Министерство образования и науки Российской Федерации

Федеральное агентство по образованию

Государственное образовательное учреждение высшего профессионального образования

« Комсомольский - на - Амуре государственный технический университет »

Факультет компьютерных технологий

Кафедра ПМИ

РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ

по дисциплине ‹‹Языки программирования››

Студент группы 8КБ-1 Д.И.Карташов

Преподаватель И.А. Трещёв

**Тема:** Изучение IDE(Integrated Development Environment) Turbo Pascal 7.0, Borland C++ 3.11, Delphi 7, Builder C++6.0 .

**Цель работы:** На четырех языках программирования (Turbo Pascal 7.0, Borland C++ 3.11, Delphi 7, Builder C++ 6.0) написать программу, которая выводит на экран имя и фамилию студента. Для написания программы использовать стандартные средства графики и простейшие геометрические фигуры.

**Содержание**

Введение

1. Блок кода

2. Блок выполнения программы

Вывод

Список использованных источников

**Введение**

Turbo Pascal — Интегрированная среда разработки программного обеспечения для платформ DOS и Windows 3.x и язык программирования в этой среде, диалект языка Паскаль от фирмы Borland.

Turbo Pascal — это среда разработки для языка программирования Паскаль. Используемый в Turbo Pascal диалект базировался на более раннем UCSD Pascal, получившем распространение, в первую очередь, на компьютерах серии Apple II. Компилирующая компонента Turbo Pascal была основана на компиляторе Blue Label Pascal, первоначально созданном в 1981 году Андерсом Хейлсбергом для операционной системы NasSys микрокомпьютера Nascom. Позднее он был переписан как Compass Pascal для операционной системы CP/M, затем как Turbo Pascal для DOS и CP/M.

Достоинства:

1.Удобная среда разработки, включающая функциональный отладчик, доступный в любой момент.

2.Контекстная справочная система, по которой можно изучать язык без обращения к сторонним источникам.

3.Высокая скорость компиляции, высокая скорость выполнения откомпилированных программ.

4.Встроенная возможность использовать вставки на языке ассемблера.

Недостатки:

1.Компилятор рассчитан на реальный режим DOS, применение которого сходит на нет. Однако в последних версиях компилятора и среды введена поддержка защищённого режима вместе с соответствующим отладчиком (TD).

2.В модуле CRT имеется ошибка (некорректный подсчёт количества циклов для функции delay, не рассчитанный на быстрые процессоры, процессоры с переменной частотой и многозадачные среды), из-за которой при запуске программы на компьютерах с тактовой частотой более 200 MHz сразу происходило аварийное завершение с сообщением «Runtime error 200 at…». Существуют разные варианты исправления модуля CRT. В варианте Клауса Хартнегга ошибка 200 не возникает, но длительность Delay на быстрых компьютерах меньше желаемой, и эта проблема по незнанию иногда тоже считается недостатком Turbo Pascal.

Borland C++ — среда программирования (IDE) на языках Си и C++ для DOS, Windows и Windows NT. Потомок Turbo C. Его дебаггер Turbo Debugger был написан для защищённого режима DOS.

C++ Builder (по-русски обычно произносят [си-плюс-плюс би́лдэр], [си би́лдэр]) - программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке C++.

C++ Builder объединяет в себе комплекс объектных библиотек (STL, VCL, CLX, MFC и др.), компилятор, отладчик, редактор кода и многие другие компоненты. Цикл разработки аналогичен Delphi. Большинство компонентов, разработанных в Delphi, можно использовать и в C++ Builder без модификации, но, к сожалению, обратное утверждение не верно.

C++ Builder содержит инструменты, которые при помощи drag-and-drop действительно делают разработку визуальной, упрощает программирование благодаря встроенному WYSIWYG - редактору интерфейса и пр.

Delphi — язык программирования, который используется в одноимённой среде разработки. Сначала язык назывался Object Pascal. Начиная со среды разработки Delphi 7.0, в официальных документах Borland стала использовать название Delphi для обозначения языка Object Pascal.

