



Учебный курс	ППО	СПО		
Волшебный компьютер	MS Office 2000 и выше	OpenOffice.org	+	+
Занимательное ЛОГОзнание 35 часов, 5—7 кл.		MSWLogo	+	+
Увлекательные уроки программирования. Pascal 70 часов, 8—9кл.	Borland Pascal 7.0, Turbo Pascal 7.0, Free Pascal 2.0 (или выше)	Borland Pascal под DosBoxкласс	+	+
Объектно-ориентированное программирование 70 часов, 10—11 кл.	Borland Developer Studio 2006 и выше	Lazarus	+	+
Программирование на языке Си 70 часов, 10—11 кл.	Turbo C		+	

Цели обучения по направлению «Программирование»:

- понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации;
- овладеть базовыми понятиями теории алгоритмов;
- научиться разрабатывать эффективные алгоритмы и реализовывать их в виде программы;
- научиться отлаживать и тестировать программы, делать выводы о работе этих программ.

Курс Занимательное ЛОГОзнание

предназначен для школьников 5-7 классов.

Рекомендуется как пропедевтический для курса «Алгоритмизация и программирование на языке Pascal».

Входит в сертификат Образовательные программы I («начального») уровня обучения.

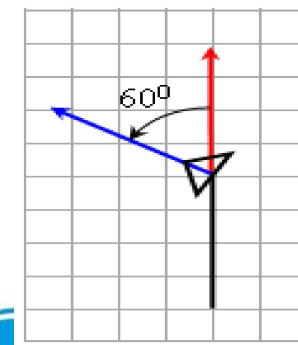


Учащийся получит навыки:

- составления алгоритмов и реализации их исполнителем;
- в дальнейшем освоении программирования на Pascal, затем на Delphi.

Курс широко использует межпредметные связи:

- математика, чтение, русский язык, окружающий мир, изобразительное искусство, музыка.



Раздел 1. Черепашня графика

Исполнитель Черепашка. Процедура. Расстояние и угол

Циклический алгоритм

Цвет. RGB-модель цвета

Вывод текста. Атрибуты шрифта

Раздел 2. Программирование с Черепашкой

Переменные. Параметры процедур

Типы данных: число, слово, список

Ветвление

Команды ввода-вывода

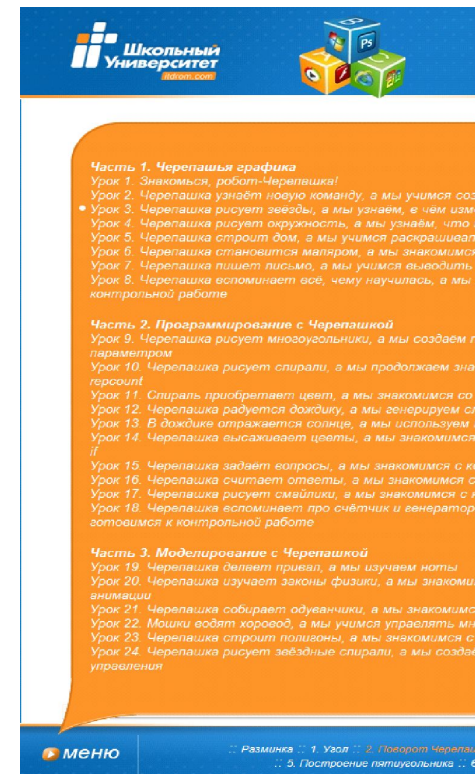
Цикл «пока»

Раздел 3. Моделирование с Черепашкой

Звук и анимация

Рекурсия

3D-моделирование



Школьный
Университет

Часть 1. Черепашня графика

- Урок 1. Знакомься, робот-Черепашка!
- Урок 2. Черепашка узнаёт новую команду, а мы учимся со...
- Урок 3. Черепашка рисует звезды, а мы узнаем, с чем изме...
- Урок 4. Черепашка рисует окружность, а мы узнаём, что л...
- Урок 5. Черепашка строит дом, а мы учимся раскрашивать...
- Урок 6. Черепашка становится маляром, а мы знакомимся...
- Урок 7. Черепашка пишет письмо, а мы учимся выводить н...
- Урок 8. Черепашка вспоминает все, чему научилась, а мы с...

Часть 2. Программирование с Черепашкой

- Урок 9. Черепашка рисует многоугольники, а мы создаем пр...
- Урок 10. Черепашка рисует спирали, а мы продолжаем знак...
- Урок 11. Спираль приобретает цвет, а мы знакомимся со с...
- Урок 12. Черепашка радуется дождю, а мы генерируем сл...
- Урок 13. В дождике отражается солнце, а мы используем л...
- Урок 14. Черепашка высказывает цветы, а мы знакомимся...
- Урок 15. Черепашка задает вопросы, а мы знакомимся с ко...
- Урок 16. Черепашка считает ответы, а мы знакомимся с н...
- Урок 17. Черепашка рисует смайлики, а мы знакомимся с н...
- Урок 18. Черепашка вспоминает про счётчик и генератор, г...

Часть 3. Моделирование с Черепашкой

- Урок 19. Черепашка делит привал, а мы изучаем ноты
- Урок 20. Черепашка изучает законы физики, а мы знакомим...
- Урок 21. Черепашка собирает одуванчики, а мы знакомимся...
- Урок 22. Машки водят хороход, а мы учимся управлять мн...
- Урок 23. Черепашка строит полигоны, а мы знакомимся с...
- Урок 24. Черепашка рисует звёздные спирали, а мы создаём...

