МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра вычислительной техники

Отчет по лабораторной работе № 5

«Множества»

по дисциплине «Функциональное и логическое программирование»

Вариант 28

Выполнил студент группы 220681:

Шайхаттаров Д.В.

Проверил:

Асс. Савенков П. А.

Тула 2020

**1. Цель работы**

Получить навык создания множества элементов при программировании в среде Пролог и выполнения операции над ними.

**2. Задание на лабораторную работу**

Создать предикат определения множества точек единичного квадрата, попадающих в заштрихованную область, представленную на рисунке:



Рисунок 1 – Единичный квадрат.

**3. Математическая постановка задачи**

Дан список координат точек, которые проверяются на нахождение в заштрихованной области.

Для выполнения этой задачи её необходимо разбить на две части:

1. Получить множество точек лежащий внутри единичного квадрата, но не лежащие внутри вписанной окружности.
2. Получить множество точек лежащих внутри квадрата в центре.

Результатом будет являться множество точек лежащих в заштрихованной области.

**4. Тестирование**

При запуске программы и вводе множеств [[0.5,0.5],[0.1,0.1],[0.75,0.75]] в консоль, выводятся следующие результаты (Рисунок 2).

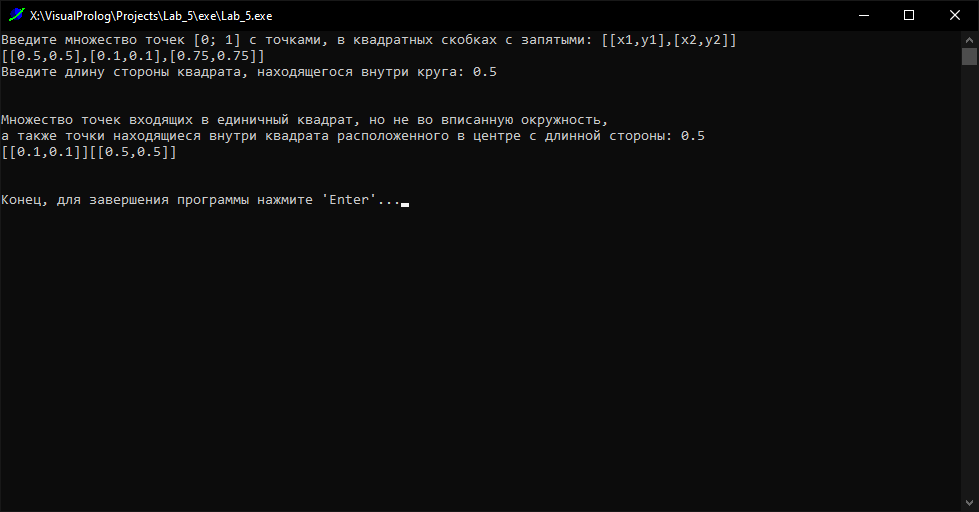


Рисунок 2 – Результат тестирования

**5. Заключение**

В ходе данной лабораторной работы был получен навык создания множества элементов при программировании в среде Пролог и выполнения операции над ними.

**Приложение**

**Исходный код программы задания**

implement main  
    open core, console, math  
  
class predicates  
    out\_dots : (real\*\* S1, real\*\* S2 [out]).  
    out\_dots1 : (real Side, real\*\* S1, real\*\* S2 [out]).  
    unique : (real\*\* S, real X, real Y, string Bool [out]) determ.  
    check : (real\*\* S, real\*\* SRes [out]).  
  
clauses  
    out\_dots([], []).  
    out\_dots([[X, Y] | T1], [[X2, Y2] | T2]) :-  
        R = sqrt(math::sqr(X - 0.5) + sqr(Y - 0.5)),  
        R > 0.5,  
        X >= 0,  
        X <= 1,  
        Y >= 0,  
        Y <= 1,  
        !,  
        X2 = X,  
        Y2 = Y,  
        out\_dots(T1, T2).  
    out\_dots([\_ | T1], T2) :-  
        out\_dots(T1, T2).  
  
    out\_dots1(\_, [], []).  
    out\_dots1(Side, [[X, Y] | T1], [[X2, Y2] | T2]) :-  
        Side < 0.5 \* sqrt(2),  
        X > 0.5 - Side / 2,  
        X < 0.5 + Side / 2,  
        Y > 0.5 - Side / 2,  
        Y < 0.5 + Side / 2,  
        !,  
        X2 = X,  
        Y2 = Y,  
        out\_dots1(Side, T1, T2).  
    out\_dots1(Side, [\_ | T1], T2) :-  
        out\_dots1(Side, T1, T2).  
  
    unique([], \_, \_, "false").  
  
    unique([[X, Y] | T1], X1, Y1, Contains) :-  
        if (X <> X1 or Y <> Y1) and ! then  
            unique(T1, X1, Y1, Contains)  
        else  
            Contains = "true"  
        end if.  
  
    check([], []).  
  
    check([[X, Y] | T1], [[X2, Y2] | T2]) :-  
        unique(T1, X, Y, Contains),  
        Contains <> "true",  
        !,  
        X2 = X,  
        Y2 = Y,  
        check(T1, T2).  
  
    check([\_ | T1], T3) :-  
        check(T1, T3).  
  
clauses  
    run() :-  
        init(),  
        write("Введите множество точек [0; 1] с точками, в квадратных скобках с запятыми: [[x1,y1],[x2,y2]]"),  
        nl,  
        Set = read(),  
        write("Введите длину стороны квадрата, находящегося внутри круга: "),  
        Side = read(),  
        out\_dots(Set, Out),  
        out\_dots1(Side, Set, Out1),  
        check(Out, Res),  
        check(Out1, Res1),  
        write(  
            "\n\nМножество точек входящих в единичный квадрат, но не во вписанную окружность,\nа также точки находящиеся внутри квадрата расположенного в центре с длинной стороны: ",  
            Side),  
        nl,  
        write(Res, Res1),  
        nl,  
        \_ = readLine(),  
        write("\n\nКонец, для завершения программы нажмите 'Enter'..."),  
        \_ = readLine().  
  
end implement main  
  
goal  
    mainExe::run(main::run).