Rust 编程语言入门

杨旭，微软MVP
Rust、Go、C#开发者

8.5 HashMap（上）

HashMap<K, V>

- 键值对的形式存储数据，一个键（Key）对应一个值（Value）
- Hash 函数：决定如何在内存中存放 K 和 V
- 适用场景：通过 K（任何类型）来寻找数据，而不是通过索引

创建 HashMap

- 创建空 HashMap：new() 函数
- 添加数据：insert() 方法
- （例子）

HashMap

- HashMap 用的较少，不在 Prelude 中
- 标准库对其支持较少，没有内置的宏来创建 HashMap
- 数据存储在 heap 上
- 同构的。一个 HashMap 中：
  - 所有的 K 必须是同一种类型
  - 所有的 V 必须是同一种类型

另一种创建 HashMap 的方式：collect 方法

- 在元素类型为 Tuple 的 Vector 上使用 collect 方法，可以组建一个 HashMap：
  - 要求 Tuple 有两个值：一个作为 K，一个作为 V
  - collect 方法可以把数据整合成很多种集合类型，包括 HashMap
    - 返回值需要显式指明类型
  - （例子）

HashMap 和所有权

- 对于实现了 Copy trait 的类型（例如 i32），值会被复制到 HashMap 中
- 对于拥有所有权的值（例如 String），值会被移动，所有权会转移给 HashMap
  - （例子）
- 如果将值的引用插入到 HashMap，值本身不会移动
  - 在 HashMap 有效的期间，被引用的值必须保持有效

访问 HashMap 中的值

- get 方法
  - 参数：K
  - 返回：Option<&V>
- （例子）

遍历 HashMap

- for 循环
- （例子）

再见