МЕНЮ

Разминка 1. Угол 2. Поворот Черепашки 3. Построение пятиугольника 4. 5. 6.

Раздел 1. Черепашья графика

Знакомьтесь, робот-Черепашка!

Школьный Университет

Занимательное ЛОГОзнание

Упражнения
Справочник
Задания

Урок 1. Знакомьтесь, робот-Черепашка!

MSWLogo Screen

File Bitmap Set Zoom Help

HELLO

Commander

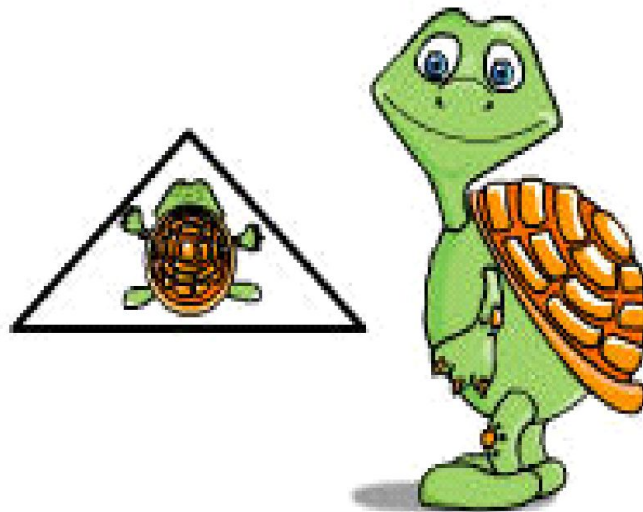
```
fd 50
lt 90
fd 30
pu
fd 20
pd
lt 90
```

Halt Trace
Pause Status
Step Reset

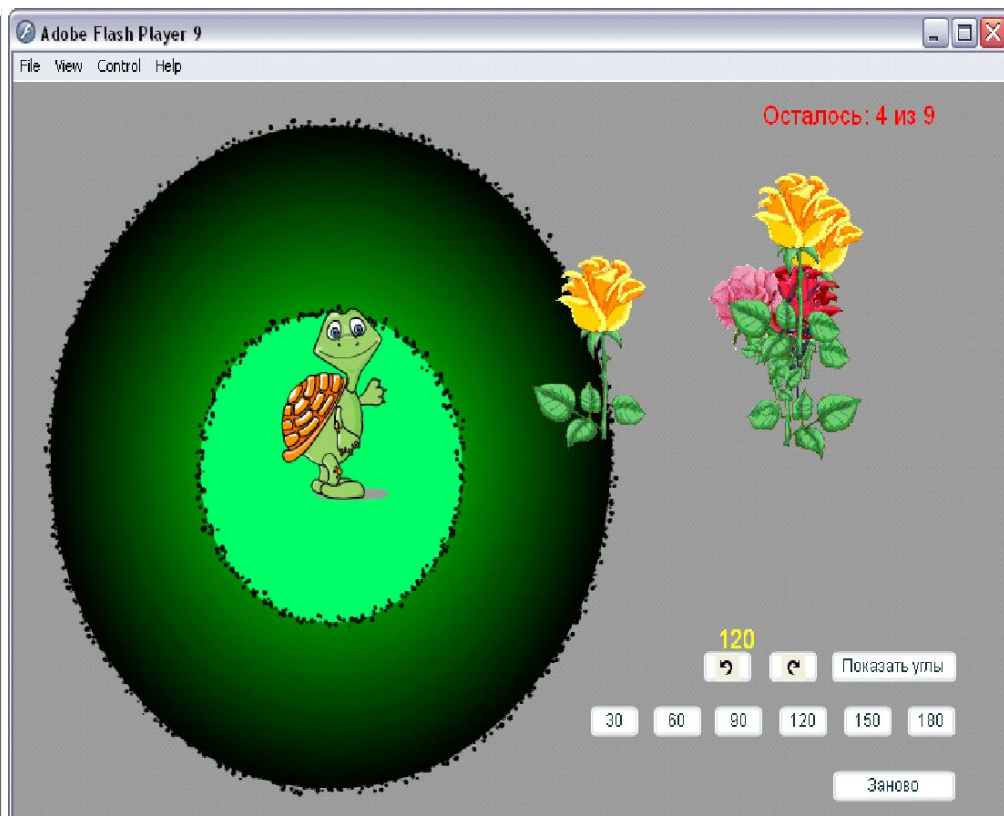
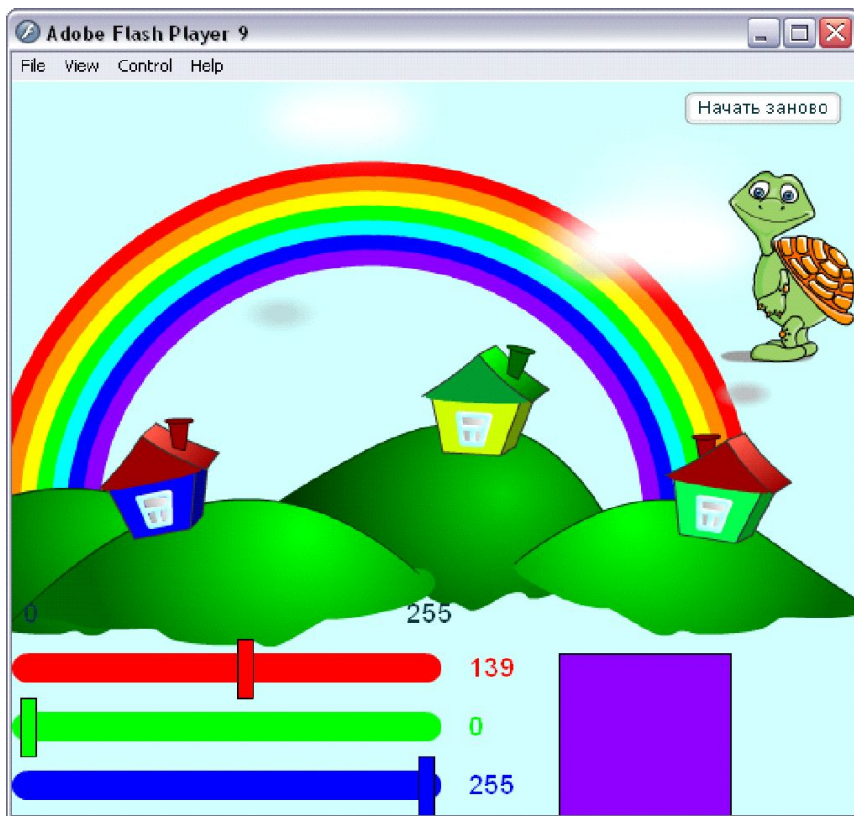
Execute Edit

