**МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ**

**Національний аерокосмічний університет ім. М.Є. Жуковського “ХАІ”**

**Кафедра 603**

**Лабораторна робота № 1**

**ЛОГІКА ПРЕДИКАТІВ**

ХАІ.603.622п.12О.050103.126331.ПЗ

**Виконав:** студент гр.622п Попков В.І.

\_\_\_\_\_\_\_\_\_\_ 09.09.13

(підпис)

**Перевірив**:

Фролова Г.О.\_\_\_\_\_\_\_\_\_\_ 09.09.13

(підпис)

**2013**

**Лабораторная работа №1. ЛОГИКА ПРЕДИКАТОВ**

**Цель работы**: изучить теоретические положения логики предикатов, программное представление предикатов; научиться описывать геометрические области сложной структуры с помощью предикатных формул.

**Задачи**

1. Ознакомиться с приведенными основными понятиями логики предикатов.
2. Описать с помощью предикатной формулы геометрическую область своего варианта. Для этого сначала выделить отдельные предикаты, описывающие ее составные элементы, а затем объединить их в общую формулу с помощью логических операций.
3. Написать программу (в произвольно выбранной среде программирования), которая будет определять принадлежность заданной точки декартовой плоскости геометрической области своего варианта. Координаты точки заданы парой действительных чисел и должны вводиться с клавиатуры. Результат должен выводиться в виде «Точка (х, у) принадлежит (не принадлежит) заданной области». Программная реализация графической иллюстрации не требуется. Каждый предикат, образующий геометрическую область, должен быть реализован в виде отдельной функции.

**Вариант 15**



P(x,y)=((x>=0)&&(x<=2)&&(y>=-2)&&(y<=0)&&(x\*x+y\*y>=1));

S(x,y)=((x>=-2)&&(x<=0)&&(y>=0)&&(y<=2)&&(x\*x+y\*y>=1));

L(x,y)=((x<=0)&&(y<=0)&&(x\*x+y\*y<=1));

D(x,y)=((x>=0)&&(y>=0)&&(x\*x+y\*y<=1)).

Заштрихованная область примет значение: P(x,y) ∨ S(x,y) ∨ L(x,y) ∨ D(x,y).

**Основы логики предикатов**

***Понятие предикат***

Несмотря на большую важность логики высказываний, ее средств оказывается недостаточно для изучения и описания простейших заключений арифметики, геометрии, не говоря уже о довольно сложных логических выводах в науке и практике. Возможности математической логики в этом смысле оказываются значительно расширенными, если вместо однородных булевых функций рассматривать функции, значения которых по-прежнему принадлежат множеству *Y* = {0,1}, а аргументы - множеству самого общего вида. Такие логические функции называют *двузначными предикатами*.

Логика предикатов предполагает логику высказываний уже известной и начинается с анализа строения простых высказываний. При этом простые высказывания выражают, что некоторые объекты (или объект) обладают определенными свойствами или же состоят между собой в некоторых отношениях.

Если имеется некоторое множество объектов *М*, то под *предикатом-свойством*, определенном на этом множестве, понимается функция на *М*, принимающая значения “ИСТИНА” (1) или “ЛОЖЬ” (0). Предикаты-свойства называются *унарными* (однородными) предикатами и считаются частным случаем *предикатов-отношений*. В зависимости от того, между каким числом объектов устанавливаются отношения, различают *двуместные* (БИНАРНЫЕ), *трехместные* (ТЕРНАРНЫЕ) и т.д., в общем случае - *n*-местные предикаты. Например, пусть *М* - множество натуральных чисел, тогда одноместным (унарным) предикатом *P*(*x*), определенным на *М*, может быть предикат-свойство *P* “быть простым числом”.

***Преобразование формул логики предикатов***

Для преобразования формул, содержащих предикаты, операции логики высказываний и кванторы, в логике предикатов используется ряд законов. Рассмотрим некоторые из них в виде примеров.

1. Вынесение кванторов за скобки:

∀x *P*(*x*) ∧ ∀x *Q*(*x*) ≡ ∀x (*P*(*x*) ∧ *Q*(*x*)); (1.5)

∃x *P*(*x*) ∨ ∃x *Q*(*x*) ≡ ∃x (*P*(*x*) ∨ *Q*(*x*)). (1.6)

2. Переход от квантора существования к квантору всеобщности и обратно:

 ≡ ∀x ; (1.7)

 ≡ ∃x . (1.8)

Равносильности (1.7) и (1.8) являются обобщениями законов де Моргана, к которым они сводятся в случае конечности множеств *М* = {*x*}.

