**Hello world**

**Часть 1. Простые объекты**

Описать объект, включающий заданные поля и методы. Написать программу, которая создает объект и тестирует его методы.

Объект – окружность. Параметры: координаты центра, радиус. Методы: процедура инициализации, процедура вывода на экран значений параметров и функция, определяющая и возвращающая по запросу длину окружности.

Текст программы

program project1;

{$APPTYPE CONSOLE}

uses

SysUtils;

type okr=object

|  |
| --- |
| okr |
| cx,cy |
| init(a,b,rad:real), print(c:real),dlina:real |

cx,cy,r:real;

procedure init(a,b,rad:real);

procedure print(C:real);

function dlina:real;

end;

procedure okr.init(a,b,rad:real);

begin

cx:=a;

cy:=b;

r:=rad;

end;

procedure okr.print(c:real);

begin

if c=0 then

writeln('X coordinate of center: ',cx:4:6,' Y coordinate of center: ',cy:4:6,' Radius: ',r:4:6)

else writeln('Length of circle: ',okr.dlina:4:6);

end;

function okr.dlina:real;

begin

Result:=2\*3.14\*r;

end;

var a:okr;

q,b,c,rad:real;

begin

readln(q,b,rad);

a.init(q,b,rad);

writeln('If you want to know parametrs of circle press 0');

witeln('If you whant to know length of circle press any number');

readln(c);

a.print(c);

readln(rad);

end.

|  |  |  |  |
| --- | --- | --- | --- |
| Входные данные | 2 2 3 0 | 2 2 3 5 | 2 2 0 1 |
| Ожидаемый результат | X coordinate of center: 2.00000  Y coordinate of center: 2.00000  Radius: 3.00000 | Length of circle: 18.840000 | Length of circle: 0.000000 |
| Полученный результат | X coordinate of center: 2.00000  Y coordinate of center: 2.00000  Radius: 3.00000 | Length of circle: 18.840000 | Length of circle:  0.000000 |

|  |
| --- |
| line |
| xbeg,ybeg,k |
| init(a,b,c:real), print(a,b,c:real), prohod(xnew,ynew:real):real |

Часть 2. Наследование

Разработать и реализовать диаграмму классов для опсианных объектов предметной области,

используя механизмы композиции.

Проверить ее на тестовом примере с демонстрацией всех возможносткй разработанных классов на конкретных данных.

Объект – линия, заданная координатами точки и углом. Умеет инициализировать свои поля, выводить свои параметры на экран и определять, проходит ли линия через точку с заданными координатами .

Объект – линия, заданная координатами точки, углом и толщиной. Умеет инициали-зировать свои поля, выводить свои параметры на экран и определять, проходит ли линия через третью точку с заданными координатами с учетом толщины линии.

program dz32;

{$APPTYPE CONSOLE}

|  |
| --- |
| fatline |
| fat |
| newinit(a,b,c,d:real), newprint(a,b,c:real), prohod(xnew,ynew:real):real |

Uses

Sysutils,Math;

type line=Object

xbeg,ybeg,k:real;

procedure init(a,b,c:real);

procedure print(a,b,c:real);

function prohod(xnew,ynew:real):real;

end;

procedure line.init(a,b,c:real);

begin

xbeg:=a;

ybeg:=b;

k:=c;

end;

procedure line.print(a,b,c:real);

begin

write('Parameters of line: ',a:3:6,' ',b:3:6,' ',c:3:6,'\*pi ');

end;

function line.prohod(xnew,ynew:real):real;

begin

if (ynew-ybeg)-(xnew-xbeg)\*tan(k\*pi)\*(xbeg+1)=0 then

result:=1

else result:=0;

end;

type fatline=object(line)

fat:real;

procedure newinit(a,b,c,d:real);

procedure newprint(a,b,c:real);

function newprohod(xnew,ynew:real):real;

end;

procedure fatline.newinit(a,b,c,d:real);

begin

line.init(a,b,c);

fat:=d;

end;

procedure fatline.newprint(a,b,c:real);

begin

line.print(a,b,c);

