第一个rust程序

use std::io;  
use rand::Rng;  
use std::cmp::Ordering;  
  
fn main() {  
 println!("Hello, world!");  
 let secret\_number=rand::thread\_rng().gen\_range(1,101);  
 println!("神秘数字是: {}", secret\_number);  
  
 loop{  
 println!("猜测一个数");  
 let mut guess = String::new();  
  
 io::stdin().read\_line(&mut guess).expect("Failed to read line");  
 //隐藏同名的旧变量，用于类型转换  
 let guess:u32= match guess.trim().parse() {  
 Ok(num)=> num,  
 Err(\_)=> continue  
 };  
 //trim:清除空格和回车 u32是无符号整数  
 println!("你猜测的数是: {}", guess);  
 match guess.cmp(&secret\_number){  
 Ordering::Less=>{println!("太小啦")},  
 Ordering::Greater=>{println!("太大啦")},  
 Ordering::Equal=>{  
 println!("一样大");  
 break;  
 }  
 }  
 }//loop死循环  
}

在第一个rust程序中，了解rust的基本语法和结构，如何创建一个rust程序以及如何构建和运行，尝试实现了一个猜数字小游戏。

1. 顶部的use std::io等，是为导入模块，std是rust的标准库。
2. 接下来是fn main(){}，这是rust的main函数，主程序在这里执行，类似C语言/c++的main函数。
3. println！是一个宏，它会接受一个字符串，这被称为格式化字符串“{}”是占位符，他会调用一个来自display特征的方法，用于显示数据类型的可读形式输出。
4. rand::thread\_rng().gen\_range(1,101); rand 是 Rust 中一个常用的随机数生成库，gen\_range 是 ThreadRng 的一个方法，用于生成指定范围内的随机数。gen\_range(1, 101) 的含义是生成一个**半开区间** [1, 101) 的随机整数。
5. loop是一个死循环，里面是1猜数字逻辑，猜中了就break退出。
6. 通过 io::stdin()，程序可以访问用户输入的数据。expect 是 Rust 中用于处理 Result 类型的一个方法。read\_line 的返回值是一个 Result 类型，表示操作可能成功（Ok）或失败（Err）。这里输入的是字符串，需要转换为u32。
7. Match是一个模式匹配，它类似于C语言里的switch，在这个例子中，parse函数返回一个枚举类型，在这里接收这个枚举类型变量，判断是否正常得到了一个num数，如果成功了，就返回num，也就是赋值给guess。
8. cmp 是 Rust 中实现 PartialOrd 或 Ord trait 的类型的方法，用于比较两个值。
9. Ordering 是 Rust 标准库中的一个枚举类型，定义在 std::cmp 模块中。

Ordering::Less：表示左边的值小于右边的值。

Ordering::Greater：表示左边的值大于右边的值。

Ordering::Equal：表示两个值相等