Санкт-Петербургский государственный университет
Кафедра системного программирования

Умные указатели

Дмитриевцев Алексей Сергеевич

Санкт-Петербург
2024

Ссылки

- Общая (&)
- Мутабельная (&mut)

Умные указатели

- Хранение на куче
- Несколько владельцев

Box<T>

Хранит данные на куче

- не можем определить размер во время компиляции
- передавать владение без копирования

Примеры

```
fn main() {
    let box = Box::new(5);
    println!("box = {box}");
}
```

Рекурсивные типы

Классический пример — cons list

```
enum List {
    Cons(i32, List),
    Nil,
}
```

Что не так с этой реализацией?

Рекурсивные типы

Можем ли мы узнать размер на этапе компиляции?

Как нам поможет Box?

Давайте попробуем починить этот пример

Box итоги

- Не требует накладных расходов
- Перенаправление и выделение на куче
- Других специальных возможностей нет

Типаж Deref

Что мы хотим?

- Умный указатель -> обычная ссылка
- &T1 -> &T2
- &mut T1 -> &mut T2
- &mut T1 -> &T2
- &T1 -> &mut T2

Типаж Drop

Что происходит с умным указателем, когда значение выходит из области видимости?

Типаж Deref

Мутабельная ссылка ВСЕГДА ОДНА, поэтому

&mut T1 -> &T2 Ok

&T2 -> &mut T1 Net

Rc

Хотим чтобы у одного значения было несколько владельцев, неизменяемое владение

Reference Counting — счетчик ссылок

Давайте посмотрим на примерах

Сказать про многопоточку

RefCell

Похож на Box, но проверяется во время выполнения. У значения единственный владелец, владения изменяемые и неизменяемые

Моки, деревья, объединение с Rc

Ссылочные циклы

Возможно ли в Rust?

Слабые и сильные ссылки

Weak<T>

Пример

```
struct Node {
    value: i32,
    parent: RefCell<Weak<Node>>,
    children: RefCell<Vec<Rc<Node>>>,
}
```

Итог

Box, Rc, RefCell, Weak

Можно сделать свои!!

А теперь немного функциональщины, если успеем

Ваши отзывы