

15std::ops::*中的关联类型

2020年3月9日 12:25

Rust的操作符可以自己定义，利用实现std::ops::*里面的trait来实现。

这里的trait都有关联类型type Output;
因为这些trait是和符号相关的操作，所以利用符号的操作返回的是Output的类型，调用函数返回的是&Output类型。

比如std::ops::Index这个trait

```
impl Index<Range<usize>> for String {  
    type Output = str;  
    fn index(&self, index: Range<usize>) -> &str;  
}
```

```
let s = String::from("hades");
```

```
let a = s[0..1]; // a是一个Output的类型，也就是str类型
```

```
let b = s.index(0..1); // a是一个&Output的类型，也就是&str类型。
```

```
let c = &s[0..]; // c是一个&Output类型，也就是&str类型。
```