

ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

студенту группы 13534/1 Стойкоски Никола
(номер (фамилия, имя, отчество)
группы)

1. Тема проекта (работы) Разработка объектно-ориентированной графической программы

2. Срок сдачи студентом законченного проекта (работы) 28 декабря

3. Исходные данные к проекту (работе) _____

Лекции по программированию

Пакет индивидуальных заданий с указанием конкретного номера задания

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов): введение, основная часть (раскрывается структура основной части), заключение, список использованных источников, приложения).

Введение с формулировкой конкретного задания

Инструкция программиста(структура программы и особенности ее применения)

Инструкция пользователя(особенности использования программой)

Полный текст программы с комментариями полей данных и методов типов объектов

Фрагмент графической картинки

Примерный объем пояснительной записки 8 страниц машинописного текста

5. Перечень графического материала (с указанием обязательных чертежей и плакатов) нет

6. Консультанты нет

7. Дата получения задания: « 10 » _____ сентября _____ 2016__ г.

Руководитель _____ Веренинов И.А
(подпись) (инициалы, фамилия)

Задание принял к исполнению _____
(подпись) (инициалы, фамилия)

(дата)

Минобрнауки России
Санкт-Петербургский государственный политехнический университет
Институт компьютерных наук и технологий
Кафедра «Информационные и управляющие системы»

**Разработка объектно-ориентированной графической
программы**
по дисциплине «Алгоритмизация и основы программирования»

Выполнил
студент гр.13534/1

Стойкоски Никола

Руководитель
доцент, к.т.н.

И.А.Веренинов

« 15 » декабря 2016 г.

Санкт-Петербург
2016

Содержание

Оглавление

1. Задание	4
2. Инструкция программиста	4
3. Инструкция оператора	4
4. Текст программы	4
5. Приложение (скриншоты)	12

1.Задание

Движение паровоза с регулируемой скоростью (от 0 до $V_{\text{кон}}$). Исходное состояние кривошипного механизма – любое. Цвета деталей и пропорции частей по усмотрению разработчика. Линейное и вращательное движения колес различного диаметра должны быть согласованы.

2.Инструкция программиста

Программа работает в среде *Turbo Pascal 7.0*. Для корректной работы требуются модули:

EGA_VGA.BGI – должно находиться в текущей папке

Graph - обеспечивает управление графическим режимом в Turbo Pascal 7.0

Crt – позволяет работать с окном программы

P11_DEP – описание объектов (колесо, паравоз, дым, дерево) и все необходимые процедуры.

3.Инструкция оператора

После запуска программы, будет показан паравоз для которого можно контролировать скорость.

С помощью клавиши со стрелкой вверх можно увеличить скорость.

С помощью клавиши со стрелкой вниз можно уменьшить скорость.

Для выхода из программы можно использовать CTRL-C.

