Rust 编程语言入门

杨旭，微软MVP
Rust、Go、C#开发者

3.3 复合类型

- 复合类型可以将多个值放在一个类型里。
- Rust 提供了两种基础的复合类型：元组（Tuple ）、数组

Tuple

- Tuple 可以将多个类型的多个值放在一个类型里
- Tuple 的长度是固定的：一旦声明就无法改变

创建 Tuple

- 在小括号里，将值用逗号分开
- Tuple 中的每个位置都对应一个类型，Tuple 中各元素的类型不必相同
- （例子）

获取 Tuple 的元素值

- 可以使用模式匹配来解构（destructure）一个 Tuple 来获取元素的值
- （例子）

访问 Tuple 的元素

- 在 tuple 变量使用点标记法，后接元素的索引号
- （例子）

数组

- 数组也可以将多个值放在一个类型里
- 数组中每个元素的类型必须相同
- 数组的长度也是固定的

声明一个数组

- 在中括号里，各值用逗号分开

数组的用处

- 如果想让你的数据存放在 stack（栈）上而不是 heap（堆）上，或者想保证有固定数量的元素，这时使用数组更有好处
- 数组没有 Vector 灵活（以后再讲）。
  - Vector 和数组类似，它由标准库提供
  - Vector 的长度可以改变
  - 如果你不确定应该用数组还是 Vector，那么估计你应该用 Vector。
- （例子）

数组的类型

- 数组的类型以这种形式表示：[类型；长度]
  - 例如： let a: [i32; 5] = [1, 2, 3, 4, 5];

另一种声明数组的方法

- 如果数组的每个元素值都相同，那么可以在：
  - 在中括号里指定初始值
  - 然后是一个“；”
  - 最后是数组的长度
- 例如： let a = [3; 5]; 它就相当于：let a = [3, 3, 3, 3, 3];

访问数组的元素

- 数组是 Stack 上分配的单个块的内存
- 可以使用索引来访问数组的元素（例子）
- 如果访问的索引超出了数组的范围，那么：
  - 编译会通过
  - 运行会报错（runtime 时会 panic）
    - Rust 不会允许其继续访问相应地址的内存

  - （例子）

再见