Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №4**

**«РАБОТА С ТИПИЗИРОВАННЫМИ ФАЙЛАМИ»**

**ПО МДК 05.02 «Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Радыгин Александр Константинович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель работы** – получить представление об организации хранения и использования информации посредством типизированных файлов, закрепить навыки создания пользовательского интерфейса.

**Задание**:

1. Сформировать структуру записи, с которой будет осуществляться работа, и согласовать её с преподавателем.
2. Разработать приложение, позволяющее извлекать набор записей из типизированного файла, визуализировать данный набор, а также позволяющее изменять данные и выполнять сохранение проделанных изменений.

**Результаты выполнения задания**

**Описание алгоритма**

**Алгоритм модуля главной формы**

В программе описан тип данных SpaceShip, являющийся записью, каждый элемент которой содержит определённую информацию о космическом корабле (модель, производитель, наличие пилотов, вместимость, дата запуска и миссия).

Процедура FormCreate отвечает за создание основного окна программы и загрузку данных из файла spaceships.dat в таблицу. Если файл не существует, процедура просто завершается.

Процедура FormClose отвечает за сохранение данных из таблицы обратно в файл spaceships.dat при закрытии программы.

Процедура bgMoveTimerTimer отвечает за анимацию фона программы, который движется вправо-вниз или влево-вверх по экрану в зависимости от своего положения.

Процедура bAddClick запускается при нажатии на кнопку "Добавить". Она открывает окно редактирования, позволяющее добавить новую запись о космическом корабле в таблицу.

Процедура bEditClick запускается при нажатии на кнопку "Изменить". Она открывает окно редактирования для изменения выбранной записи в таблице.

Процедура bDelClick запускается при нажатии на кнопку "Удалить". Она удаляет выбранную запись из таблицы после подтверждения операции.

Процедура bCatClick запускается при нажатии на кнопку "Наблюдать за котами". Она скрывает или показывает таблицу с информацией о космических кораблях и кнопки «Добавить», «Изменить», «Удалить» и «Выйти» в зависимости от состояния переменной isCatting.

Процедура bExitClick отвечает за закрытие основного окна программы при нажатии на кнопку "Выход".

**Алгоритм модуля вспомогательной формы**

Процедура eDateClick вызывается при клике на компонент eDate. Она устанавливает маску редактирования для этого компонента на "00.00.0000", что означает, что пользователь может вводить дату в формате "день.месяц.год". После этого происходит выделение всего текста в компоненте, чтобы вручную не перемещать курсор в его начало.

Процедура checkPilotChange вызывается при изменении состояния флажка checkPilot. Она включает или отключает возможность редактирования компонента eCrewCap в зависимости от того, выбран ли флажок checkPilot.

Процедура catTimerTimer вызывается по таймеру. Она перемещает изображение корабля на один пиксель вправо и на два пикселя вверх. Затем она проверяет, достиг ли корабль верхней границы окна (-600 пикселей). Если достиг, то таймер выключается, чтобы остановить движение корабля.

Процедура FormShow вызывается при отображении формы. Она устанавливает фокус на компонент eModel, устанавливает начальное положение корабля (180 пикселей слева и 600 пикселей сверху) и включает таймер catTimer, чтобы начать анимацию перемещения корабля.

**Исходный код главного модуля**

unit main;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, ExtCtrls,

Grids, edit;

type

{ TfMain }

TfMain = class(TForm)

bAdd: TButton;

bCat: TButton;

bExit: TButton;

bEdit: TButton;

bDel: TButton;

BG: TImage;

bgMoveTimer: TTimer;

SG: TStringGrid;

procedure bAddClick(Sender: TObject);

procedure bCatClick(Sender: TObject);

procedure bDelClick(Sender: TObject);

procedure bEditClick(Sender: TObject);

procedure bExitClick(Sender: TObject);

procedure bgMoveTimerTimer(Sender: TObject);

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

procedure FormCreate(Sender: TObject);

private

public

end;

type

SpaceShip = record

Model: string[100];

Manufactorer: string[100];

Manned: boolean;

CrewCapacity: integer;

Date: string[100];

