



Rust 编程语言入门



Microsoft®
Most Valuable
Professional

杨旭，微软MVP
Rust、Go、C#开发者

8.6 HashMap (下)

更新 HashMap<K, V>

- HashMap 大小可变
- 每个 K 同时只能对应一个 V
- 更新 HashMap 中的数据：
 - K 已经存在，对应一个 V
 - 替换现有的 V
 - 保留现有的 V，忽略新的 V
 - 合并现有的 V 和新的 V
 - K 不存在
 - 添加一对 K, V

替换现有的 V

- 如果向 HashMap 插入一对 KV，然后再插入同样的 K，但是不同的 V，那么原来的 V 会被替换掉
- （例子）

只在 K 不对应任何值的情况下，才插入 V

- entry 方法：检查指定的 K 是否对应一个 V
 - 参数为 K
 - 返回 enum Entry：代表值是否存在
- Entry 的 or_insert() 方法：
 - 返回：
 - 如果 K 存在，返回到对应的 V 的一个可变引用
 - 如果 K 不存在，将方法参数作为 K 的新值插进去，返回到这个值的可变引用
- （例子）

基于现有 V 来更新 V

- （例子）

Hash 函数

- 默认情况下，HashMap 使用加密功能强大的 Hash 函数，可以抵抗拒绝服务（DoS）攻击。
 - 不是可用的最快的 Hash 算法
 - 但具有更好安全性。
- 可以指定不同的 hasher 来切换到另一个函数。
 - hasher 是实现 BuildHasher trait 的类型

再见

