

# Создание анимации в среде Turbo Pascal на примере анимированной модели стрелочных часов

Исследовательская работа  
Ученицы 10 «Б» класса  
Андрук Алены

Научный руководитель:  
Баскакова Н.В.

Санкт - Петербург

2009 г.

# Цель работы

Целью данной работы является моделирование объекта «часы» в среде программирования Turbo Pascal.

Для реализации главной цели было необходимо освоить вне учебной программы графические операторы и технологии создания движущихся изображений в среде Turbo Pascal, поэтому при выполнении этой работы рассматривались и такие цели, как расширение знаний по курсу программирования на Паскале и совершенствования навыков написания и отладки программ.

# ЭТАПЫ МОДЕЛИРОВАНИЯ

## I этап: Постановка задачи

(под задачей понимается некая проблема, которую надо решить)



1. Описание задачи

2. Цель моделирования

3. Описание объекта или процесса

4. Анализ объекта



## II этап: Разработка модели

5. Разработка информационной модели

6. Разработка математической модели

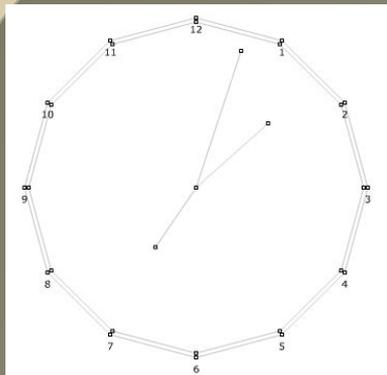
7. Разработка компьютерной модели



## III этап: Компьютерный эксперимент



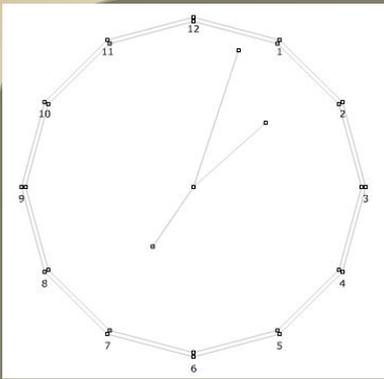
IV этап: Анализ результатов моделирования



# Постановка задачи

## Задача:

Программа «Часы» должна выводить изображение циферблата с движущимися стрелками (часовой, минутной, секундной).



# Выбор входных и выходных данных

## Входные данные:

**R** — радиус циферблата;

**X0, Y0** — координаты центра циферблата;

**d** — размер временной задержки

для демонстрации модели в реальном времени  $d=1000 \text{ ms} = 1 \text{ сек.}$ , для демонстрации модели в ускоренном режиме  $d=10000 \text{ ms}$ ;

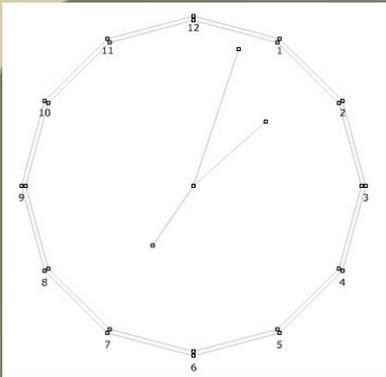
**t** — время показа анимации;

**hour, min, sec** — углы определяющие положение часовой, минутной, секундной стрелок.

**h, m, s** — значения, характеризующие текущий момент времени (количество часов, минут, секунд).

**Выходные данные:** Анимированное изображение.

# Математическая МОДЕЛЬ



## 1. Определение положения стрелок

Положение стрелок на циферблате определяется следующими формулами:

Шаг минутной и секундной стрелок равен  $6^0$ , поэтому:

