ORGANIZER: GFI VBI | POWERED BY: WEB3HACKFEST | DEV PARTNER: CODE mely VU NGUYEN {C<>[)≡R}

RUST DEVELOPER BOOTCAMP

19:30 - 21:00 | 03/07/2023

Discord, Zoom Online

Outline khoá học

Bài giảng	Bài 1: Cơ bản về Rust	8:30PM ngày 03/07	Dụng, Liên
Bài tập	Hàm , Câu điều kiện, Kiểu String và &str Dạng bài: Làm quen cú pháp, Check lỗi, viết thêm logic		Dụng, Liên
Bài giảng	Bài 2: Ownership và Borrowing trong Rust	8:30PM ngày 06/07	Dụng, Liên
Bài tập	Ownership , Borrowing, Lifetime Dạng bài: Check lỗi		Dụng, Liên
Bài giảng	Chủ đề nâng cao 1 - Zero-knowledge Proofs	8:30PM ngày 08/07	Vũ
Bài tập	Rust coding challenge 1		Vũ
Bài giảng	Bài 3: Các kiểu dữ liệu phức tạp và Kiểu Generic	8:30PM ngày 10/07	Dụng, Liên
Bài tập	Struct, Enums , Generic type Dạng bài: Check lỗi , viết thêm logic		Dụng, Liên
Bài giảng	Bài 4: Traits trong Rust	8:30PM ngày 13/07	Dụng, Liên
Bài tập	Traits Dạng bài: Check lỗi , viết thêm logic		Dụng, Liên
Bài giảng	Chủ đề nâng cao 2 - Zero-knowledge Proofs	8:30PM ngày 15/07	Vũ
Bài tập	Rust coding challenge 2		Vũ
Bài giảng	Bài 5: Xử lý lỗi và Macro	8:30PM ngày 17/07	Dụng, Liên
Bài tập	Xử lý lỗi , Macro Dạng bài: Check lỗi, viết logic		Dụng, Liên
	Kết thúc lập trình rust cơ bản		
Bài giảng	Blockchain: nền tảng và cơ hội dành cho developers (Thạnh) Interview with Dụng Interview with Vũ	7PM30 ngày 19/07	Thạnh, Vũ, Dụng và Liên

Lưu ý khoá học

- Làm bài tập đầy đủ
- Chuẩn bị câu hỏi trước bài giảng nếu có, hoặc có thể trao đổi hoặc đặt câu hỏi
- Tham gia đủ các buổi học

=> Đánh giá Kết quả cuối khoá

Basic of Rust

VBI Academy

1. Why Rust?

- Phát triển bởi Mozilla Research, 2015
- Open-source
- Đặc điểm nổi bật so với các ngôn ngữ khác :
  - An toàn và tin cậy
  - Hiệu năng cao
  - Đa luồng an toàn
- Được đánh giá là ngôn ngữ được yêu thích nhất trên Stack Overflow 2022.

1. Why Rust - Sự phổ biến rộng rãi của Rust

- Được sử dụng trong các công ty công nghệ lớn: Facebook, Microsoft, Google, Amazon, Cloudflare,...
- Các ứng dụng lớn viết bằng Rust như Dropbox, Figma, Discord, Facebook, …
- Phát triển nhiều trong phần hệ thống cần có tốc độ xử lý cao: nhân hệ điều hành, trình duyệt web, **blockchain, AI-Machine Learning**...

1. Why Rust - Sự phổ biến rộng rãi của Rust

- Rust còn được sử dụng vào việc xây dựng các dịch vụ Backend, ứng dụng web,..
- Ngoài ra nó còn có thể port sang các nền tảng khác nhau: Windows, Linux, Android, iOS... **tạo cơ hội cho việc sử dụng Rust để xây dựng nhiều ứng dụng trên các nền tảng khác nhau**

https://github.com/rust-unofficial/awesome-rust

2. Cài đặt môi trường

Link cài đặt: https://www.rust-lang.org/tools/install

Rustup cài đặt/quản lý các phiên bản rust & các công cụ hỗ trợ khác

+ Rustc: Trình biên dịch Rust
+ Bộ cài đặt toolchain (để build ra file thực thi trên nhiều nền tảng khác nhau: Linux, windows, macos, android, embedded devices.
+ Cargo: quản lý project : tạo, run, build project và quản lý các gói thư viện,...