МЕНЮ

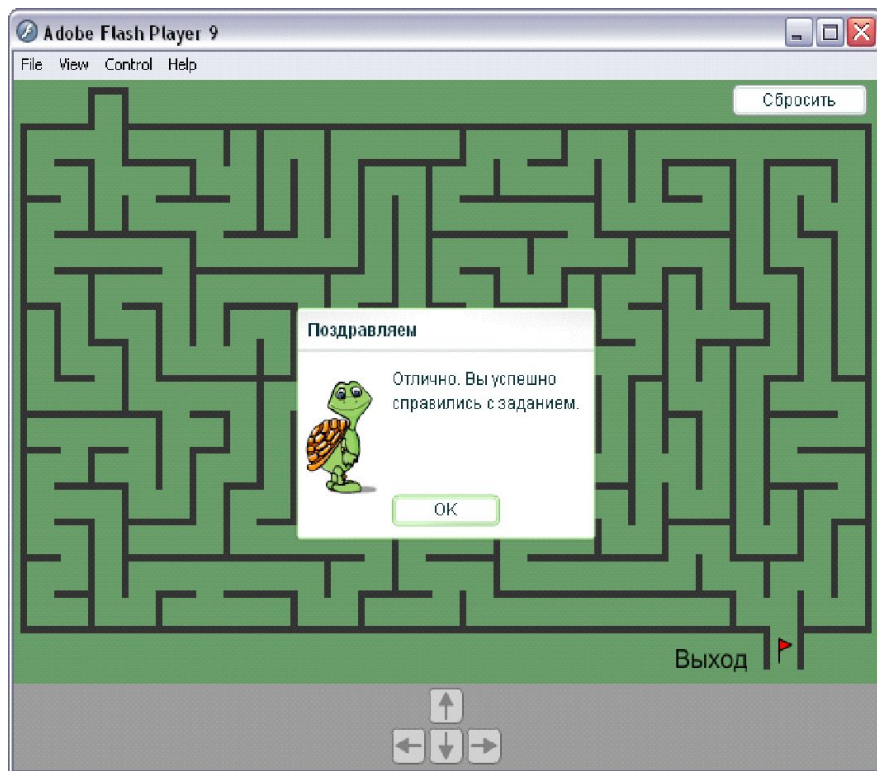
Разделы: 1. Знакомьтесь, Черепашка-робот; 2. Движение Черепашки; 3. Повороты Черепашки; 4. Голосистые; 5. Перо Черепашки; 6. Выбор цвета; 7. Тест...



Раздел 2. Программирование с Черепашкой

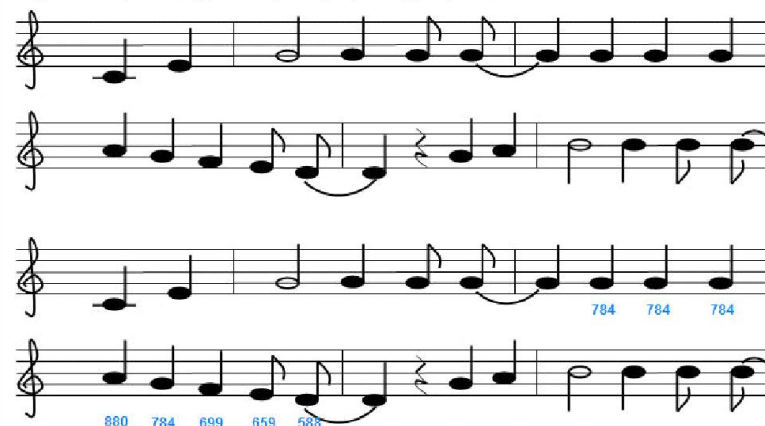


Раздел 3. Моделирование с Черепашкой



Продолжим кодирование мелодии.