Delphi — результат развития языка Турбо Паскаль, который, в свою очередь, развился из языка Паскаль. Паскаль был полностью процедурным языком, Турбо Паскаль, начиная с версии 5.5, добавил в Паскаль объектно-ориентированные свойства, а в Object Pascal динамическую идентификацию типа данных с возможностью доступа к метаданным классов (то есть к описанию классов и их членов) в компилируемом коде, также называемом интроспекцией — данная технология получила обозначение RTTI. Так как все классы наследуют функции базового класса TObject, то любой указатель на объект можно преобразовать к нему, после чего воспользоваться методом ClassType и функцией TypeInfo, которые и обеспечат интроспекцию.

Также отличительным свойством Object Pascal от С++ является то, что объекты по умолчанию располагаются в динамической памяти. Однако можно переопределить виртуальные методы NewInstance и FreeInstance класса TObject. Таким образом, абсолютно любой класс может осуществить «желание» «где хочу — там и буду лежать». Соответственно организуется и «многокучность».

**1. Блок кода**

**Рисование с помощью графики**

**Turbo Pascal 7.0**

Program myName;

uses Graph;

var

grDriver: Integer;

grMode: Integer;

ErrCode: Integer;

begin

grDriver := Detect;

InitGraph(grDriver, grMode,'D:\BP\BGI');

ErrCode := GraphResult;

if ErrCode = grOk then

begin

{Family}

Line(10, 10,10 ,50 );

Line(10,20,30,5);

Line(10,30,30,50);

Line(40,50,50,10);

Line(50,10,60,50);

Line(40,30,65,30);

Line(70,10,70,50);

Line(70,10,80,10);

Line(70,30,80,30);

Line(80,10,80,30);

Line(80,10,100,10);

Line(90,10,90,50);

Line(100,50,110,10);

Line(110,10,120,50);

Line(100,30,120,30);

Line(125,10,125,50);

Line(130,10,130,50);

Line(135,10,135,50);

Line(125,50,135,50);

Circle(155,30,18);

Line(180,10,180,50);

Circle(190,20,9);

Circle(190,40,9);

{Name}

Line(30,100,40,80);

Line(40,80,50,100);

Line(25,100,55,100);

Line(25,100,25,105);

Line(55,100,55,105);

Line(60,105,70,80);

Line(70,80,90,105);

Line(60,90,90,90);

Line(107,105,107,70);

Line(107,85,115,85);

Line(115,105,115,70);

Line(120,70,120,105);

Line(120,105,130,70);

Line(130,70,130,105);

Line(135,105,145,70);

Line(145,70,155,105);

Readln;

CloseGraph;

end

else

Writeln('Graphics error:', GraphErrorMsg(ErrCode));

end.

**Borland C++ 3.11**

#include <graphics.h>

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

int main(void)

{

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "C:\BORLANDC\BGI");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

{Family}

Line(10, 10,10 ,50 );

Line(10,20,30,5);

Line(10,30,30,50);

Line(40,50,50,10);

Line(50,10,60,50);

Line(40,30,65,30);

Line(70,10,70,50);

Line(70,10,80,10);

Line(70,30,80,30);

Line(80,10,80,30);

Line(80,10,100,10);

Line(90,10,90,50);

Line(100,50,110,10);

Line(110,10,120,50);

Line(100,30,120,30);

Line(125,10,125,50);

Line(130,10,130,50);

Line(135,10,135,50);

Line(125,50,135,50);

Circle(155,30,18);

Line(180,10,180,50);

Circle(190,20,9);

Circle(190,40,9);

{Name}

Line(30,100,40,80);

Line(40,80,50,100);

Line(25,100,55,100);

Line(25,100,25,105);

Line(55,100,55,105);

Line(60,105,70,80);

Line(70,80,90,105);

Line(60,90,90,90);

Line(107,105,107,70);

Line(107,85,115,85);

Line(115,105,115,70);

Line(120,70,120,105);

Line(120,105,130,70);

Line(130,70,130,105);

Line(135,105,145,70);

Line(145,70,155,105);

getch();

closegraph();

return 0;

}

**Delphi7**

unit GRAPH;

interface

uses

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

Dialogs, StdCtrls, ExtCtrls;

type

TForm1 = class(TForm)

