if Key=46 then begin

//Видаляємо вибраний комп’ютер

CurD:=Form1.ComboBox1.ItemIndex+1;

if ((Form1.ComboBox1.Items.Count<1)or(CurD<1)) then Exit;

if ((SR<1)or(SR>DEP[CurD].CompsCount)) then Exit;

if MessageDlg('Видалити комп’ютер',mtConfirmation,[mbOk,mbCancel],0)=mrOk then

if Form1.DelComp(CurD,SR) then

DeleteRow(SR);

end;

end;

end.

unit Unit5;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls;

type

TForm5 = class(TForm)

Edit1: TEdit;

Button1: TButton;

Button2: TButton;

procedure FormShow(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure Edit1KeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

procedure Button1Click(Sender: TObject);

procedure Edit1KeyPress(Sender: TObject; var Key: Char);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

implementation

uses Unit1;

{$R \*.dfm}

procedure TForm5.FormShow(Sender: TObject);

begin

Form1.Enabled:=False;

end;

procedure TForm5.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Form1.Enabled:=True;

end;

procedure TForm5.Edit1KeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if Key=27 then Button2.OnClick(Button2); //Вихід

end;

procedure TForm5.Button1Click(Sender: TObject);

var I:Byte;

begin

if Edit1.Text='' then begin

ShowMessage('Назва не може бути пустою!'); Exit;

end;

for i:=0 to Form1.ComboBox1.Items.Count-1 do //Перевірка на наявність відділу

if Edit1.Text=Form1.ComboBox1.Items.Strings[i] then begin

ShowMessage('Такий відділ вже є!'); Exit;

end;

I:=Form1.ComboBox1.ItemIndex+1;

if ((I<1)or(I>Count)) then Exit;

Form1.ComboBox1.Items.Strings[I-1]:=Edit1.Text;

DEP[I].Name:=Edit1.Text;

Form1.ComboBox1.ItemIndex:=I-1;

Editing:=True;

Close;

end;

procedure TForm5.Edit1KeyPress(Sender: TObject; var Key: Char);

begin

if Key=#13 then Button1.OnClick(Button1); //Переіменувати

end;

procedure TForm5.Button2Click(Sender: TObject);

begin

Close;

end;

end.

unit Unit6;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ComCtrls;

type

TForm6 = class(TForm)

TreeView1: TTreeView;

Memo1: TMemo;

Button1: TButton;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure TreeView1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form6: TForm6;

implementation

uses Unit1;

{$R \*.dfm}

procedure TForm6.Button1Click(Sender: TObject);

begin

close;

end;

procedure TForm6.Button2Click(Sender: TObject);

begin

Memo1.Lines.SaveToFile('Справка.FSH');

end;

procedure TForm6.TreeView1Click(Sender: TObject);

begin

Memo1.Lines.LoadFromFile('Справка\'+TreeView1.Selected.Text+'.FSH');

end;

end.

unit Unit7;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ExtCtrls;

type

TForm7 = class(TForm)

Panel1: TPanel;

ProgramIcon: TImage;

ProductName: TLabel;

Version: TLabel;

Copyright: TLabel;

OKButton: TButton;

procedure OKButtonClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form7: TForm7;

implementation

uses Unit1;

{$R \*.dfm}

procedure TForm7.OKButtonClick(Sender: TObject);

begin

close;

end;

end.