- Добавьте новую команду **sound**, выводящую следующие 8 нот:



Для проверки

```

Editor
File Edit Search Set Test! Help
to kartoshka :d1
  sound (list 523 :d1 659 :d1 784 :d1*2 784 :d1 784 :d1/2 784 :d1*3/2)
  sound (list 784 :d1 784 :d1 784 :d1 880 :d1 784 :d1 699 :d1 659 :d1/2 588 :d1*3/2)
end
    
```


Здоровьесберегающие технологии применяете Вы при обучении школьников 5-7 классов?

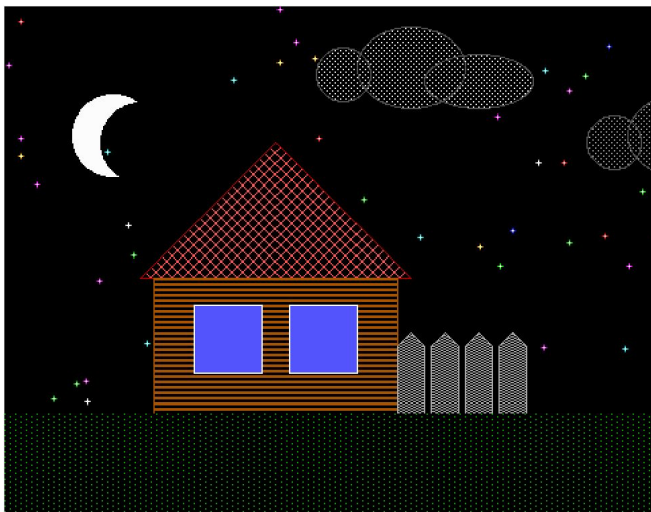
- Гимнастика на уроке
- Ограничение времени работы за компьютером
- Смена видов деятельности
- Все вышеперечисленные



Курс **Увлекательные уроки программирования** (70 часов).

Рассчитан на учащихся 9-11 классов.