Применяя отрицание к общим частям (1.7) и (1.8), имеем

 ≡ ; (1.9)

 ≡ . (1.10)

3. Дизъюнкция и конъюнкция кванторов:

(∀x *P*(*x*) ∨ ∀x *Q*(*x*)) → ∀x (*P*(*x*) ∨ *Q*(*x*)); (1.11)

(∃x *P*(*x*) ∧ ∃x *Q*(*x*)) → ∃x (*P*(*x*) ∧ *Q*(*x*)). (1.12)

4. Равносильности

∀x *P*(*x*) ∨ *C* ≡ ∀x (*P*(*x*) ∨ *C*); (1.13)

∃x *P*(*x*) ∨ *C* ≡ ∃x (*P*(*x*) ∨ *C*); (1.14)

если *С* - высказывание или предикат, не содержащие *x*. Если такое обозначение переменной в предикате *P*(*x*) произвольно, то

∀x *P*(*x*) ≡ ∀y *P*(*y*).

Следовательно, согласно равносильности (1.13) можно утверждать следующие равносильности:

∀x *P*(*x*)∨∀x *Q*(*x*) ≡ ∀x *P*(*x*)∨∀y *Q*(*y*) ≡ ∀x (*P*(*x*)∨∀y *Q*(*y*)) ≡ ∀x∀y (*P*(*x*)∨*Q*(*y*)),

т.е. имеет место равносильность

∀x *P*(*x*)∨∀x *Q*(*x*) ≡ ∀x∀y (*P*(*x*)∨*Q*(*y*)). (1.15)

Аналогично выводится равносильность

∃x *P*(*x*)∧∃x *Q*(*x*) ≡ ∃x∃y (*P*(*x*)∧*Q*(*y*)). (1.16)

Приведенные равносильности (1.15) и (1.16) позволяют любые выражения логики предикатов сопоставить с равносильными формулами некоторого стандартного вида, так называемыми *предварительными нормальными формами*.

***Основные определения и примеры предикатов***

**Определение 1.1.** *Предикатом* называется функция, аргументы которой принимают значения из произвольного множества, а сама функция – значения 1 (“истинно”) или 0 (“ложно”).

Предикат называется *n*-*местным* (*n* = 1, 2, ...), если соответствующая функция зависит от *n* аргументов.

Пусть *P(x*1, ..., *x*n*)* есть *n*-местный предикат, аргументы *x*1, ..., *x*n которого принимают значения из некоторого множества *М*. Если при *x*1 = *а*1, ..., *x*n = *а*n, где *a*i∈*M* (*i* = 1, 2, ..., *n*), *P(a*1, ..., *а*n*)* = 1 (или *P(a*1, ..., *а*n*)* = 0), то набор или точка *(a*1, ..., *а*n*)* удовлетворяет (или не удовлетворяет) предикату *P(x*1, ..., *x*n*)*, который определен на множестве *М*.

Пример 1.1

Одноместному предикату *R*(*x*) = (*x*+2 = 0) удовлетворяет точка *x* = -2, так как *R*(- 2) = (- 2 + 2 = 0), но не удовлетворяет точка *x* = 1, так как *R*(1) =(1+2 = 0)=0.

Пример 1.2

Двухместному предикату

*Q(x*,*y)* = 

точка (2,1) удовлетворяет, так как *Q*(2,1) = 1, а точка (,2) не удовлетворяет, так как *Q*(,2) = 0. Если положить *x* = 3, то получим предикат

*Q(*3,*y)* = ,

который при любом значении *y* имеет значение 0 (т.е. ложен).

Пример 1.3

*n*-местному предикату

*P(x*1, ..., *x*n*)* = *(*+ ... + ≥ 0*)*

удовлетворяет любой набор *(x*1, ..., *x*n*)∈R*n.

Если положим *x*2 = 2, *x*3= 3, ..., *x*n = *n*, то получим одноместный предикат *P*(*x*1, 2, 3, ..., *n*) = (+ 22 + ... + *n*2 ≥ 0), которому удовлетворяет каждое значение *x*1*∈R*1. Однако *n*-местному предикату *P*1*(x*1, ..., *x*n*)* = *(*+ ... + < 0*)* каждый набор (*а*1, ..., *а*n) значений аргументов *x*1, ..., *x*n не удовлетворяет.

Часто **одноместный предикат** называют свойством. Например, предикат *P*1*(x)* = *(x >* 0*)* = 1 на множестве действительных чисел определяет свойство числа *x* быть положительным.

**Нульместный предикат** не зависит ни от одной переменной, т.е. является высказыванием, например: 2 = 2 (истинное), 5 < 3 (ложное), 12 + 2 = 4 (ложное).

