Programowanie w języku Rust

Wprowadzenie i organizacja zajęć

mgr inż. Szymon Zyguła

Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska

Semestr zimowy 2026/27

Organizacyjnie

Najważniejsze zasady zaliczenia

- Obecność:
  - wykłady – nieobowiązkowe,
  - laboratoria – obowiązkowe (maksymalnie 4 nieusprawiedliwione nieobecności na ocenianych).
- Punkty:
  - 3 prace domowe/projekty: 15 pkt, 40 pkt, 15 pkt,
  - 10 laboratoriów punktowanych: 3 pkt każde,
    - 1 pkt za pracę na laboratorium (nie dotyczy dzisiaj),
    - 1 pkt za pracę w domu,
    - 1 pkt za pokazanie i odpowiedź na pytanie,
  - razem 100 pkt,
- Spóźnienia projektów: -5 pkt za każdy tydzień opóźnienia pracy domowej.
- Wszystkie zadania wykonywane indywidualnie.

Zasady co do AI

- Zakaz używania AI do generowania rozwiązań (ChatGPT, Copilot, itp.).
- Można używać AI do tłumaczenia błędów, oceny rozwiązania, propozycji innych podejść itp.

Plan wykładów w semestrze I

1. Podstawy języka: podstawy systemu cargo, typy danych, kontrola przepływu, funkcje, struktury, referencje.
2. Posiadanie, przenoszenie, kontroler zapożyczeń, okresy życia.
3. Algebraiczne typy danych, dopasowywanie wzorców, wyrażenia if-let, let-else i while-let.
4. Elementy biblioteki standardowej i obsługa błędów: kolekcje, Option, Result, łańcuchy znakowe.
5. Typy generyczne, cechy, złożone okresy życia.
6. Rozbudowane elementy systemu cargo: modularyzacja projektów, podział na biblioteki, zarządzanie zależnościami.
7. Elementy programowania funkcyjnego: funkcje wyższych rzędów, funkcje anonimowe, iteratory, operacje monadyczne, biblioteka Itertools.

Plan wykładów w semestrze II

8. Inteligentne wskaźniki i zarządzanie pamięcią: Box, Rc, Weak, RefCell, Cell, Cow, Borrow(Mut), LazyCell, OnceCell.
9. Programowanie obiektowe: obiekty cech, dynamiczna dyspozycja, wzorce projektowe.
10. Programowanie równoległe i biblioteka Rayon: wyłączność dostępu, struktury synchronizacyjne, kanały, podstawy zmiennych atomowych.
11. Zaawansowane cechy i typy.
12. Makra deklaratywne.
13. Sekcje niebezpieczne: wskaźniki, pomijanie okresu życia, PhantomData, koercje, transmutacje, nieinicjalizowana pamięć, destruktory, wycieki.
14. Makra proceduralne: wywodowe, atrybutywne, funkcyjne.
15. Inne tematy.

Strona przedmiotu

pages.mini.pw.edu.pl/~zygulas/rust/pl

Literatura

1. Klabnik, Steve, Carol Nichols, and Chris Krycho. *The Rust Programming Language*. Wyd. 3. No Starch Press, 2026.
2. Gjengset, Jon. *Rust for Rustaceans: Idiomatic Programming for Experienced Developers*. No Starch Press, 2021.
3. Blandy, Jim, Jason Orendorff, and Leonora FS Tindall. *Programming Rust*. O'Reilly Media, 2021.
4. McNamara, Tim. *Rust in Action*. Simon and Schuster, 2021.

Konsultacje

Zdalnie lub po wykładzie po wcześniejszym umówieniu.

O Języku

Chronologia

- Prace nad językiem w 2006 zaczął Graydon Hoare w wolnym czasie (pracownik Mozilla Research). motywacja: język, w którym trudno pisać niedziałający kod.
- Od 2009 Mozilla oficjalnie wspierała rozwój języka.
- Pierwsze publiczne wydanie (0.1): początek 2012.
- Wersja 1.0: 2015.
- Obecna wersja: 1.99 (wydana 28 września 2026).
- Nowa wersja co 6 tygodni.
- Obecnie sponsorowany przez Rust Foundation.

Co to za język?

- Kompilowany (backend LLVM).
- Bez garbage collection.
- Przeznaczenie i zastosowania podobne do C++.
- Mieszany paradygmat (nie jest obiektowy, nie jest funkcyjny).
- Znacznie bardziej rygorystyczny kompilator niż w C++. np. system typów wymusza poprawność referencji (nie da się zrobić „dangling reference").
- Wiele „ergonomicznych" cech (rozbudowany system makr, algebraiczne typy danych).

Gdzie jest używany?

- Firefox,
- Linux,
- Windows,
- Android,
- Discord,
- Amazon Web Services,
- Microsoft Azure,
- Dropbox,
- ...
- open source,
- naukowcy języków programowania

Zapotrzebowanie

- Zapotrzebowanie na programistów Rusta stale rośnie.
- Nadal jest bardzo niewielu doświadczonych programistów tego języka.

Pozostałe rzeczy

Środowiska programistyczne

- Rust Rover (od JetBrains)
- Visual Studio Code
- Vim / Neovim / Helix / Emacs / inny generyczny edytor

Linki

- Książka: `https://doc.rust-lang.org/stable/book/`
- Referencja biblioteki standardowej: `https://doc.rust-lang.org/stable/std/index.html`