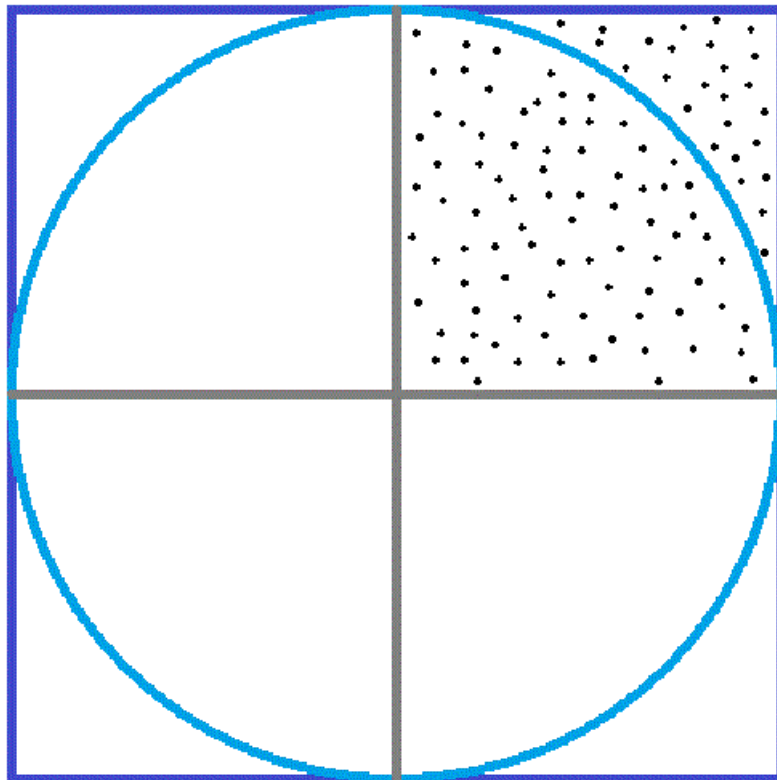


●8. モンテカルロ法

円周率の計算

乱数によって数値計算を行う一例として、円の面積計算（円周率を求める方法）を行ってみます。まず、1辺の長さが1の正方形の中にランダムに点を打っていきます。プログラムでは、このランダムに点を打つ際に乱数を用います。次に半径 r の円を表わす式は $(xの2乗)+(yの2乗)=(rの2乗)$ であるので、 $x > 0$ 、 $y > 0$ 、 $r = 1$ の時を考えると、下の図のようになります。



上の図から、 $(\text{円内の点の数}) / (\text{正方形内の点の数}) = (1/4 \text{ 円の面積}) / (\text{正方形の面積})$ となるのが分かり、円の面積が求められます。

【課題】

それでは、以上の説明からパイを求めるアルゴリズムを構築しプログラムで実装してみてください。

※試行回数は正方形内の点の数とし、キーボードより入力し1億回以下とします。

【実行例】

試行回数：100000000

PI = 3.141553332

【クラス図】

クラス名

RandomPi

フィールド定義

-x:double

-y:double

-pi:double

-kaisu:int

メソッド定義

+Pi():

+sim():void

+setKaisu(k:int):void

+getPi():double

+main():int

-:private

+:public

【C言語例】

RandPi.c

//モンテカルロ法を用いて円周率を求めるプログラム

//原点を中心とする半径 r の円（円周）の方程式は

// $x^2+y^2=r^2$

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
int main(void){
```

```
    int i;
```

```
    int count = 0;
```

```
    int kaisu = 10000000;
```

```
    double x,y,pi;
```

```
    printf("回数 : ");scanf("%d",&kaisu);
```

```
    /* 乱数の種を初期化 */
```

```
srand(time(NULL));
/* 0 から 1 の乱数を生成・出力 */
for(i=0;i<kaisu;i++){
    x = (double)rand()/RAND_MAX;
    y = (double)rand()/RAND_MAX;
    if(x*x + y*y < 1) /*ランダムに発生させた点が円内にあるか確認します*/
        count++;
}
/* 円周率を計算・出力 */
pi = (double)count / kaisu * 4; //(count/kaisu)は半径r= 1 の1/4円の面積のため、
pi*r^2で円周率はその4倍になります

printf("pi=%f¥n", pi);
return 0;
}
```