Mission: string[20];

end;

var

fMain: TfMain;

moveDir, isCatting: boolean;

adres: string;

implementation

{$R \*.lfm}

{ TfMain }

procedure TfMain.bExitClick(Sender: TObject);

begin

fMain.Close;

end;

procedure TfMain.bgMoveTimerTimer(Sender: TObject);

begin

if BG.Left = 0 then

moveDir := False;

if BG.Left = -1500 then

moveDir := True;

if moveDir then begin

BG.Top := BG.Top + 1;

BG.Left := BG.Left + 1;

end

else begin

BG.Top := BG.Top - 1;

BG.Left := BG.Left - 1;

end;

end;

procedure TfMain.FormClose(Sender: TObject; var CloseAction: TCloseAction);

var

MyShip: SpaceShip;

f: file of SpaceShip;

i: integer;

begin

if SG.RowCount = 1 then exit;

try

AssignFile(f, adres + 'spaceships.dat');

Rewrite(f);

for i := 1 to SG.RowCount-1 do begin

MyShip.Model:= SG.Cells[0, i];

MyShip.Manufactorer:= SG.Cells[1, i];

case SG.Cells[2, i] of

'Да': MyShip.Manned := True;

'Нет': MyShip.Manned := False;

end;

MyShip.CrewCapacity:= strtoint(SG.Cells[3, i]);

MyShip.Date:= SG.Cells[4, i];

MyShip.Mission:= SG.Cells[5, i];

Write(f, MyShip);

end;

finally

CloseFile(f);

end;

end;

procedure TfMain.FormCreate(Sender: TObject);

var

MyShip: SpaceShip;

f: file of SpaceShip;

i: integer;

begin

adres:= ExtractFilePath(ParamStr(0));

SG.Cells[0, 0]:= 'Модель';

SG.Cells[1, 0]:= 'Производитель';

SG.Cells[2, 0]:= 'Пилотируемый';

SG.Cells[3, 0]:= 'Вместимость';

SG.Cells[4, 0]:= 'Дата запуска';

SG.Cells[5, 0]:= 'Миссия';

SG.ColWidths[0]:= 345;

SG.ColWidths[1]:= 230;

SG.ColWidths[2]:= 230;

SG.ColWidths[3]:= 200;

SG.ColWidths[4]:= 200;

SG.ColWidths[5]:= 170;

if not FileExists(adres + 'spaceships.dat') then exit;

try

AssignFile(f, adres + 'spaceships.dat');

Reset(f);

while not Eof(f) do begin

Read(f, MyShip);

SG.RowCount:= SG.RowCount + 1;

SG.Cells[0, SG.RowCount-1]:= MyShip.Model;

SG.Cells[1, SG.RowCount-1]:= MyShip.Manufactorer;

if MyShip.Manned then

SG.Cells[2, SG.RowCount-1] := 'Да'

else

SG.Cells[2, SG.RowCount-1] := 'Нет';

SG.Cells[3, SG.RowCount-1]:= inttostr(MyShip.CrewCapacity);

SG.Cells[4, SG.RowCount-1]:= MyShip.Date;

SG.Cells[5, SG.RowCount-1]:= MyShip.Mission;

end;

finally

CloseFile(f);

end;

end;

procedure TfMain.bCatClick(Sender: TObject);

begin

SG.Visible := isCatting;

bAdd.Visible := isCatting;

bEdit.Visible := isCatting;

bDel.Visible := isCatting;

bExit.Visible := isCatting;

if isCatting then

bCat.Caption := 'Наблюдать за котами'

else

bCat.Caption := 'Вернуться к работе';

isCatting := not isCatting;

end;

procedure TfMain.bDelClick(Sender: TObject);

begin

if SG.RowCount = 1 then exit;

if MessageDlg('Требуется подтверждение',

'Вы действительно хотите удалить запись "' +

SG.Cells[0, SG.Row] + '"?',

mtConfirmation, [mbYes, mbNo, mbIgnore], 0) = mrYes then

SG.DeleteRow(SG.Row);

end;

procedure TfMain.bEditClick(Sender: TObject);

begin

if SG.RowCount = 1 then exit;

bgMoveTimer.Enabled := False;

fEdit.eModel.Text := SG.Cells[0, SG.Row];

fEdit.eManuf.Text := SG.Cells[1, SG.Row];

case SG.Cells[2, SG.Row] of

'Да': fEdit.checkPilot.Checked := True;