write(' ',fat:3:6,' ');

end;

function fatline.newprohod(xnew,ynew:real):real;

begin

if line.prohod(xnew,ynew)<>1 then begin

if abs((ybeg-ynew)-(xbeg-xnew)\*tan(k\*pi)\*(xbeg+1))/sqrt(sqr(tan(k\*pi)\*(xbeg+1)))<=fat/2 then

result:=1

else result:=0;

end

else result:=1;

end;

var first:line;

second:fatline;

a,b,c,d,e,f:real;

begin

read(a,b,c); first.init(a,b,c); first.print(a,b,c);

writeln;

readln(e,f);

if first.prohod(e,f)=1 then writeln('Yes') else

writeln('No');

read(a,b,c,d,e,f);

writeln;

second.newinit(a,b,c,d);

second.newprint(a,b,c);

if second.newprohod(e,f)=1 then writeln('Yes')

else writeln('No');

readln(a);

end.

|  |  |  |  |
| --- | --- | --- | --- |
| Входные данные | 3 3 0  4 3  3 2 0.25 2 4 3.15 | 2 2 0.25  3 3  2 2 0.25 2 3 3 | 3 4 1  3 5  3 3 1 2 3 5 |
| Ожидаемый результат | Parameters of line: 3.000000 3.000000 0.000000\*pi  Yes  Parameters of line: 3.000000 2.000000 0.250000\*pi 2.000000  Yes | Parameters of line: 2.000000 2.000000 0.250000  Yes  Parameters of line  2.000000 2.000000 0.250000\*pi 2.000000  Yes | Parameters of line: 3.000000 4.000000 1.000000\*pi  Yes  Parameters of line: 3.000000 3.000000 1.000000\*pi 2.000000  Yes |
| Полученный результат | Parameters of line: 3.000000 3.000000 0.000000\*pi  Yes  Parameters of line: 3.000000 2.000000 0.250000\*pi 2.000000  Yes | Parameters of line: 2.000000 2.000000 0.250000  No(погрешность представления тангенса)  Parameters of line  2.000000 2.000000 0.250000\*pi 2.000000  Yes | Parameters of line: 3.000000 4.000000 1.000000\*pi  No(погрешность представления тангенса)  Parameters of line: 3.000000 3.000000 1.000000\*pi 2.000000  Yes |

Часть 3. Композиция.

Разработать и реализовать диаграмму классов для описанных объектов предметной области, используя механизмы композиции. Проверить ее на тестовом примере, с демон-страцией всех возможностей разработанных классов на конкретных данных.

Объект – студент. Параметры: имя и массив экзаменационных оценок (n=5). Методы: процедура инициализации, процедура вывода на экран значений параметров и функция, определяющая и возвращающая по запросу средний балл.

Объект – группа. Параметры: количество студентов в группе и массив студентов. Методы: процедура инициализации, процедура вывода на экран значений параметров и процедура, определяющая и выводящая на экран 5 самых сильных студентов группы.

program Project3;

{$APPTYPE CONSOLE}

uses

SysUtils,Math;

type mas=array [1..5] of integer;

type student=object

name:string;

ocenki:mas;

procedure print(var x);

procedure init(var s:string;var x:mas);

function sredbal(h:mas):real;

end;

procedure student.init( var s: string; var x:mas);

var a:array[1..5] of integer absolute x;

i,t:integer;

st:string absolute s;

begin

readln(st);

for i:=1 to 5 do begin

readln(t);

a[i]:=t;

end;

writeln;

end;

procedure student.print(var x);

var i:integer;

b:array [1..5] of integer absolute x;

begin

write(name,' ');

for i:=1 to 5 do

write(b[i]:2);

writeln;

end;

function student.sredbal(h:mas):real;

var i:integer;

begin

result:=0;

for i:=1 to 5 do

result:=result+h[i]/5;

end;

type group=object

n:byte;

students:array[1..50] of student;

procedure newinit(an:byte;var astudents);

procedure newprint(an:byte;var astudents);

procedure thebest(an:byte;var astudents);

