

# 编译期常量和运行期常量

## 编译期常量

编译期常量，在编译期确定值。例如以下java代码

```
final int a = 1;
```

关键在于编译期能确定其值，简单来说，不需要复杂的计算，也不依赖程序外界的输入，仅凭对代码的肉眼观察即可判定其是否是一个常量。

## 运行期常量

在程序运行时，能确定其是一个常量。比如：

```
Random rand = new Random(47);  
final int a = rand.nextInt(20);
```

a的计算过程在程序运行时完成，a在定义后不可修改。

## C++的 const 和 constexpr

const与java的final**不区分**是编译是常量还是运行时常量，具体由编译器确定

而constexpr限定是一个编译时常量。

例如以下代码

```
int a;  
cin >> a;  
const int b = a + 1; // OK  
constexpr c = a + 1; // ERROR
```

当它们修饰函数时 `const` 表示函数运算结果的皆为运行时常量。 `constexpr` 表示的是

1. 如果传入的参数为常量值，则其为编译时常量
2. 如果传入的数据为普通变量，则其和普通函数无异

```
constexpr int foo(int i)  
{  
    return i + 5;  
}  
  
int main()  
{  
    int i = 10;
```

```
std::array<int, foo(5)> arr; // OK

foo(i); // Call is OK, 运行期常量

// But...
std::array<int, foo(i)> arr1; // Error, 模板参数要求一个编译期常量

}
```