

Giorno 8

RUST

1 Rust

Rust è un linguaggio i cui aspetti fondamentali sono:

- **Lifetimes, Ownership:** Alle aree di memoria allocate (variabili nello **heap**) è assegnato uno ed un solo **owner**.

Sia *a* il puntatore ad un'area di memoria. Se scrivo *b* = *a*, la ownership di tale area di memoria passa da *a* a *b*, che diventa l'unico puntatore a tale area di memoria. È anche possibile “prestare” (as opposed to: trasferire) un'area di memoria ad un'altra variabile, che manterrà la ownership fino al momento della sua **free** (in rust: **drop**), che avviene alla fine del blocco di definizione.

Se le variabili smettono di esistere alla fine del loro blocco di definizione, non c'è bisogno di fare **free** manualmente, o di usare un garbage collector. Possiamo quindi fare controlli statici sul **lifetime** delle variabili.

- **Traits:** = mixins, utilizzati per aggiungere funzionalità, e.g. si può definire una funzionalità di **copia** (**trait Copy**); In questo modo si può scrivere *b* = *a* (esempio di prima) ma si crea una copia della porzione di memoria, non un *alias*. Questo ci porta al prossimo punto:
- **Immutabilità** (come in OCaml)
- **Tipi di dato e pattern-matching**
- **Inferenza di tipo**
- **Generics**
- **Programmazione funzionale**

E questo è più o meno tutto quello che c'è da sapere su Rust per questo corso.