# ²íôîðìàö³éíî-äîâ³äêîâà ñèñòåìà

Ì²Í²ÑÒÅÐÑÒÂÎ ÎÑÂ²ÒÈ ² ÍÀÓÊÈ ÓÊÐÀ¯ÍÈ

ÍÀÖ²ÎÍÀËÜÍÈÉ ÓÍ²ÂÅÐÑÈÒÅÒ “ËÜÂ²ÂÑÜÊÀ ÏÎË²ÒÅÕÍ²ÊÀ”

Êàôåäðà “²íôîðìàö³éí³ ñèñòåìè òà ìåðåæ³”

**ÊÓÐÑÎÂÀ ÐÎÁÎÒÀ**

ç äèñöèïë³íè “Ïðîáëåìíî – îð³ºíòîâàí³ ìîâè ïðîãðàìóâàííÿ”

íà òåìó:

**“²íôîðìàö³éíî-äîâ³äêîâà ñèñòåìà”**

Âèêîíàâ ñòóäåíò

Ñòóäåí ãðóïè êí -16

Ëîãâèíåíêî Â. ¹0708532

Êåð³âíèê ðîáîòè Êðàâåöü Ï.Î.

ËÜÂ²Â – 2008

**Çàâäàííÿ íà êóðñîâó ðîáîòó**

**Òåìà: ²íôîðìàö³éíî- äîâ³äêîâà ñèñòåìà “Âèù³ íàâ÷àëüí³ çàêëàäè ì. Ëüâîâà”**

Çàâäàííÿ: Âèìîãè äî ïðîãðàìè:

1) ïðîãðàìà ïîâèííà áóòè ðåàë³çîâàíà íà ìîâ³ Ñ/Ñ++:

2) ïðîãðàìà ïîâèííà çàáåçïå÷óâàòè ðîáîòó ç ôàéëàìè ñòðóêòóðîâàíèõ äàíèõ (àáî áàçàìè äàíèõ ôîðìóâàííÿ, ÷èòàííÿ, ðåäàãóâàííÿ, ñîðòóâàííÿ òà ïîøóê ³íôîðìàö³¿ çà çàäàíèì êëþ÷åì.

3) ïðîãðàìà ïîâèííà ïðàöþâàòè ó â³êí³ ÌSDOS ï³ä êåðóâàííÿì îïåðàö³éíî¿ ñèñòåìè Windows;

4) ïðîãðàìà ïîâèííà ìàòè ³íòåðôåéñ ó âèãëÿä³ â³êîí òà ìåíþ äëÿ âèáîðó ðåæèì³â ¿¿ ðîáîòè, à òàêîæ ñèñòåìó ¿¿ äîïîìîãè.

5) ïåðåäáà÷èòè ââåäåííÿ/âèâåäåííÿ äàíèõ ç êëàâ³àòóðè òà ç ôàéëó

6) êåðóâàííÿ ðåæèìàìè ðîáîòè ïðîãðàìè âèêîíàòè çà äîïîìîãîþ êëàâ³àòóðè òà ìèø³.

**Çì³ñò**

Âñòóï

1. Ôîðìóëþâàííÿ çàäà÷³

2. Ìåòîäè òà çàñîáè ðîçâ’ÿçóâàííÿ çàäà÷³

3. Àëãîðèòìè ðîçâ’ÿçóâàííÿ çàäà÷³

4. Ïðîãðàìà òà ¿¿ îïèñ

4.1 Íàçâà ïðîãðàìè

4.2 Ïðèçíà÷åííÿ ïðîãðàìè

4.3 Ìîâè ïðîãðàìóâàííÿ, íà ÿêèõ íàïèñàíà ïðîãðàìà

4.4 Ëîã³÷íà ñòðóêòóðà ïðîãðàìè

4.5 Âõ³äí³ òà âèõ³äí³ äàí³

4.6 Ïðîãðàìí³ çàñîáè

4.7 Òåõí³÷í³ çàñîáè

5. Òåõíîëîã³ÿ ïðîãðàìóâàííÿ, âèêîíàííÿ òà â³äëàãîäæåííÿ ïðîãðàìè

5.1 Òåõíîëîã³ÿ ïðîãðàìóâàííÿ ïðîãðàìè

5.2 Òåõíîëîã³ÿ â³äëàãîäæåííÿ ïðîãðàìè

5.3 Òåõíîëîã³ÿ âèêîíàííÿ ïðîãðàìè

6. ²íñòðóêö³ÿ êîðèñòóâà÷åâ³

7. Êîíòðîëüíèé ïðèêëàä òà àíàë³ç ðåçóëüòàò³â êîìï’þòåðíî¿ ðåàë³çàö³¿ ïðîãðàìè

Âèñíîâêè

Ñïèñîê ë³òåðàòóðè

Äîäàòîê 1. Òåêñò ïðîãðàìè íà ìîâ³ Ñ³

Äîäàòîê 2. Ðåçóëüòàòè ðîáîòè ïðîãðàìè

**Âñòóï**

Ìåòîþ êóðñîâî¿ ðîáîòè º çàêð³ïëåííÿ òåîðåòè÷íèõ çíàíü òà ïðàêòè÷íèõ íàâè÷îê ïðîãðàìóâàííÿ, íàáóòèõ ïðè âèâ÷åíí³ äèñöèïë³í „Îñíîâè ïðîãðàìóâàííÿ” òà „Ïðîáëåìíî îð³ºíòîâàí³ ìîâè ïðîãðàìóâàííÿ”.

Â õîä³ âèêîíàííÿ êóðñîâî¿ ðîáîòè ñòóäåíòè ïîâèíí³ íàâ÷èòèñÿ ñàìîñò³éíî ïðàöþâàòè ç ë³òåðàòóðîþ, ðîçðîáëÿòè àëãîðèòìè, çä³éñíþâàòè ¿õ ïðîãðàìíó ðåàë³çàö³þ òà â³ä ëàãîäæåííÿ íà ñó÷àñíèõ êîìï‘þòåðíèõ ñèñòåìàõ.

Çà îñòàíí³ ðîêè â³äáóâñÿ ð³çêèé ñòðèáîê ó ðîçâèòêó êîìï’þòåðíî¿ òåõí³êè é ïðîãðàìíîãî çàáåçïå÷åííÿ ³ç îäíî÷àñíèì ðîçøèðåííÿ ñôåð çàñòîñóâàííÿ ïåðñîíàëüíèõ êîìï’þòåð³â.

Ïðîãðàìóâàííÿ ÿâëÿº ñîáîþ êîìïëåêñ ïî íàïèñàííþ ñèñòåìè ïðîãðàìíèõ êîä³â, ÿê³ çä³éñíþâàòèìóòü êåðóâàííÿ ò³ºþ ÷è ³íøîþ ³íôîðìàö³éíîþ ñèñòåìîþ, âçàºìîä³ÿ ç ÿêîþ çä³éñíþºòüñÿ ÷åðåç ³íôîðìàö³éí³ òåõíîëîã³¿.

Ïðîãðàìíà ðåàë³çàö³ÿ ïîñòàâëåíî¿ çàäà÷³ ìîæå âèêîðèñòîâóâàòèñÿ ÿê ïðèêëàä âèêîðèñòàííÿ ð³çíîìàí³òíèõ ãðàô³÷íèõ ôóíêö³é. À òàêîæ ÿê ïðèêëàä ïðîãðàìóâàííÿ ðîáîòè ç ÷àñîì.

Òåìà ìîº¿ êóðñîâî¿ ðîáîòè º íàäçâè÷àéíî àêòóàëüíîþ ó íàø ÷àñ, îñê³ëüêè äîâ³äêîâà ñèñòåìà çàðàç º äóæå âèëèêèì ïîì³÷íèêîì äëÿ íàâ÷àííÿ ³ ïîøóêó ³íôîðìàö³¿ ó ñâ³ò³ ³íôîðìàö³éíèõ òåõíîëîã³é.