Угол положения минутной и секундной стрелок определяется по формулам:

$$\text{Min} := 90 - m * 6;$$

$$\text{Sec} := 90 - s * 6;$$

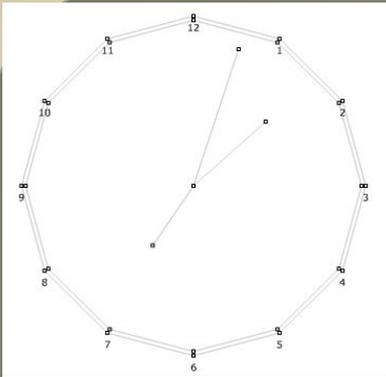
Угол положения часовой стрелки определяется количеством часов и минут:

$$\text{Hour} := 90 - h * 30 - (m \text{ div } 12) * 6,$$

где  $(h * 30)$ -угол определяемый кол-вом часов.

$(m \text{ div } 12) * 6$  – угол между соседними значениями часов

# Математическая МОДЕЛЬ



## 2. Рисование циферблата.

Для создания изображения циферблата необходимо нарисовать:

1. Белую закрашенную окружность радиусом  $R+40$ ;
2. Синюю окружность радиусом  $R$ , вдоль которой будет размещаться шкала делений;
3. Маленькую синюю закрашенную окружность радиусом  $R=5$  в центре циферблата;
4. Линии шкалы делений – 60 делений для отсчета минут (секунд), причем каждый пятый отрезок имеет большую длину и толщину, чем основные деления;
5. Вывести цифры часов напротив каждого пятого деления шкалы.

# Математическая МОДЕЛЬ

Координаты начала и конца каждого отрезка линии шкалы делений определяются по формулам:

$$\begin{aligned}x1 &:= x0 + 0.96 * r * \cos(\text{sec}); \\ y1 &:= y0 - 0.96 * r * \sin(\text{sec}); \\ x &:= x0 + r * \cos(\text{sec}); \\ y &:= y0 - r * \sin(\text{sec});\end{aligned}$$

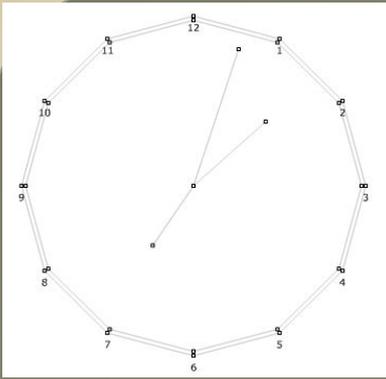
Для каждого пятого деления координаты начала отрезка линии определяются по формулам:

$$\begin{aligned}x1 &:= x0 + 0.93 * r * \cos(\text{sec}); \\ y1 &:= y0 - 0.93 * r * \sin(\text{sec});\end{aligned}$$

Координаты начала поля вывода цифр часов определяются по формулам:

$$\begin{aligned}x2 &:= x0 + 1.1 * r * \cos(\text{sec}) - 11; \\ y2 &:= y0 - 1.1 * r * \sin(\text{sec}) - 14;\end{aligned}$$

# Математическая МОДЕЛЬ



## 3. Рисование стрелок

Длина часовой стрелки:  $L_h := 0,6 * R$

Длина минутной стрелки:  $L_m := 0,8 * R$

Длина секундной стрелки:  $L_s := 0,8 * R$

Координаты концов линий-стрелок  $X$  и  $Y$  определяются координатами центра циферблата  $X_0, Y_0$ , длиной стрелки и углом ее наклона: :

$$X := \text{Round}(X_0 + L * \cos(UGOL * 2 * \text{Pi} / 360));$$

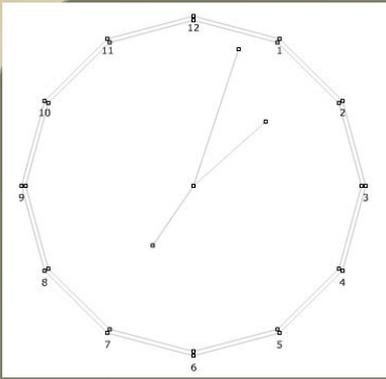
$$Y := \text{Round}(Y_0 - L * \sin(UGOL * 2 * \text{Pi} / 360));$$