'Нет': fEdit.checkPilot.Checked := False;

end;

fEdit.eCrewCap.Text := SG.Cells[3, SG.Row];

fEdit.eDate.Text := SG.Cells[4, SG.Row];

fEdit.eMission.Text := SG.Cells[5, SG.Row];

fEdit.ModalResult:= mrNone;

fEdit.ShowModal;

if (fEdit.eModel.Text <> '') and (fEdit.eManuf.Text <> '') and

(fEdit.eMission.Text <> '') and (fEdit.eDate.Text <> '') and

(fEdit.eCrewCap.Text <> '') and (fEdit.ModalResult = mrOk) then begin

SG.Cells[0, SG.Row] := fEdit.eModel.Text;

SG.Cells[1, SG.Row] := fEdit.eManuf.Text;

if fEdit.checkPilot.Checked then

SG.Cells[2, SG.RowCount-1] := 'Да'

else

SG.Cells[2, SG.Row] := 'Нет';

SG.Cells[3, SG.Row] := fEdit.eCrewCap.Text;

SG.Cells[4, SG.Row] := fEdit.eDate.Text;

SG.Cells[5, SG.Row] := fEdit.eMission.Text;

end;

bgMoveTimer.Enabled := True;

exit;

end;

procedure TfMain.bAddClick(Sender: TObject);

begin

bgMoveTimer.Enabled := False;

fEdit.eModel.Text := '';

fEdit.eManuf.Text := '';

fEdit.eMission.Text := '';

fEdit.eCrewCap.Text := '';

fEdit.eDate.Text := '';

fEdit.eDate.EditMask := '';

fEdit.checkPilot.Checked := False;

fEdit.eCrewCap.Enabled := False;

fEdit.ModalResult:= mrNone;

fEdit.ShowModal;

if (fEdit.eModel.Text <> '') and (fEdit.eManuf.Text <> '') and

(fEdit.eMission.Text <> '') and (fEdit.eDate.Text <> '') and

(fEdit.eCrewCap.Text <> '') and (fEdit.ModalResult = mrOk) then begin

SG.Cells[0, SG.RowCount-1] := fEdit.eModel.Text;

SG.Cells[1, SG.RowCount-1] := fEdit.eManuf.Text;

if fEdit.checkPilot.Checked then

SG.Cells[2, SG.RowCount-1] := 'Да'

else

SG.Cells[2, SG.RowCount-1] := 'Нет';

SG.Cells[3, SG.RowCount-1] := fEdit.eCrewCap.Text;

SG.Cells[4, SG.RowCount-1] := fEdit.eDate.Text;

SG.Cells[5, SG.RowCount-1] := fEdit.eMission.Text;

end;

bgMoveTimer.Enabled := True;

exit;

end;

end.

**Исходный код вспомогательного модуля**

unit edit;

{$mode ObjFPC}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, Buttons,

MaskEdit, ExtCtrls;

type

{ TfEdit }

TfEdit = class(TForm)

bSave: TBitBtn;

bCancel: TBitBtn;

checkPilot: TCheckBox;

eCrewCap: TEdit;

eMission: TComboBox;

eModel: TEdit;

eManuf: TEdit;

eDate: TMaskEdit;

catShip: TImage;

catTimer: TTimer;

Title: TLabel;

procedure catTimerTimer(Sender: TObject);

procedure checkPilotChange(Sender: TObject);

procedure eDateClick(Sender: TObject);

procedure FormShow(Sender: TObject);

private

public

end;

var

fEdit: TfEdit;

implementation

{$R \*.lfm}

{ TfEdit }

procedure TfEdit.eDateClick(Sender: TObject);

begin

eDate.EditMask := '00.00.0000;\_';

eDate.SelectAll;

end;

procedure TfEdit.checkPilotChange(Sender: TObject);

begin

eCrewCap.Enabled := checkPilot.Checked;

end;

procedure TfEdit.catTimerTimer(Sender: TObject);

begin

catShip.Left := catShip.Left + 1;

catShip.Top := catShip.Top - 2;

if catShip.Top = -600 then

catTimer.Enabled := False;

end;

procedure TfEdit.FormShow(Sender: TObject);

begin

eModel.SetFocus;

catShip.Left := 180;

catShip.Top := 600;

catTimer.Enabled := True;

end;

end.

