



Products

Passing lifetime to a struct

log in sign up

Ask Question

Asked 1 year, 3 months ago

Active 1 year, 3 months ago

Viewed 44 times



0



I have code like this:

```
pub trait Action {
    fn execute(&self);
}

struct AddAction<'a> {
    rp_n_calculator: &'a RpnCalculator
}

struct DeductAction<'a> {
    rp_n_calculator: &'a RpnCalculator
}

impl Action for DeductAction<'_> {
    fn execute(&self) {
        // ...
    }
}

impl Action for AddAction<'_> {
    fn execute(&self) {
        // ...
    }
}

impl<'a> RpnCalculator {
    fn actions(&self) -> Vec<Box<dyn Action + 'a>> {
        let mut actions: Vec<Box<dyn Action + 'a>> = vec![
            Box::new(AddAction { rp_n_calculator: &self }),
            Box::new(AddAction { rp_n_calculator: &self }),
            // ...
        ];
        // ...
        actions
    }
}
```

The intention of my code is that `RpnCalculator.actions()` should create some instances of some structs that implement trait `Action` and return a vector containing those instances. Those structs have a property `rp_n_calculator` which is a reference to a `RpnCalculator`. The `RpnCalculator.actions()` should put `self` (the `RpnCalculator` that creates it) into this reference.

Now the error I get is "cannot infer the appropriate lifetime". I get this error in the line where I create an instance that I add to the vector:

```
Box::new(AddAction { rp_n_calculator: &self })
```

For that reason I have `'a` in the vector declaration, but it still doesn't work.

rust reference traits lifetime

Share

Improve this question

Follow

asked Aug 30 '20 at 20:21



Damian

126 • 10

1

Does this answer your question? [How to infer an appropriate lifetime for an implementation using lifetimes for structs and impl?](#)

– hamid

Aug 30 '20 at 22:27

Add a comment

1 Answer

Active Oldest Votes



1

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Customize settings





You should probably use `fn actions(&a self) { ... }` because the `lifetime 'a` you use in `dyn Action + 'a` is related to the lifetime of the `RpnCalculator`.

[Share](#)

[Improve this answer](#)

[Follow](#)

answered Aug 30 '20 at 21:13



[prog-fh](#)

8,988 ● 1 ● 5 ● 24

[Add a comment](#)

Your Answer

Post Your Answer

By clicking "Post Your Answer", you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#).

Not the answer you're looking for? Browse other questions tagged [rust](#) [reference](#) [traits](#) [lifetime](#) or [ask your own question](#).

The Overflow Blog

- [Sequencing your DNA with a USB dongle and open source code](#)
- [Don't push that button: Exploring the software that flies SpaceX rockets and...](#)

Featured on Meta

- [Providing a JavaScript API for userscripts](#)
- [Congratulations to the 59 sites that just left Beta](#)

Linked

[How to infer an appropriate lifetime for an implementation using lifetimes for structs and impl?](#)

Related

[What is the lifetime of a static variable in a C++ function?](#)

[Construct a Vector of any Type with a Trait Constraint](#)

[Why can't I store a value and a reference to that value in the same struct?](#)

[Trait to store structs with different generic parameters](#)

[How to create a struct with a vector of references?](#)

[Having trouble understanding traits and generics in Rust](#)

[Cannot infer a lifetime for a closure returning a boxed trait that contains a reference](#)

[Store data that implements a trait in a vector](#)

Hot Network Questions

- [Have there been no deaths due to omicron in Africa?](#)
- [Bit Rot within LUKSEncryption](#)
- [Why did the JWST solar array deploy early?](#)
- [Why couldn't Smith absorb Neo in The Matrix Reloaded?](#)
- [Is the science in "Don't Look Up" realistic?](#)

[more hot questions](#)

[Question feed](#)

STACK OVERFLOW

- Questions
- Jobs
- Developer Jobs Directory
- Salary Calculator
- Help
- Mobile

PRODUCTS

- Teams
- Talent
- Advertising
- Enterprise

COMPANY

- About
- Press
- Work Here
- Legal
- Privacy Policy
- Terms of Service
- Contact Us
- Cookie Settings
- Cookie Policy

STACK EXCHANGE NETWORK

- Technology
- Culture & recreation
- Life & arts
- Science
- Professional
- Business
- API
- Data

- Blog
- Facebook
- Twitter
- LinkedIn
- Instagram