3. Tạo project Hello World

+ Tạo project : cargo new hello-world
+ Run project: cargo run
+ Build project: cargo build / cargo build --release

4. Cài đặt các extensions cần thiết cho Rust đối với VS Code

+ Rust-analyzer
+ Better TOML
+ Crates
+ Error Lens
+ CodeLLDB

Link tham khảo:

+ https://www.youtube.com/watch?v=x_iZEK6Rww4

+ https://www.youtube.com/watch?v=XiS7MB23NoE

5. Kiểu dữ liệu

Có 2 kiểu dữ liệu:

+ Scalar: lưu trữ đơn giá trị

  Ví dụ: Integer,Float , Char, Boolean

+ Compound: lưu trữ đa giá trị

  Ví dụ: Array, Tuple

Ví dụ bằng code

6. Biến

Biến là đại diện cho giá trị

+ Định dạng tên biến kiểu snake_case
+ Mặc định là kiểu immutable - không thay đổi giá trị
+ “mut” : mutable thay đổi giá trị của biến

Ví dụ bằng code

Quy ước đặt tên:
https://doc.rust-lang.org/1.0.0/style/style/naming/README.html

7. String và &str

String

+ Lưu trữ 1 chuỗi các ký tự, có kích thước động
+ let mut name = String::from(“Hello World”);

Ví dụ bằng code

&str: String slice

+ let str1 = “Hello world”;
+ let str2 = &str1[..];
+ Chỉ có quyền đọc

8. Câu điều kiện

if - else

if - else if

match

Ví dụ bằng code

9. Vòng lặp

Loop

While

For

Ví dụ bằng code

10. Hàm

+ Tập hợp các đoạn code để thực hiện một logic, nhiệm vụ nào đó
+ Các lệnh code sẽ thực hiện từ trên xuống dưới
+ Hàm có tham số, hàm không có tham số, hàm có giá trị trả về, hàm không trả về (void), hàm lồng hàm

Tại vì sao phải sử dụng Function:

+ Chức năng lớn -> chia ra các chức năng nhỏ hơn -> dễ kiểm soát logic
+ Có thể sử dụng lại cho các mục đích khác nhau
+ Thường các thư viện sẽ định nghĩa các function -> 1 mục đích nhất định -> developer có thể sử dụng (https://doc.rust-lang.org/std/index.html)

10. Hàm

```rust
fn function_with_param(x: u32, y: u32) -> u32{
    x+y
}
```

```rust
fn function_without_param() {
    println!("Hello World");
}
```

10. Hàm

Một số lưu ý khi sử dụng hàm

+ Tham số hàm định nghĩa kiểu dữ liệu gì, thì khi sử dụng lại hàm đó phải đúng kiểu dữ liệu của tham số
+ Đối với hàm có trả về, thì kết thúc hàm phải trả về đúng kiểu dữ liệu mà hàm đã định nghĩa
+ Có 2 cách để trả giá trị hàm : **biểu thức hoặc biến ko có dấu chấm phẩy** ở đoạn cuối của hàm hoặc dùng từ khoá **return**

11. Closure

+ Hàm bí danh (hàm không có tên)
+ Sử dụng ||
+ Ví dụ

```
fn main() {
    let my_closure = || println!("This
is a closure");
    my_closure();
}
```

Một số đặc trưng của Rust

Zero Cost Abstraction (Generics , Iterators, Collections)

+ No runtime cost, compile cost

```rust
fn main() {
    let numbers = vec![5, 2, 8, 1, 3];

    // Using iterator abstraction to find the maximum element
    let max_element = numbers.iter().max();

    match max_element {
        Some(&max) => println!("Maximum element: {}", max),
        None => println!("The vector is empty."),
    }
}
```

Một số đặc trưng của Rust

+ Ownership (bài sau)

```
fn main(){
    let x = String::from("Hello world");
    let y = x;
    println!("x:{}",x);

}
```

```
error[E0382]: borrow of moved value: `x`
```

Một số đặc trưng của Rust

Polymorphism (Bài sau)

+ Rust không phải là object oriented programming language
+ Rust là system programming language

https://www.quora.com/Why-does-Rust-call-itself-a-%E2%80%9Csystems-programming%E2%80%9D-language
https://www.reddit.com/r/rust/comments/zbj9io/why_is_rust_always_advertised_as_system/

Một số đặc trưng của Rust

Macros (Bài sau)

+ Code trong code
+ ví dụ: println!, vec![]

Một số đặc trưng của Rust

Memory Management

+ No garbage collection

https://marketsplash.com/tutorials/rust/rust-garbage-collection/#:~:text=Rust's%20memory%20management%20techniques%20ensure,of%20memory%20is%20properly%20managed.

11. Thực hành

https://github.com/CocDap/Rust-Bootcamp-2023/tree/class-1-exercises

Form submission:
https://forms.gle/DSWCiu5LQ9tpnpK2A

Cảm ơn mọi người đã lắng nghe