МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СЄВЄРОДОНЕЦЬКЕ ВИЩЕ ПРОФЕСІЙНЕ УЧИЛИЩЕ

ЗАТВЕРДЖУЮ:

Заступник директора з УВР

\_\_\_\_\_\_\_\_\_\_\_\_\_ О.П. Лимар

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" 2010 р.

КАЛЕНДАРНИЙ ПЛАН

НА КУРСОВУ РОБОТУ СТУДЕНТА ГРУПИ № Т-09-4

Козирєв Артем Миколайович

Спеціальність: 5.091504 "Обслуговування комп’ютерних та інтелектуальних систем та мереж"

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Найменування етапів дипломної кваліфікаційної роботи | Термін виконання етапів роботи | Примітка |
| 1 | Постановка задачі |  |  |
| 2 | Математичний опис задачі |  |  |
| 3 | Розробка алгоритму програми |  |  |
| 4 | Структура програми, опис її структурних компонентів |  |  |
| 5 | Описання початкових даних, на які розрахована програма |  |  |
| 6 | Розробка інструкції оператору |  |  |
| 7 | Тестування програми та її оптимізація |  |  |
| 8 | Оформлення пояснювальної записки до курсової роботи |  |  |

Випускник

Керівник роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СЄВЄРОДОНЕЦЬКЕ ВИЩЕ ПРОФЕСІЙНЕ УЧИЛИЩЕ

ВІДГУК

НА КУРСОВУ РОБОТУ СТУДЕНТА ГРУПИ № Т-09-4

Козирєв Артем Миколайович

Спеціальність: 5.091504 "Обслуговування комп’ютерних та інтелектуальних систем та мереж"

1. Тема роботи "Розробка програми за допомогою мови програмування Delphi"
2. Керівник роботи Руднєв Максим Миколайович
3. Загальна характеристика курсової роботи:
4. Попередня оцінка:
5. Робота захищена з оцінкою:

Керівник роботи М. М. Руднєв

"\_\_\_\_"\_\_\_\_\_\_\_\_\_\_\_2010 р.

**СОДЕРЖАНИЕ**

РЕФЕРАТ

ВВЕДЕНИЕ

1. Анализ технического задания и постановка задачи проектирования

2. Разработка алгоритма программы

3. ОПИСАНИЕ СТРУКТУРЫ ПРОГРАММЫ

4. ОПИСАНИЕ ВХОДНЫХ ДАННЫХ

5. ОПИСАНИЕ ВЫХОДНЫХ ДАННЫХ

6. ИНСТРУКЦИЯ ОПЕРАТОРУ

ВЫВОДЫ

Перечень ссылок

ПРИЛОЖЕНИЕ А

ПРИЛОЖЕНИЕ Б

##### РЕФЕРАТ

Пояснительная записка к курсовой работе содержит:

страниц - 24;

рисунков - 7;

таблиц – 1 ;

приложений – 2.

Цель работы: разработать программу, на языке Delphi, работы с клиентами, с данными: номер по порядку, наименование клиента, вид услуги, дата начала работы с клиентом и срок абонентской платы, предусмотреть ввод, редактирование и удаление данных

В курсовой работе создана программа, которая показывает таблицу работы с клиентами, в которой указаны: номер по порядку, наименование клиента, вид услуги, дата начала работы с клиентом и срок абонентской платы за услугу. При двойном щелчке на какой либо из ячеек можно вводить, редактировать и удалять данные. При нажатии на кнопку "Записать", данные из ячейки в таблице записываются в файл "text" на жестком диске.

В первом разделе курсовой работы выполнен анализ технического задания, выделены функции, которыми должно обладать разрабатываемое приложение, а также сформулированы требования к нему, произведена постановка задачи на проектирование.

Во втором разделе выполняется анализ задачи проектирования, анализируется вид исходных данных и приводится словесный алгоритм работы программы.