4.Текст программы

```
program p11;
uses P11_DEP, graph, crt;
```

Type

```
GameController = Object    {golovnoi obyekt}
  gd, gm, pg: integer;      {dlya graficeskii rezim}
  n_smoke, target, distance: integer; {tekushee kolicestvo dim, mesto gde vznikaet}
  _train: Train;            {paravoz}
  smokes: array[1..MAX_SMOKE] of Smoke; {dim}
  trees : array[1..5] of Tree; {derevyia}
  procedure handleUserInput(var vx:real); {uvelicit ili umensit skorost, po trebovaniu polzovatel}
  procedure run;             {nacalo igra}
  procedure initObjects(vx:real;tx:integer); {inicializatsiya obektov}
  procedure initDisplay; {inicializatsiya monitor}
  procedure spawnSmoke; {novyy puzirek dima}
  procedure drawBackground; {risuet fon}
  procedure drawShini; {risuet shini}
  procedure StartFrame; {menyaet activePage, I cistit ekran}
  procedure EndFrame; {menyaet visualPage, I delay}
end;
procedure gameController.run;
```

```

var i: integer;
begin
  while true do
    begin
      StartFrame;
      handleUserInput(_train.vx);

      _train.getParams;
      if _train.checkRange = false then
        initObjects(_train.vx, -200);

        spawnSmoke;
        drawBackground;
        _train.putFrame;
        EndFrame;
      end;
    end;
  procedure gameController.drawBackground;
  var i: integer;
  begin
    for i:=1 to n_smoke do
      begin
        smokes[i].getParams;
        smokes[i].putFrame;
      end;
    drawShini;
    for i:=1 to 5 do
      trees[i].putFrame;
    end;
  procedure gameController.drawShini;
  var i: integer;
  begin
    SetFillStyle(1, green);
    SetColor(8);
    bar(0,GROUND_Y-40,640,350);
    line(0,GROUND_Y,640,GROUND_Y);
    line(0,GROUND_Y+2,640,GROUND_Y+2);
    SetFillStyle(1, brown);
    i := 5;
    while i < 640 do
      begin
        bar(i,GROUND_Y+1,i+5, GROUND_Y+5);
        i := i+ 20;
      end;
    end;
  procedure gameController.spawnSmoke;
  begin
    distance := distance + trunc(_train.vx/dt);
    if (distance >= target) and (n_smoke < MAX_SMOKE) then
      begin
        target := target + S_TICK;
        n_smoke := n_smoke+1;
      end;
    end;
  end;
end;

```

```

    smokes[n_smoke].init(_train.wheels[1].x, _train.vx);
end;
end;
procedure gameController.initObjects(vx:real;tx:integer);
var i: integer;
begin
    n_smoke := 0;
    _train.vx := vx;
    _train.init(tx);
    trees[1].Init(20+random(100), 20+random(20));
    for i:=2 to 5 do
        trees[i].Init(trees[i-1].x+30+random(200), 20+random(20));
    if tx = 50 then begin
        distance := 50;
        target := 51;
    end;
end;
end;
procedure gameController.initDisplay;
begin
    initgraph(gd, gm, "");
    gm := 1;
    setGraphMode(gm);
    pg := 0;
    setbkcolor(cyan);
end;
procedure gameController.startFrame;
begin
    setActivePage(pg);
    cleardevice;
end;
procedure gameController.endFrame;
begin
    setVisualPage(pg);
    delay(1000 div fps);
    pg := 1-pg;
end;
procedure gameController.handleUserInput(var vx: real);
var key: char;
begin
    if keypressed then
        begin
            key := readkey;
            Case key of
                #72: vx := vx+0.03;
                #80: vx := vx-0.1;
            end;
            if vx < 0 then
                vx := 0;
            if vx > V_MAX then
                vx := V_MAX;
            end;
        end;
end;
end;

```

```

var
game: gameController;

begin
  game.initDisplay;
  game.initObjects(0, 50);
  game.run;
end.

```

Модуль P11 DEP:

```
unit P11_DEP;
```

```
INTERFACE
```

```
uses graph;
```

```
const
```

```

V_MAX = 5; MAX_SMOKE = 50; {maskimalnaya skorost, max kolicestvo dim}
GROUND_Y = 300; TREE_Y = 230; {viota gde nahoditsya zemlya, derevya}
SMOKE_VY = 0.3; S_TICK = 30; {skorost dima, novii puzirek na kazdiy X pikselov}

```

```

CON_LEN = 35; BOX_LEN = 15; {dlina korobka v kolesniy mehanizam}
BOXH_D2 = 5; CSHAFT_LEN = 60; {visota korobka v kolesniy mehanizam}
EPS = 0.00001; PI_2 = 6.28; {epsilon, 2*pi}

```

```

BR = 20; SR = 10; {radius dlya bolshoe I malenkoe koleso}
WR = 14; FPS = 20; {vnutrenii radius okruznost koleso, kadrov v sekunde}
C = 3; dt = 0.5; {konstanta dlya uglovaya skorost, vreme kotoroe prohodit kazii cikl}

