درس ۹

Rust

Vec · HashMap · iter

main.rs

```
let top: Vec<_> = scores.iter()
  .filter(|s| **s > 90)
  .collect();
```

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی Rust

درس ۹

collection ها، closure ها و iterator ها

Vec، HashMap، closure ها، adapter های iterator و collect



Vec

Vec یک آرایهٔ قابل رشد روی heap است. macro vec! یکی را با مقدارها می‌سازد. push به آخر ...

وقتی یک تابع فقط لازم دارد بخواند، Vec را به عنوان یک slice قرض بدهید.

main.rs

```rust
let mut scores = vec![90, 85, 77];
scores.push(92);
scores.len();                    // 4
scores[0];                       // 90
scores.get(10);                  // None, no panic
scores.sort();
scores.retain(|&s| s >= 80);     // keep 85, 90, 92
```

01

HashMap

HashMap جفت‌های کلید-مقدار را نگه می‌دارد. API entry شمردن و گروه‌بندی را یک سطر می‌سازد: ...

ترتیب پیمایش تعریف نشده است. وقتی کلیدهای مرتب لازم دارید از BTreeMap استفاده کنید.

main.rs

```rust
use std::collections::HashMap;

let mut counts: HashMap<&str, u32> = HashMap::new();
for w in text.split_whitespace() {
    *counts.entry(w).or_insert(0) += 1;
}
if let Some(n) = counts.get("kabul") { println!("{n}"); }
```

02

closure ها

یک closure یک تابع بی‌نام است که پارامترهای آن بین دو خط عمودی نوشته می‌شوند. می‌تواند ...

closure، move را مجبور می‌کند مالکیت آنچه را می‌گیرد بگیرد، که برای thread ها لازم دارید.

main.rs

```rust
let bonus = 5;
let add_bonus = |s: i32| s + bonus;       // captures bonus
add_bonus(90);                            // 95

let mut count = 0;
let mut inc = || count += 1;              // FnMut
inc(); inc();

let owned = move || println!("{bonus}");
```

03

adapter های iterator

scores.iter() — iterator

filter — |s| **s >= 80

map — |s| s * 2



04

یکجا کردن همه

اینجا رکوردهای شاگردان را می‌خوانیم، آن‌هایی را که کامیاب شدند نگه می‌داریم، اوسط نمره‌ها ...

into_iter یک collection را مصرف می‌کند، iter آن را قرض می‌گیرد، و iter_mut آن را به شکل ...

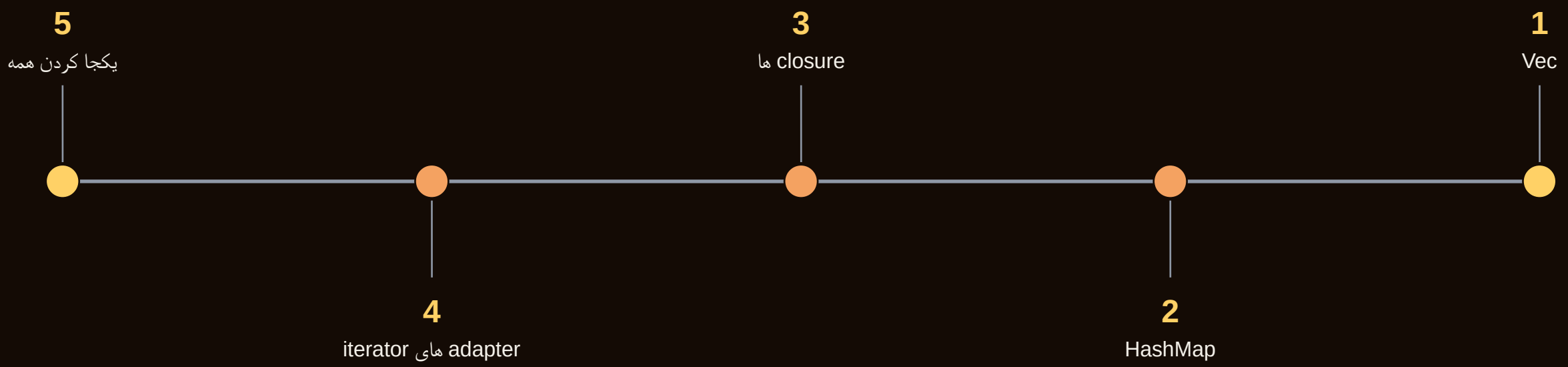
RUST PROGRAMMING · 9

main.rs

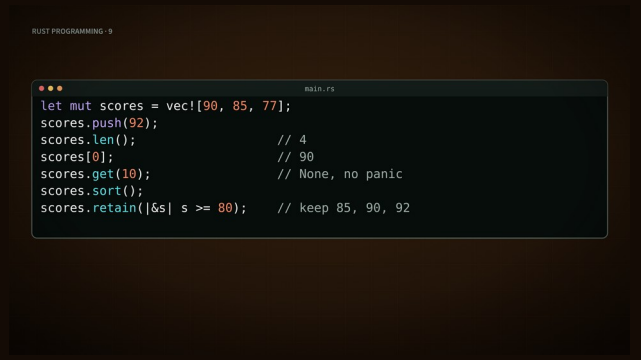
```rust
let mut by_city: BTreeMap<&str, Vec<f64>> = BTreeMap::new();
for s in students.iter().filter(|s| s.score >= 50.0) {
    by_city.entry(&s.city).or_default().push(s.score);
}
for (city, v) in &by_city {
    let avg = v.iter().sum::<f64>() / v.len() as f64;
    println!("{city:<10} {avg:.1}");
}
```

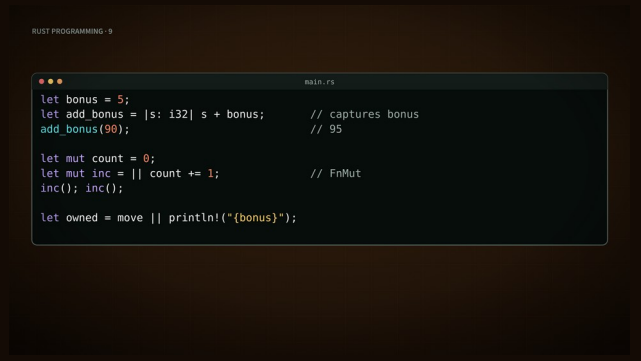
05

نقشهٔ درس


5
یکجا کردن همه
4
iterator های adapter
3
ها closure
2
HashMap
1
Vec

منابع و مطالعهٔ بیشتر





تشکر از تماشای شما

MyWebUniversity.com

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری

@MyWebUniversity-Dari · @MyWebUniversity