

18包和模块的关系

2020年3月9日 12:29

在Rust中有包的概念、crate的概念、模块的概念、路径的概念。

1、package：包，一个包的管理工具就是Cargo，它允许我们构建、测试和分享crate。

一个package中可以拥有一个或者多个的crate，Cargo.toml就是用来描述如何构建一个或者多个crate的文件。

注意：一个包中，最多只能有一个库项目crate。（通过Cargo.toml中的[lib] name = "", path = ""指定的）。

一个包可以带有零个或一个库crate和任意多个二进制crate。一个包中必须带有至少一个crate。

2、crate：是一个许多模块组成的属性结构，形成了库或者二进制的可运行的项目。

3、modules：模块，用来控制作用域的。模块存在于crate中。组成了crate。

4、路径：是一个命名，用::表示路径。

```
crate
├── sound
│   ├── instrument
│   │   └── woodwind
│   └── voice
└── nihao
    └── haha
```

```
mod sound {
    mod instrument {
        mod woodwind {
            fn clarinet() {
                // 函数体
            }
        }
    }

    mod voice {

    }
}

mod nihao {
    mod haha {

    }
}

fn main() {
```

```
}
```

注意： 这里的main函数和sound模块和nihao模块都定义在了一个叫做根模块的空间中了，
意义就在于，写进根模块空间的内容才能被编译器所编译，所以尽管sound模块和nihao模块
都不是pub的，但是main函数是能访问这两个模块的。

一般来说父亲模块是私有的，外部的模块是不能访问父亲模块里面的一切内容的，但是孩子模块能访问父亲模块的空间内容。
如果父亲模块是pub的，孩子模块是私有的，外部的模块只能访问到父亲模块的曾经，而不能访问到孩子模块空间。

注意use： use后面的名字并不是包名称，而是crate的名称。

比如Cargo.toml中，我们定义了hades的库，但是在[lib] name="hello"，那么我们应该使用use Hello::*，而不是
use hades::.

crate, self, super,

self:表示当前模块

super:父模块

crate: crate根

pub(crate): 可见性到当前的crate层,不能向外继续pub use了.

pub(in XXX):可见性只在特定的crate中,注意,这个的XXX是一个crate,而不是一个模块.

pub(self):可见性限制于当前模块,不能向外pub use.

pub(super):可见性是上一层.