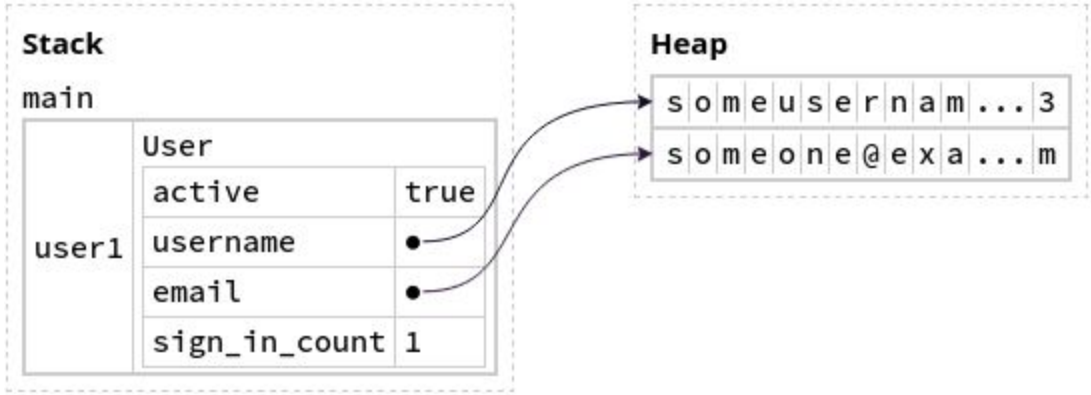
Strukture i enumeracije

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

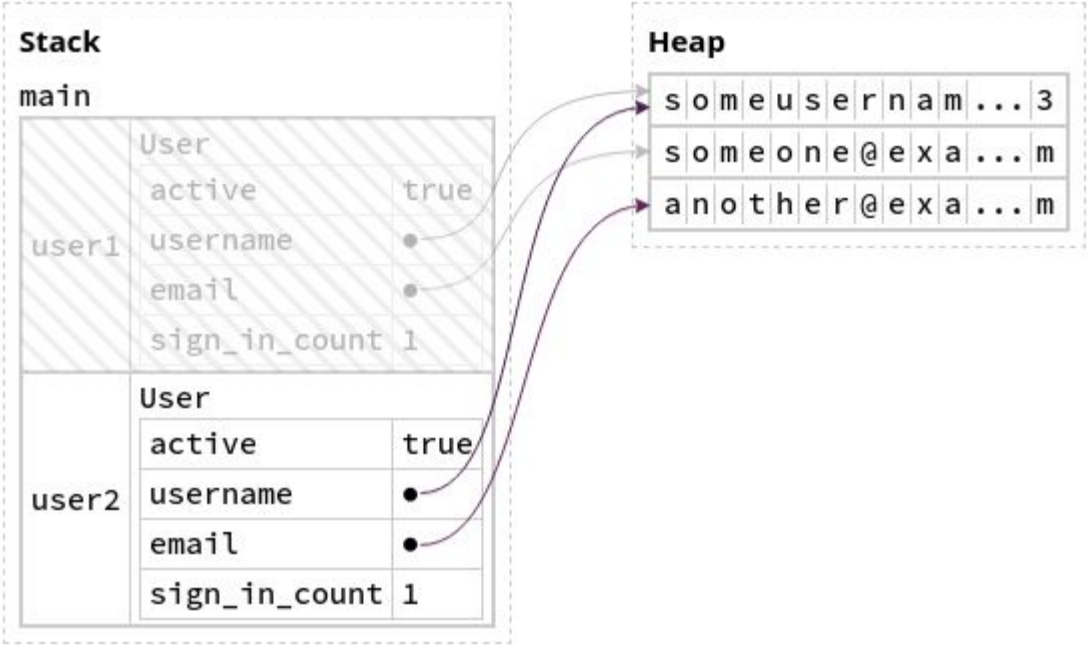
Strukture

- Strukture imaju više polja, povezanih u celinu.
- Slično torkama, ali sa imenovanim poljima.
- Polja mogu biti različiti tipovi podataka.
- Jasnije i fleksibilnije od torki.
- Ključna reč `struct`.
- Pristup putem tačka operatora (`.`).
- Mutabilnost na nivou strukture:
  - Nije moguće postaviti određena polja kao nepromenljiva.
- Primer:
  - *01_structure.rs*