Где:

$L$  – длина стрелки

$UGOL$  - угол положения стрелки (подставляются значения углов часовой, минутной или секундной стрелки - hour, min, sec)

# Математическая МОДЕЛЬ

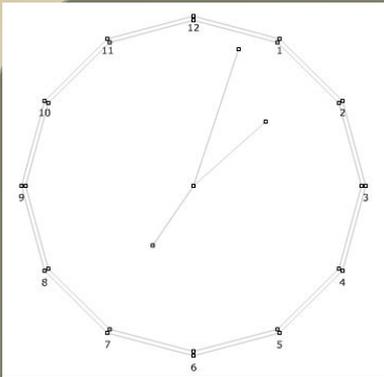


## 4. Реализация анимированного изображения.

1. Сначала рисуется циферблат, который в дальнейшем остается неизменным.
2. Создание эффекта движения в Паскале реализуется следующим образом:
  - В каждый текущий момент(шаг = 1 сек) рисуются черным цветом секундная, минутная и часовая стрелки.
  - После временной задержки, 1000 миллисекунд =1секунде, стрелки рисуются еще раз, но цвет линии белый. Тем самым исходное изображение стирается.
  - Рисуется новое положение стрелок.

# Компьютерный эксперимент

## Программа «Стрелочные часы»



program chasy;

uses graph, crt;

{ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ}

var r,s,ss,m,h,x0,y0,x,y,x1,y1,x2,y2,n,t,tmax,d,driver,mode:integer;

k: string[2];

ls, lm, lh, sec, ssec, min, hour : real;

{r - радиус, x0, y0 – координаты центра, x, y, x1, y1 – координаты отрезков делений шкалы циферблата, x2, y2 – координаты цифр циферблата, s – количество секунд, sec – угол секундной стрелки, m – количество минут, min – угол минутной стрелки, h – количество часов, hour – угол часовой стрелки, ls, lm, lh – длины секундной, минутной и часовой стрелок, n – счетчик цифры циферблата, k - текст цифры циферблата, d – временная задержка, tmax— время показа анимации;

## {ПРОЦЕДУРА РИСОВАНИЯ СТРЕЛОК}

```
Procedure strelka (ugol,dlina:real);  
  Var x,y:integer;  
  
  begin  
    x:=Round(x0+dlina*cos(ugol*pi/180));  
    y:=Round(y0-dlina*sin(ugol*pi/180));  
    Line(x,y,x0,y0);  
  end;
```

## {НАЧАЛО ОСНОВНОЙ ПРОГРАММЫ}

```
begin  
  writeln('vvedite chas: h=');   readln(h);  
  writeln('vvedite minuty: m='); readln(m);  
  writeln('vvedite secundy: ss='); readln(ss);  
  writeln('vvedite vremya zaderzhki (1000) d='); readln(d);  
  writeln('vvedite vremya pokaza (1800,3600 ili 7200) tmax='); readln(tmax);  
  r:=180;  
  x0:=320;  
  y0:=240;  
  driver:=0;  
  mode:=0;  
  initgraph(driver,mode,'c:/bp/bgi');  
  cleardevice;
```

## {РИСОВАНИЕ ЦИФЕРБЛАТА}

```
setbkcolor(blue);  
setcolor(white);
```

### {циферблат – белая закрашенная окружность}

```
setfillstyle(solidfill, white);  
fillellipse(x0,y0,r+40,r+40);
```

### {синяя окружность внутри белой закрашенной окружности}

```
setcolor(blue);  
ellipse(x0,y0,0,360,r,r);
```

### {маленькая синяя закрашенная окружность в центре циферблата}

```
setfillstyle(solidfill,blue);  
fillellipse(x0,y0,5,5);  
sec:=0;n:=0;
```