**1. Ôîðìóëþâàííÿ çàäà÷³**

Ó õîä³ äàíî¿ êóðñîâî¿ ðîáîòè ÿ ïîâèíåí áóâ çðîáèòè ³íòåðôåéñ ïðîãðàìè ÿêèé áè çàáåçïå÷óâàâ ä³àëîã: êîì’þòåð-êîðèñòóâà÷ äëÿ ðîáîòè ç áàçîþ äàíèõ (â ìîìó âèïàäêó áàçà äàíèõ ïðî äîâ³äêîâó ñèñòåìó íàâ÷àëüíèõ çàêëàä³â ì. Ëüâîâà). Äàíèé ³íòåðôåéñ ìàâ ñêëàäàòèñÿ ç òàêîãî ìåíþ: ³íôîðìàö³þ ïðî íàâ÷àëüí³ çàêëàäè; ïîøóê: âèâîäèòü äàí³ ïðî îäèí íàâ÷àëüíèé çàêëàä; à òàêîæ âèõ³ä (exit).

ß çðîáèâ òàê ùîá êîðèñòóâà÷ ñàì âèáèðàâ òó ê³ëüê³ñòü äàíèõ ÿêèõ â³í õî÷å ïîäàòè íà âõ³ä, à íå ñòàâèâ íà ê³ëüê³ñòü âõ³äíèõ äàíèõ ïåâí³ îáìåæåííÿ;

Ìåòîä ÷èòàííÿ äàíèõ âèãëÿäàº â íàñòóïíèì ÷èíîì:

Êîðèñòóâà÷ ââîäèòü êëþ÷îâå ñëîâî(íàïðèêëàä íàçâó çàêëàäó) ³ éîìó êîìï'þòåð ïîêàçóº âñ³ äàí³ ïðî çàäàíèé çàêëàä.

Äàíà ïðîãðàìà ìîæå êîìï³ëþâàòèñÿ ó áóäü-ÿêîìó ñåðåäîâèù³ Ñ/Ñ++, ïî÷èíàþ÷è ç Borland C++ 3.1, àáî Turbo C. Ñàìó ïðîãðàìó ÿ ðåàë³çóâàâ íà ìîâ³ Ñ.

Òàêîæ ¿¿ ìîæíà âèêëèêàòè ó îïåðàö³éí³é ñèñòåì³ Windows ³ âîíà áóäå ïðàöþâàòè ó â³êí³ MSDOS.

**2. Ìåòîäè òà çàñîáè ðîçâ’ÿçóâàííÿ çàäà÷³**

Íèæ÷å áóäóòü íàâåäåí³ òåîðåòè÷í³ â³äîìîñò³, ÿê³ äîïîìîæóòü êðàùå óñâ³äîìèòè ñïåöèô³êó ðåàë³çàö³¿ çàäà÷³.

Ñèìâîëüíà êîíñòàíòà ñêëàäàºòüñÿ ç îäíîãî ñèìâîëà ASCII ì³æ àïîñòðîôàìè (''). Ñòðèíãè º îäíèì ç íàéá³ëüø êîðèñíèõ òà âàæëèâèõ òèï³â äàíèõ ìîâè Ñ. Ñèìâîëüíèé ðÿäîê (ñòðèíã) - öå ìàñèâ ñèìâîë³â, çàìêíåíèé ó ëàïêè ("). Â³í ìàº òèï char. Íóëüîâèé ñèìâîë (\0) àâòîìàòè÷íî äîäàºòüñÿ îñòàíí³ì áàéòîì ñèìâîëüíîãî ðÿäêà òà âèêîíóº ðîëü îçíàêè éîãî ê³íöÿ. Ê³ëüê³ñòü åëåìåíò³â ó ìàñèâ³ äîð³âíþº ê³ëüêîñò³ ñèìâîë³â ó ñòðèíãó ïëþñ îäèí, îñê³ëüêè íóëüîâèé ñèìâîë òàêîæ º åëåìåíòîì ìàñèâà. Êîæíà ñòðèíãîâà êîíñòàíòà, íàâ³òü ó âèïàäêó, êîëè âîíà ³äåíòè÷íà ³íø³é ñòðèíãîâ³é êîíñòàíò³, çáåð³ãàºòüñÿ ó îêðåìîìó ì³ñö³ ïàì'ÿò³. ßêùî íåîáõ³äíî ââåñòè ó ðÿäîê ñèìâîë ëàïîê ("), òî ïåðåä íèì òðåáà ïîñòàâèòè ñèìâîë çâîðîòíîãî ñëåøó (\). Ó ñòðèíã ìîæóòü áóòè ââåäåí³ áóäü-ÿê³ ñïåö³àëüí³ ñèìâîëüí³ êîíñòàíòè, ïåðåä ÿêèìè ñòî¿òü ñèìâîë \.

Ïðîòîòèïè âñ³õ ôóíêö³é, ùî ïðàöþþòü ç ðÿäêàìè ñèìâîë³â, ì³ñòÿòüñÿ ó ôàéë³ string.h. Âñ³ ôóíêö³¿ ïðàöþþòü ç ðÿäêàìè, ùî çàê³í÷óþòüñÿ íóëüîâèì ñèìâîëîì.

Â ïðîãðàìóâàíí³ ìàñèâ (àíãë. array) — îäíà ç íàéïðîñò³øèõ ñòðóêòóð äàíèõ, ñóêóïí³ñòü åëåìåíò³â ïåðåâàæíî îäíîãî òèïó äàíèõ, âïîðÿäêîâàíèõ çà ³íäåêñàìè, ÿê³ çàçâè÷àé ðåïðåçåíòîâàí³ íàòóðàëüíèìè ÷èñëàìè, ùî âèçíà÷àþòü ïîëîæåííÿ åëåìåíòà â ìàñèâ³.

Ìàñèâ ìîæå áóòè îäíîâèì³ðíèì (âåêòîðîì), òà áàãàòîâèì³ðíèì (íàïðèêëàä, äâîâèì³ðíîþ òàáëèöåþ), òîáòî òàêèì, äå ³íäåêñîì º íå îäíå ÷èñëî, à êîðòåæ (ñóêóïí³ñòü) ç äåê³ëüêîõ ÷èñåë, ê³ëüê³ñòü ÿêèõ ñï³âïàäàº ç ðîçì³ðí³ñòþ ìàñèâà

Â ïåðåâàæí³é á³ëüøîñò³ ìîâ ïðîãðàìóâàííÿ ìàñèâ º ñòàíäàðòíîþ âáóäîâàíîþ ñòðóêòóðîþ äàíèõ.

Â ïðîãðàìóâàíí³ òà êîìï'þòåðíèõ íàóêàõ ñòðóêòóðè äàíèõ — öå ñïîñîáè îðãàí³çàö³¿ äàíèõ â êîìï'þòåðàõ. ×àñòî ðàçîì ç³ ñòðóêòóðîþ äàíèõ ïîâ'ÿçóºòüñÿ ³ ñïåöèô³÷íèé ïåðåë³ê îïåðàö³é, ÿê³ ìîæóòü áóòè âèêîíàíèìè íàä äàíèìè, îðãàí³çîâàíèìè â òàêó ñòðóêòóðó.

Ïðàâèëüíèé ï³äá³ð ñòðóêòóð äàíèõ º íàäçâè÷àéíî âàæëèâèì äëÿ åôåêòèâíîãî ôóíêö³îíóâàííÿ â³äïîâ³äíèõ àëãîðèòì³â ¿õ îáðîáêè. Äîáðå ïîáóäîâàí³ ñòðóêòóðè äàíèõ äîçâîëÿþòü îïòèì³çóâàòè âèêîðèñòàííÿ ìàøèííîãî ÷àñó òà ïàì'ÿò³ êîìï'þòåðà äëÿ âèêîíàííÿ íàéá³ëüø êðèòè÷íèõ îïåðàö³é.