Strukture

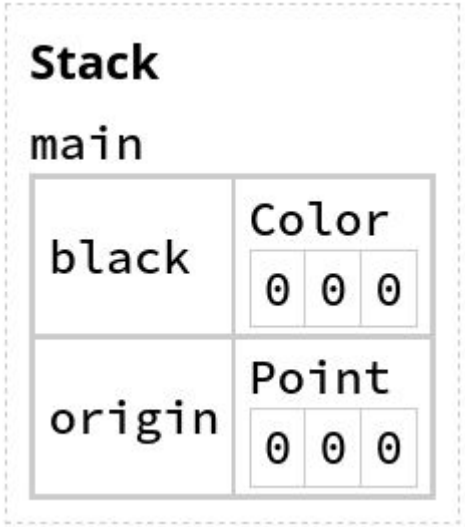

Stack
main
user1
User
active
true
username
email
sign_in_count
1
Heap
someusernam...3
someone@exa...m

Strukture


Stack
main
user1
User
active
true
username
email
sign_in_count
1
user2
User
active
true
username
email
sign_in_count
1
Heap
someusernam...3
someone@exa...m
another@exa...m

Struktura torke

- Struktura torke omogućava grupisanje vrednosti bez imenovanja polja.
- Kreira se zaseban tip, za razliku od standardnih torki.
- Pristup preko indeksa:
  - Kao i kod torki.
- Korisno za razaznavanje sličnih podataka.
- Primer:
  - *02_tuple_structure.rs*



Jediničnolika struktura bez polja

- Struktura bez polja, slično ().
- Korisno za nešto što ne zahteva podatke.
- Instance dele tip ali ne sadrže vrednosti.
- Primer:
  - *03_unit_structure.rs*

Vlasništvo u strukturama

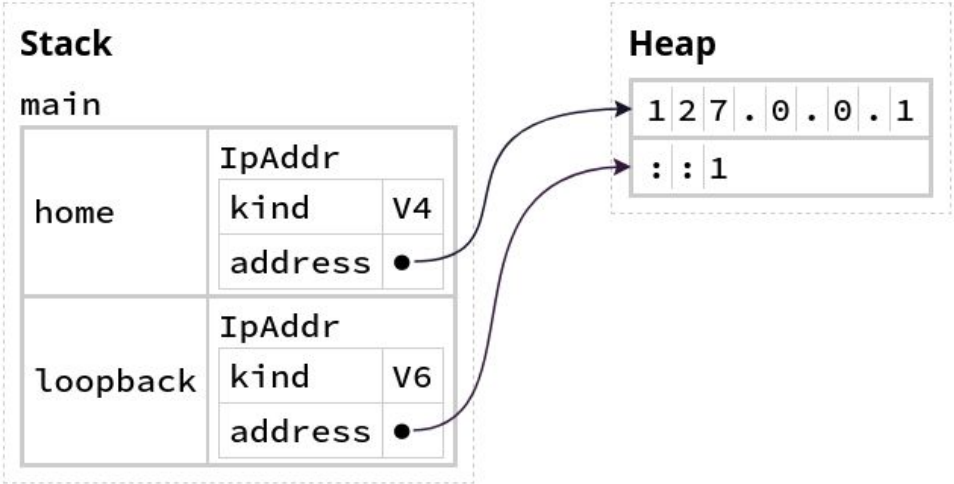
- Rust strukture obično koriste tipove sa vlasništvom kao što je String.
  - Umesto pozajmljenih tipova kao što je &str.
- Ovaj pristup obezbeđuje da svaka instanca strukture poseduje svoje podatke.
  - Podaci ostanu važeći dokle god je i struktura važeća.
- Strukture mogu čuvati reference na podatke koji su u vlasništvu nekog drugog.
  - Za to su potrebni eksplicitni životni vekovi (lifetimes).
- Životni vekovi obezbeđuju da podaci na koje struktura referiše ostanu važeći koliko i sama struktura.
- Primer:
  - *04_struct_string_slices.rs*

Pozajmljivanje polja u strukturama

- Vodi se računa o vlasništvu i pozajmljivanju na nivou:
  - Strukture,
  - Polja.
- Ukoliko se pozajmi polje strukture:
  - Suštinski je pozajmljeno:
    - Polje,
    - Sama struktura, ali ne i njena ostala polja.
- Primer:
  - *05_struct_field_borrowing.rs*