end;

procedure group.newinit(an:byte; var astudents);

var d:array [1..50] of student absolute astudents;

i:byte;

begin

n:=an;

for i:=1 to an do begin

students[i].init(d[i].name,d[i].ocenki);

end;

end;

procedure group.newprint(an:byte; var astudents);

var i:integer;

a:array [1..50] of student absolute astudents;

begin

for i:=1 to an do begin

students[i].print(a[i].ocenki);

end;

writeln;

end;

procedure group.thebest(an:byte;var astudents);

var j,i:integer;

f:student;

a:array[1..50] of student absolute astudents;

begin

for j:=1 to 5 do begin

for i:=0 to an-j do begin

if a[i].sredbal(a[i].ocenki)>a[i+1].sredbal(a[i+1].ocenki) then begin

f:=a[i+1];

a[i+1]:=a[i];

a[i]:=f;

end;

end;

end;

for i:=0 to 4 do begin

a[an-i].print(a[an-i].ocenki);

end;

end;

var st:student;

gr:group;

k,i:integer;

s:string;

x:mas;

begin

st.init(st.name,st.ocenki);

st.print(st.ocenki);

readln(k);

if k=1 then

writeln(st.sredbal(st.ocenki):3:6);

readln(k);

gr.newinit(k,gr.students);

gr.newprint(k,gr.students);

gr.thebest(k,gr.students);

readln;

|  |
| --- |
| student |
| name,ocenki  50 |
| init(var s,var x),print(var x), sredbal(h:mas):real |

|  |
| --- |
| group |
| n,students |
| newinit(an:byte,var astudents), newprint(an:byte, var astudents), thebest(an:byte, var sstudents) |

end.

|  |  |  |  |
| --- | --- | --- | --- |
| Входные данные | y 4 4 4 5 4  1  5  a 5 5 5 5 5  b 4 4 3 3 5  c 3 3 5 5 5  d 3 4 4 4 5  e 3 3 3 3 3 | t 4 4 3 5 4  1  6  a 4 3 3 4 5  b 4 5 4 2 4  c 2 5 5 4 5  d 3 4 4 5 5  e 5 5 5 5 4  f 4 4 4 5 5 | k 3 3 5 5 4  1  4  a 3 3 4 5 5  b 3 5 5 5 3  c 5 5 5 5 5  d 4 4 5 5 5 |
| Ожидаемый результат | y 4 4 4 5 4  4.200000  a 5 5 5 5 5  b 4 4 3 3 5  c 3 3 5 5 5  d 3 4 4 4 5  e 3 3 3 3 3  a 5 5 5 5 5  c 3 3 5 5 5  d 3 4 4 4 5  b 4 4 3 3 5  e 3 3 3 3 3 | t 4 4 3 5 4  4.000000  a 4 3 3 4 5  b 4 5 4 2 4  c 2 5 5 4 5  d 3 4 4 5 5  e 5 5 5 5 4  f 4 4 4 5 5  e 5 5 5 5 4  f 4 4 4 5 5  d 3 4 4 5 5  c 2 5 5 4 5  b 4 5 4 2 4 | k 3 3 5 5 4  4.000000  a 3 3 4 5 5  b 3 5 5 5 3  c 5 5 5 5 5  d 4 4 5 5 5  c 5 5 5 5 5  d 4 4 5 5 5  b 3 5 5 5 4  a 3 3 4 5 5 |
| Полученный результат | y 4 4 4 5 4  4.200000  a 5 5 5 5 5  b 4 4 3 3 5  c 3 3 5 5 5  d 3 4 4 4 5  e 3 3 3 3 3  a 5 5 5 5 5  c 3 3 5 5 5  d 3 4 4 4 5  b 4 4 3 3 5  e 3 3 3 3 3 |  | k 3 3 5 5 4  4.000000  a 3 3 4 5 5  b 3 5 5 5 3  c 5 5 5 5 5  d 4 4 5 5 5  c 5 5 5 5 5  d 4 4 5 5 5  b 3 5 5 5 4  a 3 3 4 5 5 |