Hecke 是 Julia 下的计算代数数论包,

一、创建与赋值

1、预定义整环和有理数域

```
x = ZZ(2)
y = QQ(1//3)
```

2、Z 上的剩余环

```
p = 5
Rp = ResidueField(ZZ, p)
x = Rp(12) # x = 2
y = Rp(13) # y = 3
z = x * y  #z = 1
```

3、二次域

```
d = -3
F, t = quadratic_field(d) # t是单位元, d 是无平方因子数
# 这种赋值方式，要保证[]中的元素数量和域的次数一致，二次域是 2 个
x = F([1, 2])
y = F([2, 1])
#或者
x = 1 + 2t
y = 2 + t
#这种赋值方式可以支持多项式形式，系统会自动计算
z = 1 +2t + 3t^2
```

$x = 1 + 2\sqrt{-3}$
$y = 2 + \sqrt{-3}$
$z = -8 + 2\sqrt{-3}$

4、分圆域

```
C7, a = cyclotomic_field(7)
# 这种赋值方式，要保证[]中的元素数量和域的次数一致，n 次分圆域是 ϕ(n) 个。
x = C7([1, 2, 3, 4, 5, 6])
y = C7([6, 5, 4, 3, 2, 1])
#或者
x = 1 + 2a + 3a^2 + 4a^3 + 5a^4 + 6a^5
y = 6 + 5a + 4a^2 + 3a^3 + 2a^4 + a^5
#同样可以
z = 1 + 2a + 3a^2 + 4a^3 + 5a^4 + 6a^5 +7a^6
```

$x = 1 + 2z_7 + 3z_7^2 + 4z_7^3 + 5z_7^4 + 6z_7^5$
$y = 6 + 5z_7 + 4z_7^2 + 3z_7^3 + 2z_7^4 + z_7^5$
$z = -6 - 5 * z_7 - 4 * z_7^2 - 3 * z_7^3 - 2 * z_7^4 - z_7^5$
(z_7 满足$x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 = 0$)

5、一般代数数域

```
Qx, x = QQ["x"] #有理数域上的通用多项式环
#定义于 x^4+x^3+2x+1 上的代数数域，多项式要不可约，a 是单位元
N, a = NumberField(x^4+x^3+2x+1, "a")
n1 = N([1, 2, 3, 4])
#或者
n2 = 1 + 2a + 3a^2 + 4a^3 + 5a^4
```

$n_1 = 1 + 2a + 3a^2 + 4a^3$
$n_2 = -4 - 8a + 3a^2 - a^3$
(这里实际上是带入 a^4 = -1 - 2a - a^3计算获得)
如果要定义代数整数环，多项式首项系数要是 1，其他项系数是整数。

6、多项式

```
Zx, x = ZZ["x"] #定义整系数多项式
a = x^127 + 1
b = x^7 + 1
c = mod(a, b)
p = 5
Rp = residue_field(ZZ, p) #定义模p环上多项式
Rx, x = Rp["x"]
a = x^127 + 1
b = x^7 + 1
c = mod(a, b)
```

7、两个特殊值

```
c1 = one(F) # F 下的 1
c1 = one(x) # F 下的元素 x 对应的 1
c0 = zero(F) # F 下的 0
c0 = zero(x) # F 下的元素 x 对应的 0
```

8、次数，范数

```
d = degree(F) #域 F 的次数
d = degree(x) # x 的次数
n = norm(x) # x 的范数
```

8、获得变量的类型

```
F, t = quadratic_field(-3)
x = F([1, 2])
parent(x)
parent(x)([1,1])
```

二、加减乘乘方按照通常运算进行

```
z = x + y
z = x - y
z = x * y
z = x^3
```

需要注意的是，两个操作数要在相同的域中

三、模运算

```
z = mod(x + y, 5)
z = mod(x * y, 5)
```

下面实现的是 x^p % n 模幂运算

```
function fpowermod(x, p , n)
    @assert p >= 0
    p == 0 && return one(x)
    b = x
    t = ZZ(prevpow(BigInt(2), BigInt(p)))
    r = one(x)
    while true
        if p >= t
            r = mod(r * b, n)
            p -= t
        end
        t >>= 1
        t <= 0 && break
        r = mod(r * r ,n)
    end
    return r
end
```

下面是二次域整数的共轭数模 n

```
function qfconjmod(x, n)
    re = coeff(x, 0)
    ir = mod(ZZ(-coeff(x, 1)), n)
    return parent(x)([re, ir])
end
```

四、各项系数

```
   x = F([1, 2])
   coefficients(x) #返回各项系数
   coeff(x, 0) #结果是 1
   coeff(x, 1) #结果是 2
   coeff(x, 2) #因为只有2项，大于等于 2 结果是 0
```