Kolekcije, šabloni, argumenti komandne linije, rukovanje datotekama

Paralelne i distribuirane arhitekture i jezici
Zimski semestar, školska 2025./26.

Branislav Ristić

Vec

- Linearna struktura.
- `Vec::new()` za prazan vektor i makro `vec![]` za inicijalizaciju sa vrednostima.
  - Rust zaključuje tip na osnovu početnih vrednosti, bez potrebe za eksplicitnom anotacijom.
- Indeksiranje van opsega izaziva paniku, dok `get` vraća `None`.
  - Omogućava bezbednije rukovanje greškama.
- Koristiti enumeracije za vektore sa mešotivitm tipovima.
- Pravila vlasništva važe u okviru vektora.
- Primer:
  - *01_vector.rs*
  - *02_vector.rs*
  - *03_vector.rs*
  - *04_vector.rs*

HashMap

- Nelinearna struktura.
  - Ključ, par vrednosti.
- `HashMap::new()` za kreiranje nove mape.
  - Ne postoji macro.
- Omogućena podrška za iteriranje.
- Pravila vlasništva važe u okviru `HashMap`-a.
- Primer:
  - *05_hashmap.rs*
  - *06_hashmap.rs*
  - *07_hashmap.rs*
  - *08_hashmap.rs*

Pattern matching

- Šabloni (engl. *patterns*) predstavljaju specijalnu sintaksu u okviru Rust-a.
- Pomoću match izraza moguće je prilagoditi kontrolu toka.
- Šabloni se sastoje iz kombinacija:
  - Literala
  - Destruktuiranih nizova, enuma, struktura ili torki
  - Varijabli
  - Džoker karaktera
  - Placeholder-a
    - Imenovani džoker karakteri
- Kako bi se koristio šablon, on se poredi sa nekom vrednošću.

Pattern matching - Upotreba

- Šabloni nalaze primenu u sledećim slučajevima:
  - `match` ruke
  - `if let`
  - `while let`
  - `for` petlje
  - `let` iskaz
  - parametri funkcije
- Primer:
  - *10_patterns.rs*

Pattern matching - Refutability

- Šabloni mogu biti:
  - Opovrgljivi (engl. *refutable*),
    - šablon nije uvek u mogućnosti da se poklopi.
  - Neopovrgljivi (engl. *irrefutable*),
    - šablon je uvek u mogućnosti da se poklopi,
- Primer:
  - *11_patterns.rs*

Pattern matching - Upotreba

- Šabloni mogu biti nalaze široku primenu u Rust programskom jeziku.
- Primer:
  - *12_patterns.rs*

Argumenti komandne linije

- Koristi se `std::env::args` za pristup argumentima komandne linije.
- `args()` vraća iterator koji sadrži argumente prosleđene programu.
  - Pretvaranje iteratora u kolekciju metodom `.collect()`.
- Prvi argument predstavlja ime ili putanju programa.
- Sledeći argumenti su ulazi koje je korisnik prosledio.

Rukovanje datotekama

- Modul `std::fs` omogućava interakciju sa fajlovima.
- Zajedničke funkcije:
  - `fs::read_to_string`: Čita ceo sadržaj fajla u `String`.
  - `fs::write`: Piše `String` ili bajtove u fajl, prepisujući postojeći sadržaj.
  - `fs::append`: Dodaje sadržaj postojećem fajlu koristeći opcije.
- Koristiti `expect` ili `unwrap` za obradu potencijalnih grešaka.
- Primer:
  - *13_minigrep.rs*

Zadatak

- Detaljno proučiti poglavlje 12 dostupno na *rust-book.cs.brown.edu*

Izvori

- Rust Community. “The Rust Programming Language - the Rust Programming Language.” Rust-Lang.org, 2018, doc.rust-lang.org/book/.
- Crichton, Will. “Experiment Introduction - the Rust Programming Language.” Brown.edu, rust-book.cs.brown.edu/.
- Rust Community. “Tour of Rust - Let’s Go on an Adventure!” Tourofrust.com, tourofrust.com/.
- Rust Team. “Rust Programming Language.” Rust-Lang.org, 2018, www.rust-lang.org/.

Kolekcije, šabloni, argumenti komandne linije, rukovanje datotekama

Paralelne i distribuirane arhitekture i jezici
Zimski semestar, školska 2025./26.

Branislav Ristić