В третьем разделе рассматриваются алгоритмы разработанной программы, а также приводится их описание. К реализованной программе разработана инструкция пользователя, которая приведена в четвертом разделе.

Алгоритмы процедур приведены в приложении А.

Листинг программы содержится в приложении Б.

Результат тестового запуска приведен в приложении В.

Программа разработана с использованием языка Delphi.

**ТЕКСТОВЫЙ ФАЙЛ, РАБОТА С КЛИЕНТОМ, ДИАЛОГОВОЕ ОКНО, ЦИКЛИЧЕСКИЕ КОНСТРУКЦИИ**

ВВЕДЕНИЕ

Данный курсовой проект был разработан в среде Delphi 7 с базовым языком программирования Delphi. Среди множества языков Delphi является наиболее универсальным и легко изучаемым языком. На сегодня Delphi поддерживает все современные возможности объектно-ориентированного программирования в такой мощной системе проектирования как Delphi 7.

Цель курсовой работы – закрепление и демонстрация знаний, полученных при изучении курса.

Задача работы заключается в том, чтобы правильно составить алгоритм поставленной задачи по обработке текстовой информации, разработать и отладить программу, реализующую разработанный алгоритм.

В ходе выполнения курсового проекта необходимо выполнить следующие этапы:

- анализ технического задания и постановка задачи проектирования;

- выделение функций, которые выполнять разрабатываемая программа;

- разработка алгоритма программы, предназначенной для выполнения функций, сформированных на предыдущем этапе;

- составление программы согласно разработанным алгоритмам на языке Delphi;

- отладка программы на тестовых наборах;

- составление и оформление пояснительной записки.

Теоретическая и практическая значимость заключается в закреплении навыков составления алгоритмов поставленных задач и программирования на языке Delphi.

1. Анализ технического задания и постановка задачи проектирования

Согласно заданию необходимо составить программу для работы с клиентами, в которой будут пункты: номер по порядку, наименование клиента, вид услуги, дата начала работы с клиентом, срок абонентской платы. Так же в задании сказано, что необходимо предусмотреть ввод, редактирование и удаление данных

Как видим, задание сводится к созданию базы данных для работы с клиентами, в качестве чего служит текстовый файл. А также, согласно заданию, вывод на экран базы данных в табличной форме и программное редактирование данных.

В программе следует проанализировать полученную из файла текстовую информацию и вывести на экран в виде таблицы.

Подытожив все вышесказанное можно сформулировать требования к разрабатываемому программному обеспечению и выполнить постановку задания на проектирование. Так, согласно заданию, программа должна быть реализована в среде Delphi и должна выполнять следующие функции:

- вывод на экран информации о задании и назначении программы;

- чтение из файла данных и вывод их на экран;

- ввод, редактирование, удаление данных.

- сохранение изменений в текстовый файл.

Последующие разделы будут посвящены решению поставленных задач и разработке приложения с вышеперечисленными функциями.

2. Разработка алгоритма программы

В предыдущем разделе были сформулированы требования к разрабатываемой программе и к ее функциям. Анализируя требования к программе, можно разработать алгоритм разрабатываемого приложения.

Так, поскольку задание состоит в создании базы данных, целесообразно для хранения информации табличный тип данных, а также создать текстовый файл.

Далее необходимо считать информацию из текстового файла и занести ее в таблицу в соответствующие столбцы, для этого необходимо выполнение следующего условия: отделить вид данных (например, номер по порядку, от даты абонентской платы);

Для выполнения этого условия необходимо первоначально открыть файл для чтения. Затем прочитать строку из файла и разделить вид данных при помощи разделителя.

