**Индивидуальные задания на программирование**

***Цель работы:*** *изучение структуры, свойств и видов объектов; изучение способов доступа к полям и правил вызова методов объектов; получение навыков объектно-ориентированного программирования на языке C++. Освоение процесса инсталлирования среды программирования, необходимой для решения индивидуального задания; Освоение процесса проведения занятий по обучению применения программно-методических комплексов.*

***Задание на программирование:*** *используя технологию объектно-ориентированного программирования разработать два варианта программы, реализующей движущийся графический объект в соответствии с индивидуальным заданием:*

*- с использованием статического объекта;*

*- с использованием динамического объекта.*

***Порядок выполнения работы:***

1) Разработать иерархию и структуру объектов, связанных на принципах наследования, в соответствии с индивидуальным заданием. Дерево наследования должно содержать не менее трех уровней.

2) Описать типы объектов и методы обработки их полей.

3) Составить две программы на языке C++, реализующие движение графического объекта по заданной траектории: в виде динамического объекта и в виде статического объекта описанного типа.

4) Оформить отчет о работе в составе: постановка задачи, математическая модель, спецификации функций, текст программы, контрольные примеры.

***Индивидуальное задание***

Движение закрашенного прямоугольника по треугольному контуру с изменением цвета при изменении направления движения.

***Примеры программ***

**//Движение прямоугольника по треугольному контуру.**

**//Динамические объекты.**

#include <stdio.h>

#include <graphics.h>

#include <iostream.h>

#include <process.h>

#include <string.h>

#include <dos.h>

class graphworld

 {int driver, mode, grerror, colb, bkcl ;

 char path[80] ;

 public:

 graphworld() ;

 void closegraphworld() ;

 } ;

graphworld::graphworld()

 {strcpy(path, "c:\\turboc30\\bgi") ;

 driver = 0 ;

 initgraph(&driver, &mode, path) ;

 grerror = graphresult() ;

 if(grerror != grOk)

 {cout<<"\nОшибка открытия графического режима" ;

 abort() ;

 }

 setcolor(RED) ;

 setbkcolor(BLACK) ;

 cleardevice() ;

 }

void graphworld::closegraphworld()

 {cleardevice() ;

 closegraph() ;

 }

class location

 {protected:

 int x, y ;

 public:

 location(int initx, int inity) ;

 int getx() ;

 int gety() ;

 } ;

location::location(int initx, int inity)

 {x = initx ;

 y = inity ;

 }

int location::getx()

 {return x ;

 }

int location::gety()

 {return y ;

 }

class point:public location

 {protected:

 int visible ;

 void setvisible(int pr) ;

 public:

 point(int initx, int inity) ;

 ~point() ;

 virtual void show() ;

 virtual void hide() ;

 int getvisible() ;

 void moveto(int nx, int ny) ;

 } ;

point::point(int initx, int inity):location(initx, inity)

 {

 }

point::~point()

 {hide() ;

 }

void point::moveto(int nx, int ny)

 {hide() ;

 x=nx ;

 y=ny ;

 show() ;

 }

void point::setvisible(int pr)

 {visible = pr ;

 }

void point::show()

 {putpixel(x, y, getcolor()) ;

 setvisible(1) ;

 }

void point::hide()

 {putpixel(x, y, getbkcolor()) ;

 setvisible(0) ;

 }

class pramoug:public point

 {int dx, dy ;

 public:

 pramoug(int initx, int inity, int initdx, int initdy) ;

 ~pramoug() ;

 void show() ;

 void hide() ;

 } ;

pramoug::pramoug(int initx, int inity, int initdx, int initdy):

 point(initx, inity)

 {dx = initdx ;

 dy = initdy ;

 }

void pramoug::show()

 {line(x, y, x, y+dy) ;

 line(x, y + dy, x + dx, y + dy) ;

 line(x + dx, y + dy, x + dx, y) ;

 line(x, y, x + dx, y) ;

 }

void pramoug::hide()

 {int r ;

 r = getcolor() ;

 setcolor(getbkcolor()) ;

 line(x, y, x, y + dy) ;

 line(x, y + dy, x + dx, y + dy) ;

 line(x + dx, y + dy, x + dx, y) ;

 line(x, y, x + dx, y) ;

 setcolor(r) ;

 }

pramoug::~pramoug()

 {hide() ;

 }

int main()

{graphworld world ;

 pramoug \*pt ;

 int x = 150, y = 100 ;

 printf("Press ENTER, please...") ;

 getchar() ;

 cleardevice() ;

 pt = new pramoug(x, y, 200, 100) ;

 delay(750) ;

 pt -> show() ;

 for(int i = 3; i > 0 ; i--)