Ïî õîäó âèêîíàííÿ äàíî¿ êóðñîâî¿ ðîáîòè íàéá³ëüøà ïîòðåáà âèíèêàëà ó âèêîðèñòàíí³ ñòàíäàðòíèõ ãðàô³÷íèõ ôóíêö³é, ÿê³ âõîäÿòü ó äèðåêòèâó <graphics.h>, à òàêîæ ó á³áë³îòå÷íèõ ôóíêö³ÿõ êîíñîëüíîãî ââåäåííÿ - âèâåäåííÿ, ÿê³ çíàõîäÿòüñÿ ó äèðåêòèâ³ <conio.h>.

Ïðèêëàäè ôóíêö³é êîíñîëüíîãî ââåäåííÿ – âèâåäåííÿ, ïðîòîòèïè ÿêèõ çíàõîäÿòüñÿ ó <conio.h>:

*int getch(void);* Ç÷èòóº 1 ñèìâîë ç êëàâiàòóðè áåç âiäîáðàæåííÿ íà åêðàíi i áåç íàòèñêàííÿ êëàâiøi <Enter>.

*int kbhit(void);* Ïîâåðòàº íåíóëüîâå çíà÷åííÿ, ÿêùî ó áóôåði êëàâiàòóðè º íåç÷èòàíi ñèìâîëè, iíàêøå ïîâåðòàº 0.

Äåÿê³ ôóíêö³¿ äëÿ ðîáîòè ç åêðàííèìè â³êíàìè

*void window(int x1, int y1, int x2, int y2);* Ñòâîðþº âiêíî ç êîîðäèíàòàìè âóçëîâèõòî÷îê (x1,y1)-(x2,y2).

*void gotoxy(int x, int y);*

Ïåðåìiùóº êóðñîð â ïîçèöiþ ç êîîðäèíàòàìè x, y.

*void textcolor(int c);*

Âñòàíîâëþº êîëið ñèìâîëiâ.

*void textbackground(int c*);

Âñòàíîâëþº êîë³ð ôîíó (äîïóñêàþòüñÿ êîëüîðè 0-7).

*void clrscr(void);*

Î÷èùàº åêðàí àáî â³êíî.

Ïîáóäîâà ìåíþ ïðîãðàìè òà ìîæëèâ³ñòü âèáîðó âàð³àíò³â ó íüîìó çàáåçïå÷óºòüñÿ çà äîïîìîãîþ òàêèõ îïåðàòîð³â ìîâè Ñ³, ÿê switch òà case, à òàêîæ îïåðàòîðó break, ÿêèé çàáåçïå÷óº âèõ³ä ç âèùåíàâåäåíèõ îïåðàòîð³â.

**3. Àëãîðèòìè ðîçâ’ÿçóâàííÿ çàäà÷³**

Áëîê – ñõåìà ôóíêö³¿ void cursor (int state)



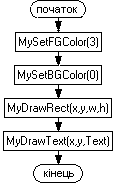
(Ðèñ. 1)

Áëîê – ñõåìà ôóíêö³¿ void ramka



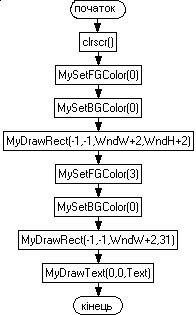
(Ðèñ. 2)

Áëîê – ñõåìà ôóíêö³¿ strukt value (int x, int y, int w, int h, char \*Text) (äèâ. Ðèñ.3)



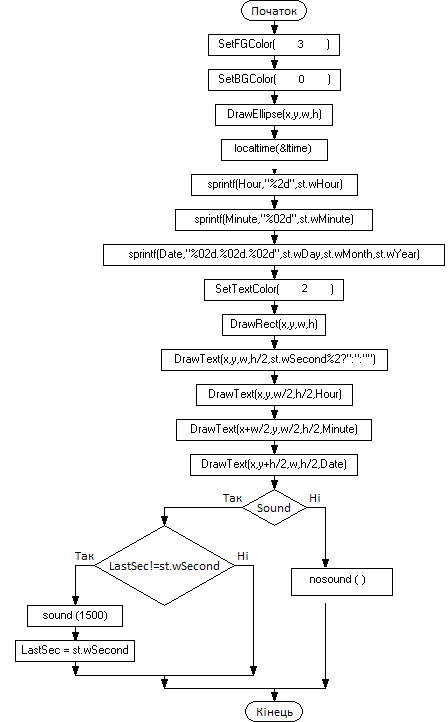
(Ðèñ. 3)

Áëîê – ñõåìà ôóíêö³¿ (äèâ. Ðèñ. 4)



(Ðèñ. 4)

Áëîê – ñõåìà ôóíêö³¿ f =fopen(“ c: \\ vuz.My “,”w+”);



**4. Ïðîãðàìà òà ¿¿ îïèñ**

**4.1 Íàçâà ïðîãðàìè**

²íôîðìàö³éíà - äîâ³äêîâà ñèñòåìà Âèù³ íàâ÷àëüí³ çàêëàäè ì. Ëüâîâà

**4.2 Ïðèçíà÷åííÿ ïðîãðàìè**

Ðîáîòà ç ôàéëàìè ñòðóêòóðîâàíèõ äàíèõ (àáî áàçàìè äàíèõ ôîðìóâàííÿ, ÷èòàííÿ,ðåäàãóâàííÿ, ñîðòóâàííÿ òà ïîøóê ³íôîðìàö³¿ çà çàäàíèì êëþ÷åì. Ìîæå áóòè âèêîðèñòàíà äëÿ ïîêàçó ³íôîðìàö³¿ ïîøóêîâî¿ ñèñòåìè ïðî íàâ÷àëüí³ çàêëàäè íà ïðèêëàä³ ìåòîäèêè ïðî âèù³ íàâ÷àëüí³ çàêëàäè. Ñôåðà çàñòîñóâàííÿ òà âèêîðèñòàííÿ º íåîáìåæåíîþ. Ìîæå áóòè ïðèäàòíîþ äëÿ áóäü-ÿêèõ îðãàí³çàö³é, ï³äïðèºìñòâ, íàâ÷àëüíèõ çàêëàä³â.

**4.3 Ìîâè ïðîãðàìóâàííÿ, íà ÿêèõ íàïèñàíà ïðîãðàìà**

Ïðîãðàìà íàïèñàíà íà ìîâ³ Ñ³.

Ìîâà Ñ³ º ìîâîþ ñåðåäíüîãî ð³âíÿ. Ìîâà Ñ³ ìàº çàñîáè ñèñòåìíîãî ïðîãðàìóâàííÿ, âêëþ÷àº ó ñåáå óñ³ êîíñòðóêö³¿ ñó÷àñíèõ àëãîðèòì³÷íèõ ìîâ, íàéêðàùå âðàõîâóº ìîæëèâîñò³ ñó÷àñíèõ êîìï’þòåðíèõ ñèñòåì, à òàêîæ º ìîá³ëüíîþ ìîâîþ.

**4.4 Ëîã³÷íà ñòðóêòóðà ïðîãðàìè**

Äàíà ïðîãðàìà ñêëàäàºòüñÿ ç äâîõ ôàéë³â: îñíîâíîãî ôàéëó kyrsova.cpp òà ôàéëó kurs.h.

