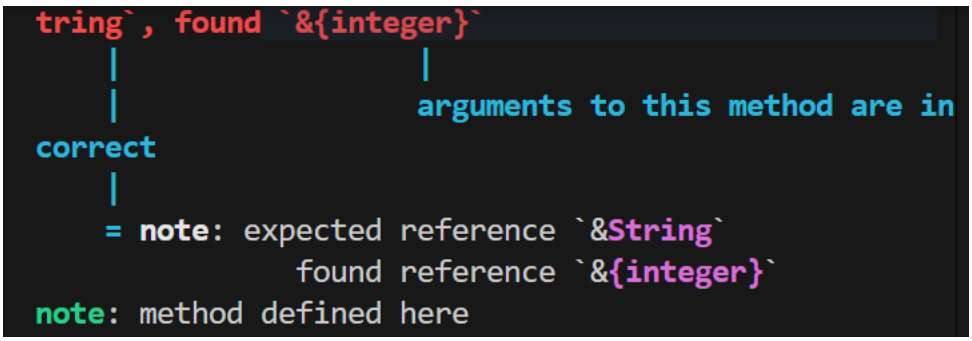
猜数游戏—比较猜测数字和神秘数字

Rust 是静态 强类型语言 还有类型推断的能力



此处展示了 match 结构中字符串和整数类型不匹配产生的报错

在前边重新定义 guess 变量 覆盖掉之前字符串的 guess 变成 u32 形式的 guess

```
11      let mut guess = String::new();
12
13      io::stdin().read_line(&mut guess).expect("无法读取行");
14      //定义了新变量来隐藏前边的变量guess 叫做shadow
15      let guess:u32 = guess.trim().parse().expect("Please type a number");
16      //trim表示去掉前后的空白 parse把字符串解析成数值类型的
17      //变量后边加:u32是指定变量类型 无符号证书类型
18      println!("你猜测的数是:{}",guess);
19
20      match guess.cmp(&secret_number){
21          Ordering::Less =>println!("Too small"),
22          Ordering::Greater => println!("To big"),
23          Ordering::Equal =>println!("You win!"),
24
25      }
```

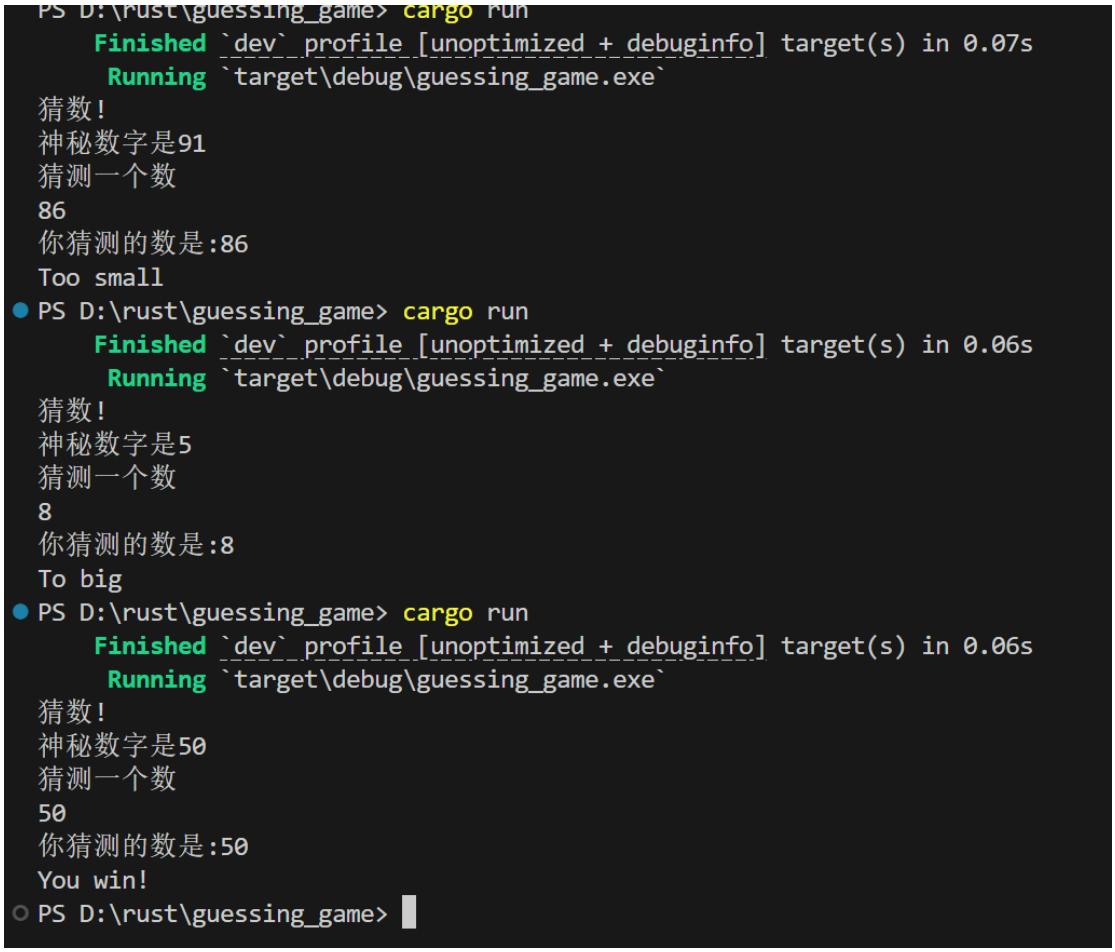
Match 表达式

Rust 中的 match 表达式有些类似其他语言中的 switch 和 case 语句，=>左边是需要匹配的模式，=>右边是待执行的代码。需要注意的是，**match 表达式必须枚举每一种可能**，所以一般在结尾使用通配符"_"来代表其他情况。

Trim 方法：裁边的一个功能

Parse 方法：在 Rust 中，parse 方法用于**将字符串解析为特定的数据类型**。这个方法通常是通过为类型实现 std::str::FromStr trait 来提供的。当字符串的格式符合预期的数据类型时，parse 方法能够**安全地转换字符串为该类型**。如果字符串不符合格式要求，parse 方法会返回一个 Result 类型，其中包含一个错误信息（Err），或者转换成功的值（Ok）。

成功运行


PS D:\rust\guessing_game> cargo run
Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.07s
Running `target\debug\guessing_game.exe`
猜数!
神秘数字是91
猜测一个数
86
你猜测的数是:86
Too small
PS D:\rust\guessing_game> cargo run
Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.06s
Running `target\debug\guessing_game.exe`
猜数!
神秘数字是5
猜测一个数
8
你猜测的数是:8
To big
PS D:\rust\guessing_game> cargo run
Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.06s
Running `target\debug\guessing_game.exe`
猜数!
神秘数字是50
猜测一个数
50
你猜测的数是:50
You win!
PS D:\rust\guessing_game>