Составим следующий алгоритм:

1. Привязать файл, к конкретной файловой переменной.
2. Открыть файл для чтения.
3. Пока курсор не станет на позицию конца файла считывать поочередно строки из файла. Если курсор станет на позицию конца файла, то переходим к пункту 7.
4. Взять конкретную считанную строку и приступить к выделению в ней слова, т.е. стать на первый символ в строке.
5. Проверить символ на сходство с символом разделителя - "|", который задан, пока не наступит конец считанной строки. Если наступил конец строки, то перейдем к пункту 3.
6. Если символ – разделитель то запишем предыдущие символы в строку 1 столбца 1, и вернемся на пункт 5 с уже с этого места.
7. Если по окончании цикла не найден разделитель "|", то запишем его и создадим новую строку.
8. Конец считывания из файла, и занесение данных в таблицу.

В конце следует проанализировать проверен файл полностью или нет и если полностью, то записать все данные из файла в таблицу.

Так как созданную базу данных необходимо редактировать необходимо выполнение еще одного условия: ввод, редактирование и удаление данных из программы. Для выполнения этого условия необходимо опциях компоненты StringGrid, изменить параметр goEditing с False на True. А так же создать кнопку обеспечивающую сохранение в текстовый файл изменений сделанных в таблице.

Детальное описание разработанной программы, алгоритмов и их программной реализации приведено в последующих разделах.

3. ОПИСАНИЕ СТРУКТУРЫ ПРОГРАММЫ

## Описание переменных

В разделе описаний переменных описаны переменные, назначение которых приведено в таблице 3.1.

Таблица 3.1– Описание переменных программы

|  |  |  |
| --- | --- | --- |
| Наименование | Тип | Назначение |
| f | TextFile | Переменная текстового файла, из которого производится считывание, а так же заносятся изменения, базы данных |
| str, t | String | Строковые переменные для чтения из файла |
| k, i, j, g | integer | Переменные счетчики для циклов |
| t | string | Переменная, в которую считываются данные из файла |
| str1,str2,str3,str4,str5 | string | Строковые переменные для записи в файл |
| Fname | String | Строка, в которой хранится путь к исходному файлу |

## Описание вспомогательной процедуры

В программе приведено описание одной вспомогательной процедуры.

Процедура **FormCreate** отвечает за открытие и заполнение компоненты stringgrid.

Процедура **Button1Click** для занесения изменений, сделанных в таблице, в текстовый файл.

## Алгоритм основной программы

Алгоритм выполнения основного тела программы приведен на рисунке А.1 в приложении А. Так в процедуре в первую очередь осуществляется привязка файла к файловой переменной и открытие текстового файла для чтения текстовой информации.

Затем производится чтение информации из файла построчно.

Затем в каждой строке выделяются слова.

Затем осуществляется поиск в выделенных словах разделителя.

Далее, если найден разделитель, происходит запись данных из файла в таблицу программы.

В ходе выполнения работы использовались следующие операции работы с текстовыми файлами:

AssignFile (<Имя файловой переменной>,<Имя файла>);

Имя файла задается либо строковой константой, либо через переменную типа Sting. Имя файла должно соответствовать правилам работающей в данный момент операционной системы. Если строка имени пустая, то связь файловой переменной осуществляется со стандартным устройством ввода-вывода.

Reset (<Имя файловой переменной>);

Открывается файл для чтения из него информации, ему присваивается имя, заданное процедурой Assign.

После работы с файлом он, как правило, должен быть закрыт процедурой CloseFile (<Имя файловой переменной>). Чтение информации производит оператор Readln(f,str). После его выполнения строка из файла заносится f в переменную строкового типа данных str.

4. ОПИСАНИЕ ВХОДНЫХ ДАННЫХ

Исходными данными для выполнения является файл, который должен содержать, или в который нужно внести, данные о работе с клиентами.

5. ОПИСАНИЕ ВЫХОДНЫХ ДАННЫХ

Выходными данными являются:

- изменение данного текстового файла;

- заполненная таблица о работе с клиентами, которая содержит следующие пункты: номер по порядку, наименование клиента, вид услуги, дата начала работы с клиентом, срок абонентской платы.