Äàíà ïðîãðàìà ìàº íàñòóïíó ñòðóêòóðó (ï³ä ñòðóêòóðîþ ðîçóì³þòüñÿ ôóíêö³¿ ñêëàäîâèõ ÷àñòèí ïðîãðàìè):

Ôóíêö³¿ ôàéëó kurs.h

Void ramka (int kyt \_vl\_y, int dov\_x, int dov\_y) - ôóíêö³ÿ äëÿ âñòàíîâëåííÿ ðàìêè.

void cursor (int state) êóðñîð ïðîãðàìè.

void ramka(int kyt\_vl\_x,int kyt\_vl\_y,int dov\_x,int dov\_y) - ôóíêö³ÿ äëÿ âñòàíîâëåííÿ ðàìêè êóðñîð â ðàìö³

Ïàðàìåòðè int kyt\_vl\_x,int kyt\_vl\_y,int dov\_x,int dov\_y â³äïîâ³äàþòü çà êîîðäèíàòè ðàìêè.

void refresh\_list()- ôóíêö³ÿ äëÿ âèâîäó ñòâîðåíîãî ôàéëó.

void insert\_list() - ôóíêö³ÿ äëÿ ñòâîðåíîãî ôàéëó ³ çàïèñ íà íüîãî ³íôîðìàö³¿

void MyDrawText(int x,int y,char \*Text) - ôóíêö³ÿ äëÿ âèâîäó òåêñòîâî¿ ³íôîðìàö³¿ íà åêðàí.

void save\_list()- ôóíêö³ÿ äëÿ ñîõðÿíÿííÿ ôàéëó

void delete\_list() - ôóíêö³ÿ äëÿ âèäàëåííÿ ëèñòà çàïèñó ÿêèé áóäå çíàõîäèòèñü ó ìåíþ

void sort\_list(int vub)- ôóíêö³ÿ äëÿ ñîðòóâàííÿ ëèñò³â çàïèñ³â ÿêà áóäå çàïèñàíà ó ìåíþ

Ôóíêö³¿ ôàéëó kyrsova.cpp

void new\_window(int vub) - ôóíêö³ÿ, ÿêà ñòâîðþº ïîòð³áíèé ôàéë

void open\_list() - ôóíêö³ÿ äëÿ â³äêðèòòÿ ôàéëó

void main\_menu() - ôóíêö³ÿ äëÿ ïîáóäîâè ìåíþ.

void main() - îñíîâíà ôóíêö³ÿ, ÷åðåç ÿêó âçàºìîä³þòü óñ³ ³íø³ ôóíêö³¿.

Óñ³ ôóíêö³¿ ó äàí³é ïðîãðàì³ º âçàºìîïîâ’ÿçàí³. Îáì³í ³íôîðìàö³ºþ ì³æ íèìè â³äáóâàºòüñÿ ÿê ÷åðåç ëîêàëüí³, òàê ³ ÷åðåç ãëîáàëüí³ çì³íí³, à òàêîæ ÷åðåç ñïèñîê ôàêòè÷íèõ-ôîðìàëüíèõ ïàðàìåòð³â.

**4.5 Âõ³äí³ òà âèõ³äí³ äàí³**

Âõ³äíèìè äàíèìè ó äàí³é ïðîãðàì³ º ìåíþ è ñîðòóâàííÿ.

Âèõ³äíèìè äàíèìè â äàí³é ïðîãðàì³ º ìèòîäè÷êà ïðî çàêëàäè, ÿêèé ïîêàçóº ³íôîðìàö³¿.

**4.6 Ïðîãðàìí³ çàñîáè**

Áàæàíî ìàòè ñåðåäîâèùå Borland C++ 3.1 àáî Turbo C. À òàêîæ âñòàíîâëåíó îïåðàö³éíó ñèñòåìó Windows. Íà ³øèõ ÎÑ äàíà ïðîãðàìà íå òåñòóâàëàñÿ.

Ïðîãðàìà ïîâèííà ïðàöþâàòè ó â³êí³ MS-DOS ï³ä êåðóâàííÿì Windows.

Àáî çàïóñêàòèñÿ ç ñàìîãî ñåðåäîâèùà.

Äëÿ ðîáîòè ïðîãðàìè íåîáõ³äíà ÎÑ: DOS, Windows 1,2,3; Windows 95/98/ME; Windows 2000; Windows XP;

Äëÿ êîìï³ëþâàííÿ ïðîåêòó íåîáõ³äíèé êîìï³ëÿòîð Borland Ñ (C++), á³áë³îòå÷í³ ôóíêö³¿: *stdio.h*, *conio.h*, *math.h*.,*, stdlib.h,.*

**4.7 Òåõí³÷í³ çàñîáè**

Ïðîãðàìà íå ïîòðåáóº ïîòóæíîãî êîìï’þòåðà, òîìó ç ëåãê³ñòþ ï³äå ïðàêòè÷íî íà óñ³õ ìîäåëÿõ.

**5.Òåõíîëîã³ÿ ïðîãðàìóâàííÿ, âèêîíàííÿ òà â³äëàãîäæåííÿ ïðîãðàìè**

**5.1 Òåõíîëîã³ÿ ïðîãðàìóâàííÿ ïðîãðàìè**

Âèêîðèñòàííÿ ìîäåë³ ó ðîçðîáö³ òà âèð³øåíí³ ñêëàäíèõ çàäà÷ º îäíèì ç íàéâàæëèâ³øèõ ìåòîä³â. Ìîäåëü – öå çàì³ùóâà÷ îá’ºêòó äîñë³äæåííÿ, ùî çíàõîäèòüñÿ ç íèì ó òàê³é â³äïîâ³äíîñò³, ÿêà äîçâîëÿº îòðèìàòè íîâå çíà÷åííÿ ïðî îá’ºêò.

Îñê³ëüêè ìîäåëü ´ðóíòóºòüñÿ íà àíàëîã³¿, òî âîíà ãóáèòü ñåíñ çàñîáó ï³çíàííÿ ÿê ó âèïàäêó òîòîæíîñò³ ìîäåë³ òà îá’ºêòó äîñë³äæåííÿ, òàê ³ ó âèïàäêó äóæå âåëèêèõ â³äì³ííîñòåé ì³æ íèìè. Òàêèì ÷èíîì ìîäåëþâàííÿ ïîâ’ÿçàíå ç³ ñïðîùåííÿì, àáñòðàãóâàííÿì ïðîòîòèïó â³ä ðÿäó éîãî âëàñòèâîñòåé, îçíàê, ñòîð³í. Ïðîòå íàäì³ðíî ñïðîùåíà ìîäåëü ìîæå ïðèçâåñòè äî íåâ³äïîâ³äíîñò³ ç äîñë³äæóâàíèì îá’ºêòîì, ùî óíåìîæëèâèòü äîñë³äæåííÿ éîãî ç äîïîìîãîþ òàêî¿ ìîäåë³. Ç ³íøîãî áîêó, âðàõóâàííÿ â ìîäåë³ ÿêîìîãà á³ëüøî¿ ê³ëüêîñò³ âëàñòèâîñòåé äîñë³äæóâàíîãî îá’ºêòó ïðèçâîäèòü äî óñêëàäíåííÿ ïðîöåñó äîñë³äæåííÿ.

**5.2 Òåõíîëîã³ÿ âèêîíàííÿ ïðîãðàìè**

Âèêîíàííÿ ïðîãðàìè âèêîíóºòüñÿ ÷àñòèíàìè. Ñïî÷àòêó ñòâîðþþòüñÿ ï³äïðîãðàìè, ÿê³ ïîëåãøóþòü ïðîãðàìóâàííÿ íåîáõ³äíî¿ çàäà÷³, ³ ÿê³ áóäóòü âèêîðèñòîâóâàòèñÿ â òåêñò³ îñíîâíî¿ ïðîãðàìè. Â íàøîìó âèïàäêó òàêèìè ï³äïðîãðàìàìè º çâè÷àéí³ ãðàô³÷í³ ôóíêö³¿:

¨   Îïèñ âñ³õ ãðàô³÷íèõ ïðèì³òèâ³â;

¨   Ñòâîðåííÿ çîáðàæåííÿ ãîëîâíîãî ìåíþ.

