Rust

Don’t panic!

Göran Anderson
initgoran@gmail.com

Innan vi börjar…

- Vilka tider gäller för kursen?
  - Startar 09:00
  - Kafferast (~10:00 & ~14:30)
  - Lunch (12:00 – 13:00)
  - Slutar cirka 16:30
- Grundkurs i Rust
  - Men programmeringsvana krävs
- Kursens källkod:
  - https://gitlab.com/initgoran/rustkurs

Historik (axplock)

- 1950-tal Fortran, Lisp, Cobol
- 1960-tal BASIC, SQL
- 1970-tal C, Shell script
- 1980-tal C++, Objective-C, Perl
- 1990-tal Python, Java, JavaScript, PHP
- 2000-tal C#

Modern programmering

- Go, Rust, Kotlin, Swift, ...
- Bygger ofta på LLVM
- Ny syn på bl.a.
  - Asynkron programmering
  - Felhantering
  - Objektorientering

LLVM

- Modulär infrastruktur för hantering av kod
- Excellent statisk kodanalys
- Lätt att skapa nya språk
- Chris Lattner, Apple, alla andra står bakom
  - utom kanske Microsoft och Red Hat

Varför Rust?

- Snabbast
- Robust(**!**)
  - Lite som C++ (fast tvärtom)
- Modernt
  - Parallellisering, Unicode
- Bra kompilator

Varför inte Rust?

- Lågnivå
- Svårt att lära sig
- Inga exceptions(!)
- Tidsödande
  - På projektledarspråk: *Dyrt*
- Ont om utvecklare

När är Rust rätt?

- Höga prestandakrav
- Ersättare för C/C++
- Maskinnära
- Kodbibliotek
- Spel

När är Rust fel?

- Måttliga prestandakrav
- Begränsad utvecklingsbudget
- Enkla uppgifter

Rust

- Ett ”värde” är ett stycke data
  - I andra språk kallas det för ”objekt”
- Stöder inte objektorienterat arv
  - I stället används ”trait”
- Inga exceptions
  - Fel måste propageras

Rustprogrammering

- Bevisa för kompilatorn att programmet fungerar.
- T.ex. att värden
  - initaliseras innan de används,
  - används bara när de finns,
  - inte ändras samtidigt.

Mutable

- Variabler är *immutable* per default
  - Dvs. kan inte ändra sitt värde
- Ändringsbara variabler deklareras som *mutable*

Ägare

- Varje värde har exakt en ägare
- Ett värde kan äga andra värden
- När ägaren försvinner tas dess värden bort
  - drop
- Ägarskap kan överföras
  - move

Kopiering

- Enkla datatyper tillåter kopiering
  - The ”Copy” trait

Move

- Tilldelning överför ägarskap
  - för typer som inte är Copy
- Även vid funktionsparametrar
- Och returvärden

Referenser

- Värden kan lånas ut via referenser
- Referensen kan bara användas under värdets livstid
  - inga "null pointer exception"
  - kontrolleras vid kompileringen
- Mutable referenser är exklusiva

Nog talat...

SHOW US THE CODE!