6. ИНСТРУКЦИЯ ОПЕРАТОРУ

Разработанная программа представляет собой исполняемый файл Kozirev.exe размером 430 Кбайт. В программе реализован алгоритм программы для работы с клиентами, которая содержит следующие пункты: номер по порядку, наименование клиента, вид услуги, дата начала работы с клиентом, срок абонентской платы. Предусмотрено ввод, редактирование и удаление данных. После запуска программы появляется окно, изображенное на рисунке 4.1.

После этого пользователь должен заполнить таблицу. На рисунке 4.2 показан пример диалогового окна с заполненной таблицей.

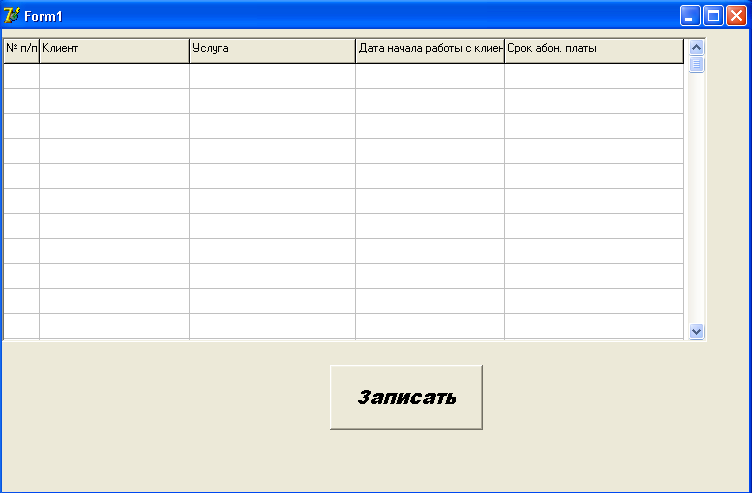


Рисунок 4.1 – Главное окно программы

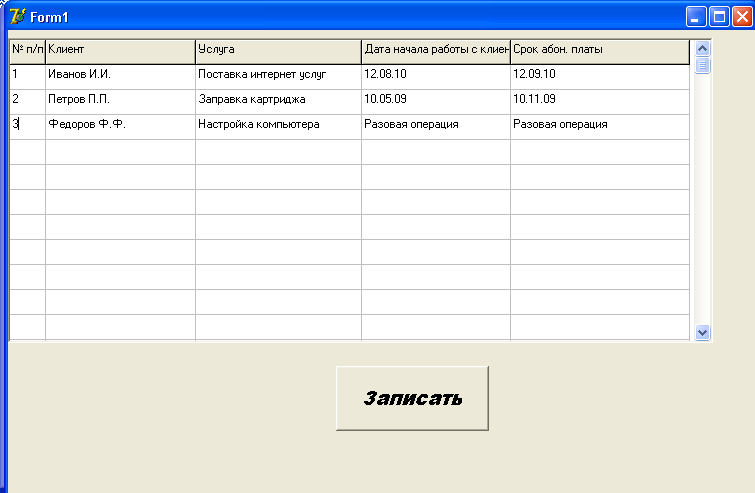


Рисунок 4.2 – Пример заполненной таблицы

После заполнения таблицы необходимо сохранить внесенные изменения в текстовый файл. Что осуществляется по нажатию кнопки "Записать". Пример этого действия показан на рисунке 4.3.и 4.4.

Пример сообщения о нахождении заглавного символа приведен на рисунке 4.3.

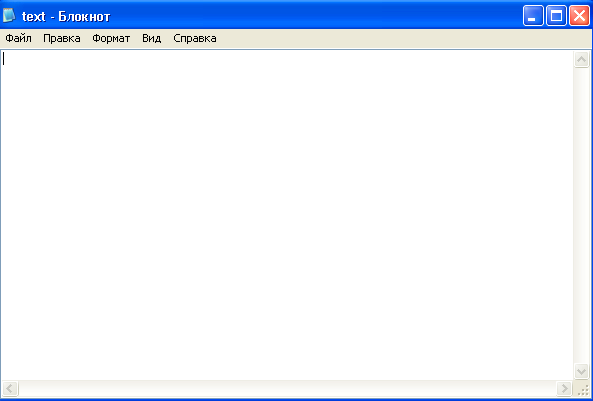


Рисунок 4.3– Пример текстового фала до нажатия кнопки "Записать"