¨   Ñòâîðåííÿ çîáðàæåííÿ êóðñîðà

¨   Âèçíà÷åííÿ ì³ñöÿ ðîçòàøóâàííÿ îá'ºêòà;

¨   Ñòâîðåííÿ ôàéëó

¨   Çàáåçïå÷åííÿ ðóõó êóðñîðà

¨   Çàáåçïå÷åííÿ âèêîíàííÿ áàçàìè äàíèõ ôîðìóâàííÿ, ÷èòàííÿ, ðåäàãóâàííÿ, ñîðòóâàííÿ òà ïîøóê ³íôîðìàö³¿ çà çàäàíèì êëþ÷åì

¨   Ñòâîðåííÿ ïðîãðàìíîãî ìåíþ;

¨   Çàáåçïå÷åííÿ êåðóâàííÿ ïðîãðàìîþ çà äîïîìîãîþ êëàâ³àòóðè.

**5.3 Òåõíîëîã³ÿ â³äëàãîäæåííÿ ïðîãðàìè**

Äëÿ òîãî, ùîá â³äëàãîäèòè ïðîãðàìó, ïîòð³áíî âèêîðèñòàòè ïîêðîêîâå ï³äêëþ÷åííÿ ï³äïðîãðàì (òàê ìè øâèäøå çðîçóì³ºìî ó ÿê³é ÷àñòèí³ ïðîãðàìè º ïîìèëêè ³ ïîñë³äîâíî âèïðàâèìî ö³ ïîìèëêè). Òîáòî ñïî÷àòêó ìè çàïèñóºìî ï³äïðîãðàìè â îñíîâíèé òåêñò ïðîãðàìè, ïîò³ì, ïðè íàÿâíîñò³ ïîìèëîê, ìè çì³íþºìî ï³äïðîãàìè äî òàêîãî âèäó, ÿêîãî âèìàãàº öåé òåêñò ³ ñèíòàêñèñ ìîâè Ñ³. Ïîò³ì ï³äëàøòîâóºìî ¿õ ï³ä òåêñò ïðîãðàìè ³ ïåðåâ³ðÿºìî ¿õ ðîáî÷ó åôåêòèâí³ñòü. Íàìàãàþòüñÿ çðîáèòè íàéêðàùó åôåêòèâí³ñòü øëÿõîì çì³íè òåêñòó ï³äïðîãðàì.

Ï³ñëÿ óñï³øíîãî ï³ä‘ºäíàííÿ ãîëîâíèõ ï³äïðîãðàì ï³ä‘ºäíóþòüñÿ ³íø³ ï³äïðîãðàìè(òàêèì ÷èíîì, ÿê ³ îñíîâí³ ï³äïðîãðàìè). Íàìàãàþòüñÿ ñòâîðèòè íàéêðàùó âçàºìîä³þ øëÿõîì ïåðåñòàíîâêè àáî çì³íè òåêñòó ï³äïðîãðàì.

**6. ²íñòðóêö³ÿ êîðèñòóâà÷åâ³**

²íòåðôåéñ ïðîãðàìè ðåàë³çîâàíèé ó ãðàô³÷íîìó ðåæèì³ ç âèêîðèñòàííÿì ôóíêö³é çàãîëîâî÷íîãî ôàéëó include<string.h>. ²íòåðôåéñ º çðó÷íèì ó êîðèñòóâàíí³ ³ íå ïîòðåáóº çíà÷íèõ ðåñóðñ³â êîìï’þòåðà äëÿ ðîáîòè ó íüîìó.

Êåðóâàííÿ ïðîãðàìîþ çä³éñíþºòüñÿ ïîâí³ñòþ çà äîïîìîãîþ êëàâ³àòóðè.

ª äâà ñïîñîáè ïî çàïóñêó ïðîãðàìè íà âèêîíàííÿ:

1) Çàïóñê ÷åðåç ñåðåäîâèùå ïðîãðàìóâàííÿ

Öåé ñïîñ³á çðó÷íèé òèì, ùî ïðè íàÿâíîñò³ áàæàíü êîðèñòóâà÷à ùîñü çì³íèòè ó ïðîãðàì³ ìîæíà îäðàçó æ âèéòè ç ïðîãðàìè ó ñåðåäîâèùå ïðîãðàìóâàííÿ ³ çì³íèòè ïåâíèé êîä, ïðè öüîìó â³äðàçó æ â³äêîìï³ëþâàòè íîâîñòâîðåíèé êîä.

Ïåâíîþ íåçðó÷í³ñòþ º òå, ùî íå âñ³ ìîæóòü ìàòè ñàìå ñåðåäîâèùå ïðîãðàìóâàííÿ ÷è êîìï³ëÿòîð. Äëÿ çàïóñêó ïðîãðàìè ïîòð³áíî ìàòè ùîíàéìåíøå ñåðåäîâèùå Borland C++ 3.1. Íà ïîïåðåäí³õ âåðñ³ÿõ ïðîãðàìà íå òåñòóâàëàñÿ.

Ùîá çàïóñòèòè ïðîãðàìó íà âèêîíàííÿ ³ç ñåðåäîâèùà ïðîãðàìóâàííÿ ïîòð³áíî íàòèñíóòè êëàâ³ø³ <Ctrl>+F9. Àáî çä³éñíèòè òàêó ïîñë³äîâí³ñòü ä³é ó ìåíþ ñàìîãî ñåðåäîâèùà: Menu (F10) -> Run (<Ctrl>+F9).

2) Çàïóñê ïðîãðàìè ÷åðåç îïåðàö³éíó ñèñòåìó

Íà ì³é ïîãëÿä, öåé ñïîñ³á º íàáàãàòî çðó÷í³øèì â³ä ïîïåðåäíüîãî. Ùîá çàïóñòèòè ïðîãðàìó ÷åðåç îïåðàö³éíó ñèñòåìó ïîòð³áíî çíàéòè ôàéë KYRSOVA.EXE ³ êë³êíóòè ïî íüîìó ìèøêîþ. Â ðåçóëüòàò³ ïðîãðàìà áóäå çàïóùåíà ó â³êí³ MSDOS ï³ä êåðóâàííÿì îïåðàö³éíî¿ ñèñòåìè.

Íåäîë³êîì öüîãî ñïîñîáó º òå, ùî çì³íè ó ïðîãðàìíèé êîä ïðîãðàìè âíîñèòè íå ìîæíà.

Òàê ÷è ³íàêøå, ï³ñëÿ çàïóñêó ïðîãðàìè íà âèêîíàííÿ íà ãîëóáîìó ôîí³ åêðàíó ç’ÿâëÿºòüñÿ ìåíþ. Ïî çàìîâ÷óâàííþ, âñòàíîâëåíèì ó ïðîãðàìíîìó êîä³ ïðîãðàìè, çàâæäè ç’ÿâëÿºòüñÿ ìåíþ. Ùîá âíåñòè áóäü-ÿê³ çì³íè êîðèñòóâà÷ó ïîòð³áíî ñêîðèñòàòèñÿ ìåíþ ïðîãðàìè. Ïðî äîñòóïí³ çì³íè áóäå ðîçêàçàíî íèæ÷å.

Ïî ñåðåäèí³ áóäå ïîêàçàíî ìåíþ ç òðüîõ ïóíêò³â new list, open list exit, çà äîïîìîãîþ ÿêèõ ìîæíà ó áóäü-ÿêèé ìîìåíò çì³íèòè íàñòðîéêè ó ïðîãðàì³ ÷è âèéòè ç ïðîãðàìè. Ôîðìà îñíîâíîãî ìåíþ ïîêàçàíà íèæ÷å (äèâ. Ìàë. 1).