**Результат выполнения программы**

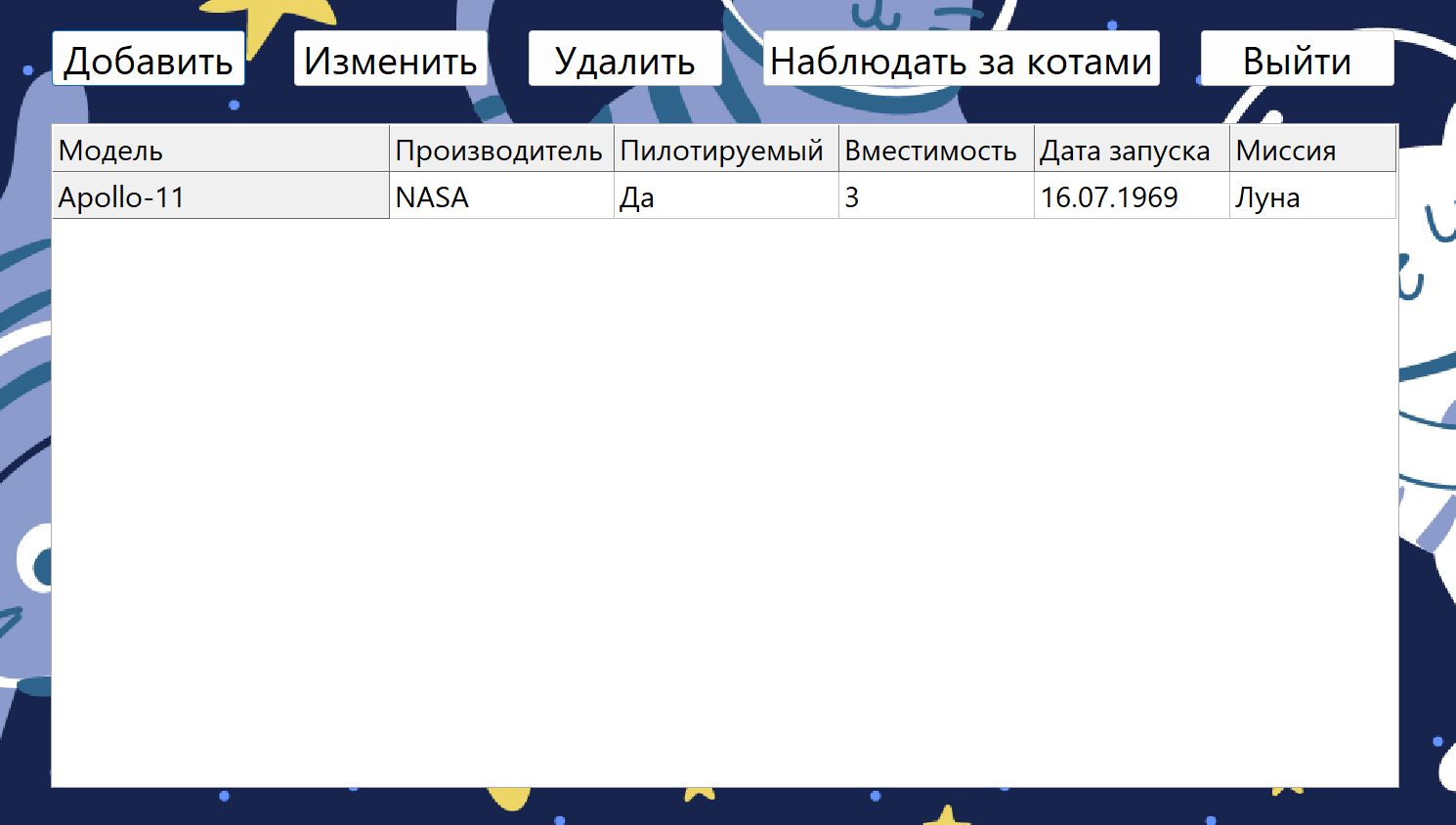


Рис. 1 – главная форма

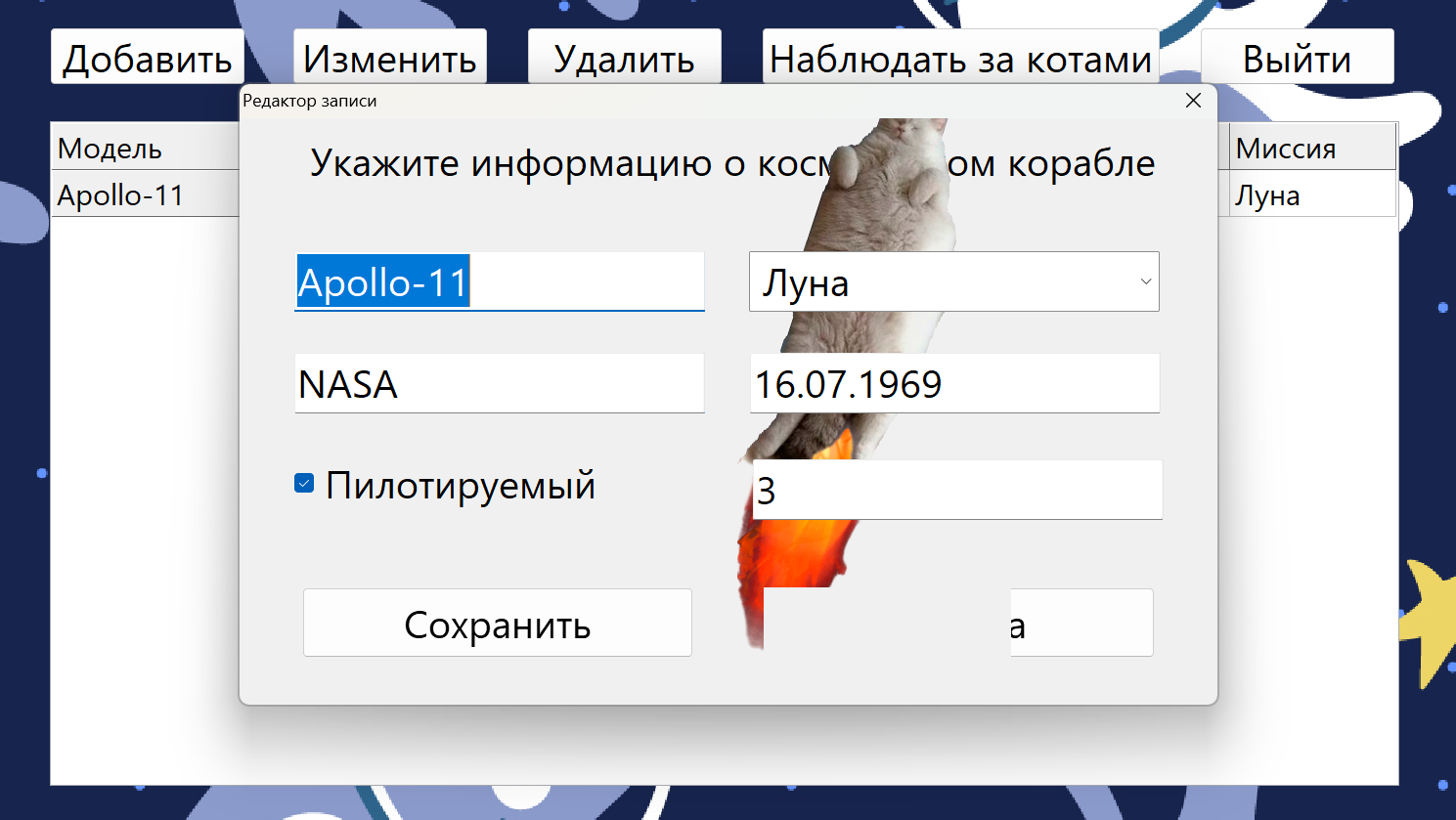


Рис. 2 – окно редактора записи

**Вывод**

В ходе выполнения данной лабораторной работы была разработана программа, представляющая собой инструмент для визуализации и редактирования хранимой в типизированном файле информации о космических кораблях. Написанные для корректного функционирования приложения функции реагируют на действия пользователя, управляя графическим интерфейсом и преобразовывая вводимые данные.

Таким образом, эта работа помогла закрепить навыки в области программирования на языке Pascal, работы с графическим интерфейсом пользователя и обработки событий, а также ознакомиться с организацией хранения и использования информации посредством типизированных файлов.