**Определение 1.2.** Для *n* ≥ 1 *n*-местный предикат *P(x*1, ..., *x*n*)* на некотором множестве *М* называется *тождественно-истинным*, если для любого набора (*а*1, ..., *а*n)∈*М* значений аргументов *x*1, ..., *x*n *P(а*1, ..., *а*n*)* = 1; *тождественно-ложным*, если для любого набора (*а*1, ..., *а*n)∈*М* *P(а*1, ..., *а*n*)* = 0; *выполнимым*, если существует хотя бы один набор значений аргументов, для которого *P(а*1, ..., *а*n*)* = 1. Например, предикат *P(x*1, ..., *x*n*)* (см. пример 6.3) - тождественно- истинный, *P*1*(x*1, ..., *x*n*)* - тождественно-ложный, а предикат *Q(x*,*y)* (см. пример 6.2) - выполнимый. Истинное высказывание есть нульместный тождественно-истинный предикат, а ложное - нульместный тождественно-ложный предикат.

**Определение 1.3.** Из двух *n*-местных предикатов *R(x*1,...,*x*n*)* и *Q(x*1,...,*x*n*)* на одном и том же множестве предикат *Q(x*1,...,*x*n*)* называется ***следствием*** *R(x*1,...,*x*n*)*, если любой набор значений переменных удовлетворяющих  *R(x*1,...,*x*n*)*, удовлетворяет и *Q(x*1,...,*x*n*)*.

Сокращенно следствие обозначается *R(x*1, ..., *x*n*)* ⇒ *Q(x*1, ..., *x*n*)*.

*Например*:

1) из предикатов *R(n)* (*n* делится на 4) и *Q(n)* (*n* делится на 2) имеем *Q(n)* - следствие *R(n)* на множестве натуральных чисел, т.е. *R(n)*⇒*Q(n)*.

2) предикат *Q*1*(x*1,*x*2*)* = (*x*1 - *x*2) ≥ 5 является следствием *R*1*(x*1,*x*2*)* = (*x*1 - *x*2) = 5 на множестве действительных чисел, но не наоборот, т.е. *R(x*1,*x*2*)* ⇒ *Q(x*1,*x*2*)*, но не *Q*1⇒ *R*1.

3) каждый из предикатов *R*2*(n)* = (*n* + 2 = 4) и *Q*2*(n)* = (*n* = 2) является следствием другого на множестве натуральных чисел, т.е. *Q*2⇒*R*2 и *R*2⇒*Q*2.

Из определения 1.3 следует:

а) любой предикат на данном множестве есть следствие тождественно-ложного предиката на этом же множестве;

б) тождественно-истинный предикат - следствие любого предиката на данном множестве.

**Определение 1.4.** Предикаты *R(x*1, ..., *x*n*)* и *Q(x*1, ..., *x*n*)* на одном и том же множестве называются ***равносильными***, если их значения совпадают для любого набора значений переменных *x*1, ..., *x*n, и обозначаются: *R(x*1, ..., *x*n*)* ≡ *Q(x*1, ..., *x*n*)*. Например, *R*2*(n)* ≡ *Q*2*(n)*, где *R*2*(n)* и *Q*2*(n)* - предикаты из предшествующего примера 3. *R(x*,*y)* ≡ *Q(x*,*y)*, если *R(x*,*y)* = (*x*2 - *y*2 ≥ 1), *Q(x*,*y)* = ((*x* - *y*)(*x* + *y*) ≥ 1) на множестве действительных чисел.

Из определения 1.4 следует, что равносильные уравнения, системы уравнений, неравенства и системы неравенств задают равносильные предикаты.

При постановке наборов значений переменных предикаты образуют высказывания, следовательно, над предикатами можно производить операции логики высказываний и в результате получать новые предикаты.

***Предикатные формулы. Равносильные формулы***

Вернемся к предикатным формулам и поясним их подробнее.

Формулы логики предикатов определяются индуктивно.

1. Базис индукции: всякая пропорциональная буква есть формула; если *P* есть *n*-местная предикатная буква, *n* > 0 и *x*1, ..., *x*n - предметные переменные, то *P*(*x*1, ..., *x*n) есть формула (простая).

2. Если *А* и *В* - формулы, то формулами являются и следующие комбинации символов:

⎤ *A*, *(A*∧*B)*, *(A*∨*B)*, *(A*⊃*B)*, *(A*⇔*B)*.

3. Если *А* - формула и *х* - предметная (индивидуальная) переменная, то ∀*хА* и ∃*хА* есть формулы. Здесь формулу *А* называют *областью действия кванторов* ∀*х*, ∃*х*.

Скобки в формулу вводятся автоматически и в тех же целях, что и в пропозициональные формулы. Об опускании скобок принимаются те же соглашения, что и в логике высказываний. Опускаются скобки, заключающие в себе простые формулы. Примеры формул: *R(x)*, *P(x*,*y)*, *Q(x*,*y*,*z)* - простые, ∀*x(*∃*y(R(x*,*y)*∨*Q(x*,*y)))*, ⎤ *Q(x*,*x)*, ⎤ *x(*⎤ *P(x*,*x)*⇔ *P(x*,*y)* = *Q(y))* и т.п.