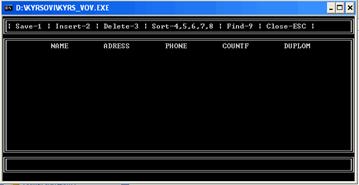
(Ìàë. 1)

Â³äêðèâàåòüñÿ ãîëîâíå ìåíþ äå ïîÿâëÿþòüñÿ 6 ïóíêò³â ÿê³ ñîõðàíÿþòü âèäàëÿþòü ñîðòóþòü ³ çàêðèâàþòü



(ìàë. 2)

Òåïåð äåòàëüí³øå ïîãîâîðèìî ïðî ìîæëèâîñò³ íàëàøòóâàíü ïðîãðàìè. Ùîá çàéòè â íàñòðîéêè ïðîãðàìè ïîòð³áíî ç êëàâ³àòóðè íàòèñíóòè ‘1’. Êîðèñòóâà÷ ïðè öüîìó ïîáà÷èòü òàêó åêðàííó ôîðìó ìåíþ (äèâ. Ìàë. 2).



(Ìàë. 3)

Ïðè íàòèñíåíí³ 1 ñîõðàíÿº, Ïðè íàòèñíåíí³ 2 âñòàâëÿº, 3 âèäàëÿº,

4 ñîðòóº ïî íàçâ³ çàêëàäó, 5 ñîðòóº ïî àäðåñó âèùîãî íàâ÷àëüíîãî çàêëàäó, 6 ñîðòóâàííÿ çà òåëåôîíîì íàâ÷àëüíîãî çàêëàäó. 7 ñîðòóâàííÿ çà 8 ñîðòóâàííÿ äâîñòîðîíí³é.

9 çíàéòè çà êðèòåð³ÿìè ³ ôàéëà.

Ùîá óñï³øíî âèéòè ç ïðîãðàìè ïîòð³áíî â îñíîâíîìó ìåíþ íàòèñíóòè ‘3’. Ï³ñëÿ öüîãî êîðèñòóâà÷ âèéäå àáî â ïðîãðàìíå ñåðåäîâèùå (ÿêùî ïðîãðàìà çàïóñêàëàñÿ ÷åðåç ñàìå ñåðåäîâèùå), àáî â îïåðàö³éíó ñèñòåìó (ÿêùî ïðîãðàìà çàïóñêàëàñÿ â îïåðàö³éí³é ñèñòåì³)

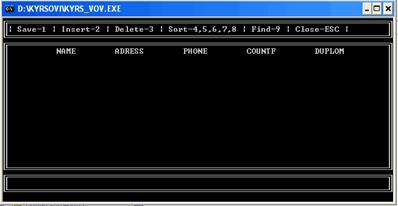
**7. Êîíòðîëüíèé ïðèêëàä òà àíàë³ç ðåçóëüòàò³â êîìï’þòåðíî¿ ðåàë³çàö³¿ ïðîãðàìè**

Îòæå, ìåí³ ó êóðñîâ³é ðîáîò³ ïîòð³áíî áóëî ðåàë³çóâàòè äîâ³äêîâó ñèñòåìó ïðî âèù³ íàâ÷àëüí³ çàêëàäè ì Ëüâîâà. Öÿ ïðîãðàìà âèêîíóº ç÷èòóâàííÿ òà âñòàíîâëþâàííÿ ³íôîðìàö³¿. Çîáðàæåííÿ õîäó ïîâèííî áóòè âèêîíàíî. Ìîâîþ íàïèñàííÿ äàíîãî ïðîåêòó ìîãëà áóòè Ñ àáî Ñ++.

Ï³ä ÷àñ ðåàë³çàö³¿ òà âèêîíàííÿ äàíî¿ ïðîãðàìè âõ³äíèìè äàíèìè äëÿ íå¿ áóëè ïîòî÷í³ ìåòîäè÷í³ âêàç³âö³.

Â ÿêîñò³ êîíòðîëüíîãî ïðèêëàäó íèæ÷å íàâåäåíî çí³ìîê åêðàíó ó ìîìåíò çîáðàæåííÿ ìåíþ (äèâ. Ìàë. 4

.



(Ìàë. 4)

Ïðè çì³í³ âõ³äíèõ äàíèõ ðåçóëüòàò º ïðàâèëüíèì. Çì³íà ðîçì³ðó çàäàºòüñÿ òàê, ÿê âêàçàâ êîðèñòóâà÷. Ïðè âèêîíàíí³ äàíî¿ ïðîãðàìè íå ïîòð³áíî ìàòè âåëèêèé îá’ºì îïåðàòèâíî¿ ïàì’ÿò³, ùî º äóæå âåëèêèì ïëþñîì.

Çàâèñàíü, â³äõèëåíü ÷è ãëþê³â ïðîãðàìè íå áóëî ïîì³÷åíî.

**Äîäàòîê 1. Òåêñò ïðîãðàìè íà ìîâ³ Ñ³**

#include<conio.h>

#include<stdio.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

/\*-------------------cursor--------------------\*/

void cursor(int state){

if (state==0)

asm{

mov ah,1

mov ch,0x20

int 0x10

}

else

asm {

mov ah,1

mov ch,5

mov cl,7

int 0x10

}

}

/\*-------------------cursor--------------------\*/

/\*--------------------Ramka-----------------------------------\*/

void ramka(int kyt\_vl\_x,int kyt\_vl\_y,int dov\_x,int dov\_y)

{

int i1;

for(i1=kyt\_vl\_x;i1<dov\_x;i1++){

gotoxy(i1,kyt\_vl\_y-1);

putchar(205);

}

for(i1=kyt\_vl\_y;i1<dov\_y;i1++){

gotoxy(dov\_x,i1);

putchar(186);

}

for(i1=dov\_x-1;i1>kyt\_vl\_x-1;i1--){

gotoxy(i1,dov\_y);

putchar(205);

}

for(i1=dov\_y-1;i1>kyt\_vl\_y-1;i1--){

gotoxy(kyt\_vl\_x-1,i1);

putchar(186);

}

gotoxy(kyt\_vl\_x-1,kyt\_vl\_y-1);

putchar(201);

gotoxy(dov\_x,kyt\_vl\_y-1);

putchar(187);

gotoxy(dov\_x,dov\_y);

putchar(188);

gotoxy(kyt\_vl\_x-1,dov\_y);

putchar(200);

}

/\*--------------------Ramka-----------------------------------\*/

int page=0;

int page\_count=0;

int pos=0;

struct value

{

char name[30];

char adress[30];

char phone[15];

char countf[5];

char duplom[30];

};

struct mai

{

int count;

value m[100];

};

mai list;

void refresh\_list()

{

int li=6;

int dest;

page\_count = 0;

int tmpl=list.count;

while (tmpl-15>0)

{

tmpl=tmpl-15;

page\_count++;

}

if (page\*15+15>list.count)

{

dest=list.count;

}

else

{

dest=page\*15+14;

}

for (int i=0;i<15;i++)

{

gotoxy(9,i+6);

printf("  ");

}

for (i=page\*15;i<=dest;i++)

{

gotoxy(12,li);

printf(list.m[i].name);

gotoxy(24,li);

printf(list.m[i].adress);

gotoxy(38,li);

printf(list.m[i].phone);

gotoxy(51,li);

printf(list.m[i].countf);

gotoxy(65,li);

printf(list.m[i].duplom);

li++;

}

}

void insert\_list()

{

char s[100];

list.count++;

gotoxy(1,1);

printf("%d",list.count);

gotoxy(3,23);

printf("Enter Name ");

scanf("%s",&s);

strcat(list.m[list.count].name,s);

gotoxy(3,23);

printf(" ");

gotoxy(3,23);

printf("Enter Adress ");

scanf("%s",&s);

strcat(list.m[list.count].adress,s);

gotoxy(3,23);

printf(" ");

gotoxy(3,23);

printf("Enter Phone ");

scanf("%s",&s);

strcat(list.m[list.count].phone,s);

gotoxy(3,23);

printf("                                             ");

gotoxy(3,23);

printf("Enter Count Fakultet ");

scanf("%s",&s);

strcat(list.m[list.count].countf,s);

gotoxy(3,23);

printf("                                                      ");

gotoxy(3,23);

printf("Enter Duplom ");

scanf("%s",&s);

strcat(list.m[list.count].duplom,s);

gotoxy(3,23);

printf("                                                      ");

refresh\_list();