### {рисование делений шкалы циферблата}

```
for s:=0 to 59 do BEGIN  
    sec:=(90-s*6)*pi/180;  
    x:=Round(x0+r*cos(sec));  
    y:=Round(y0-r*sin(sec));  
    x1:=Round(x0+0.96*r*cos(sec));  
    y1:=Round(y0-0.96*r*sin(sec));  
    setlinestyle(SolidLn,0,1);
```

{рисование каждого пятого деления шкалы циферблата более длинным и с большей толщиной линии}

```
if s mod 5 = 0 then begin x1:=Round(x0+0.93*r*cos(sec));  
    y1:=Round(y0-0.93*r*sin(sec));  
    setlinestyle(SolidLn,0,3);
```

{вывод текста цифр циферблата}

```
x2:=Round(x0+1.1*r*cos(sec)-11);  
y2:=Round(y0-1.1*r*sin(sec)-14);  
if s=0 then k:='12' else begin  
n:=n+1; Str(n:2,k); end; end;
```

```
Settextstyle(1,horizdir,1);
```

```
OuttextXY(x2,y2,k);
```

```
Line(x,y,x1,y1);
```

```
end;
```

{ДВИЖЕНИЕ СТРЕЛОК}

```
ls:=0.85*r; lm:=0.8*r; lh:=0.6*r;
```

```
for t:=0 to tmax do begin
```

```
ssec:=90-ss*6;
```

```
min:=90-m*6;
```

```
hour:=90-h*30-(m div 12)*6;
```

{рисование стрелок}

```
setcolor(blue); setlinestyle(0,0,3);
```

```
strelka(min,lm);
```

```
strelka(hour,lh);
```

```
setlinestyle(0,0,1);
```

```
strelka(ssec,ls);
```

```
{временная задержка}
```

```
    delay(d);  
    setcolor(white);  
    setlinestyle(0,0,3);  
    strelka(min,lm);  
    strelka(hour,lh);  
    setlinestyle(0,0,1);  
    strelka(ssec,ls);  
    setfillstyle(solidfill,blue);  
    fillellipse(x0,y0,5,5);
```

```
{задание нового значения секунд (минут, часов)}
```

```
    ss:=ss+1;  
    if (ss div 60)=1 then begin ss:=0; m:=m+1;end;  
    if (m div 60)=1 then begin m:=0; h:=h+1;end;  
    end;
```

```
{рисование стрелок по завершении показа анимации – чтобы изображение часов на  
экране не пропадало}
```

```
    setcolor(blue);  
    setlinestyle(0,0,3);  
    strelka(min,lm);  
    strelka(hour,lh);  
    setlinestyle(0,0,1);  
    strelka(ssec,ls);  
    readln;  
    end.
```

# Анализ результатов моделирования

Написанная мной программа выводит на экран компьютера стрелочные часы. Но, чтобы данные часы шли точно, показывали реальное время, необходимо произвести их калибровку. На практике временные интервалы при демонстрации анимации зависят от тактовой частоты работы компьютера, поэтому калибровку данной модели на каждом компьютере приходится осуществлять в ручную.

В Delphi существует специальная функция «Timer», считывающая текущее значение времени с таймера компьютера, что дает возможность создания модели часов с точным соответствием времени реальному .

# Список литературы

1. Культин Н.Б Turbo Pascal в задачах и примерах. — СПб.: ВHV, 2006.
2. Культин Н.Б Delphi в задачах и примерах. — СПб.: ВHV, 2005.
3. Макарова Н.В. Информатика. Учебник для 7-9 кл. — СПб.: Питер, 2001.
4. Меженный О.А. Turbo Pascal. — М., СПб., Киев: Диалектика, 2001.
5. <http://graphics.cs.msu.su/courses/cg/library/> — все, что связано с компьютерной графикой, обработкой изображений и мультимедиа