Курс входит в состав образовательных программ на получение сертификата "Программист" (10 класс).



```
File Edit Search Run Compile Debug Tools Options Window Help
Ivanou\pr5.pas 1
Ivanou\pr6.pas 3-[↑]
[ ]
PROGRAM pr6;
USES Graph;
VAR
  gd, gm: integer;
  i, j: integer;
Procedure Star(x, y, c: integer);
Begin
  PutPixel(x, y, 15);
  PutPixel(x+1, y, c);
  PutPixel(x-1, y, c);
  PutPixel(x, y+1, c);
  PutPixel(x, y-1, c);
  PutPixel(x+2, y, c-8);
  PutPixel(x-2, y, c-8);
  PutPixel(x, y+2, c-8);
  PutPixel(x, y-2, c-8);
End;
BEGIN
  randomize;
  gd:=UGA;
  gm:=7;
  InitGraph(gd, gm, '');
  for i:=1 to 80 do
    Star<random<640>, random<480>, random<7>+9>;
1:1 =
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```



Урок 11. И получилась звёздная дорожка, или Цикл с параметром

Для начала сделаем звёзды разноцветными.

- Вы, конечно, помните, что в Pascal доступны 16 цветов (с номерами от 0 до 15). Какой командой можно загадать случайное число из этого промежутка? `random (``)`

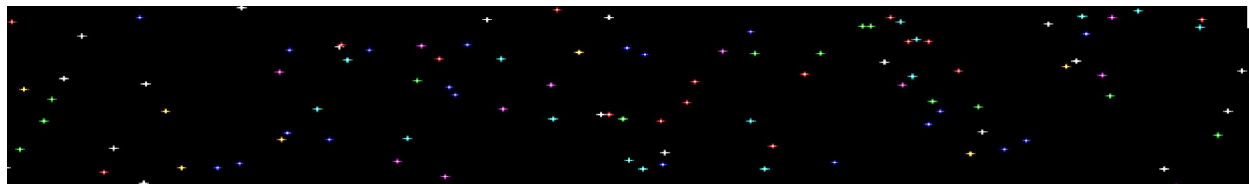
Проверить

Молодец! Верно!

- Измените программу так, чтобы на экран выводились точки случайного цвета.

📌 Для проверки

- Запустите её:



Структура курса

Раздел 1. Основы алгоритмизации

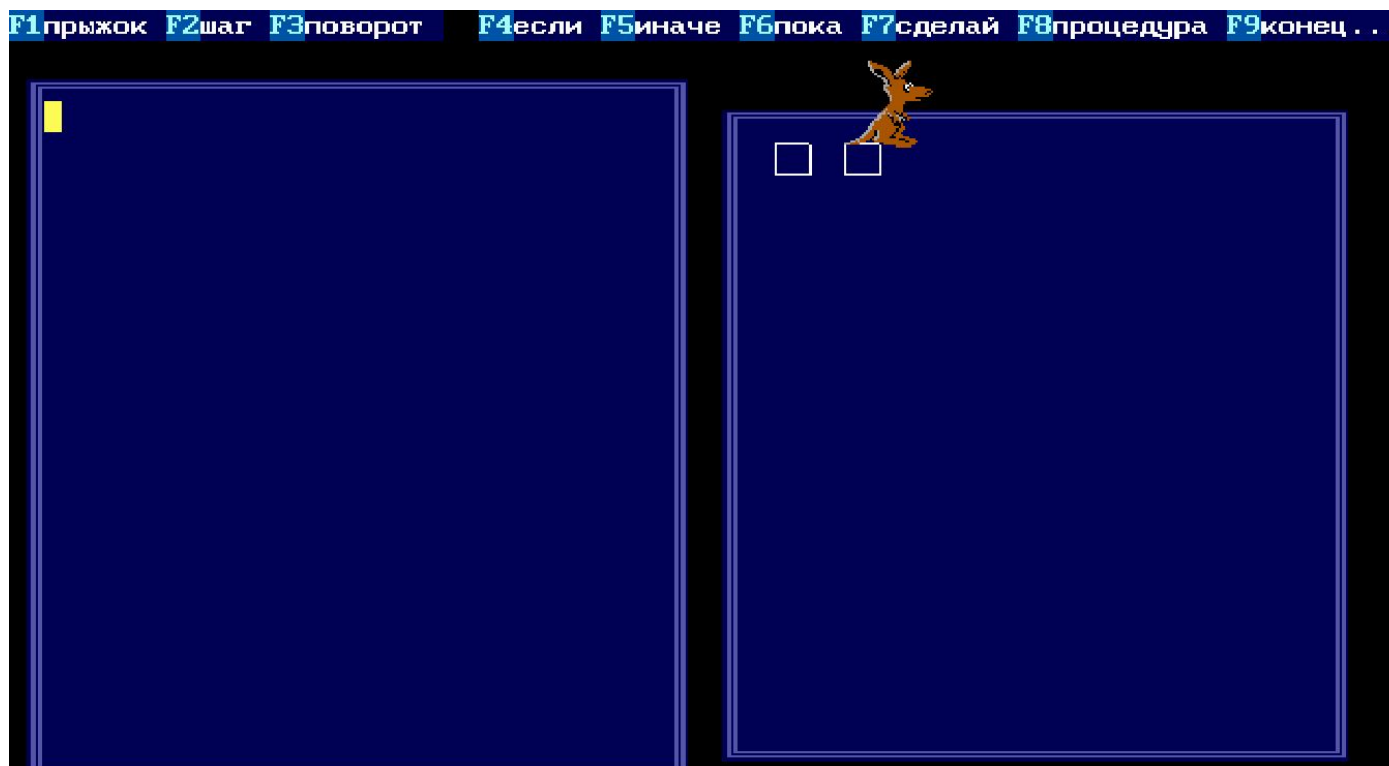
Раздел 2. Знакомство с Pascal

Раздел 3. Основные алгоритмические конструкции

Раздел 4. Структурированные типы данных

Раздел 1. Основы алгоритмизации

Кубики, ступеньки, или Знакомство с исполнителем



Раздел 2. Знакомство с Pascal

Структура программы на языке Pascal

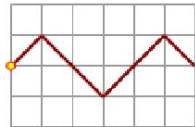
Числовые типы данных

Подпрограммы

Использование графического модуля

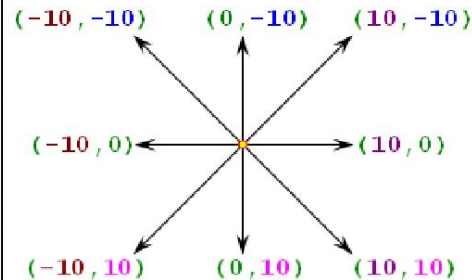
Задание 7

- Создайте процедуру, выводящую на экран «птичку» (размер клетки **10** пикселей, координаты выделенной точки **(x, y)** передаются в процедуру как параметры):



```

Procedure Ptichka(x,y:integer);
Begin
  MoveTo(  ,  );
  LineRel(  ,  );
  LineRel(  ,  );
  LineRel(  ,  );
  LineRel(  ,  );
End;
    