}

void save\_list()

{

FILE \*f;

f=fopen("ñ:\\vuz.my","w+");

for (int i=0;i<=list.count;i++)

{

char tmp[150];

strcpy(tmp,list.m[i].name);

strcat(tmp,",");

strcat(tmp,list.m[i].adress);

strcat(tmp,",");

strcat(tmp,list.m[i].phone);

strcat(tmp,",");

strcat(tmp,list.m[i].countf);

strcat(tmp,",");

strcat(tmp,list.m[i].duplom);

fprintf(f,"%s\n",tmp);

}

fclose(f);

}

void delete\_list()

{

if (page\*15+pos<=list.count)

{

for (int i=page\*15+pos;i<list.count;i++)

{

list.m[i]=list.m[i+1];

}

list.count--;

refresh\_list();

}

}

void sort\_list(int vub)

{

value tmpv;

if (vub==0)

{

for (int i=0;i<=list.count;i++)

{

for (int j=0;j<=list.count;j++)

{

if (strcmp(list.m[i].name,list.m[j].name)<0)

{

          tmpv=list.m[i];

          list.m[i]=list.m[j];

          list.m[j]=tmpv;

}

}

}

}

if (vub==1)

{

for (int i=0;i<=list.count;i++)

{

for (int j=0;j<=list.count;j++)

{

if (strcmp(list.m[i].adress,list.m[j].adress)<0)

{

          tmpv=list.m[i];

          list.m[i]=list.m[j];

          list.m[j]=tmpv;

}

}

}

}

if (vub==2)

{

for (int i=0;i<=list.count;i++)

{

for (int j=0;j<=list.count;j++)

{

if (strcmp(list.m[i].phone,list.m[j].phone)<0)

{

          tmpv=list.m[i];

          list.m[i]=list.m[j];

          list.m[j]=tmpv;

}

}

}

}

if (vub==3)

{

for (int i=0;i<=list.count;i++)

{

for (int j=0;j<=list.count;j++)

{

if (strcmp(list.m[i].countf,list.m[j].countf)<0)

{

          tmpv=list.m[i];

          list.m[i]=list.m[j];

          list.m[j]=tmpv;

}

}

}

}

if (vub==4)

{

for (int i=0;i<=list.count;i++)

{

for (int j=0;j<=list.count;j++)

{

if (strcmp(list.m[i].duplom,list.m[j].duplom)<0)

{

          tmpv=list.m[i];

          list.m[i]=list.m[j];

          list.m[j]=tmpv;

}

}

}

}

refresh\_list();

}

void find\_list()

{

int vub;

mit\_1:

gotoxy(3,23);

printf("                                    ");

gotoxy(3,23);

printf("FIND: | Name-1 | Adress-2 | Phone-3 | CountF-4 | Duplom-5 | ");

cursor(1);

scanf("%d",&vub);

if((vub<1)||(vub>5)){goto mit\_1;}

gotoxy(3,23);

printf("                                    ");

gotoxy(3,23);

printf("Find Text: ");

char s[50];

scanf("%s",&s);

for (int i=0;i<=list.count;i++)

{

char s1[50];

switch(vub)

{

case 1: strcpy(s1,list.m[i].name);break;

case 2: strcpy(s1,list.m[i].adress);break;

case 3: strcpy(s1,list.m[i].phone);break;

case 4: strcpy(s1,list.m[i].countf);break;

case 5: strcpy(s1,list.m[i].duplom);break;

}

if (strcmp(s1,s)==0)

{

gotoxy(3,pos+6);

printf(" ");

int j=i;

int l=0;

while (j-15>=0)

{

j=j-15;

l++;

}

page=l;

pos=j;

refresh\_list();

gotoxy(3,pos+6);

printf(">");

gotoxy(3,23);

printf("Press ENTER ");

mit\_2:

int c=getche();

if (c!=13){goto mit\_2;}

}

}

cursor(0);

gotoxy(3,23);

printf("                                    ");

}

void new\_window(int vub)

{

clrscr();

cursor(0);

ramka(2,2,80,3);

ramka(2,5,80,21);

ramka(2,23,80,24);

gotoxy(2,2);

printf("| Save-1 | Insert-2 | Delete-3 | Sort-4,5,6,7,8 | Find-9 | Close-ESC |");

gotoxy(12,5);

printf("NAME ADRESS PHONE COUNTF DUPLOM");

if (vub==1)

{

refresh\_list();

}

pos=0;

mi2:

gotoxy(3,23);

int c=getche();

switch (c)

{

case 49:save\_list(); break;

case 50:insert\_list(); break;

case 51:delete\_list(); break;

case 52:sort\_list(0); break;

case 53:sort\_list(1); break;

case 54:sort\_list(2); break;

case 55:sort\_list(3); break;

case 56:sort\_list(4); break;

case 57:find\_list(); break;

case 72:

          if (pos!=0)

          {

          gotoxy(3,pos+6);

          printf(" ");

          pos--;

          gotoxy(3,pos+6);

          printf(">");

          }

          else

          {

          if (page!=0)

          {

          page--;

          gotoxy(3,pos+6);

          printf(" ");

          pos=14;

          refresh\_list();

          }

          }

          break;

case 80:

          if (pos!=14)

          {

          gotoxy(3,pos+6);

          printf(" ");

          pos++;

          gotoxy(3,pos+6);

          printf(">");

          }

          else

          {

          if (page\_count>page)

          {

          gotoxy(3,pos+6);

          printf(" ");

          page++;

          pos=0;

          refresh\_list();

          }

          }

          break;

case 27:goto mi1;break;

}

goto mi2;

mi1:

}

void open\_list()

{

clrscr();

FILE \*f;

f=fopen("ñ:\\vuz.my","r");

char \*s;

list.count=-1;

while(! feof(f))

{

list.count++;

fscanf(f,"%s\n",s);

// printf("%s\n",s);

char \*p;

p = strtok(s,",");

int l=0;

while (p)

{

switch(l)

{

case 0:strcpy(list.m[list.count].name,p);break;

case 1:strcpy(list.m[list.count].adress,p);break;

case 2:strcpy(list.m[list.count].phone,p);break;

case 3:strcpy(list.m[list.count].countf,p);break;

case 4:strcpy(list.m[list.count].duplom,p);break;

}

l++;

p = strtok(NULL,",");

}

}

fclose(f);

}

void main\_menu()

{

mit\_3:

clrscr();

ramka(2,2,80,24);

ramka(20,8,60,16);

gotoxy(35,10);

printf("New list - 1");

gotoxy(35,12);

printf("Open list - 2");

gotoxy(35,14);

printf("Exit - 3 ");

cursor(0);

mit\_1:

gotoxy(45,14);

char c=getche();

if ((c!='1')&&(c!='2')&&(c!='3'))

{

goto mit\_1;

}

switch (c)

{

case '1':page\_count=0;page=0;list.count=-1; new\_window(0);goto mit\_3; break;

case '2':open\_list();new\_window(1);goto mit\_3; break;

case '3':goto mit2; break;

}

goto mit\_1;

mit2:

}

void main()

{

list.count=-1;

main\_menu();

//

//

open\_list();

// new\_window();

char s[10];

// scanf("%s",&s);

}

**Äîäàòîê 2. Ðåçóëüòàòè ðîáîòè ïðîãðàìè**

Ïî÷àòîê ïðîãðàìè



Ñòâîðèâ ôàéë

**Âèñíîâêè**

Ï³ñëÿ âèêîíàííÿ êóðñîâî¿ ðîáîòè ÿ îòðèìàâ íåîáõ³äí³ çíàííÿ òà ïðàêòè÷í³ íàâè÷êè ðîáîòè ç ³ç ðîáîòîþ ç ôàéëàìè, íàâ÷èâñÿ ïðîãðàìóâàòè ðîçãàëóæåí³ òà öèêë³÷í³ ïðîöåñè, çðîçóì³â ÿê ïðàöþâàòè ç ÷àñîì òà áóäóâàòè ìåíþ ïðîãðàì.