 {

 {do

 {x += 3 ; y++ ;

 pt -> moveto(x, y) ;

 delay(5) ;

 }

 while(!(y >= 200)) ;

 do

 {x--; y++;

 pt -> moveto(x, y) ;

 delay(5) ;

 }

 while(!(y >= 400)) ;

 do

 {x--; y -= 3 ;

 pt -> moveto(x, y) ;

 delay(5) ;

 }

 while(!(y <= 100)) ;

 }

 }

 delete pt ;

 world.closegraphworld() ;

 return 0 ;

}

**//Движение прямоугольника по треугольному контуру.**

**//Статические объекты.**

#include <stdio.h>

#include <graphics.h>

#include <iostream.h>

#include <process.h>

#include <string.h>

#include <dos.h>

class graphworld

 {int driver, mode, grerror, colb, bkcl ;

 char path[80] ;

 public:

 graphworld() ;

 void closegraphworld() ;

 };

graphworld::graphworld()

 {strcpy(path, "c:\\turboc30\\bgi") ;

 driver = 0 ;

 initgraph(&driver, &mode, path) ;

 grerror = graphresult() ;

 if(grerror != grOk)

 {cout<<"\nОшибка открытия графического режима" ;

 abort() ;

 }

 setcolor(RED) ;

 setbkcolor(BLACK) ;

 cleardevice() ;

 }

void graphworld::closegraphworld()

 {cleardevice() ;

 closegraph() ;

 }

class location

 {protected:

 int x, y ;

 public:

 location(int initx, int inity) ;

 int getx() ;

 int gety() ;

 } ;

location::location(int initx, int inity)

 {x = initx ;

 y = inity ;

 }

int location::getx()

 {return x ;

 }

int location::gety()

 {return y ;

 }

class point:public location

 {protected:

 int visible ;

 void setvisible(int pr) ;

 public:

 point(int initx, int inity) ;

 ~point() ;

 virtual void show() ;

 virtual void hide() ;

 int getvisible() ;

 void moveto(int nx, int ny) ;

 } ;

point::point(int initx, int inity):location(initx, inity)

 {

 }

point::~point()

 {hide() ;

 }

void point::moveto(int nx, int ny)

 {hide() ;

 x = nx ;

 y = ny ;

 show() ;

 }

void point::setvisible(int pr)

 {visible = pr ;

 }

void point::show()

 {putpixel(x, y, getcolor()) ;

 setvisible(1) ;

 }

void point::hide()

 {putpixel(x, y, getbkcolor()) ;

 setvisible(0) ;

 }

class pramoug:public point

 {int dx, dy ;

 public:

 pramoug(int initx, int inity, int initdx, int initdy) ;

 ~pramoug() ;

 void show() ;

 void hide() ;

 } ;

pramoug::pramoug(int initx, int inity, int initdx, int initdy):

 point(initx, inity)

 {dx = initdx ;

 dy = initdy ;

 }

void pramoug::show()

 {line(x, y, x, y + dy) ;

 line(x, y + dy, x + dx, y + dy) ;

 line(x + dx, y + dy, x + dx, y) ;

 line(x, y, x + dx, y) ;

 }

void pramoug::hide()

 {int r ;

 r = getcolor() ;

 setcolor(getbkcolor()) ;

 line(x, y, x, y + dy) ;

 line(x, y + dy, x + dx, y + dy) ;

 line(x + dx, y + dy, x + dx, y) ;

 line(x, y, x + dx, y) ;

 setcolor(r) ;

 }

pramoug::~pramoug()

 {hide() ;

 }

int main()

{graphworld world ;

 int x = 150 ,

 y = 100 ;

 printf("Press ENTER, please...") ;

 getchar() ;

 cleardevice() ;

 pramoug pt(x, y, 200, 100) ;

 delay(750) ;

 pt.show() ;

 for(int i = 3 ; i > 0 ; i--)

 {

 {do

 {x += 3 ; y++ ;

 pt.moveto(x, y) ;

 delay(5) ;

 }

 while(!(y >= 200)) ;

 do

 {x-- ; y++ ;

 pt.moveto(x, y) ;

 delay(5) ;

 }

 while(!(y >= 400)) ;

 do

 {x-- ; y -= 3 ;

 pt.moveto(x, y) ;

 delay(5) ;

 }

 while(!(y <= 100)) ;

 }

 }

 world.closegraphworld() ;

 return 0 ;

}