```



Раздел 3. Основные алгоритмические конструкции

Циклы

Условный оператор.

Оператор выбора

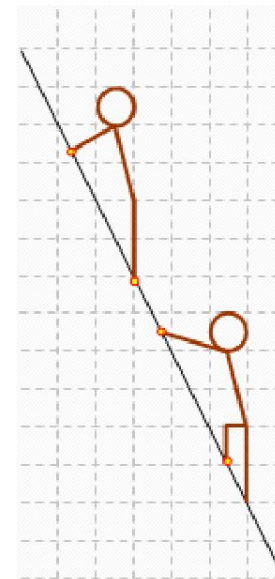
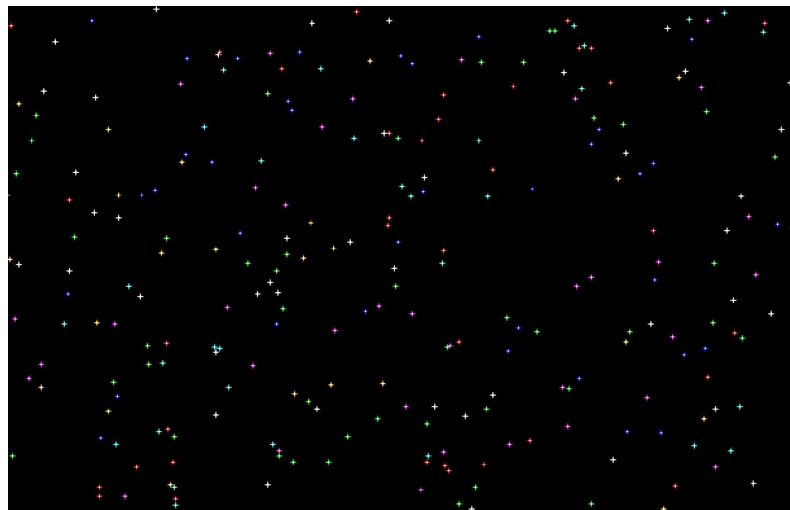
Средства отладки программ

Константы

Компьютерная анимация

Задание 4**

Создайте программу, выводящую на экран лезущего в гору человечка:



Раздел 4. Структурированные типы данных

Массивы

Типизированные константы

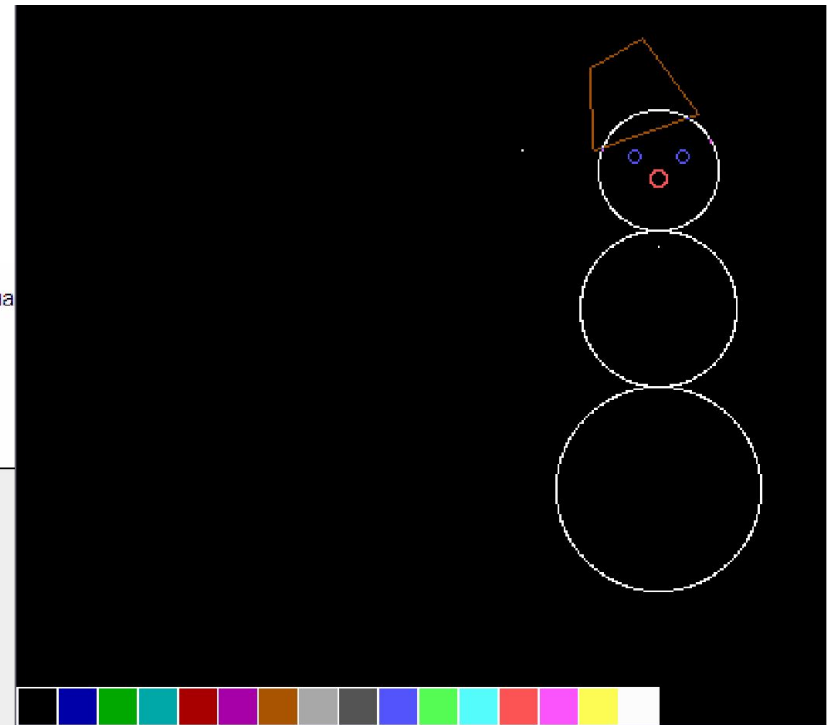
Строковый тип данных

Записи

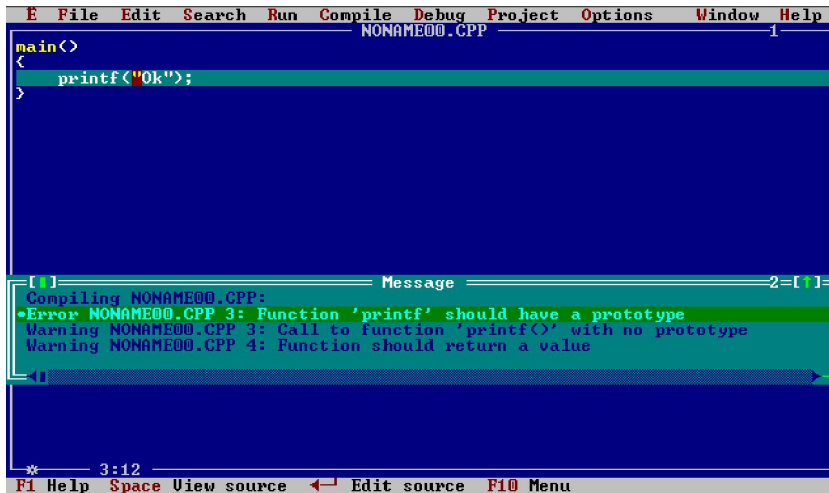
Файловый тип данных

- Добавим массив `buf`, в котором будем сохранять цвет точек в том месте экрана нашего передвигающегося объекта:
 - объявите новый массив того же типа и размера, что и массив-константа `u`:

```
...  
VAR  
  gd, gm: integer;  
  x, y, i, j: integer;  
  c: char;  
  buf: array[1..n, 1..m] of integer;  
...
```



Курс **Программирование на языке Си** содержит исчерпывающие сведения касающиеся теоретических основ программирования, и массу практических заданий.