**Определение 1.7.** Вхождение переменной в данную формулу называется ***связанным***, если она стоит за знаком ∀ или ∃ либо находится в области какого-либо квантора по той же переменной; в противном случае вхождение переменной в данную формулу называется ***свободным***.

Переменная называется ***свободной*** в данной формуле, если, по крайней мере, одно ее вхождение в данной формуле свободно, и ***связанной***, если по крайней мере одно ее вхождение связано. Формула, не содержащая свободных переменных, называется замкнутой. Например, в формуле ∀*xR(x*,*y)* оба вхождения связаны, а единственное вхождение *y* свободно. В формуле ∀*xP(x*,*y)*⊃∃*yQ(x*,*y)* первые два вхождения *х* связаны, а третье свободно, т.е. в этой формуле *х* является одновременно свободной и связанной переменной.

**Листинг программы**

#include <iostream>;

#include <math.h>// подключение математической библиотеки

#include <Windows.h>

using namespace std;

void P(float x, float y,float &res)//функция отвечающая за заштрихованную область фигуры в 1 четверти

{

if((x>=0)&&(x<=2)&&(y>=-2)&&(y<=0)&&(x\*x+y\*y>=1))

res=1;

return;

}

void S(float x, float y,float &res)//функция отвечающая за заштрихованную область фигуры в 2 четверти

{

if((x>=-2)&&(x<=0)&&(y>=0)&&(y<=2)&&(x\*x+y\*y>=1))

res=1;

return;

}

void L(float x, float y,float &res)//функция отвечающая за заштрихованную область фигуры в 3 четверти

{

if((x<=0)&&(y<=0)&&(x\*x+y\*y<=1))

res=1;

return;

}

void D(float x, float y,float &res)//функция отвечающая за заштрихованную область фигуры в 4 четверти

{

if((x>=0)&&(y>=0)&&(x\*x+y\*y<=1))

res=1;

return;

}

int main()

{

SetConsoleOutputCP(1251);

float x,y;

float res;// переменная отвечающая за попадание точки в заштрихованную область

cout<<"";

cout<<"Input x--";

cin>>x;

cout<<"Input y--";

cin>>y;

res=0;//инициализация переменной

P(x, y,res);

S(x, y,res);

L(x, y,res);

D(x, y,res);

if(res==1){cout<<" The dot posseses to the defined range ";}

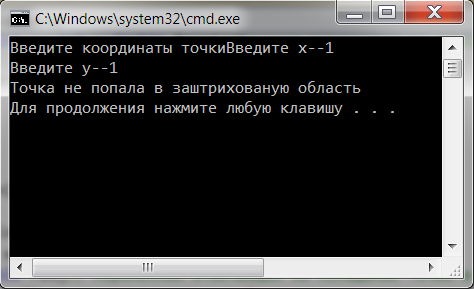
else cout<<"The doesn't posses to the defined range "<<endl;

return 0;

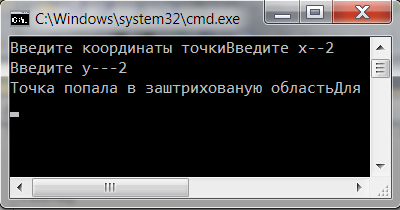
}

**Результаты работы программы**

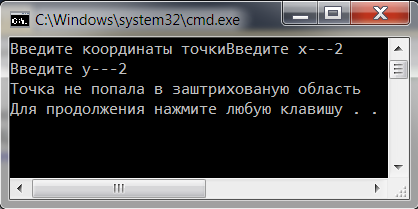
1. Проверка попадания точки в заштрихованную область D.

****

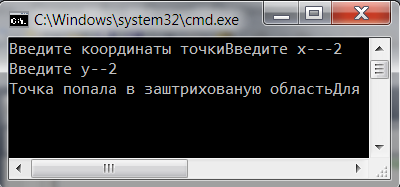
1. Проверка попадания точки в заштрихованную область P.

****

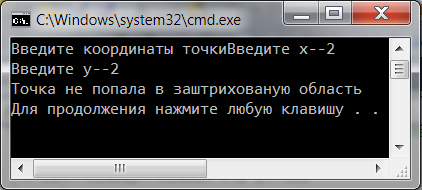
1. Проверка исключения, попадания точки в заштрихованную область L.

****

**4)** Проверка попадания точки в заштрихованную область S.

****

**5)** Проверка исключения, попадания точки в заштрихованную область D.

****

**Вывод:** выполнив эту лабораторную работу я описал геометрические области сложной структуры с помощью предикатных формул, изучил теоретические положения логики предикатов, программное представление предикатов.