Button1: TButton;

Image1: TImage;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

with Image1 do begin

{К}

Canvas.MoveTo(140,110);

Canvas.LineTo(140,170);

Canvas.MoveTo(170,110);

Canvas.LineTo(140,140);

Canvas.LineTo(170,170);

{А}

Canvas.MoveTo(145,90);

Canvas.LineTo(160,30);

Canvas.LineTo(175,90);

Canvas.MoveTo(155,60);

Canvas.LineTo(170,60);

{Р}

Canvas.MoveTo(145,90);

Canvas.LineTo(160,30);

Canvas.LineTo(175,90);

Canvas.MoveTo(155,60);

Canvas.LineTo(170,60);

{А}

Canvas.MoveTo(180,90);

Canvas.LineTo(160,35);

Canvas.LineTo(175,90);

Canvas.MoveTo(155,60);

Canvas.LineTo(170,60);

{Ш}

Canvas.MoveTo(60,30);

Canvas.LineTo(60,90);

Canvas.LineTo(90,30);

Canvas.LineTo(90,90);

{О}

Canvas.Ellipse(225,30,255,90);

{В}

Canvas.MoveTo(265,30);

Canvas.LineTo(265,90);

Canvas.Ellipse(265,30,295,60);

Canvas.Ellipse(265,60,300,90);

{Д}

Canvas.MoveTo(100,110);

Canvas.LineTo(100,170);

Canvas.LineTo(130,110);

Canvas.LineTo(130,170);

Canvas.MoveTo(170,110);

Canvas.LineTo(140

{А}

Canvas.MoveTo(140,110);

Canvas.LineTo(140,170);

Canvas.MoveTo(170,110);

Canvas.LineTo(140,140);

Canvas.LineTo(170,170);

{Н}

Canvas.MoveTo(180,110);

Canvas.LineTo(180,170);

Canvas.LineTo(210,110);

Canvas.LineTo(210,170);

{И}

Canvas.MoveTo(220,110);

Canvas.LineTo(250,110);

Canvas.MoveTo(235,110);

Canvas.LineTo(235,170);

{Л}

Canvas.MoveTo(260,170);

Canvas.LineTo(275,110);

Canvas.LineTo(290,170);

Canvas.MoveTo(270,140);

end;

end;

end.

**Builder C++ 6.0**

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

Form1->Image1-> Canvas->

Form1->Image1-> Canvas->MoveTo(60,30);

Form1->Image1-> Canvas->LineTo(60,90);

Form1->Image1-> Canvas->LineTo(90,30);

Form1->Image1-> Canvas->LineTo(90,90);

Form1->Image1-> Canvas->MoveTo(100,30);

Form1->Image1-> Canvas->LineTo(100,90);

Form1->Image1-> Canvas->Ellipse(100,30,130,60);

Form1->Image1-> Canvas->Ellipse(100,60,135,90);

Form1->Image1-> Canvas->MoveTo(145,90);

Form1->Image1-> Canvas->LineTo(160,30);

Form1->Image1-> Canvas->LineTo(175,90);

Form1->Image1-> Canvas->MoveTo(155,60);

Form1->Image1-> Canvas->LineTo(170,60);

Form1->Image1-> Canvas->MoveTo(185,30);

Form1->Image1-> Canvas->LineTo(185,90);

Form1->Image1-> Canvas->MoveTo(185,60);

Form1->Image1-> Canvas->LineTo(215,60);

Form1->Image1-> Canvas->MoveTo(215,30);

Form1->Image1-> Canvas->LineTo(215,90);

Form1->Image1-> Canvas->Ellipse(225,30,255,90);

Form1->Image1-> Canvas->MoveTo(265,30);

Form1->Image1-> Canvas->LineTo(265,90);

Form1->Image1-> Canvas->MoveTo(60,110);

Form1->Image1-> Canvas->LineTo(60,170);

Form1->Image1-> Canvas->MoveTo(60,140);

Form1->Image1-> Canvas->LineTo(90,140);

Form1->Image1-> Canvas->MoveTo(90,110);

Form1->Image1-> Canvas->LineTo(90,170);