```
main<>
<
printf("Ok");
>
```

Message

```
Compiling NONAME00.CPP:
Error NONAME00.CPP 3: Function 'printf' should have a prototype
Warning NONAME00.CPP 3: Call to function 'printf()' with no prototype
Warning NONAME00.CPP 4: Function should return a value
```

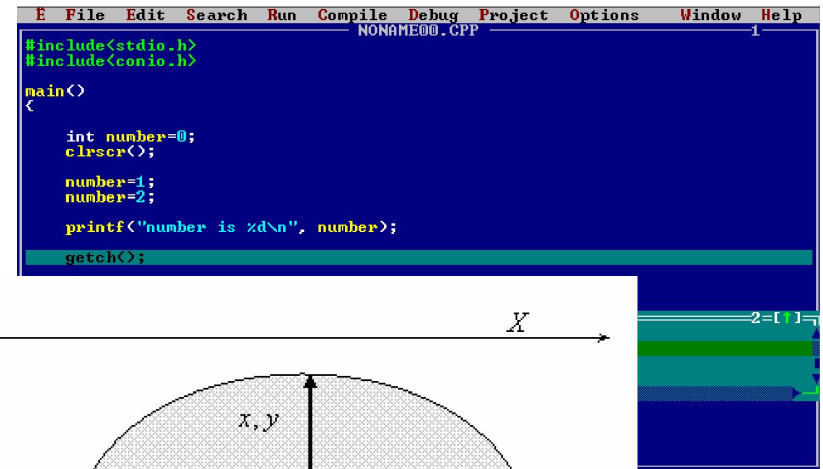
3:12

F1 Help Space View source Edit source F10 Menu

```
1.  /*change of symbols
2.  during a print */
3.
4.  #include<stdio.h>
5.  #include<conio.h>
6.
7.  main()
8.  {      //creating display one
9.         clrscr();          //to clear display
10.        printf("@");      //printing
11.        getch();          //to stop display
12.
13.        //creating display two
14.        clrscr();          //4istka ekrana
15.        printf("%");      //raspe4atka
16.        getch();          //zaderzka ekrana
17.
18.    }
```


Язык Си - первый шаг к серьёзному программированию

- Часть 1. Общие понятия языка Си
- Часть 2. Основы языка Си
- Часть 3. Математика в Си
- Часть 4. Служба помощи и отладка программ
- Часть 5. Графика
- Часть 6. Логические операторы



```
E File Edit Search Run Compile Debug Project Options Window Help
NONAME00.CPP 1
#include<stdio.h>
#include<conio.h>
main()
{
    int number=0;
    clrscr();
    number=1;
    number=2;
    printf("number is %d\n", number);
    getch();
}
```

The diagram shows a Cartesian coordinate system with a horizontal X-axis and a vertical Y-axis. The origin is labeled (0,0). A shaded oval is drawn in the first quadrant. A point (x,y) is marked on the oval. A horizontal line segment from the origin to the point is labeled x, r. A vertical line segment from the origin to the point is labeled x, y.

Компьютерные игры на языке программирования Си

Часть 1. Меню и не только

Часть 2. Циклы и сложные
программы

Часть 3. Мышиная возня

Часть 4. Служба помощи
и отладка программ

Часть 5. Графика

Пример. Напишите программу, подключающую манипулятор мышь.

Программа № 1

```
1.  #include<graphics.h>
2.  #include<dos.h>
3.
4.  union REGS regs;
5.
6.  main()
7.  {
8.      int gdriver = DETECT, gmode;
9.
10.     initgraph( &gdriver, &gmode, "c:/tc/bgi");
11.     /*****/
12.     while( !kbhit() )
13.     {
14.         regs.x.ax=1;
15.         int86(0x33,&regs,&regs);
16.     }
17.     /*****/
18.
19.     closegraph();
20. }
```

Курс **Основы объектно-ориентированного программирования** содержит исчерпывающие сведения касающиеся теоретических основ программирования, и массу практических заданий.

Курс построен так что обучение программированию основывается на написании различных полезных программ.

Курс входит в сертификат «Программист».

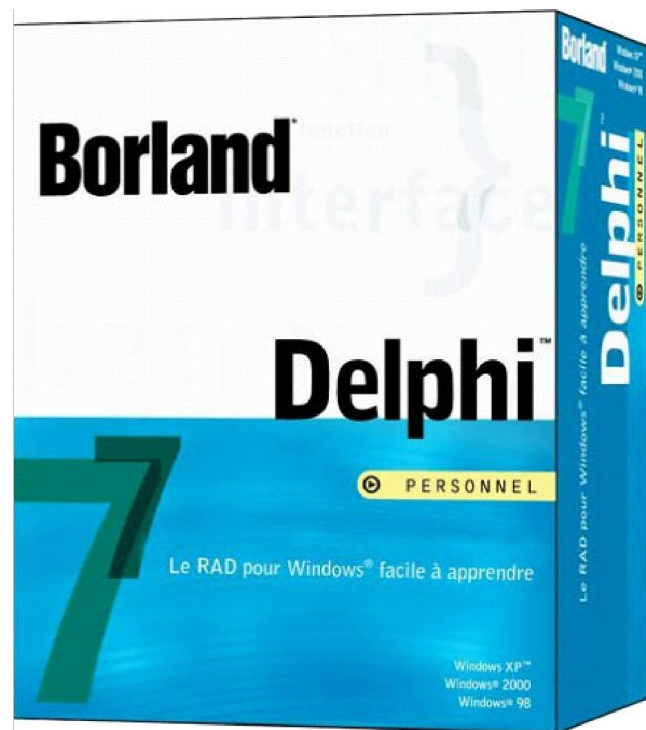
Основными **целями курса** являются:

- сформировать у школьников целостное представление о разработке приложений с помощью объектно-ориентированных подходов в программировании;
- реализовать способности учащихся в ходе создания различных приложений;
- сформировать элементы информационной компетенции по отношению к знаниям, умениям и опыту создания приложений с использованием объектно-ориентированных технологий.

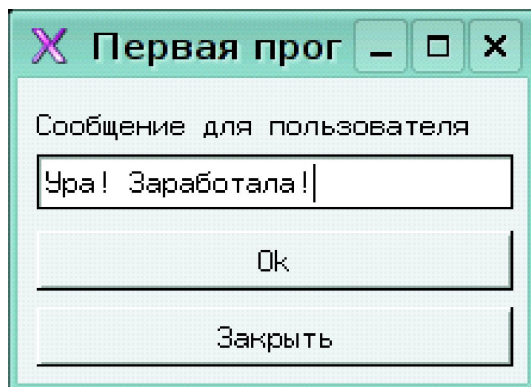
Курс можно изучить в двух различных средах:

В ОС Windows: **Delphi, Lazarus**

В ОС Linux: **Lazarus**



Курс построен так что первые шаги основываются на самых простых заданиях.



Первая прог

Сообщение для пользователя

Ура! Заработала!

Ok

Закреть

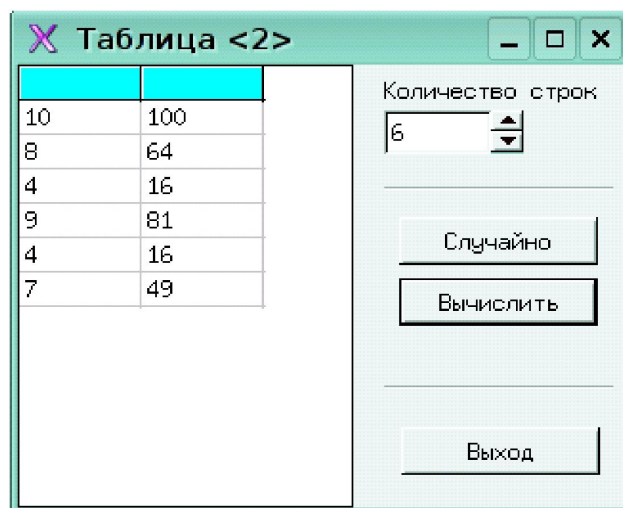


Таблица <2>

10	100
8	64
4	16
9	81
4	16
7	49

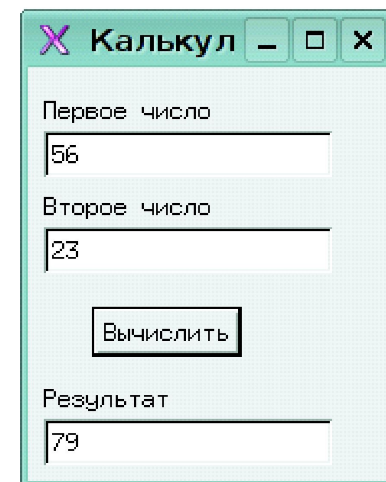
Количество строк

6

Случайно

Вычислить

Выход



Калькул

Первое число

56

Второе число

23

Вычислить

Результат

79

