**1.为每个cpp文件都写一个对应的头文件,其中放置对应的cpp文件中的需要向外部公开的函数声明或变量声明、常量声明，最好使用包含警戒。**

本例子中mymain.cpp中由于只有一个main函数(main函数是默认的入口函数)，可以不写mymain.h.

其它：

与**Func1.cpp 对应的func1.h**

#ifndef FUNC1H

#deinfe FUNC1H

int f1(int num);

#endif

与**Func2.cpp 对应的func2.h**

#ifndef FUNC2H

#deinfe FUNC2H

int g1(int num);

int g2(int num);

#endif

若某个cpp文件中需要访问其它cpp中的函数，只需在其中include 相应的头文件

如：func2.cpp中用到发f1函数，需在func2.cpp中加入#include "func1.h"

**对于常数/枚举常数，也可放在一个单独的头文件中，如global.h,哪个cpp中要用到这些常量，就在其中#include "global.h"。**

#ifndef GLOBALH

#deinfe GLOBALH

const int YEARDAYS=365;

const enum WEEKDAY {MON=1,TUE,WED,THU,FRI,SAT,SUN};

#endif

**//mymain.cpp**

#include <iostream.h>

#include "func2.h"

int main(int argc,char \* argv[])

{

cout<<g2(10)<<endl;

return 0;

}

**//func1.cpp**

#include "func1.h"

#include "func2.h"

int f1(int num)

{

return g1(num\*2);

}

**//func2.cpp**

#include "func2.h"

#include "func1.h"

int g1(int num)

{

return num\*num;

}

int g2(int num)

{

return f1(num\*3);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**//5.2洗牌程序**

#include <limits.h> //定义INT\_MAX和ULONG\_MAX常量

#include <windows.h> //定义GetTickCount()函数,其返回从开机到现在经过的毫秒数

class TRandom

{ public:

TRandom (long seed=0) {mSeed=(seed?seed: GetTickCount()); }

void Seed(long seed=0) {mSeed=(seed?seed: GetTickCount());}

int Integer() {return Next();}

int Integer(int min,int max) {return min+Next()%(max-min+1);}

double Real() {return double(Next())/double(INT\_MAX);}

private:

void Change() {mSeed=(314159265\*mSeed+13579)%ULONG\_MAX;}

int Next()

{

int loops=mSeed%3;

for (int i=0;i<=loops;i++)

Change ();

return int (mSeed/2); }

unsigned long mSeed;

};

int main()

{

int cards[54];

for(int i=0;i<54;++i)

cards[i]=i+1;

TRandom rand; //创建伪随机数发生器的一个实例对象

//以下是洗牌过程

for(int j=0;j<54;++j)

{

int pos=rand.Integer(j,53); //pos是一个[j,53]之间的随机数

//交换cards[j]和cards[pos];

int temp=cards[j];

cards[j]=cards[pos];

cards[pos]=temp;

}

//输出洗牌结果

for (int k=0;k<54;++k)

cout<<"Cards["<<k<<"]="<<cards[k]<<endl;

return 0;

}

采用伪随机数洗牌，无论洗多少张牌，排列的可能都是2的32次方种可能。因为伪随机数发生器给定，应用的算法给定，对指定的种子(seed),产生的伪随机数数列就确定了，也就是洗牌后的结果定了。而在32位平台上，seed是无符号长整形，最多有2的32次方种可能，它远远小于54的阶层。这也是商业加密算法不使用伪随机数加密的原因。

**//5.4投针法程序（仅供参考，不精确估计，李河作）**

#include <iostream>

using namespace std;

#include <limits.h> //声明INT\_MAX和ULONG\_MAX常量

#include <windows.h> //声明GetTickCount()函数,其返回从开机到现在经过的毫秒数

//#include <stdlib.h> //声明rand和srand函数

//#include <time.h> //声明time函数

class TRandom

{

public:

//缺省使用系统时间(开机后经过的毫秒数)为种子

TRandom(long seed = 0) { mSeed = (seed ? seed : GetTickCount()); }

//也可以定义自己的种子

void Seed(long seed = 0) { mSeed = (seed ? seed : GetTickCount()); }

//取得一个随机的整数

int Integer() { return Next(); }

//取得一个在指定范围内的随机整数

int Integer(int min, int max) { return min + Next() % (max - min + 1); }

//取得一个随机的（0到1之间的）小数

double Real() { return double(Next()) / double(INT\_MAX); }

private:

//使用调和算法

void Change() { mSeed = (314159265 \* mSeed + 13579) % ULONG\_MAX; }

//取得伪随机数发生器的随机数序列中的下一个随机整数

int Next()

{

int loops = mSeed % 3;

for (int i = 0; i <= loops; i++)

Change();

return int(mSeed / 2);

}

//随机数发生器的种子

unsigned long mSeed;

};

int main()

{

TRandom t;

long int j = 0;

double x;

double y;

long int maxNum = 1000000;//可变的

for (long int i = 0; i < maxNum;i++)

{

x = t.Real();

y = t.Real();

if ((fabs(sqrt(x\*x+y\*y)-1.0) < 1.0e-8)||(sqrt(x\*x+y\*y)<1.0))

j++;