Îòðèìàíà ïðîãðàìà º ÿê³ñíîþ, øâèäêîþ ïðè âèêîíàíí³ ³ íå ïîòðåáóº çíà÷íèõ ðåñóðñ³â êîìï’þòåðà äëÿ ¿¿ çàïóñêó. ßêèõîñü íåäîë³ê³â ÿ íå ïîì³òèâ.

Äàíà ïðîãðàìà ìîæå çãîäèòèñÿ äëÿ äóæå áàãàòüîõ êîðèñòóâà÷³â, ï³äïðèºìñòâ, íàâ÷àëüíèõ óñòàíîâ. Âîíà º óí³âåðñàëüíîþ, îñê³ëüêè ãîäèííèê º íåâ³ä’ºìíèì àòðèáóòîì áóäü-ÿêî¿ êîìï’þòåðíî¿ ñèñòåìè. Òîìó àêòóàëüí³ñòü äàíî¿ ïðîãðàìè º íàäçâè÷àéíî âåëèêîþ.

Íà ö³é êóðñîâ³é ðîáîò³ ÿ îñâî¿â ìåòîäè ïðîãðàìóâàííÿ íà ìîâ³ ñ³ ³ âèêîíàâ çàâäàííÿ ïðî: ²íôîðìàö³éíî- äîâ³äêîâà ñèñòåìà “Âèù³ íàâ÷àëüí³ çàêëàäè ì. Ëüâîâà”

**Ë³òåðàòóðà**

1) Êðàâåöü Ï. Î. Ìåòîäè÷í³ âêàç³âêè äî âèêîíàííÿ ëàáîðàòîðíèõ ðîá³ò ç äèñöèïë³íè “Ïðîáëåìíî-îð³ºíòîâàí³ ìîâè ïðîãðàìóâàííÿ”, 2005.

2) Êðàâåöü Ï. Î. Ìåòîäè÷í³ âêàç³âêè äî âèêîíàííÿ êóðñîâî¿ ðîáîòè ç äèñöèïë³íè “Ïðîáëåìíî-îð³ºíòîâàí³ ìîâè ïðîãðàìóâàííÿ”, 2005

3) Êóëüòèí Í. Ñ, Ñ++ â çàäà÷àõ è ïðèìåðàõ, Ñàíêò-Ïåòåðáóðã 2005.

4) Ðè÷àðä Õåçôèëä, Ëîóðåíñ Êèðáè Èññêóñòâî ïðîãðàììèðîâàíèÿ íà Ñ, Ìîñêâà, Ñàíêò-Ïåòåðáóðã 2001

5) Ïàâëîâñêàÿ Ò. À., Ùóïàê Þ. À. Ñ,Ñ++ - ïðîãðàììèðîâàíèå íà ÿçûêå âèñîêîãî óðîâíÿ (Ñòðóêòóðíîå ïðîãðàììèðîâàíèå), Ñàíêò-Ïåòåðáóðã 2003

6) Ðîìàíîâ Å. Ë. Ïðàêòèêóì ïî ïðîãðàììèðîâàíèþ íà Ñ++, Ñàíêò-Ïåòåðáóðã 2004

Äèÿê ².². Ïðîïåäåâòèêà ïðèêëàäíîãî ïðîãðàìóâàííÿ – Êè¿â, ²ÑÄÎ, 1994 – 176 ñ.

Ôðîëîâ Ã.Ä., Êóçíåöîâ Ý.È. Ýëåìåíòû èíôîðìàòèêè. – Ì.: Âûñø. øê., 1989. - 304 ñ.

Êóøíèðåíêî À.Ã., Ëåáåäåâ Ò.Â. Ïðîãðàììèðîâàíèå äëÿ ìàòåìàòèêîâ. – Ì.: Íàóêà, 1988 – 384 ñ.

Ñ.À.Àáðàìîâ, Å.Â.Çèìà. Íà÷àëà èíôîðìàòèêè. – Ì.: Íàóêà, 1989. - 256 ñ.

Â.È.Ïèëüùèêîâ. Ñáîðíèê óïðàæíåíèé ïî ÿçûêó Ïàñêàëü. – Ì.: Íàóêà, 1989. – 160 ñ.

Ñ.À.Àáðàìîâ, Ò.Ò.Ãíåçäèëîâà, Å.Í.Êàïóñòèíà. Çàäà÷è ïî ïðîãðàììèðîâàíèþ. – Ì.: Íàóêà, 1988. – 224 ñ.

Â.Ì.Áðÿáðèí. Ïðîãðàììíîå îáåñïå÷åíèå ïåðñîíàëüíûõ ÝÂÌ. – Ì.: Íàóêà, 1989. – 272 ñ.

Â. Ý.Ôèãóðíîâ. IBM PC äëÿ ïîëüçîâàòåëÿ. Èçä. øåñòîå. – Ì.: “Èíôà Ì”, 1995. – 432 ñ.

Â.Ï.Ãðèáàíîâ, Ñ.Â.Äðîáèí, Â.Ä.Ìåäâåäåâ. Îïåðàöèîííûå ñèñòåìû. Ì.: “Ôèíàíñû è ñòàòèñòèêà”. 1990. – 230 ñ.

Â.È.Êàñüÿíîâ, Â.Ê.Ñàáåëüôåëüä. Ñáîðíèê çàäàíèé ïî ïðàêòèêóìó íà ÝÂÌ. – Ì.: “Íàóêà”, 1986. – 272 ñ.

Í.Ï.Òðèôîíîâ, Å.È.Ïàñõèí. Ïðàêòèêóì íà ÝÂÌ. – Ì.: Íàóêà, 1982. – 288 ñ.

Ì.Äàíñìóð, Å.Äåéâèñ. Îïåðàöèîííàÿ ñèñòåìà UNIX è ïðîãðàììèðîâàíèå íà ÿçûêå ÑÈ. – Ì.: “Ðàäèî è ñâÿçü”, 1989. – 192 ñ.

Â.Â.Ôàðîíîâ. Òóðáî Ïàñêàëü. – ÌÂÒÓ, Ì., 1992. – 304 ñ.

Å.À.Çóåâ. ßçûê ïðîãðàììèðîâàíèÿ TURBO PASCAL 6.0. – Ì. “Óíèòåõ”, 1992. – 298 ñ.

À.Åïàíåøíèêîâ, Å.Åïàíåøíèêîâ. Ïðîãðàìèðîâàíèå â ñðåäå Turbo Pascal 7.0. -3-å èçä., - Ì.: "ÄÈÀËÎÃ-ÌÈÔÈ", 1997. - 288 ñ.

Delphi 2. Ýíöèêëîïåäèÿ ïîëüçîâàòåëÿ: Ïåð. ñ àíãë./×àðëüç Êàëâåðò. - Ê.: ÍÈÏÔ "ÄèàÑîôò Ëòä.", 1996. - 736 ñ.

Èñïîëüçîâàíèå Delphi 3. Ñïåöèàëüíîå èçäàíèå.: Ïåð. ñ àíãë. - Ê.: Äèàëåêòèêà, 1997. - 768 ñ.

Delphi 4. Áèáëèÿ ðàçðàáîò÷èêà: Ïåð. ñ àíãë. - Ê.; ÑÏá.: Äèàëåêòèêà, 1998. - 672.