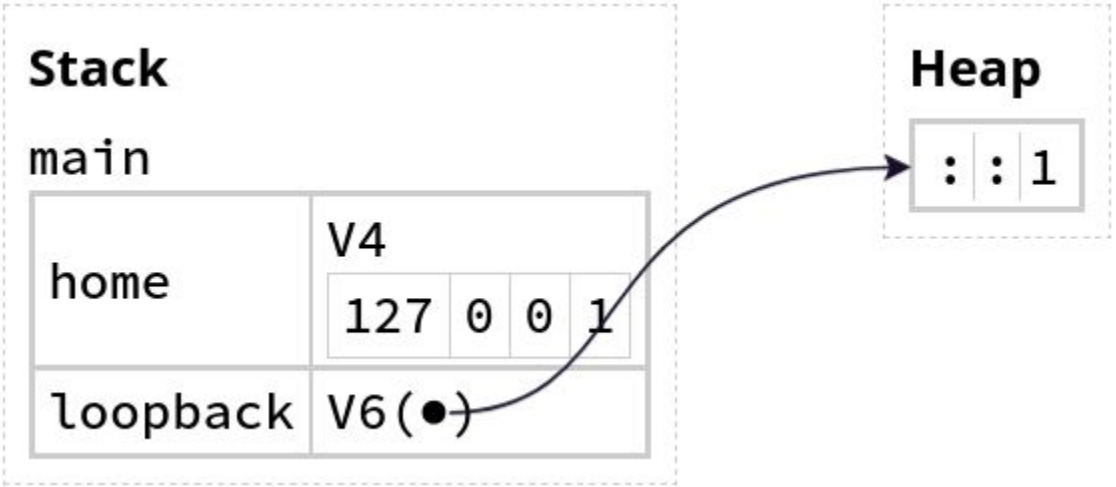
Enumeracije

- Ukoliko se vrednost promenljive uzima iz nekog konačnog skupa vrednosti:
  - Koji nije nepraktično nabrojati.
    - Koristiti enumeracije.
- Primer:
  - *06_enum.rs*



Enumeracije

- Primer:
  - *07_enum_tuple.rs*
  - *07a_enum_copy.rs*



Enumeracije

- Primer:
  - *08_enum_multiple.rs*

Generički tipovi u definicijama funkcija

- Generički tipovi omogućavaju fleksibilnost u parametrima i povratnim vrednostima.
  - Smanjuju potrebu za ponavljanjem koda.
- Generički tipovi se dodaju unutar uglastih zagrada <T> posle imena funkcije
  - T se koristi kao oznaka za tip.
- Omogućava definisanje funkcija koje mogu da rade sa bilo kojim tipom T.
- Da bi se omogućile određene operacije, kao što su poređenja, mogu se dodati osobine (traits).
  - PartialOrd.
- Primer:
  - *09_generic_intro.rs*
  - *10_generic_func.rs*

Generički tipovi u definicijama struktura i metoda

- Moguće je napraviti strukturu koja u sebi sadrži proizvoljan tip.
- Sintaksa `StructureName<T>`.
- Primer:
  - *11_generic_struct.rs*
  - *12_generic_struct_multiple.rs*
  - *13_generic_struct_multiple.rs*

Generički tipovi u enumeracijama

- Moguće je (i veoma je često) da enumeracija u sebi sadrži podatak
- Neki od dominantnih enumeracija:
  - `Option<T>`
  - `Result<T, E>`
- Primer:
  - *14_generic_enums.rs*

Performanse generičkih tipova?

- Monomorfizam:
  - Za svaki generički tip se ponaosob generiše implementacija.

```
enum Option_i32 {
    Some(i32),
    None,
}

enum Option_f64 {
    Some(f64),
    None,
}

fn main() {
    let integer = Option_i32::Some(5);
    let float = Option_f64::Some(5.0);
}
```

Pattern matching

- `if let`
- Primeri:
  - *15_pattern_matching.rs*
  - *16_pattern_matching.rs*
  - *17_pattern_matching_ownership.rs*
  - *18_if_let.rs*

Zadatak

- Detaljno proučiti poglavlje o vlasništvu 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/.

Strukture i enumeracije

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