```

```
Type
```

```
GameObject = Object {standartnoe opisanie igrovie ob'ekti}
```

```

  x,y : integer; {x,y poziciya}
  xr,yr,vx: real; {x,y poziciya (real), i skorost}
  procedure getParams; {viichislit sledushii x,y,vx..}
  procedure putFrame; {postavit ob'ekt v kadre}
  procedure move;virtual; {peremestit ob'ekt/ menyaet x,y,vx...}
  procedure draw;virtual; {risuyet ob'ekt}
  Constructor init; {inicializaciya ob'ekta}

```

```
end;
```

```
Wheel = Object(GameObject) {koleso}
```

```

  r: integer; {radius}
  a,da,w: real; {vedushii ugol, peremeshenie ugla, uglovaya skorost}
  procedure getXY(var x1,y1: integer); {zagonyaet x->x1, y->y1}
  procedure incAngle; {uvelicit vedushii ugol}
  procedure draw;virtual; {risuyet koleso}
  procedure move;virtual; {peremeshaet koleso/ meyaet X poziciya}
  Constructor Init(x1,y1,r1:integer; a1:real); {inicializaciya}

```

```
end;
```

```
Train = Object(GameObject) {paravoz}
```

```

  wheels: array[1..3] of wheel; {kolesa}
  procedure drawConnector; {risuet mehanizm}

```

```

procedure drawBody; {risuet telo paravoza}
procedure draw;virtual; {risuet paravoz}
procedure move;virtual; {peremeshaet paravoz po X poziciya}
function checkRange:boolean;virtual; {proveryaet esli paravoz v granici ekrana}
Constructor Init(sx: integer);{inicializaciya}
end;
Tree = Object(GameObject){derevo}
  r: integer;{radius listya}
  procedure draw;virtual; {risyet derevo}
  Constructor Init(x1,r1: integer);{inicializaciya}
end;
Smoke = Object(GameObject){dim}
  r: integer;{radius puzirek}
  procedure move;virtual;{peremeshaet po X,Y poziciya}
  procedure draw;virtual;{risuet puzirek}
  Constructor Init(x1:integer;vx1:real);{inicializaciya}
end;

```

IMPLEMENTATION

```

procedure GameObject.getParams;
begin
  move;
end;
procedure GameObject.putFrame;
begin
  draw;
end;
procedure GameObject.move;begin end;
procedure GameObject.draw;begin end;
Constructor GameObject.init;begin end;
Constructor Smoke.Init(x1:integer;vx1:real);
begin
  x := x1+100;
  y := 190;
  yr := y;
  r := 3 + trunc(2.5*vx1);
end;
procedure Smoke.move;
begin
  yr := yr - (SMOKE_VY*3/dt);
  y := trunc(yr);
  x := x - random(2) + random(2);
  if (random(8) = 4) then
    r := r + random(2);
end;
procedure Smoke.draw;
begin
  SetColor(white);
  SetFillStyle(1, white);
  circle(x,y,r);
  floodfill(x,y,white);
end;

```



```

Constructor Tree.Init(x1,r1: integer);
begin
  x := x1; r := r1;
end;
procedure Tree.draw;
begin
  setcolor(green);
  setFillStyle(1,green);
  circle(x, TREE_Y, r);
  floodfill(x, TREE_Y, green);
  setColor(brown);
  SetLineStyle(0,0,ThickWidth);
  line(x,TREE_Y,x,TREE_Y+40);
  line(x-10,TREE_Y+5,x,TREE_Y+20);
  line(x+10,TREE_Y+5,x,TREE_Y+20);
  SetLineStyle(0,0,0);
end;
constructor Wheel.Init(x1,y1,r1:integer;a1:real);
begin
  x := x1; y := y1;
  r := r1; a := a1;
  xr := x; yr := y;
end;
procedure Wheel.getXY(var x1,y1:integer);
begin
  x1 := x; y1 := y;
end;
procedure Wheel.incAngle;
begin
  a := a + da;
  if a > PI_2 then
    a := a - PI_2;
end;
procedure Wheel.move;
begin
  w := (vx+c)/(r/(vx+EPS));
  da := w*dt - w/(r/2);
  xr := xr + vx/dt;
  x := trunc(xr);
  incAngle;
end;
procedure Wheel.draw;
var i: integer;ang:real;
begin
  setColor(red);
  circle(x,y,r+4);
  circle(x,y,r+3);
  da := PI_2/8;
  for i:=1 to 8 do
  begin
    line(x, y, trunc(x+r*cos(a)), trunc(y+r*sin(a)));
    incAngle;
  end;
end;