```
procedure TForm1.ButtonMathClick(Sender  
begin  
    editResult.Text := editNum1.Text +  
end;
```


С усложнением создаваемых программ учащийся всё больше и больше знакомится с интерфейсом среды разработки и языком программирования

арифметические действия
 и: Result = A <мат.операция,

```

ТForm1 = class(TForm)
  MemoOut: TMemo;
  Panel1: TPanel;
  ButtonNew: TButton;
    
```

```

unit Unit1;
{$mode objfpc}{$H+}

interface
  OpenFile
  Font
uses
  Classes, SysUtils, FileUtil, LResources, Forms,
  Controls, Graphics, Dialogs,
  ExtCtrls, StdCtrls, Menus, ComCtrls;
type
  { TForm1 }
    
```

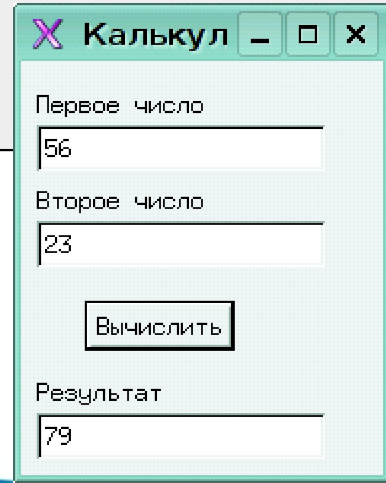
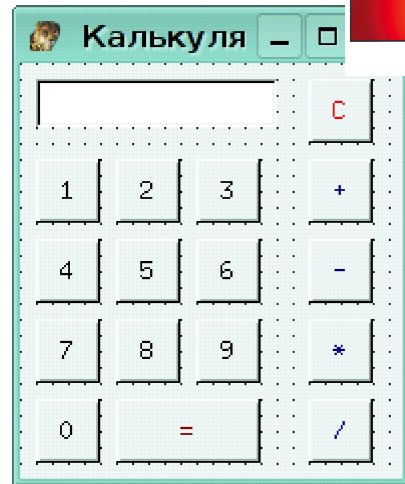
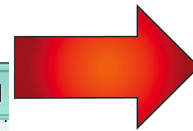
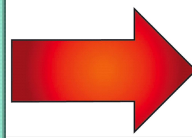
```

editOut.Text := editOut.Text +
editOut.Text := editOut.Text +
    
```

Воспользуемся этим чудесным параметром

```

procedure TForm1.SBtn1Click(S
begin
  editOut.Text := editOut.Te
end;
    
```

Постепенно переходя к созданию весьма
сложных — но очень интересных приложений.

