

Министерство науки и высшего образования Российской Федерации  
Муромский институт (филиал)  
Федерального государственного бюджетного образовательного учреждения высшего  
образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет \_\_\_\_\_ ФИТ \_\_\_\_\_

Кафедра \_\_\_\_\_ ПИН \_\_\_\_\_

## *ЛАБОРАТОРНАЯ РАБОТА №9*

По \_\_\_\_\_ Основы алгоритмизации и программирования \_\_\_\_\_

Руководитель

Привезенцев Д.Г.  
\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись) (дата)

Студент \_\_\_\_\_ ПИН - 121 \_\_\_\_\_  
(группа)

Ермилов М.В.  
\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись) (дата)

Муром 2021

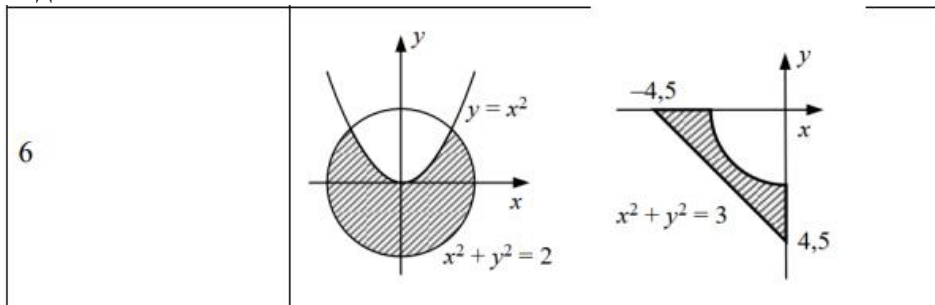
## Лабораторная работа №9

**Тема:** Приближенное вычисление площади фигуры методом МонтеКарло

**Цель работы:** изучение метода Монте-Карло (метода статистических испытаний) на примере вычисления площади фигуры.

Ход работы:

**Задание**



Код:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include "lib.h"

void main(){
    srand(time(NULL));

    R1 = sqrt(2);
    L2 = 4.5;
    S1 = (2 * M_PI) - ((float) 1/3 + M_PI / 2);
    S01 = pow(R1*2, 2);
    S2 = (pow(4.5, 2) / 2) - ((3 * M_PI) / 4);
    S02 = pow(4.5, 2);

    printf("Вычисление фигуры №1\nЕё площадь:%f\n\n", S1);
    func(100, 1);
    func(1000, 1);
    func(10000, 1);
    func(100000, 1);
    func(1000000, 1);
    func(10000000, 1);
}
```

					МИ ВлГУ 09.03.04						
Изм.	Лист	№ докум.	Подпись	Дата							
Разраб.	Ермилов М.В.				Приближенное вычисление площади фигуры методом МонтеКарло			Лит.	Лист	Листов	
Провер.	Привезенцев Д.Г.									2	6
Реценз.								ПИН-121			
Н. Контр.											
Утверд.											

```

printf("\nВычисление фигуры №2\nЕё площадь:%f\n\n",S2);
func(100, 2);
func(1000, 2);
func(10000, 2);
func(100000, 2);
func(1000000, 2);
func(10000000, 2);
printf("\nКонец работы");
getchar();
}

```

```

Вычисление фигуры №1
Её площадь:4.379056

100 точек:
Площадь получилась: 4.239999
отклонение: 0.139056
попавшие точки: 53

1000 точек:
Площадь получилась: 4.423999
отклонение: -0.044944
попавшие точки: 553

10000 точек:
Площадь получилась: 4.439199
отклонение: -0.060144
попавшие точки: 5549

100000 точек:
Площадь получилась: 4.383039
отклонение: -0.003984
попавшие точки: 54788

1000000 точек:
Площадь получилась: 4.374479
отклонение: 0.004576
попавшие точки: 546810

10000000 точек:
Площадь получилась: 4.381268
отклонение: -0.002212
попавшие точки: 5476585

Вычисление фигуры №2
Её площадь:7.768806

100 точек:
Площадь получилась: 7.897500
отклонение: -3.518444
попавшие точки: 39

1000 точек:
Площадь получилась: 7.998750
отклонение: -3.619695
попавшие точки: 395

10000 точек:
Площадь получилась: 7.699050
отклонение: -3.319994
попавшие точки: 3802

100000 точек:
Площадь получилась: 7.700468
отклонение: -3.321412
попавшие точки: 38027

1000000 точек:
Площадь получилась: 7.777559
отклонение: -3.398504
попавшие точки: 384077

10000000 точек:
Площадь получилась: 7.764358
отклонение: -3.385303
попавшие точки: 3834251

Конец работы

```

Рис. 1 - пример работы программы

### Дополнительный код

Для работы с данной лабораторной, была создана и использована библиотека для функций чтобы не засорять основной файл

Было создано 2 файла “lib.h” для хранения названий функций и “lib.c” где находятся сами функции

“lib.h”

```
float R1, L2, S1, S01, S2, S02;

typedef struct point_t Point;

float RandFloat(float min, float max);
Point PointGeneration(float Xmin, float Xmax, float Ymin, float Ymax);

float PointSquare(int n, int true, float S0);

int PointCheck1(Point point);
int PointCheck2(Point point);

void func(int n, int Shape);

void ConsolePercent(int Max, int I);
void ConsolePercentClear();

#include "lib.c"
```

					МИ ВлГУ 09.03.04	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

“lib.c”

```
float R1, L2, S1, S01, S2, S02;

typedef struct point_t{ float x, y; }Point;

float RandFloat(float min, float max){ return (((float)rand())/((float)(RAND_MAX))) *
(max - min)) + min; }

Point PointGeneration(float Xmin, float Xmax, float Ymin, float Ymax){
    Point a;
    a.x = RandFloat(Xmin, Xmax);
    a.y = RandFloat(Ymin, Ymax);
    return a;
}

int PointCheck1(Point point){
    if((pow(point.x,2) + pow(point.y,2)) <= 2){
        if(point.y > 0){
            if(pow(point.x,2) <= point.y){
                return 0;
            }
        }
        return 1;
    }
    return 0;
}

int PointCheck2(Point point){
    if((pow(point.x,2) + pow(point.y,2)) < 3){
        return 0;
    }
    point.y += 4.5;
    point.x *= -1;
    if(point.x > point.y){
        return 0;
    }
    return 1;
}

float PointSquare(int n, int true, float S0){
    return (float) ((float) true / n) * S0;
}

void func(int n, int Shape){
    printf("%i точек:\n", n);
    int true = 0;
    for(int i = 0; i < n; i++){
        switch (Shape){
```

					МИ ВлГУ 09.03.04	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

```

        case 1:
            true += PointCheck1(PointGeneration(-R1, R1, -R1, R1));
            break;
        case 2:
            true += PointCheck2(PointGeneration(-L2, 0, -L2, 0));
            break;
    }
}

float S;
switch (Shape){
    case 1:
        S = PointSquare(n, true, S01);
        break;
    case 2:
        S = PointSquare(n, true, S02);
        break;
}

printf("Площадь получилась: %f\n", S);
printf("отклонение: %f\n", S1 - S);
printf("попавшие точки: %i\n\n", true);
}

```

					МИ ВлГУ 09.03.04	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6