Rust 编程语言入门

杨旭，微软MVP
Rust、Go、C#开发者

11.10 集成测试

集成测试

- 在 Rust 里，集成测试完全位于被测试库的外部
- 目的：是测试被测试库的多个部分是否能正确的一起工作
- 集成测试的覆盖率很重要

tests 目录

- 创建集成测试：tests 目录
- tests 目录下的每个测试文件都是单独的一个 crate
  - 需要将被测试库导入
- 无需标注 #[cfg(test)]，tests 目录被特殊对待
  - 只有 cargo test，才会编译 tests 目录下的文件
- （例子）

运行指定的集成测试

- 运行一个特定的集成测试：cargo test 函数名
- 运行某个测试文件内的所有测试：cargo test --test 文件名
- （例子）

集成测试中的子模块

- tests 目录下每个文件被编译成单独的 crate
  - 这些文件不共享行为（与 src 下的文件规则不同）
- tests 目录下 子目录里的文件不会被编译为单独的 crate
- （例子）

针对 binary crate 的集成测试

- 如果项目是 binary crate，只含有 src/main.rs 没有 src/lib.rs：
  - 不能在 tests 目录下创建集成测试
  - 无法把 main.rs 的函数导入作用域
- 只有 library crate 才能暴露函数给其它 crate 用
- binary crate 意味着独立运行

再见