```

```

    end;
end;
procedure Train.move;
var i: integer;
begin
    for i:=1 to 3 do
        begin
            wheels[i].vx := vx;
            wheels[i].getParams;
        end;
        x := wheels[1].x-2*BR;
    end;
procedure Train.draw;
var i: integer;
begin
    drawBody;
    for i := 1 to 3 do
        with wheels[i] do
            putFrame;
        end;
    drawConnector;
end;
function Train.checkRange:boolean;
begin
    checkRange := true;
    if wheels[1].x > 700 then
        checkRange := false;
    end;
end;
Constructor Train.init(sx:integer);
begin
    wheels[1].Init(sx ,GROUND_Y-BR,BR,1);
    wheels[2].Init(sx+50 ,GROUND_Y-BR,BR,1);
    wheels[3].Init(sx+130,GROUND_Y-SR,SR,1);
    y := GROUND_Y - 5*BR - 5;
end;

procedure Train.drawConnector;
var x1,y1,x2,y2,p1x,p1y,p2x,p2y: integer;a1:real;
begin
    a1 := wheels[1].a;
    wheels[1].getXY(x1,y1);
    wheels[2].getXY(x2,y2);
    p1x := trunc(x1 + WR*cos(a1));
    p1y := trunc(y1 + WR*sin(a1));
    p2x := trunc(x2 + WR*cos(a1));
    p2y := trunc(y2 + WR*sin(a1));
    {details}
    setColor(white);
    circle(p1x,p1y, 2);
    circle(p2x,p2y, 2);
    circle(p2x+CON_LEN+BOX_LEN div 2,y2, 4);
    {fixed shaft}
    rectangle(x2+30,y2-BOXH_d2-1,x2+CSHAFT_LEN,y2+BOXH_d2+1);

```

```

{mechanism}
rectangle(p2x+CON_LEN,y2-BOXH_d2, p2x+CON_LEN+BOX_LEN, y2+BOXH_d2);
line(p2x,p2y,p2x+CON_LEN+BOX_LEN div 2, y2);
{between wheels}
line(p1x,p1y,p2x,p2y);
end;
procedure Train.drawBody;
var i: integer;
begin
  setFillStyle(9,1);
  {kabina}
  bar(x+10, y+5, x+2*BR, y+3*BR);
  {behind wheels}
  bar(x+10, y+3*BR, x+15, y+5*BR-5);
  {behind/around wheels}
  bar(x+10, y+3*BR, x+190, y+4*BR);
  for i:=1 to 2 do
    with wheels[i] do
      bar(x-r+8, y+5, x+r-8, y-r+5);
  for i:=1 to 2 do
    with wheels[i] do
      bar(x+r+3, y-r+10, x+r+8, y+ 5);
  with wheels[2] do
    bar(x+CSHAFT_LEN, y-BOXH_d2, x+CSHAFT_LEN+35, y+BOXH_d2);
  {fuel tank}
  SetFillStyle(1, red);
  SetColor(red);
  bar(x+2*BR, y+30, x+170, y+3*BR);
  line(x+170,y+30,x+170,y+3*BR);
  line(x+170,y+3*BR, x+180,y+3*BR);
  line(x+170,y+30,x+180,y+40);
  line(x+180,y+40,x+180,y+3*BR);
  floodfill(x+171,y+40,red);
  {roof}
  bar(x, y, x+2*BR, y+5);
  {crank box}
  with wheels[2] do
    bar(x+30,y-BOXH_d2-1,x+CSHAFT_LEN,y+BOXH_d2+1);
  {odzaci}
  setFillStyle(1, 1);
  bar(x+130,y,x+150,y+30);
  bar(x+80,y+10,x+100,y+30);
  {window}
  bar(x+20,y+15,x+2*BR-10,y+2*BR);
end;
begin
end.

```

5. Приложение (скриншоты)