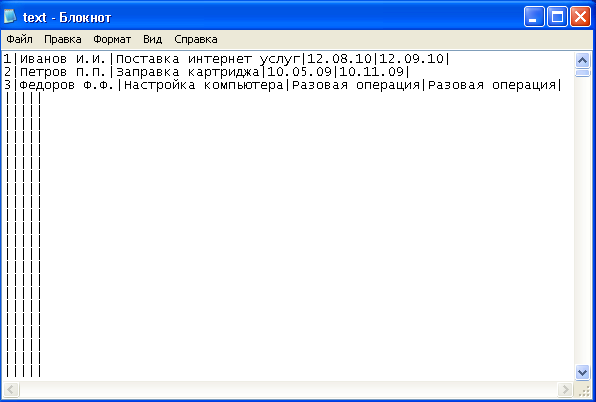


Рисунок 4.4- Пример текстового фала после нажатия кнопки "Записать".

Результат заполнения таблицы заносится в текстовый файл служащий базой данных для работы с клиентами.

Функционирование программы полностью соответствует заданию.

ВЫВОДЫ

Данная курсовая работа была выполнена в полном соответствии поставленному заданию и отлажена в среде Delphi 7.0. В ходе выполнения курсовой работы была разработана программа для работы с клиентами, с данными: номер по порядку, наименование клиента, вид услуги, дата начала работы с клиентом и срок абонентской платы, предусмотреть ввод, редактирование и удаление данных

В ходе выполнения курсового проекта были проведен - анализ технического задания и постановка задачи проектирования.

Были выделены функции, которыми должна обладать разрабатываемая программа.

Был разработан алгоритм программы, предназначенной для выполнения сформированных функций.

Согласно разработанному алгоритму была составлена и отлажена программа алгоритмическом языке программирования Delphi. В ходе разработки программы проводилось ее тестирование и отладка на тестовых наборах. Для написанной программы разработана инструкция оператора с указанием порядка выполняемых действий .

Результаты работы оформлены в виде пояснительной записки.

В ходе разработки программы были закреплены навыки работы:

- командами ввода-вывода как на экран, так и в текстовые файлы,

- использования процедур,

- команды обработки массивов,

- команды для реализации циклов с заданным числом повторений;

- команды проверки условий и реализации ветвлений.

В результате выполнения данной курсовой работы, мы убедились в широких возможностях языка программирования Delphi, закрепили практические навыки программирования в cреде Delphi.

Перечень ссылок

1. Зуев Е.А. Программирование на языке Delphi 6.0,7.0. – М.: Радио и связь, Веста, 1993.
2. Фаронов В.В. Delphi 7.0. Начальный курс. - М.: Нолидж, 2000.

ПРИЛОЖЕНИЕ А

**Алгоритм программы**

**Процедура FormCreate**

**Процедура Button1Click**

Рисунок А.1 – Алгоритм программы

Rewrite (f)

нет

t:=t+str[k];

Fname:= 'd:/text.txt';

AssignFile (f,fname);

{$I-}

reset(f);

{$I+}

if IOResult <> 0 then

да

нет

t:='';

j:=1;

StringGrid1.Cells[0,0]:='№ п/п';

StringGrid1.Cells[1,0]:='Клиент';

StringGrid1.Cells[2,0]:='Операция';

StringGrid1.Cells[3,0]:='Дата начала работы с клиентом';

StringGrid1.Cells[4,0]:='Срок абон. платы';

Readln(f,str);

i:=0;

while not EOF(f) do begin

for k:=1 to Length(str) do begin

for k:=1 to Length(str) do begin

да

StringGrid1.Cells[i,j]:=t;

i:=i+1;

t:='';

end

end;

j:=j+1;

end;

