

泛型 2

Generic Type Part II

Rust

```
trait Bar<T> {  
    fn get(&self) -> &T;  
}  
  
struct MyStruct<D> {  
    pub data: D,  
}  
  
impl<D> MyStruct<D> {  
    pub fn new(data: D) -> Self {  
        Self { data }  
    }  
}  
  
impl<D> Bar<D> for MyStruct<D> {  
    fn get(&self) -> &D {  
        &self.data  
    }  
}
```

```
trait Bar<T> {  
    fn get(&self) -> &T;  
}
```

2 implementations

```
struct MyStruct<D> {  
    pub data: D,  
}
```

```
impl<D> MyStruct<D> {  
    pub fn new(data: D) -> Self {  
        Self { data }  
    }  
}
```

```
impl<D> Bar<D> for MyStruct<D> {  
    fn get(&self) -> &D {  
        &self.data  
    }  
}
```

► Run | Debug

```
fn main() {  
    let _my_struct_i32: MyStruct<i32> = MyStruct::new(data: 10);  
    let _my_struct_str: MyStruct<&str> = MyStruct::new(data: "Hello, MWC 2023!");  
    let _my_struct_str: MyStruct<String> = MyStruct::new(data: "Hello, MWC 2023!".to_string());  
}
```