Form1->Image1-> Canvas->MoveTo(100,110);

Form1->Image1-> Canvas->LineTo(100,170);

Form1->Image1-> Canvas->LineTo(130,110);

Form1->Image1-> Canvas->LineTo(130,170);

Form1->Image1-> Canvas->MoveTo(140,110);

Form1->Image1-> Canvas->LineTo(140,170);

Form1->Image1-> Canvas->MoveTo(170,110);

Form1->Image1-> Canvas->LineTo(140,140);

Form1->Image1-> Canvas->LineTo(170,170);

Form1->Image1-> Canvas->MoveTo(180,110);

Form1->Image1-> Canvas->LineTo(180,170);

Form1->Image1-> Canvas->LineTo(210,110);

Form1->Image1-> Canvas->LineTo(210,170);

Form1->Image1-> Canvas->MoveTo(220,110);

Form1->Image1-> Canvas->LineTo(250,110);

Form1->Image1-> Canvas->MoveTo(235,110);

Form1->Image1-> Canvas->LineTo(235,170);

Form1->Image1-> Canvas->MoveTo(260,170);

Form1->Image1-> Canvas->LineTo(275,110);

Form1->Image1-> Canvas->LineTo(290,170);

Form1->Image1-> Canvas->MoveTo(270,140);

Form1->Image1-> Canvas-> LineTo(280,140);

}

//---------------------------------------------------------------------------

**Вывод текста на экран**

**Turbo Pascal 7.0**

program name;

uses crt;

begin

clrscr;

write('Karasov Danil ');

readln;

end.

**Borland C++ 3.11**

#include<stdio.h>

int main (void)

{

printf("'Karasov Danil ");

return 0;

}

**Delphi7**

unit Unit1;

interface

uses

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

Dialogs, StdCtrls;

type

TForm1 = class(TForm)

Label1: TLabel;

Button1: TButton;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

Label1.Caption:='Карташов Данил';

end;

end.

**Borland C++ 3.11**

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

Form1->Label1->Caption = "Карташов Данил";