CloseFile(f);

Рисунок А.2 – Алгоритм основной процедуры программы

Fname:= 'd:/text.txt';

AssignFile (f,fname);

Rewrite (f);

i:=0;

j:=1

for g:=1 to 1000 do begin

begin

str1:='';

str2:='';

str3:='';

str4:='';

str5:='';

end;

if StringGrid1.Cells[0,j] <> '' then begin

нет

да

str1:=StringGrid1.Cells[0,j];

str2:=StringGrid1.Cells[1,j];

str3:=StringGrid1.Cells[2,j];

str4:=StringGrid1.Cells[3,j];

str5:=StringGrid1.Cells[4,j];

end;

j:=j+1;

write(f,str1+'|');

write(f,str2+'|');

write(f,str3+'|');

write(f,str4+'|');

write(f,str5+'|');

writeln(f,'');

end;

CloseFile(f);

Рисунок А.3 – Алгоритм вторичной процедуры программы

ПРИЛОЖЕНИЕ Б

**Листинг программы**

1. unit Unit1;
2. interface
3. uses
4. Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
5. Dialogs, StdCtrls, Grids;
6. type
7. TForm1 = class(TForm)
8. Button1: TButton;
9. StringGrid1: TStringGrid;
10. procedure Button1Click(Sender: TObject);
11. procedure FormCreate(Sender: TObject);
12. private
13. { Private declarations }
14. public
15. { Public declarations }
16. end;
17. var
18. Form1: TForm1;
19. implementation
20. {$R \*.dfm}
21. procedure TForm1.Button1Click(Sender: TObject);
22. var
23. f:TextFile;
24. i,j,g:integer;
25. Fname,str1,str2,str3,str4,str5: string;
26. begin
27. Fname:= 'd:/text.txt';
28. AssignFile (f,fname);
29. Rewrite (f);
30. i:=0;
31. j:=1;
32. for g:=1 to 1000 do begin
33. begin
34. str1:='';
35. str2:='';
36. str3:='';
37. str4:='';
38. str5:='';
39. end;
40. if StringGrid1.Cells[0,j] <> '' then begin
41. str1:=StringGrid1.Cells[0,j];
42. str2:=StringGrid1.Cells[1,j];
43. str3:=StringGrid1.Cells[2,j];
44. str4:=StringGrid1.Cells[3,j];
45. str5:=StringGrid1.Cells[4,j];
46. end;
47. j:=j+1;
48. write(f,str1+'|');
49. write(f,str2+'|');
50. write(f,str3+'|');
51. write(f,str4+'|');
52. write(f,str5+'|');
53. writeln(f,'');
54. end;
55. CloseFile(f);
56. end;
57. procedure TForm1.FormCreate(Sender: TObject);
58. //Открытие и заполнение компоненты stringgrid
59. var
60. f: TextFile;
61. str,t,Fname: string;
62. k,i,j:integer;
63. begin
64. Fname:= 'd:/text.txt';
65. AssignFile (f,fname);
66. {$I-}
67. reset(f);
68. {$I+}
69. if IOResult <> 0 then
70. Rewrite (f);
71. t:='';
72. j:=1;
73. StringGrid1.Cells[0,0]:='№ п/п';
74. StringGrid1.Cells[1,0]:='Клиент';
75. StringGrid1.Cells[2,0]:='Операция';
76. StringGrid1.Cells[3,0]:='Дата начала работы с клиентом';
77. StringGrid1.Cells[4,0]:='Срок абон. платы';
78. while not EOF(f) do begin
79. Readln(f,str);
80. i:=0;
81. for k:=1 to Length(str) do begin
82. if str[k] = '|' then begin
83. StringGrid1.Cells[i,j]:=t;
84. i:=i+1;
85. t:='';
86. end
87. else
88. t:=t+str[k];
89. end;
90. j:=j+1;
91. end;
92. CloseFile(f);
93. end;
94. end.

нт