}

double pi = 4.0\*j /maxNum;

cout << "pi=" <<pi<< endl;

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**//6骰子**

***//Dice.h***

#ifndef DICEH

#define DICEH

class Dice

{

public:

static const enum SIDE { FRONT = 0, LEFT, TOP, RIGHT, BOTTOM, BACK };

static const char \* Names[];//在Cpp文件中初始化

public:

Dice();

int Cast();

int FaceValue(SIDE side) const; //返回指定面的值

private:

void Turn();

void TurnX();

void TurnY();

void TurnZ();

private:

int values[6];

};

#endif

***//dice.cpp***

#include <iostream>

#include <stdlib.h>

#include <time.h>

#include "Dice.h"

//初始化Names数组

const char \* Dice::Names[] = { "FRONT","LEFT","TOP","RIGHT","BOTTOM","BACK" };

Dice::Dice()

{

values[FRONT] = 1;

values[TOP] = 2;

values[RIGHT] = 3;

values[LEFT] = 4;

values[BOTTOM] = 5;

values[BACK] = 6;

//初始化随机数发生器，也可放在入口函数main中。

srand(time(0)); // VC6.0中这样使用

}

//---------------------------------------------------------------------------

int Dice::Cast()

{

Turn();

return FaceValue(FRONT);

}

//---------------------------------------------------------------------------

void Dice::Turn()

{

//取得0－3的随机数:可用rand()%4

int num;

//取得0－3的随机数

num = rand()%4; VC6.0中这条语句

for (int i = 0; i < num; i++)

TurnX();

//取得0－3的随机数

num = rand()%4; VC6.0中为这条语句

for (int i = 0; i < num; i++)

TurnY();

//取得0－3的随机数

num = rand()%4; VC6.0中为这条语句

for (int i = 0; i < num; i++)

TurnZ();

}

//---------------------------------------------------------------------------

int Dice::FaceValue(SIDE side) const

{

return values[side];

}

//---------------------------------------------------------------------------

//按X轴旋转90度

void Dice::TurnX()

{

int temp = values[FRONT];

values[FRONT] = values[TOP];

values[TOP] = values[BACK];

values[BACK] = values[BOTTOM];

values[BOTTOM] = temp;

}

//---------------------------------------------------------------------------

//按Y轴旋转90度

void Dice::TurnY()

{

int temp = values[FRONT];

values[FRONT] = values[RIGHT];

values[RIGHT] = values[BACK];

values[BACK] = values[LEFT];

values[LEFT] = temp;

}

//---------------------------------------------------------------------------

//按Z轴旋转90度

void Dice::TurnZ()

{

int temp = values[LEFT];

values[LEFT] = values[TOP];

values[TOP] = values[RIGHT];

values[RIGHT] = values[BOTTOM];

values[BOTTOM] = temp;

}

//---------------------------------------------------------------------------

***//MyMain1.cpp 一个骰子***

#include <iostream>

using namespace std;

#include "dice.h"

//一个骰子。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

int main(int argc, char\* argv[])

{

Dice aDice;

for (int i = 1; i < 10; i++) {

cout << "第" << i << "次掷得的骰子正面值是：" << aDice.Cast() << endl;

for (int s = Dice::FRONT; s <= Dice::BACK; ++s)

{

Dice::SIDE side = (Dice::SIDE)s;

cout << Dice::Names[side] << "面=" << aDice.FaceValue(side) << endl;

}

}

return 0;

}

***//MyMain2.cpp 两个骰子***

#include <iostream>

using namespace std;

#include "dice.h"

//两个骰子。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

int main(int argc, char\* argv[])

{

Dice aDice;

Dice aDice2;

for (int i = 1; i < 10; i++) {

cout << "第" << i << "次掷得第一个骰子的正面值是：" << aDice.Cast() << endl;

cout << endl;

cout << "第" << i << "次掷得第二个骰子的正面值是：" << aDice2.Cast() << endl;

cout << endl;

for (int s = Dice::FRONT; s <= Dice::BACK; ++s)

{

Dice::SIDE side = (Dice::SIDE)s;

cout <<"第一个骰子的"<< Dice::Names[side] << "面=" <<

aDice.FaceValue(side) << endl;

}

cout << endl;

for (int s = Dice::FRONT; s <= Dice::BACK; ++s)

{

Dice::SIDE side = (Dice::SIDE)s;

cout << "第二个骰子的" << Dice::Names[side] << "面=" <<

aDice2.FaceValue(side) << endl;

}

cout << endl;

}

return 0;

}

***//在VC6.0下，随机数函数使用的例子如下：***

#include <iostream.h>

#include <stdlib.h> //用于随机函数:srand()、rand()

#include <time.h> //用于time()

//---------------------------------------------------------------------------

// void srand(unsigned long seed ); 功能：设置随机数发生器的种子。

// int rand(); 功能：返回0到2的32次方的一个伪随机整数

// int main(void)

// {

// time\_t t;

// srand((unsigned) time(&t)); //此句用系统时间作为种子。

// cout<<"Ten random numbers from 0 to 99")<<endl;

// for(int i=0; i<10; i++)

// cout<< rand() % 100 <<endl;

// return 0;

// }

//---------------------------------------------------------------------------