



Rust 编程语言入门



Microsoft®
Most Valuable
Professional

杨旭，微软MVP
Rust、Go、C#开发者

12. 项目实例:

— 命令行程序

本章内容

- 12.1 接收命令行参数
- 12.2 读取文件
- 12.3 重构: 改进模块和错误处理
- 12.4 使用 TDD（测试驱动开发）开发库功能
- 12.5 使用环境变量
- 12.6 将错误消息写入标准错误而不是标准输出

标准输出 vs 标准错误

- 标准输出: stdout
 - println!
- 标准错误: stderr
 - eprintln!

二进制程序关注点分离的指导性原则

- 将程序拆分为 main.rs 和 lib.rs，将业务逻辑放入 lib.rs
- 当命令行解析逻辑较少时，将它放在 main.rs 也行
- 当命令行解析逻辑变复杂时，需要将它从 main.rs 提取到 lib.rs

经过上述拆分，留在 main 的功能有：

- 使用参数值调用命令行解析逻辑
- 进行其它配置
- 调用 lib.rs 中的 run 函数
- 处理 run 函数可能出现的错误

测试驱动开发

TDD (Test-Driven Development)

- 编写一个会失败的测试，运行该测试，确保它是按照预期的原因失败
- 编写或修改刚好足够的代码，让新测试通过
- 重构刚刚添加或修改的代码，确保测试会始终通过
- 返回步骤 1，继续

再见

