C++ STL中的数值组件主要包括随机数及其分布、复数、数值数组，全局数值函数。

随机数：

例：程序stl\_test88

// 每次运行程序

// 得到的随机数不一样

*random\_device* rd;

// 引擎

// create default engine as source

// of randomness

*default\_random\_engine* dre(rd());

// 线性分布

// use engine to generate integral

// numbers between 10 and 20(both included)

*uniform\_int\_distribution*<int> di(10, 20);

for (int i = 0; i < 20; ++i)

{

*cout* << di(dre) << " ";

}

*cout* << *endl*;

*cout* << *endl*;

// 线性分布

// use engine to generate floating-point

// numbers between 10.0 and 20.0

// (10.0 included, 20.0 not included)

*uniform\_real\_distribution*<double> dr(10, 20);

for (int i = 0; i < 8; ++i)

{

*cout* << dr(dre) << " ";

}

*cout* << *endl*;

*cout* << *endl*;

// 洗牌

// use engine to shuffle elements

*vector*<int> v = {1, 2, 3, 4, 5, 6, 7, 8, 9};

*shuffle*(v.*begin*(), v.*end*(), dre);

for (int i = 0; i < v.*size*(); ++i)

{

*cout* << v[i] << " ";

}

*cout* << *endl*;

// 两次洗牌结果不同

*shuffle*(v.*begin*(), v.*end*(), dre);

for (int i = 0; i < v.*size*(); ++i)

{

*cout* << v[i] << " ";

}

*cout* << *endl*;

第1次运行输出为：

20 10 18 17 20 17 14 17 20 14 20 11 11 12 16 14 16 10 14 20

19.0456 10.0466 13.316 10.2864 19.4737 13.4315 12.4724 10.0856

9 3 7 6 5 1 4 2 8

2 8 7 3 5 6 1 4 9

第2次运行输出为：

18 10 14 18 19 11 12 19 14 20 13 18 15 18 17 14 20 18 10 10

13.051 14.2941 17.6592 17.863 11.6908 14.8149 12.6939 12.9921

7 8 4 5 2 6 9 1 3

8 3 4 7 9 2 6 5 1

复数：

例：程序stl\_test89

// complex number with real and imaginary parts

// 4.0 + 3.0i

*complex*<double> c1(4.0, 3.0);

// 极坐标复数

// create complex number from polar coordinates

*complex*<float> c2(*polar*(5.0f, 0.75f));

*cout* << "c1: " << c1 << *endl*;

*cout* << "c2: " << c2 << *endl*;

// print complex numbers as polar coordinates

*cout* << "c1: magnitude: " << *abs*(c1)

<< " (squared magnitude: " << *norm*(c1) << ") "

<< " phase angle: " << *arg*(c1) << *endl*;

*cout* << "c2: magnitude: " << *abs*(c2)

<< " (squared magnitude: " << *norm*(c2) << ") "

<< " phase angle: " << *arg*(c2) << *endl*;

// 共轭

*cout* << "c1 conjugated: " << *conj*(c1) << *endl*;

*cout* << "c2 conjugated: " << *conj*(c2) << *endl*;

*cout* << "4.4 + c1 \* 1.8 " << 4.4 + c1 \* 1.8 << *endl*;

*cout* << "c1 + c2: "

<< c1 + *complex*<double>(c2.*real*(), c2.*imag*()) << *endl*;

// real = c1.real \* c2.real - c1.imag \* c2.imag

// imag = c1.real \* c2.imag + c1.imag \* c2.reals

*cout* << "c1 \* c2: "

<< c1 \* *complex*<double>(c2.*real*(), c2.*imag*()) << *endl*;

输出为：

c1: (4,3)

c2: (3.65844,3.40819)

c1: magnitude: 5 (squared magnitude: 25) phase angle: 0.643501

c2: magnitude: 5 (squared magnitude: 25) phase angle: 0.75

c1 conjugated: (4,-3)

c2 conjugated: (3.65844,-3.40819)

4.4 + c1 \* 1.8 (11.6,5.4)

c1 + c2: (7.65844,6.40819)

c1 \* c2: (4.4092,24.6081)