}

//---------------------------------------------------------------------------

**2. Блок выполнения программы**

**Рисование с помощью графики**

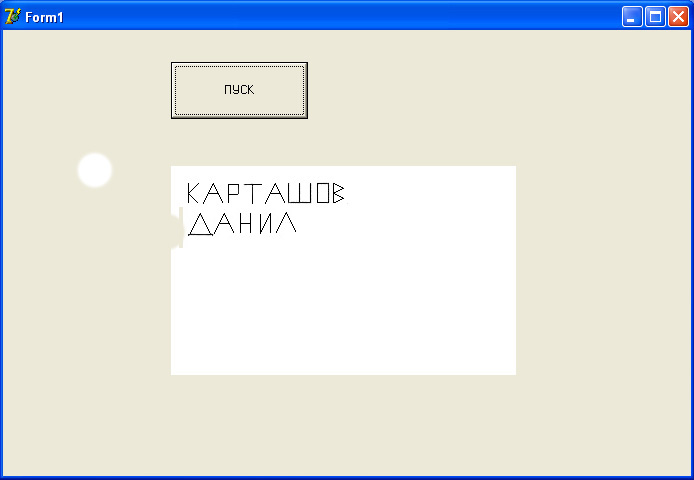
Turbo Pascal 7.0



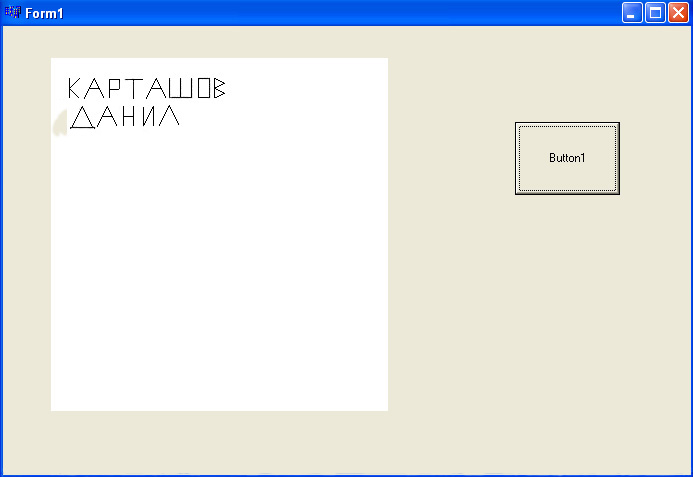
Borland C++ 3.11



Delphi7



Builder C++ 6.0

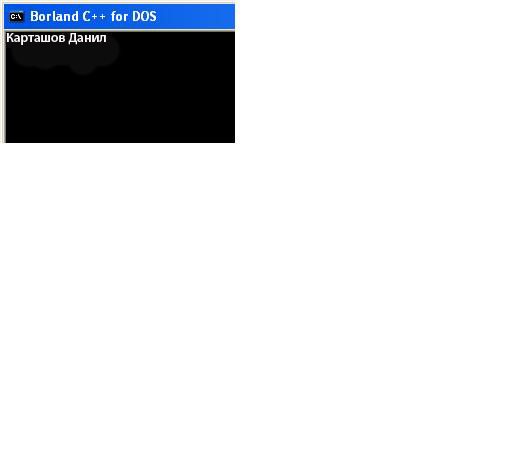


**Вывод текста на экран**

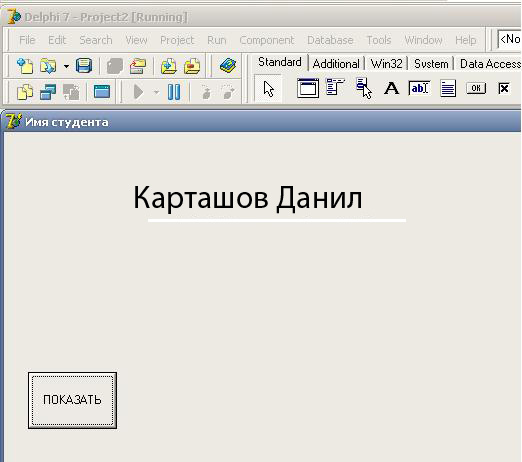
Turbo Pascal 7.0



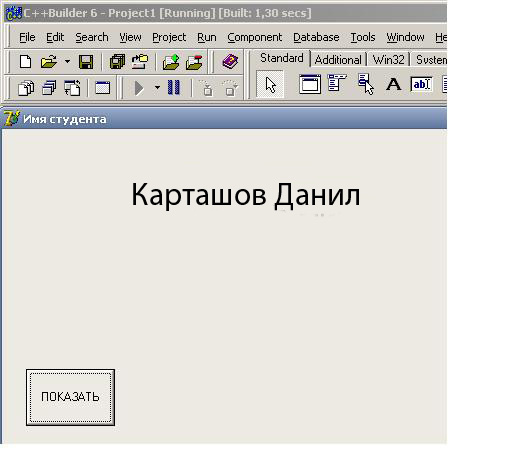
Borland C++ 3.11



Delphi7



Builder C++ 6.0



**Вывод**

Была проведена работа по написанию программы в четырех языковых средах, выводящей имя и фамилию студента на экран по средствам графики с использование простейших геометрических фигур.

**Список использованных источников**

1. Джаррод Холингворт, Боб Сворт, Марк Кэшмэн, Поль Густавсон Borland C++ Builder 6. Руководство разработчика = Borland C++ Builder 6 Developer's Guide. — М.: «Вильямс», 2004. — С. 976.

2. Нил Дж. Рубенкинг. Язык программирования Delphi для «чайников». Введение в Borland Delphi 2006 = Delphi for Dummies. — М.: Диалектика, 2007. — 336 с.

3. Хавьер Пашеку. Программирование в Borland Delphi 2006 для профессионалов = Delphi for .NET Developer’s Guide. — М.: Вильямс, 2006. — 944 с.

4. Е.А. Зуев. Программирование на языке Turbo Pascal 6.0, 7.0, М.:Веста,Радио и связь, 1993, — С.376.

5. Моргун Александр Николаевич. Справочник по Turbo Pascal для студентов. — М.: Диалектика, 2006. — С. 608.