

Rust

Rust on vuonna 2006 alkunsa saanut järjestelmäohjelmointikieli. Kieli pyrkii olemaan nopea ja turvallinen vaihtoehto alhaisen tason ohjelmointiin. Kielen kehitys on täysin yhteisöveitoista ja sen kääntäjä sekä standardikirjasto ovat avointa lähdekoodia.

* Tyypijärjestelmä

Rustin tyypitys on vahva ja staattinen, mutta poiketen esimerkiksi Javasta Rustissa tapahtuu tyypipäättelyä: esimerkiksi muuttujille ei ole pakollista määritellä eksplisiittisesti tyyppiä, vaan kääntäjä päättelee arvolle sopivan tyyppin. Rustissa ei tapahdu epäsuoria tyyppimuunnoksia ollenkaan, mutta suorat tyyppimuunnokset ovat sallittuja.

* Älykkäät osoittimet

Rust tarjoaa ainoana turvallisena vaihtoehtona muistiviittauksia varten C++:an tapaan älykkäitä osoittimia, jotka mahdollistavat osoittimien automaattisen tuhoamisen. Älykkäät osoittimet toteutetaan useimmiten käyttäen tietueita. Rustin standardikirjasto tarjoaa käytettäväksi myös valmiita toteutuksia.

* Omistus

Muistinhallinta tapahtuu Rustissa omistussuhteita käyttäen. Rustissa ei ole siis roskienkeruuta ja turvallisuuden nimissä muistia ei varata ja vapauteta käsin. Rustissa jokaisella arvolla on omistaja, omistajia voi olla vain yksi ja kun omistaja poistuu ohjelman suorituksen näkyvyysalueelta, omistettu arvo poistetaan muistista. Omistuksesta johtuen esimerkiksi olioviite voi olla vain yhden nimetyn arvon omistuksessa kerrallaan.

Rust on perinteisen järjestelmäohjelmoinnin lisäksi hyvä web-kehitykseen, sillä Rust kääntyy suoraan WebAssembly:ksi. Rust tulee kasvattamaan suosiotaan tulevaisuudessa sen intohimoisen käyttäjäkunnan ansiosta. Hyvää: Suorituskyky, cargo, turvallisuus, funktionaalisuus. Pahaa: merkkijonojen kanssa työskentely (sisäänrakennettu `str` ja `std::String`).

Rust esimerkki:

```
fn main() {
    let text = "Kiitos KURSSISTA Kaikki";
    let s = 3;

    println!("{}", text);
    println!("Ciphared: {}",
            caesar(&text, s));
}

fn caesar(text:&str, s:i16) -> String {
    let mut result = String::from("");

    for i in text.chars() {
        if i.is_uppercase() {
            let x = (((i as i16) + s - 65)
                    % 26 + 65).abs() as u8;
            result.push(x as char);
        } else {
            let x = (((i as i16) + s - 97)
                    % 26 + 97).abs() as u8;
            result.push(x as char);
        }
    }
    return result;
}
```